BioData Mining

REVIEW Open Access

Using graph theory to analyze biological networks

Georgios A Pavlopoulos[1,2*], Maria Secrier[3], Charalampos N Moschopoulos[4,5], Theodoros G Soldatos[6], Sophia Kossida[5], Jan Aerts[2], Reinhard Schneider[3,7] and Pantelis G Bagos[1]

* Correspondence: pavlopou@embl.de
[1]Department of Computer Science and Biomedical Informatics, University of Central Greece, Lamia, 35100, Greece
Full list of author information is available at the end of the article

Abstract

Understanding complex systems often requires a bottom-up analysis towards a systems biology approach. The need to investigate a system, not only as individual components but as a whole, emerges. This can be done by examining the elementary constituents individually and then how these are connected. The myriad components of a system and their interactions are best characterized as networks and they are mainly represented as graphs where thousands of nodes are connected with thousands of vertices. In this article we demonstrate approaches, models and methods from the graph theory universe and we discuss ways in which they can be used to reveal hidden properties and features of a network. This network profiling combined with knowledge extraction will help us to better understand the biological significance of the system.

Keywords: biological network clustering analysis, graph theory, node ranking

Introduction

The theory of complex networks plays an important role in a wide variety of disciplines, ranging from computer science, sociology, engineering and physics, to molecular and population biology. Within the fields of biology and medicine, potential applications of network analysis include for example drug target identification, determining a protein's or gene's function, designing effective strategies for treating various diseases or providing early diagnosis of disorders. Protein-protein interaction (PPI) networks, biochemical networks, transcriptional regulation networks, signal transduction or metabolic networks are the highlighted network categories in systems biology often sharing characteristics and properties.

Protein-protein interaction (PPI) networks [1] mainly hold information of how different proteins operate in coordination with others to enable the biological processes within the cell. Despite the fact that for the majority of proteins the complete sequence is already known, their molecular function is not yet fully determined. Predicting protein function is still a bottleneck in computational biology research and many experimental and computational techniques have been developed in order to infer protein function from interactions with other biomolecules. Large-scale and high-throughput techniques can detect proteins that interact within an organism. Among them, the most well-known are the pull down assays [2], tandem affinity purification (TAP) [3], yeast two-hybrid (Y2H) [4], mass spectrometry [5], microarrays [6] and phage display [7]. Some very well-known datasets that have been recently produced by employing the aforementioned techniques and that are widely used are the Tong [8], Krogan [9], DIP [10], MIPS [11], Gavin 2002 [5] and Gavin 2006 [12] datasets. Besides the various experimental methods, a variety of large biological databases that contain information concerning PPI data is already available and most of them are organism specific. Some well-known databases are the Yeast Proteome Database (YPD) [13], the Munich Information Center for Protein Sequences (MIPS) [14], the Molecular Interactions (MINT) database [15], the IntAct database [16], the Database of Interacting Proteins (DIP) [10], the Biomolecular Interaction Network Database (BIND) [17], the BioGRID database [18], the Human Protein Reference Database (HPRD) [19], the HPID [20] or the DroID [21] for Drosophila. Two additional well-documented services based on text mining analysis are the Stitch [22] and String [23] databases.

Regulatory networks (GRNs) contain information concerning the control of gene expression in cells. This process is modulated by many variables, such as transcription factors [24], their post-translational modifications or association with other biomolecules [25]. Usually, these networks use a directed graph representation in an effort to model the way that proteins and other biological molecules are involved in gene expression and try to imitate the series of events that take place in different stages of the process. They often exhibit specific motifs and patterns concerning their topology. Data collection, data integration and analysis techniques give now the possibility to study gene regulatory networks in a larger scale [26]. Protein-DNA interaction data is collected in databases like JASPAR [27], TRANSFAC [28,29] or B-cell interactome (BCI) [30], while post-translational modification can be found in databases like Phospho.ELM [31], NetPhorest [32] or PHOSIDA [33].

Signal transduction networks often use multi-edged directed graphs to represent a series of interactions between different bioentities such as proteins, chemicals or macromolecules and to investigate how signal transmission is performed either from the outside to the inside of the cell, or within the cell. Environmental parameters change the homeostasis of the cell and, depending on the circumstances, different responses can be triggered. Similarly to GRNs, these networks also exhibit common patterns and motifs concerning their topology [34]. Databases that store information about signal transduction pathways are MiST [35], TRANSPATH [36], etc.

Metabolic and biochemical networks [37] are powerful tools for studying and modelling metabolism in various organisms. As metabolic pathways, we consider a series of chemical reactions occurring within a cell at different time points. The main role within a metabolic network is played by the enzymes, since they are the main determinants in catalyzing biochemical reactions. Often, enzymes are dependent on other cofactors such as vitamins for proper functioning. The collection of pathways, holding information about a series of biochemical events and the way they are correlated, is called a metabolic network. Modern sequencing techniques allow the reconstruction of the network of biochemical reactions in many organisms, from bacteria to human [38,39]. Among the several databases holding information about biochemical networks some of the most popular are the Kyoto Encyclopedia of Genes and Genomes (KEGG) [40], EcoCyc [41], BioCyc [42] and metaTIGER [43]. Several methods have also been discovered to analyze the pathway structure of metabolic networks [44-48].

Many computer readable formats are available to describe biological networks. The *Systems Biology Markup Language (SBML)* [49] is an XML-like machine-readable language, that is able to represent models to be analyzed by a computer. SBML can represent metabolic networks, cell signaling pathways, regulatory networks, and many other kinds of systems [50]. Other file formats that can represent biological networks are the *Proteomics Standards Initiative Interaction (PSI-MI)* [51], *Chemical Markup Language* (*CML*) [52,53] for chemicals or *BioPAX* [54] for pathways. Secondary formats that can also be used in similar ways are the *Cell Markup Language* [55] which is an XML-like machine-readable language mainly developed for the exchange of computer-based mathematical models or the *Resource Description Framework, RDF* which is a language for the representation of information about resources on the World Wide Web [56,57].

After having given a short overview of how data can be produced either experimentally or retrieved from various databases and which formats are available for each type of network, we further emphasize on the computational analysis as defined in graph theory. We finally conclude by describing which properties of the ones discussed below characterize the various networks.

Graph Theory and Definitions

To introduce the basic concepts of graph theory, we give both the empirical and the mathematical description of graphs that represent networks as they are originally defined in the literature [58,59].

Undirected single graph

A graph G can be defined as a pair (V, E) where V is a set of vertices representing the nodes and E is a set of edges representing the connections between the nodes. We define as $E = \{(i, j)| \ i, j \in \ V\}$ the single connection between nodes i and j. In this case, we say that i and j are ***neighbors***. A *multi-edge connection* consists of two or more edges that have the same endpoints. Such multi-edges are especially important for networks in which two elements can be linked by more than one connection. In such cases, each connection indicates a different type of information. This is an important feature since there are networks such as protein-protein interaction networks in which two proteins might be evolutionary related, co-occur in the literature or co-express in some experiments, resulting by this way in three different connections, each one with a different meaning. An example of PPI database that takes into account the different types of interactions between proteins is String [23].

Directed graph

A directed graph is defined as an ordered triple $G = (V, E, f)$, where f is a function that maps each element in E to an ordered pair of vertices in V. The ordered pairs of vertices are called ***directed edges, arcs or arrows***. An edge $E = (i, j)$ is considered to have direction from $\boldsymbol{i}$ to $\boldsymbol{j}$. Directed graphs are mostly suitable for the representation of schemas describing biological pathways or procedures which show the sequential interaction of elements at one or multiple time points and the flow of information throughout the network. These are mainly metabolic, signal transduction or regulatory networks [34].

Weighted graph

A weighted graph is defined as a graph $G = (V, E)$ where V is a set of vertices and E is a set of edges between the vertices $E = \{(u, v) \mid u, v \in \ V\}$ associated with it a weight function *w: E→R*, where *R* denotes the set of all real numbers. Most of the times, the weight w_{ij} of the edge between nodes *i* and *j* represents the relevance of the connection. Usually, a larger weight corresponds to higher reliability of a connection. Weighted graphs are currently the most widely used networks throughout the field of bioinformatics. As an example, relations whose importance varies are frequently assigned to biological data to capture the relevance of co-occurrences identified by text mining, sequence or structural similarities between proteins or co-expression of genes [23,60].

Bipartite graph is an undirected graph $G = (V, E)$ in which V can be partitioned into 2 sets V_1 and V_2 such that $(u,v) \in E$ implies either $u \in V_1$ and $v \in V_2$ OR $v \in V_1$ and $u \in V_2$. Applications of this type of graph to visualization or modeling of biological networks range from representation of enzyme-reaction links in metabolic pathways to ontologies or ecological connections, as discussed in [61] or [62].

If $G = (V, E)$ is a graph, then $G_1 = (V_1, E_1)$ is called a ***subgraph*** or if $V_1 \subseteq V$ and $E_1 \subseteq E$, where each edge in E_1 is incident with vertices in V_1.

Examples and shapes describing the aforementioned graph types can be found in Figure 1. The most common data structures that are used to make these networks computer readable are adjacency matrices or adjacency lists. The following section provides a short mathematical description of these data structures.

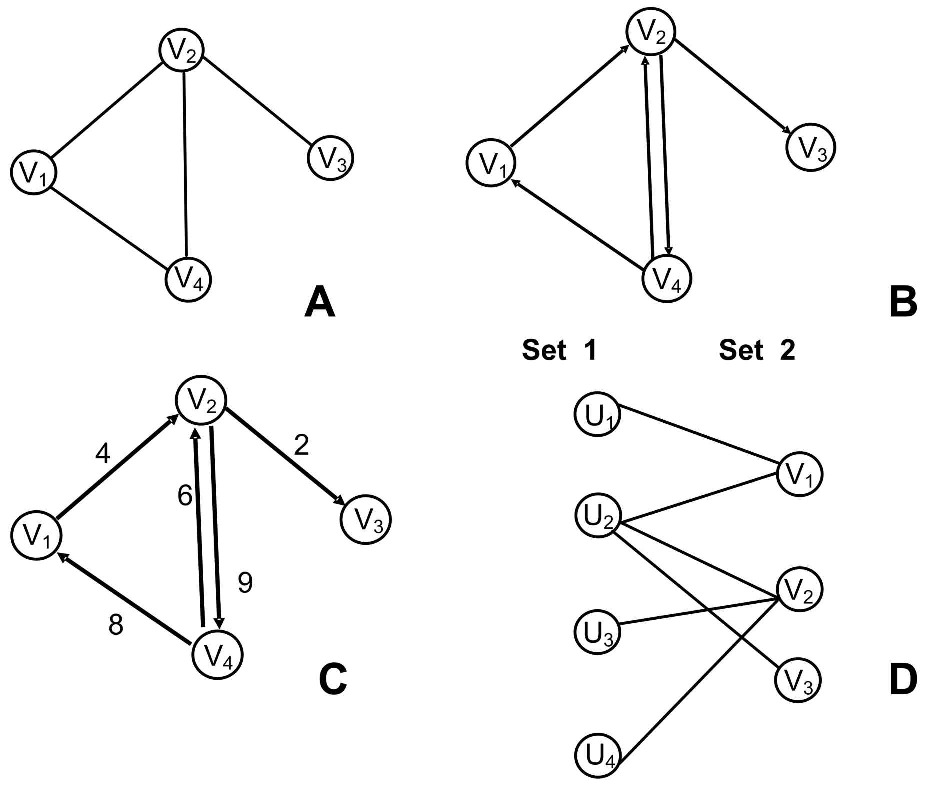


Figure 1 Undirected, Directed, Weighted, Bipartite graphs. A. Undirected Graph: $V = \{V_1, V_2, V_3, V_4\}$, $|V| = 4$, $E = \{(V_1, V_2), (V_2, V_3), (V_2, V_4), (V_4, V_1)\}$, $|E| = 4$. B. Directed Graph: $V = \{V_1, V_2, V_3, V_4\}$, $|V| = 4$, $E = \{(V_1, V_2), (V_2, V_3), (V_2, V_4), (V_4, V_1), (V_4, V_2)\}$, $|E| = 5$. C. Weighted Graph: $V = \{V_1, V_2, V_3, V_4\}$, $|V| = 4$, $E = \{(V_1, V_2, V_4), (V_2, V_3, V_2), (V_2, V_4, V_9), (V_4, V_1, V_8), (V_4, V_2, V_6)\}$, $|E| = 5$. D. Bipartite graph: $V = \{U_1, U_2, U_3, U_4, V_1, V_2, V_3\}$, $|V| = 7$, $E = \{(U_1, V_1), (U_2, V_1), (U_2, V_2), (U_2, V_3), (U_3, V_2), (U_4, V_2)\}$, $|E| = 6$.

The ***degree of a node*** in an undirected graph is the number of connections or edges the node has to other nodes and is defined as $deg(i) = k(i) = |N(i)|$ where $N(i)$ is the number of the neighbors of node *i*. If a network is directed, then each node has two different degrees, the ***in-degree*** $deg_{in}\ (i)$ which is the number of incoming edges to node *i*, and the ***out-degree*** $deg_{out}(i)$ which is the number of outgoing edges from node *i*. The ***total connectivity*** of a network is defined as $C = \frac{E}{N(N-1)}$ where E is the number of edges and N the total number of nodes. The connectivity structure of biological networks is often informative with respect to reaction interplay and reversibility, compounds that structure the network, like in metabolism, or trophic relationships, like in food-web networks. Such connectivity profiles can be detected based on mixture models using software like MixNet [63].

Data Structures

The two main data structures used to store network graph representations are described below.

Adjacency matrix

Given a graph $G = (V, E)$ the adjacency matrix representation consists of a $|V|x|V| = nxn$ matrix $A = (a_{ij})$ such that $a_{ij} = 1$ if $(i, j)\in V$ or $a_{ij} = 0$ or otherwise $A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$, $n = |V|$. In the case where we have weighted graphs $a_{ij} = w_{ij}$ *if* $(i, j)\in V$ or $a_{ij} = 0$ otherwise. For **undirected graphs** the matrix is symmetric because $a_{ij} = a_{ji}$. The aforementioned rule does not apply to directed graphs, because in that case the upper and the lower triangle parts of the matrix reveal the direction of the edges. Adjacency matrices require space of $\Theta(|V|^2)$ and are best suited for dense and not for sparse graphs. For an all-against-all symmetric data set, only the upper or the lower triangular part of the matrix is necessary, which requires $\Theta(|V|)$ amount of memory to be allocated. This data structure is more efficient for cluttered networks, where the density of the connections between elements is relatively high. In the case of a fully connected graph where all nodes are connected with each other, adjacency matrices are highly suggested. To reduce memory allocation to half for larger scale data, a symmetric *2D* matrix A can be stored as a *1D* matrix B, where $A[i,j] = B[\frac{i(i-1)}{2} + j]$ if the first element is α_{11} like for example in Matlab platform or $A[i,j] = B[\frac{i(i+1)}{2} + j]$ if the first element is α_{00} like in most programing languages. Matrix B currently hosts the lower part of matrix A. If for example A is a 3 × 3 matrix starting from element α_{11}, $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$ then matrix B is defined as $B = \{\alpha_{11},\alpha_{21},\alpha_{22},\alpha_{31},\alpha_{32},\alpha_{33}\}$. The 1D array will be of size $\frac{n(n+1)}{2}$ including the diagonal.

Adjacency list

Given a graph $G = (V, E)$ the adjacency list representation consists of an array *Adj* of $|E|$ elements where for each $e \in E$ $Adj(0, e) = i \in V$. Adjacency lists require space Θ $(|V| + |E|)$ and are preferable for sparse graphs with a low density of connections. An example of how these data structures represent a graph is given in Figure 2.

Network Properties

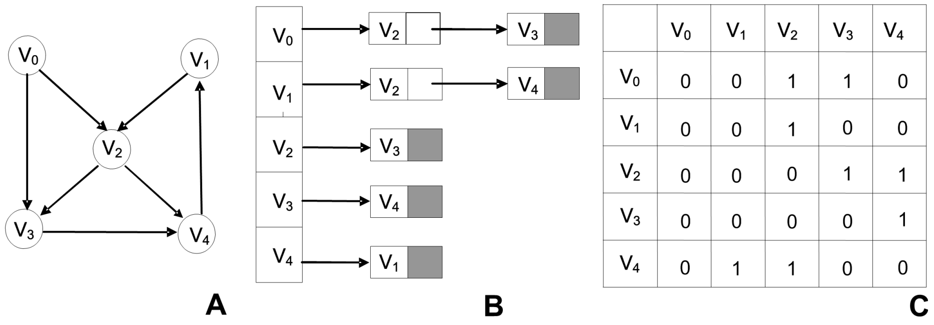
Looking at different network properties can provide valuable insight into the internal organization of a biological network, the repartition of molecules among cellular processes, as well as the evolutionary constraints that have shaped an organism's protein, metabolic or regulatory network into a functional, feasible structure. In the following, we give a short description of the main properties that are commonly analyzed in networks.

The ***graph density*** shows how sparse or dense a graph is according to the number of connections per node set and is defined as $density = \frac{2|E|}{|V|(|V|-1)}$. A ***sparse graph*** is a graph where $|E| = O(|V|^k)$ and $2 > k > 1$ or otherwise when $|E|$ “ $|V|$. ***Dense*** is a graph where $|E|$ “ $|V|^2$. It has been argued that biological networks are generally sparsely connected, as this confers an evolutionary advantage for preserving robustness. This has been observed for a series of organisms: the transcriptional regulatory networks of *S. cerevisiae*, *E. coli*, *D. melanogaster* all have connectivity densities lower than 0.1 [64].

In the mathematical field of graph theory, a ***complete graph*** is a simple graph in which every pair of distinct vertices is connected by a unique edge. The complete graph on n vertices has $\frac{n(n-1)}{2}$, $n = |V|$ number of edges and it is a regular graph of degree $|V|$ - ***1***.

Graph Isomorphism

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two undirected graphs. A function f: $V_1 \text{->} V_2$ is called isomorphism if f is an edge-preserving bisection, such that for all $a, b \in V_1$, $(a, b) \in E_1$ if and only if $(f(a), f(b)) \in E_2$. When such function exists, then G_1 and G_2 are called isomorphic. An example is shown in Figure 3.



	V0	V1	V2	V3	V4
V0	0	0	1	1	0
V1	0	0	1	0	0
V2	0	0	0	1	1
V3	0	0	0	0	1
V4	0	1	1	0	0

Figure 2 Data structures. A. A Directed Graph: A random graph consisting of five nodes and six directed edges. B. Adjacency List: The data structure which represents the directed graph using lists. C. Adjacency Matrix: The data structure which represents the directed graph using a 2D matrix. The zeros represent the absence of the connection whereas the ones represent the existence of the connection between two nodes. The matrix is not symmetric since the graph is directed.

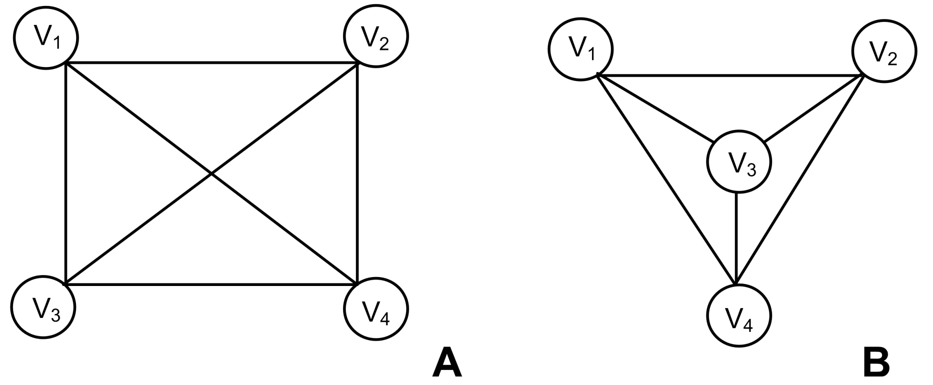


Figure 3 Graph Isomorphism. *$V = \{V_1, V_2, V_3, V_4\}$, $|V| = 4$, $E = \{(V_1, V_2), (V_1, V_3), (V_1, V_4), (V_2, V_3), (V_2, V_4), (V_3, V_4)\}$, $|E| = 6$*. Graphs *A* and *B* have different topology but they are isomorphs. The graph is fully connected and every node is connected to any other so that it forms a fully connected clique.

A ***walk*** is a pass through a specific sequence of nodes *$(v_1, v_2,..., v_L)$* such that *$\{(v_1, v_2), (v_2, v_3),..., (v_{L-1}, v_L)\} \subseteq E$*. A ***simple path*** is a walk with no repeated nodes. A ***cycle*** is a walk *$(v_1, v_2,..., v_L)$* where $v_1 = v_L$ with no other nodes repeated and $L > 3$, such that the last node is the same with the first one. A ***trail*** is a path where no edge can be repeated. A graph is called ***cyclic*** if it contains a cycle. In any other case it is called ***acyclic***. All of the aforementioned can be found as an example in Figure 4. A ***complete graph*** is a graph in which every pair of nodes is ***adjacent***. If *(i, j)* is an edge in a graph *G* between nodes *i* and *j*, we say that the vertex *i* is *adjacent* to the vertex *j*. An undirected graph is ***connected*** if one can get from any node to any other node by following a sequence of edges. A directed graph is ***strongly connected*** if there is a directed path from any node to any other node. This does not require an all-against combination. The ***distance*** *$\delta(i, j)$* from *i* to *j* is the length of the *shortest path* from *i* to *j* in *G*. If no such path exists, then we set $\delta(i, j) = \infty$ assuming that the nodes are so far between each other so they are not connected. Practically, for the distance $\delta(i, j) = \infty$ we can

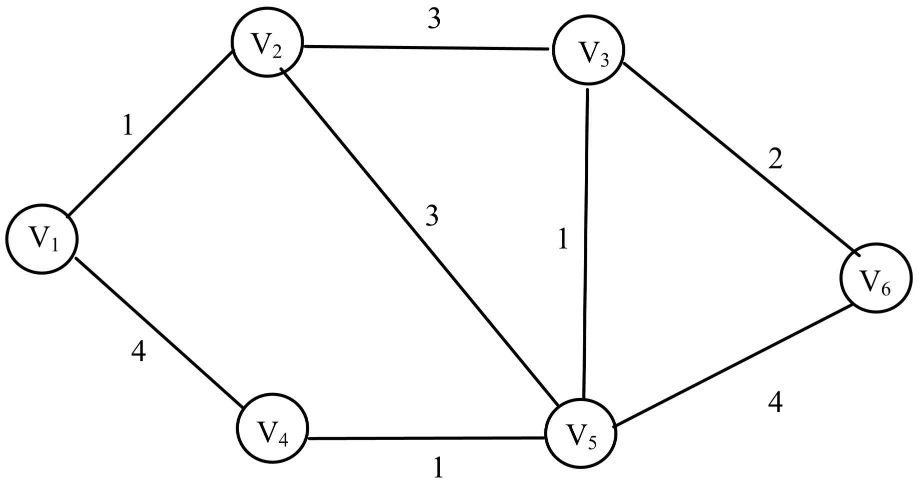


Figure 4 Walks, simple paths trails and cycles in graphs. A *walk* is a sequence of nodes *e.g.* (V_2, V_3, V_6, V_5, V_3). A *simple path* is a walk with no repeated nodes, e.g. (V_1, V_4, V_5, V_2, V_3). A trail is a walk where no edges are repeated e.g. (V_1, V_2, V_3 V_6). A *cycle* is a walk (V_1, V_2,..., V_L) where $V_1 = V_L$ with no other nodes repeated and L>3, e.g. (V_1, V_2, V_5, V_4, V_1).

use the maximum weight of the graph by adding one. Thus $\delta(i, j) = \infty = (max_{d(i,\ j)}+1)$. To define the shortest path problem we can briefly say that it is the methodology of finding a path between two nodes such that the sum of the weights of its constituent edges is minimized. The ***average path length*** and the ***diameter*** of a graph *G* are defined to be the average and maximum value of $\delta(i, j)$ taken over all pairs of distinct nodes, $i, j \in V(G)$ which are connected by at least one path. More specifically, the average path length of a network is the average number of edges or connections between nodes, which must be crossed in the shortest path between any two nodes. It is calculated as $\delta = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} \delta_{\min}(i,j)$ where $\delta_{min}(i, j)$ is the minimum distance between nodes *i* and *j*. The diameter of a network is the longest shortest path within a network. The ***diameter*** is defined as $D = \max_{i,j} \delta_{\min}(i,j)$. The most common algorithms for calculating the shortest paths are ***Dijkstra***'s greedy algorithm [65] and ***Floyd's*** dynamic algorithm [66]. *Dijkstra's* algorithm has running time complexity $O(N^2)$ where *N* is the number of vertices and returns the shortest path between a source vertex *i* and all other vertices in the network. *Floyd's* algorithm has running time complexity $O(N^3)$ and requires an all-against-all matrix that contains the distances of every node in the network to every other node in the network.

A ***clique*** in an undirected graph *G* is a subgraph *G'* which is complete. An independent set in a graph is a subset of the vertices such that no pair of vertices is an edge in the graph. The size of a clique comes from the number of vertices it contains. A ***maximal clique*** is a clique that cannot be extended by including one more adjacent vertex, i.e. a clique which does not exist exclusively within the vertex set of a larger clique. A maximum clique is a clique of the largest possible size in a given graph. The clique problem refers to the problem of finding the largest clique in any graph *G*. This problem is *NP*-complete, and as such, many consider that it is unlikely that an efficient algorithm for finding the largest clique of a graph exists. Figure 3b shows a clique. A very famous method to find maximal cliques in a graph is the so-called Bron-Kerbosch algorithm [67]. Detection and analysis of these structures has found many biological applications: identifying groups of consistently co-expressed genes in microarray datasets, finding cis regulatory motifs or matching three-dimensional structures of molecules [68,69]. Several tools have been developed for clique identification, like Clique Finder within the Arabidopsis Co-expression Tool server [70] or MIClique [68]. Bioconductor [71] provides a large collection of software for clique analysis.

Clustering Coefficient is the measurement that shows the tendency of a graph to be divided into clusters. A cluster is a subset of vertices that contains lots of edges connecting these vertices to each other. Assuming that *i* is a vertex with degree $deg(i) = k$ in an undirected graph *G* and that there are *e* edges between the *k* neighbors of *i* in *G*, then the *Local Clustering Coefficient* of *i* in *G* is given by $C_i = \frac{2e}{k(k-1)}$. Thus, C_i measures the ratio of the number of edges between the neighbors of *i* to the total possible number of such edges, which is $k(k\text{-}1)/2$. It takes values as $0 \leq C_i \leq 1$. The ***average Clustering Coefficient*** of the whole network $C_{average}$ is given by $C_{average} = \frac{1}{N} \sum_{i=1}^{N} \frac{E_i}{k_i(k_i-1)}$ where $N=|V|$ is the number of vertices. The closer the local clustering coefficient is to 1, the more likely it is for the network to form clusters.

Obviously, a clique would come with local clustering coefficient equal to 1. An example showing how local clustering coefficient is calculated is shown in Figure 5.

Biological networks have a significantly higher average clustering coefficient compared to random networks, which proves their modular nature. Indeed, many cellular processes are governed by subsets of biomolecules that form an interaction module. Since cellular processes are linked, the modules tend to be linked as well, but the linking molecules are often few, such that the module overlap is quite low [72,73].

Centralization is the measurement that shows whether a network has a star-like topology or whether the nodes of the network have on average the same connectivity. The closer the centralization is to 1, the more likely is the network to have a star-like topology. The closer to 0, the more likely it is that the nodes of the network have on average the same connectivity (for example a square, where every node is connected with 2 neighbors). It is calculated as

$$Centralization = \frac{n}{n-2}\left(\frac{\max(k)}{n-1} - Density\right),$$

$$Centralization \approx \frac{\max(k)}{n-1} - Density$$

Network Motifs represent patterns in complex networks occurring significantly more often than in randomized networks [74]. They consist of subgraphs of local interconnections between network elements. A motif is a small connected graph *G'*. A *match G'* of a motif in graph *G* is a graph *G" which* is isomorphic to *G'* and a subgraph of *G*. Signal transduction and gene regulatory networks tend to be described by various motifs [72,75]. Although motif determination gives lots of information concerning the properties and the characteristics of a network, it does not necessarily reveal evidence about its function and the function of its components [76]. However, some motifs have been found to be associated with optimized biological functions, like in the case of positive and negative feedback loops, oscillators or bifans [73]. Figure 6 shows the most common motifs that are found in various networks.

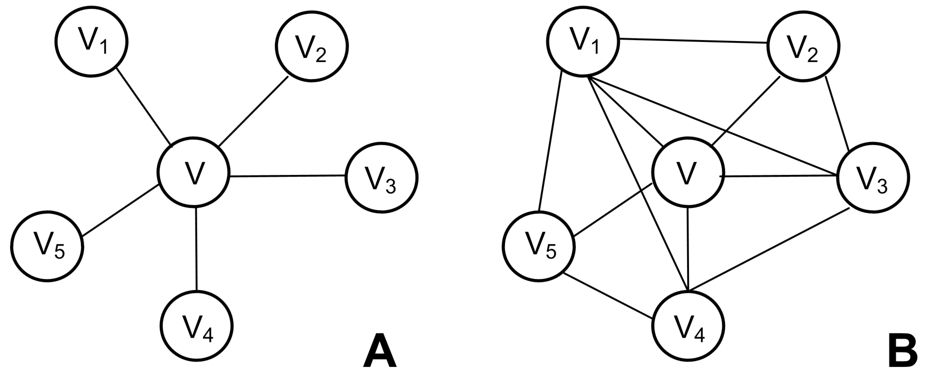


Figure 5 Clustering Coefficient. A) Node *V* behaves like a hub but it has clustering coefficient *C = 0*. B) Node *V* comes with a high clustering coefficient. The maximum number of potential connection is given by E_{max}=*|V|(|V|-1)/2* where *|V| = 5* is the number of the neighbors of node V, thus E_{max} *= 10*. The neighbors of node V are connected with 7 edges between each other, *E = {*$(V_1, V_2), (V_2, V_3), (V_3, V_4), (V_4, V_5), (V_5, V_1), (V_1, V_3), (V_1, V_4)$*}*. The clustering coefficient of node V is *C =* E_V/E_{max} *= 7/10 = 0.7.*

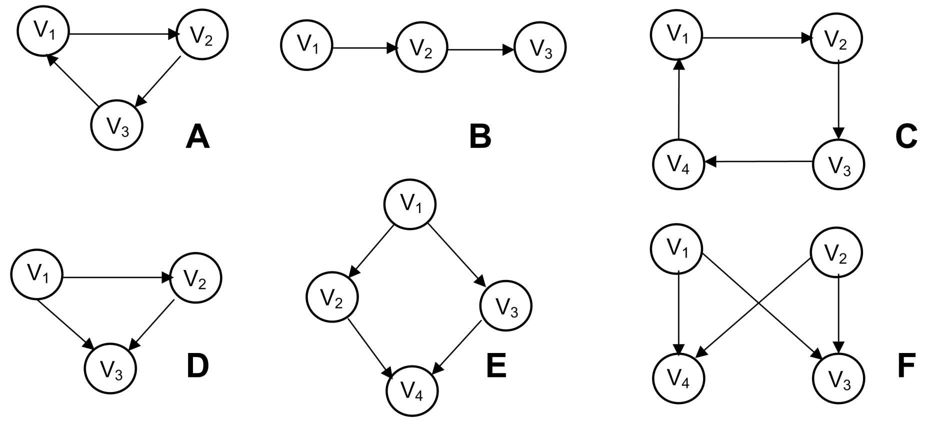


Figure 6 Network Motifs. Some common network motifs. A) *Feed-forward loop*. Type of networks: protein, neuron, electronic. B) *Three chain*. Type of network: food webs. C) *Four node feedback*. Type of network: gene regulatory, electronic. D) *Three node feedback*. Type of network: gene regulatory, electronic. E) *Bi-parallel*. Type of network: gene regulatory, biochemical. F) *Bi-Fan*. Type of networks: protein, neuron, electronic [74].

Network Centralities and Node Ranking

This section shows how nodes can be ranked or sorted according to their properties, depending on the question asked. In biological networks, it is important for example to detect central nodes or intermediate nodes that affect the topology of the network, depending of course on the biological question. Such a question would be to find the molecules in a biological pathway that are not necessarily central but have a crucial biological role in signal transduction or in PPI networks, to detect such nodes that interact with many other proteins or find molecules that are crucial for stimulating the expression of genes.

Degree Centrality shows that an important node is involved in a large number of interactions. For a node i, the degree centrality is calculated as $C_d(i) = deg(i)$. For directed graphs, each node is obviously characterized by two degree centralities. These are $C_{d\ in}(i) = deg_{in}(i)$ *and* $C_{d\ out}(i) = deg_{out}(i)$. Nodes with very high degree centrality are called ***hubs*** since they are connected to many neighbors (see Figure 5). Scale-free networks tend to contain hubs. The removal of such central nodes has great impact on the topology of the network. It has been shown that biological networks tend to be robust against random perturbations, but disruption of hubs often leads to system failure [77,78].

Closeness Centrality indicates important nodes that can communicate quickly with other nodes of the network. Let $G = (V, E)$ be an undirected graph. Then, the centrality is defined as $C_{clo}(i) = \frac{1}{\sum_{t \in V}^{|V|} dist(i,j)}$ where $dist(i, j)$ denotes the distance or else the shortest path p between the nodes i and j. An example is shown in Figure 7. Closeness centrality has been used to identify the top central metabolites in genome-based large-scale metabolic networks [79], to compare unicellular and multicellular eukarya, to rank pathways and obtain a perspective on the evolution of metabolic organization [80]. A decrease in closeness centrality of components has been observed as a consequence of increased distance between pathways throughout evolution [80]. It has been

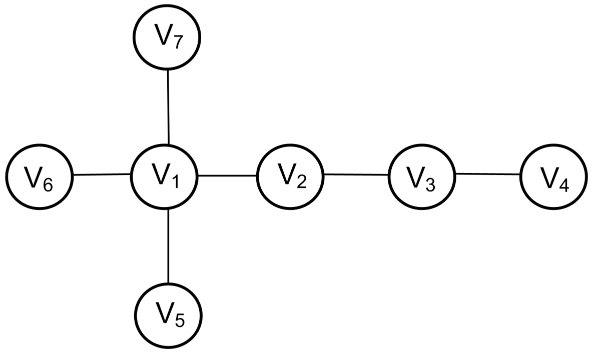


Figure 7 Closeness and Betweeness centralities. **Closeness centrality.** ***V_1***: $d_1 = 4 \times 1 + 1 \times 2 + 1 \times 3 = 9$, $C_{clo}(1) = 6/9$. V_1 accesses 4 nodes (V_2, V_5, V_6, V_7) with step 1, 1 node (V_3) with step 2 and 1 node (V_4) with step 3. 6 nodes can be accessed in total by V_1. ***V_2***: $d_2 = 2 \times 1 + 4 \times 2 = 10 > d_1$, $C_{clo}(2) = 6/10$. V_2 accesses 2 nodes (V_1, V_3) with step 1 and 4 nodes (V_4, V_5, V_6, V_7) with step 2. 6 nodes can also be accessed in total by V_2. As a result, V_1 is more central than node V_2 since $d1>d_2$. Betweenness centrality. $N_p(1) = 12$ shortest paths that pass through node V_1. The paths from the starting to the ending node are $\{V_2\text{-}V_5, V_2\text{-}V_6, V_2\text{-}V_7, V_3\text{-}V_5, V_3\text{-}V_6, V_3\text{-}V_7, V_4\text{-}V_5, V_4\text{-}V_6, V_4\text{-}V_7, V_5\text{-}V_6, V_5\text{-}V_7, V_6\text{-}V_7\}$. $N_p(2) = 8$ shortest paths that pass through node V_2. The paths are $\{V_1\text{-}V_3, V_1\text{-}V_4, V_3\text{-}V_5, V_3\text{-}V_6, V_3\text{-}V_7, V_4\text{-}V_5, V_4\text{-}V_6, V_4\text{-}V_7\}$. $N_p(3) = 5$ $\{V_1\text{-}V_4, V_2\text{-}V_4, V_4\text{-}V_5, V_4\text{-}V_6, V_4\text{-}V_7\}$. $N_p(4) = N_p(5) = N_p(6) = N_p(7) = 0$. $N_p = 25$ the total sum of shortest paths that pass through the nodes, thus $N_p= N_p(1)+N_p(2)+N_p(3)+N_p(4)+N_p(5)+N_p(6)+N_p(7)$. The centralities are $C_b\ (1) = 12/25 = 0.48$, $C_b\ (2) = 8/25 = 0.32$, $C_b\ (3) = 5/25 = 0.20$, $C_b\ (4) = C_b\ (5) = C_b\ (6) = C_b\ (7) = 0$, thus node V_1 is more central.

chosen as the best centrality measure that can be used extract the metabolic core of a network [81].

Betweenness Centrality shows that nodes which are intermediate between neighbors rank higher. Without these nodes, there would be no way for two neighbors to communicate with each other. Thus, *betweenness centrality* shows important nodes that lie on a high proportion of paths between other nodes in the network. For distinct nodes $i, j, w \in V(G)$, let σ_{ij} be the total number of shortest paths between i and j and $\sigma_{ij}(w)$ be the number of shortest paths from i to j that pass through w. Moreover, for $w \in V(G)$, let $V(i)$ denote the set of all ordered pairs, (i, j) in $V(G) \times V(G)$ such that i, j, w are all distinct. Then, the Betweenness Centrality is calculated as $C_b(w) = \sum_{(i,j) \in V(w)} \frac{\sigma_{ij}(w)}{\sigma_{ij}}$. An example is shown in Figure 7. Proteins with high betweenness centralities have been termed “bottlenecks”, for their role as key connector proteins with essential functional and dynamic properties [73], for example metabolites that control the flux between two big metabolic modules. Calculation of this centrality measure is discussed in [82] and [83] and their properties within the PPI network of yeast are detailed in [84].

Eigenvector Centrality ranks higher the nodes that are connected to important neighbors. Let $G = (V, E)$ be an undirected graph and A the adjacency matrix of network G. The eigenvector centrality is the eigenvector $C_{eiv\text{-}}$ of the largest eigenvalue λ_{max} in absolute value such that $\lambda C_{eiv} = AC_{eiv}$. Formally, if A is the adjacency matrix of a network G with $V(G) = \{v_1,..., v_n\}$, and $\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|$, then the eigenvector centrality $C_{eiv}(v_i)$ of the node v_i is given by the i^{th} coordinate x_i of a normalized eigenvector that satisfies the condition $Ax=\rho(A)x$. Such algorithms can be used for efficient page ranking on the web. In biology this centrality measurement has been used, among others, to identify synthetic genetic interactions [85], gene-disease associations [86] or network hubs [77].

Eccentricity Centrality is the measure that shows how easily accessible a node is from other nodes. Let $G = (V, E)$ be an undirected graph. The eccentricity centrality is calculated as $C_{ecc} = \frac{1}{\max\{dist(i,j)\}}$ where *dist(i, j)* is the shortest path between nodes *i* and *j*. The eccentricity C_{ecc} of a vertex *V* is the greatest distance between v and any other vertex. An example is shown in Figure 8. In biological networks, proteins or other bioentities with high eccentricity are easily functionally reachable by other components of the network, and thus can readily perceive changes in concentration of other enzymes or molecules they are linked to. In contrast, those proteins that have lower eccentricities will often play a marginal functional role in the system [87].

Subgraph Centrality is the measure that ranks nodes according to the number of subgraphs of the overall network in which the node participates, with more weight given to small subgraphs. Let $G = (V, E)$ be an undirected graph and *A* the adjacency matrix of network *G*. The subgraph centrality of a node is calculated as $C_{sg} = \sum_{k=0}^{\infty} \frac{(A^k)_{ii}}{k!}$. Subgraph centrality analysis has been used to study essential proteins in proteomic maps [77], to compute the degree of folding of protein chains [88], to understand the molecular structure of drug-like compounds [89] or to zoom into the topological environment of certain nodes in PPI networks of several organisms [90].

Matching Index is the measure that shows how similar two nodes are within the network. Two vertices that are functionally similar do not always have to be connected. The matching index M_{ij} measures the "similarity" of two nodes and is based on the number of common neighbors shared by nodes *i* and *j*. It is calculated as

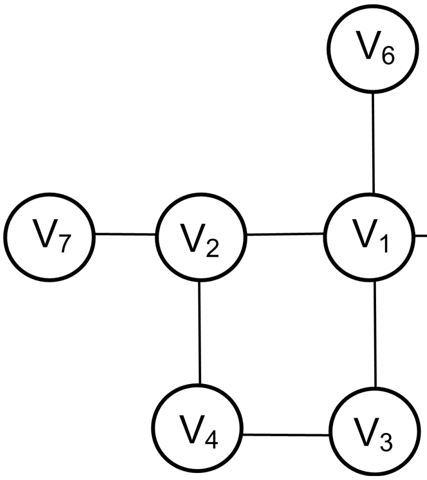


Figure 8 Eccentricity Centrality. $\boldsymbol{V_1}$: 4 × 1, 2 × 2; V_1 accesses 4 nodes *(V_2, V_3, V_5, V_6)* with step 1 and 2 nodes *(V_4, V_7)* with step 2. The step represents the shortest path. The maximum shortest path $d_{max} = 2$. $\boldsymbol{V_2}$: 3 × 1, 3 × 2; Similarly V_2 accesses 3 nodes *(V_4, V_7, V_1)* with step 1 and 3 nodes *(V_3, V_5, V_6)* with step 2. The maximum shortest path $d_{max} = 2$. $\boldsymbol{V_3}$: 2 × 1, 3 × 2, 1 × 3; Similarly V_3 accesses 2 nodes *(V_1, V_4)* with step 1, 3 nodes *(V_2, V_5, V_6)* and one node *(V_7)* with step 3. The maximum shortest path $d_{max} = 3$. $\boldsymbol{V_4}$: 2 × 1, 2 × 2, 2 × 3; The maximum shortest path $d_{max}=3$. $\boldsymbol{V_5}$: 1 × 1, 3 × 2, 2 × 3; The maximum shortest path $d_{max} = 3$. $\boldsymbol{V_6}$: 1 × 1, 3 × 2, 2 × 3; The maximum shortest path $d_{max} = 3$. $\boldsymbol{V_7}$: 1 × 1, 2 × 2, 3 × 3; The maximum shortest path $d_{max} = 3$. As a result, the ordering of the nodes according to C_{ecc} : (V_1,V_2), (V_3,V_4,V_5,V_6,V_7).

$M_{ij} = \frac{\sum common\_neighbors}{\sum total\_number\_of\_neighbors}$ or $M_{ij} = \frac{\sum_{k,l}^{N} A_{ik}A_{jl}}{k_i + k_j - \sum_{k,l}^{N} A_{ik}A_{jl}}$. An example is shown in Figure 9. The matching index is often used to cluster different components of a biological network according to some property. For instance, it has been used to describe spatial growth in brain networks during development [91] or to predict the connectivity of primate cortical networks [92].

Further centrality measurements and their application to the study of PPIs in yeast are introduced in [85]. A discussion about how centrality correlates with lethality in biological networks can be found in [93]. The coupling between centrality and essentiality has also been investigated in several eukaryotic protein networks [94]. It is very often the case that studies of a particular network involve the analysis and comparison of several centrality measures, for instance to study pleiotropy in human genetic diseases [87], to compare PPI and transcriptional regulation networks [95] or to test hub essentiality [77]. Tools that have implemented functionality for exploring the different types of centralities previously mentioned in biological networks and not only are CentiBiN [96], Visone [97], Pajek [98], VisANT [99]. In most of the cases, however, only a limited selection of centrality measures is available.

Network Topology

The topology of the network often reveals information about its biological significance. Often, networks follow patterns and rules and have a specific topology that allows scientists to go through a deeper investigation towards knowledge extraction.

Scale-free or otherwise real world networks describe natural networks like online communities (i.e Facebook) where the nodes are the people and the edges the connection between them, or networks such as the World Wide Web (www) where the nodes are individual web pages and the links are hyperlinks. Many biological networks also have scale-free properties, with nodes representing bioentities and edges the

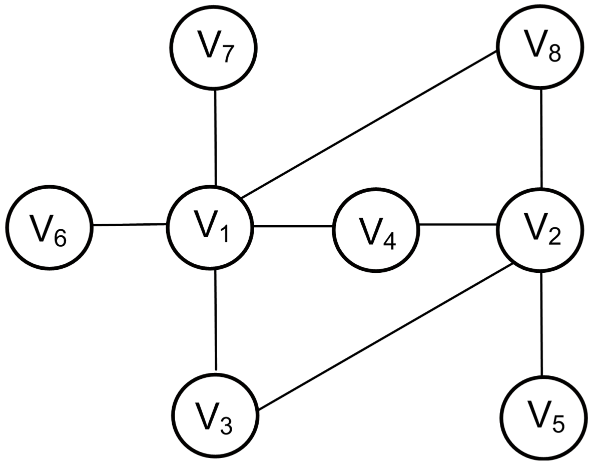


Figure 9 Matching Index. $\boldsymbol{V_1}$ is connected with 5 nodes $(V_3, V_4, V_6, V_7, V_8)$. $\boldsymbol{V_2}$ is connected with 4 nodes (V_3, V_4, V_5, V_8). $\boldsymbol{V_3}$ is connected with 2 nodes (V_1, V_2). $\boldsymbol{V_4}$ is connected with 3 nodes (V_1, V_2). $\boldsymbol{V_5}$ is connected with 1 node (V_2). $\boldsymbol{V_6}$ is connected with 1 node (V_1). $\boldsymbol{V_7}$ is connected with 1 node (V_1). $\boldsymbol{V_8}$ is connected with 2 nodes (V_1, V_5). Node V_1 and V_2 are connected with 3 common nodes $(V_3, V_4, V_8)$and in total with 6 distinct neighbors $(V_3, V_4, V_8, V_5, V_6, V_7)$. The matching index will then be $M_{1,2} = 3/6 = 0.5$, thus V_1 and V_2 are functionally similar even though they are not connected.

interactions between them (like proteins that interact physically or metabolites that take part in the same reaction) [73,93,100]. Assuming that *k* is the number of links originating from a given node and *P(k)* the probability that the degree of a randomly chosen vertex equals *k*, a scale-free network exhibits a power law distribution $P(k) \sim k^{-\gamma}$ where γ denotes the ***degree exponent***. A scale-free network can be constructed by progressively adding nodes to an existing network and introducing links to existing nodes with preferential attachment so that the probability of linking to a given node *i* is proportional to the number of existing links k_i that the node has. Thus the connectivity of one node *i* to any other node *j* should approximately follow the rule: $P(links\_to\_note\_i) \sim \frac{k_i}{\sum_j k_j}$.

The ***degree distribution P(k)*** has become one of the most prominent characteristics in network topology. In terms of numerical estimation, a more reliable property, very similar to the previous, is the ***cumulative degree distribution*** $\boldsymbol{P_c(k)}$. For a power law distribution $P(k) \sim k^{-\gamma}$ the cumulative degree distribution is of the form $P(k) \sim k^{(-\gamma-1)}$ and describes the probability of a random chosen node in the network to have a degree *greater* than *k*. Even though lots of research has been done on power law analysis in biological networks, it is still not an established approach widely accepted by the scientific community [101].

To visually represent the properties of the network we usually rank the vertices according to their degree and then plot the degree versus the rank of each vertex. Another representation is to create a histogram by plotting the vertices of the graph sorted according to their degree using a logarithmic scale. A third and very popular representation is to plot the degrees of the nodes sorted versus either their degree distribution *P(k)* or their cumulative degree distribution $P_c(k)$. An interesting analysis of most of these properties in various PPI, metabolic or transcriptional networks of several organisms (*S. cerevisiae*, *H. pylori*, *C. elegans*) can be found in [100].

A network is called ***assortative*** if the vertices with higher degree have the tendency to connect with other vertices that also have high degree of connectivity; one such category is social networks [102]. If the vertices with higher degree have the tendency to connect with other vertices with low degree then the network is called ***disassortative***. This is characteristic to most molecular interaction networks, where hubs have the tendency to link to nodes with fewer interaction partners rather than to other hubs [103,104]. Newman [102] discusses this property for protein interaction networks, neural networks and food webs.

To correlate the degrees of two nodes *i* and *j* we use a joint probability distribution $P(k_i, k_j) = P(k_i)P(k_j)$. A more straightforward way is to use the Pearson's Correlation Coefficient (PCC), which quantifies the correlation or linear dependence between two variables (in this case, the degrees of two nodes). In other words, it measures to which extent one variable increases/decreases as the other increases. PCC (*r-value*) between two nodes is defined as the covariance of the two nodes divided by the product of their standard deviations. For the entire network, the assortativity coefficient is the measure of how assortative or disassortative a network is overall. If *M* is the number of edges, and x_i and y_i the degrees of the vertices at either ends of edge *i*, the assortativity coefficient *r* is calculated as follows [102].:

$$r = \frac{M^{-1}\sum_i x_i y_i - \left[M^{-1}\sum_i \frac{1}{2}(x_i + y_i)\right]^2}{M^{-1}\sum_i \frac{1}{2}(x_i{}^2 + y_i{}^2) - \left[M^{-1}\sum_i \frac{1}{2}(x_i + y_i)\right]^2}, \text{ with } i = 1 \ldots M$$

This is equivalent to the Pearson correlation coefficient of the degrees at either ends of an edge. The range of the *r*-values is between *+1* and *-1, r <0* corresponding to a disassortative network whereas r > 0 to an assortative one. Another way to correlate degrees is to calculate the ***average neighbor degree***. For each vertex *i*, the average degree of its neighbor is calculated as $k_{i,nn} = \frac{1}{k_i}\sum_{j=1}^{N_V} A_{ij}k_j$. The values are then averaged for all vertices with the same degree *k*, showing the average neighbor degree $k_{nn}(k)$.

Network Models

Several topological models have been built to describe the global structure of a network, as introduced below.

Erdös-Rényi model for random graphs [105]

This model was mainly introduced to describe the properties of a random graph. The simple model of a network involves taking a number of vertices *N* and connecting nodes by selecting edges from the *N(N-1)/2* possible edges randomly. The degree distribution for this model is given by a binomial distribution. The probability of a vertex to have degree *k* is $P(k) \approx e^{-\langle k\rangle}\frac{\langle k\rangle^k}{k!}$, where $\langle k\rangle$ is the average connectivity of the network. For small *P* probabilities, the network seems to be disconnected and consists of many isolated components whereas for *P >log(N)/N* almost all vertices are connected.

Watts and Strogatz model [106]

This model was introduced to describe networks that follow the small world topology. This type of topology characterizes many biological networks, like metabolic networks where it often happens that paths of few (three-four) reactions link most metabolites. As a consequence, local changes in metabolite concentration local perturbations in these networks will propagate throughout the entire network. In this model, the frequency of nodes *P(k)* with *k* connections follows a power-law distribution equation $P(k) \sim k^{-\gamma}$, in which most nodes are connected with small proportion of other nodes and a small proportion of nodes are highly connected. Thus each vertex is connected to *N/2* nearest neighbors. In exponential networks the probability that a node has a high number of connections is very low.

Barabasi-Albert model [107]

This model describes scale-free networks and it is one of the most basic models since it describes most of the biological networks [37,108]. The concept behind this model is to reveal information about the dynamics of the network, especially from an evolutionary perspective. The networks are built to mimic gene duplication events, such that they expand continuously by addition of new nodes and the new nodes attach preferentially to sites that are already well connected [109]. Initially we start with small number of nodes m_0. At each step, a new node $m < m_0$ is added and gets linked to the existing network. The probability that a new node is now connected to node i is $P(k_i) = \frac{k_i}{\sum_j k_j}$ where k_i is the connectivity of node i. The rate of connecting new nodes to node i is $\frac{\partial k_i}{\partial t} = \Delta k \frac{k_i}{\sum_j k_j} = m \frac{k_i}{2mt} = \frac{k_i}{2t}$. The connections are time-dependent so $k_i(t) = m\sqrt{\frac{t}{t_i}}$ where t_i is the time point when node i enters the network. The probability that a node has degree smaller than k is $t_i > \frac{m^2 t}{k^2}$. The probability density of the network is $P(k) = \frac{\partial p(k_i(t) < k)}{\partial k}$ or $P(k) = \frac{2m^2 t + 1}{m_0 + tk^3} \sim k^{-3}$, such that the model produces a power law distribution of $\gamma = 3$.

Cluster Analysis and Visualization

Cluster analysis [110] aims at classifying a set of observations into two or more mutually exclusive *unknown* groups based on combinations of variables. Thus, cluster analysis is usually presented in the context of *unsupervised* classification [111]. It can be applied to a wide range of biological study cases, such as microarray, sequence and phylogenetic analysis [112]. The purpose of clustering is to group different objects together by observing common properties of elements in a system. In biological networks, this can help identify similar biological entities, like proteins that are homologous in different organisms or that belong to the same complex and genes that are co-expressed [113,114].

It is generally difficult to predict behavior and properties based on observations of behaviors or properties of other elements in the same system, therefore various approaches for cluster analysis emerge. Clustering algorithms may be *Exclusive, Overlapping, Hierarchical* or *Probabilistic*. In the first case, data are grouped in an exclusive way, so that a certain element can be assigned to only one group (exclusively). On the other hand, the overlapping clustering uses fuzzy sets to cluster data, so that each point may belong to two or more clusters with different degrees of membership. A hierarchical clustering algorithm organizes data in hierarchies and is based on the union between the two nearest clusters; it is commonly used for microarray and sequence analysis [115]. A more analytical categorization of clustering algorithms can be found at [110,116].

An important component of a clustering algorithm is the distance measure between data points. If all the components of the data instance vectors have the same physical units, it is then possible that the simple Euclidean distance metric is sufficient to successfully group similar data instances. One example is to cluster cities on a map, since in this case Euclidean distance represents real natural distances. However, for higher dimensional data the Euclidean distance can sometimes be misleading. In that case, a popular measure is the ***Minkowski metric*** and is calculated as $d(i,j) = \left(\sum_{k=1}^{D} |x_{i,k} - x_{j,k}|^p\right)^{\frac{1}{p}}$ where D is the dimensionality of the data. The *Euclidean* can be calculated if we set $p = 2$, while *Manhattan* metric has $p = 1$. There are no general theoretical guidelines for selecting a measure for a given application.

Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. There are two different strategies to organize data. These are the *agglomerative* and the *divisive*: ***Agglomerative***: It is a "bottom-up" approach. Each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. ***Divisive***: This is a "top-down" approach. In this case, all of the observations start by forming one cluster, and then split recursively as one moves down the hierarchy. Some of the most common tree based clustering algorithms that organize data in hierarchies are the Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [117,118], Neighbor Joining [112,119] and Hierarchical Clustering [120,121], all of which represent their clusters as tree structures. The results of hierarchical clustering are usually presented in a dendrogram. Figure 10 shows an example of how genes can be clustered.

Let n_r be the number of clusters and x_{ri} is the ith object in cluster r and cluster r is formed from clusters p and q. In the following, we describe the different methods used to calculate distances between clusters in hierarchical clustering.

Single linkage calculates the smallest distance between objects in the two clusters to merge them: $d(r, s) = \min(dist(x_{ri}, x_{sj}))$, $i \in (i, \ldots, n_r)$, $j \in (1, \ldots. n_s)$.

Complete linkage calculates the largest distance between objects in the two clusters to merge them: $d(r, s) = \max(dist(x_{ri}, x_{sj}))$, $i \in (i, \ldots, n_r)$, $j \in (1, \ldots. n_s)$.

Average linkage uses the average distance between all pairs of objects in any two clusters: $d(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist(x_{ri}, x_{sj})$. This algorithm is also known as *Unweighted Pair Group Method with Arithmetic Mean (**UPGMA**)* [117,118].

Centroid linkage finds the Euclidean distance between the centroids of the two clusters: $d(r,s) = ||\overline{x_r} - \overline{x_s}||_2, \overline{x_r} = \frac{1}{n_r} \sum_{i=1}^{n_r} x_{ri} \cdot || \, ||_2$ is the Euclidean distance.

Median linkage uses the Euclidean distance between weighted centroids of the two clusters, $d(r,s) = ||x_r - x_s||_2, x_r, x_s$ are weighted centroids for the clusters r and s. If cluster r was created by combining clusters p and q, $\mathrm{x_r}$ is defined recursively as $x_r = \frac{1}{2}(x_p + x_q)x_r$.

Single or complete linkages are the fastest of the linkage methods. However, single linkage tends to produce stringy clusters, which is not always preferable. The centroid or average linkage produce better results regarding the accordance between the produced clusters and the structure present in the data. These methods require much more computations. Average linkage and complete linkage may be the preferred methods for microarray data analysis [115].

Ward's linkage finds the incremental sum of squares; that is, the increase in the total within-cluster sum of squares as a result of joining two clusters. The within-cluster sum of squares is defined as the sum of the squares of the distances between all objects in the cluster and the centroid of the cluster. The sum of squares measure is equivalent to the following distance measure $d(r,s) = \sqrt{\frac{2n_r n_s}{(n_r + n_s)}} ||\overline{x_r} - \overline{x_s}||_2$,

where $|| \, ||_2$ is the Euclidean distance and $\overline{x_r}, \overline{x_s}$ are the centroids of clusters r and s and n_r and n_s are the number of elements in clusters r and s.

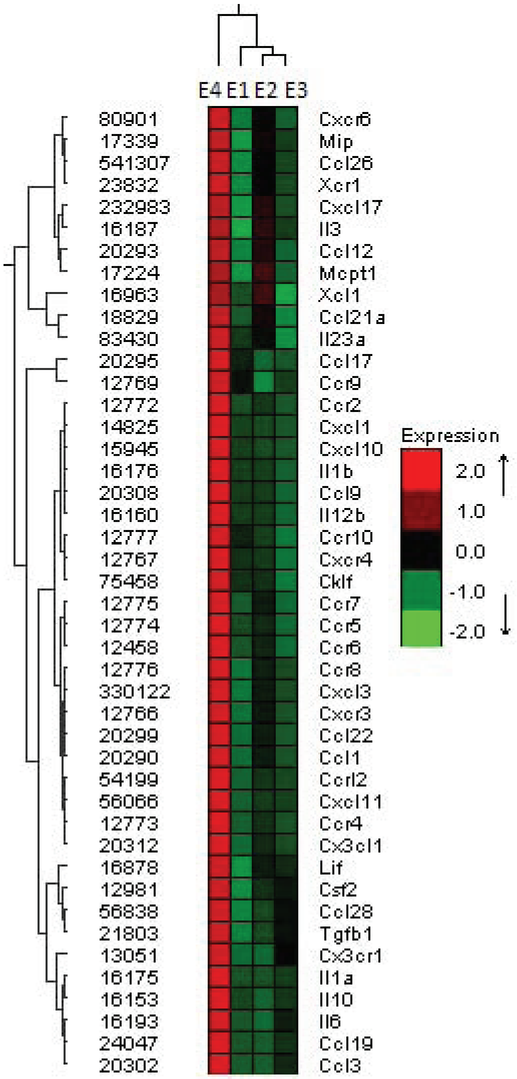


Figure 10 Average linkage hierarchical clustering example. The expression of 44 genes was measured in 4 experiments (E_1, E_2, E_3, E_4). The genes were classified according to their coexpression levels. The Pearson Correlation Coefficient was used (*r*-value) to analyze gene set signal values. Genes were clustered according to the *r*-value correlation matrix using the Average Linkage Hierarchical clustering method. The tree on the left clusters the expressions of the genes whereas the tree on top of the figure clusters the profiles of the experiments. Thus experiments E_2 and E_3 are similar and closely related.

Weighted average linkage uses a recursive definition for the distance between two clusters. If cluster r was created by combining clusters p and q, the distance between r and another cluster s is defined as the average of the distance between p and s and the distance between q and s: $d(r,s) = \frac{(d(p,s) + d(q,s))}{2}$.

Neighbor Joining [112,119] was initially proposed for finding pairs of operational taxonomic units (OTUs) that minimize the total branch length at each stage of clustering of OTUs starting with a star-like tree. The branch lengths as well as the topology of a parsimonious tree can quickly be obtained by using this method [112].

Known platforms that already share the tree-based algorithms described above are the Hierarchical Clustering Explorer (HCE) [122,123], MEGA [124-127] or TM4 [128].

A recent review article shows which file formats, visualization techniques and algorithms can be used for tree analysis [129].

Another category of clustering algorithms tries to cluster data in separate groups by identifying common properties that the nodes of a network share. Different strategies exist, like for example trying to find dense areas in a graph or areas where message exchange between nodes is easier or to identify strongly connected components or clique-like areas etc. Many of such algorithms have been used in different case studies like for example to identify protein families [130], to detect protein complexes in PPI networks [131,132], or for finding patterns and motifs in a sequence [133]. Even though many more exist, some of the most famous algorithms are given below.

Markov Clustering [134] (MCL) algorithm is a fast and scalable unsupervised clustering algorithm based on simulation of stochastic flow in graphs. The MCL algorithm can detect cluster structures in graphs by a mathematical bootstrapping procedure which takes into account the connectivity properties of the underlying network. The process deterministically computes the probabilities of random walks through a graph by alternating two operations: expansion, and inflation of the underlying matrix. The principle behind it is that random walks on a graph are likely to get locked within dense subgraphs rather than move between dense subgraphs via sparse connections. In other words, higher length paths are more often encountered between nodes in the same cluster than between nodes within different clusters, such that the probabilities between nodes in the same complex will typically be higher in expanded matrices. Clusters are identified by alternating expansion and inflation until the graph is partitioned into subsets so that there are no longer paths between these subsets [135,136].

***k*-Means** [137] is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. K-means and its modifications are widely used for gene expression data analysis [138]. It is a supervised method and users need to predefine the number of clusters. Its complexity is $O(nlk)$ where k is the number of clusters, n the size of the dataset and l the loops of the algorithm. The k-means algorithm is one of the simplest and fastest clustering algorithms. However, it has a major drawback: the results of the k-means algorithm may change in successive runs because the initial clusters are chosen randomly.

Affinity Propagation [139] takes as input measures of similarity between pairs of data points and simultaneously considers all data points as potential candidates. Real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges.

Restricted Neighborhood Search Cluster Algorithm [140]: It tries to find low cost clustering by composing first an initial random clustering. Later it iteratively moves one node from one cluster to another in a random way trying to improve the clustering cost.

Spectral clustering [141]: This algorithm tries to find clusters in the graph such that the nodes within a cluster are connected with highly-similar edges and the connections between such areas are weak, constituted by edges with low similarity. The aim is to identify these tightly coupled clusters, and cut the inter-cluster edges. Figure 11 shows an example of protein complex prediction from PPI yeast dataset [12].

Despite the great variety of clustering techniques, many articles directly compare the various clustering methodologies like [135] and [142]. Very often we encounter articles

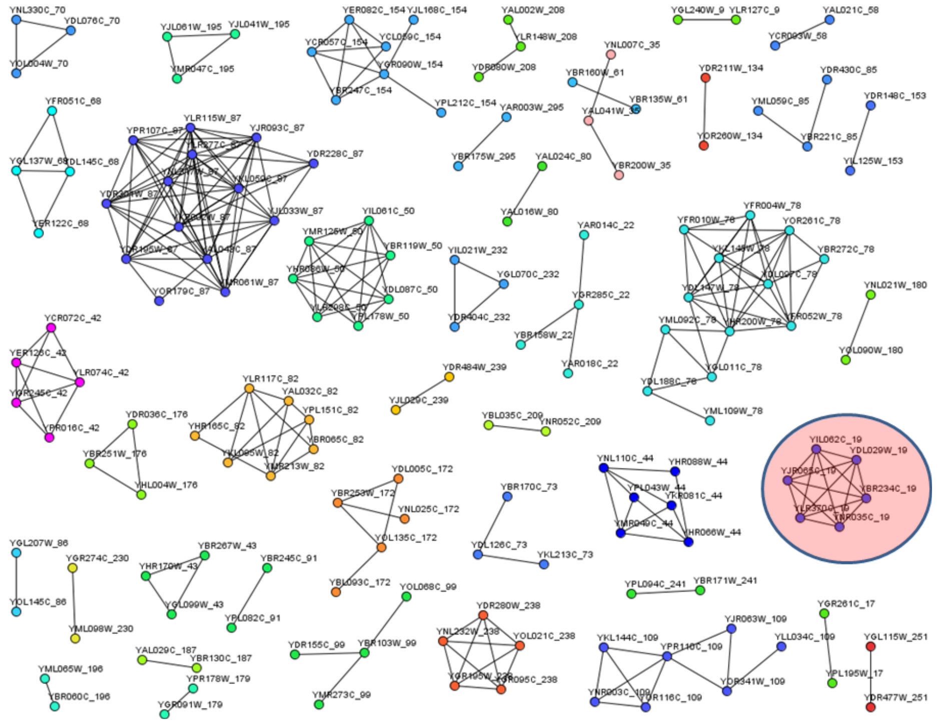


Figure 11 Predicting protein complexes from PPI networks. Protein complexes predicted after applying Spectral clustering algorithm and filtering the results in a yeast protein-protein dataset [12] using the jClust application [146]. The budding yeast Arp2/3 complex that is highlighted was successfully predicted.

that compare similar algorithms using different datasets and come to very diverse conclusions and results i.e [142,143].

Concerning the visualization of networks, the availability of clustering techniques and their complex configuration/combination, today to a large extent, there is a lack of visualization platforms or tools that are able to integrate a variety of more advanced algorithms and the development implementation of such implementations emerges [144]. Platforms that share clustering algorithms are the Network Analysis Tool (NEAT) [145] and jClust [146] but they are still poor in the variety of methods they offer. Software like ArrayCluster [147] and MCODE [60] is often used in analysis of gene expression profiles and coexpression detection. Many visualization tools [144] such as Medusa [148], Cytoscape [149], Pajek [98] and many others [144] visualize networks in both 2D and 3D, but very few of them like Arena3D [150] try to bridge the gap between clustering analysis and visualization.

Discussion

Protein-protein interaction (PPI) networks [1] are very diverse and it is difficult to come to general conclusions about their properties, mainly because data are generated from different sources both computationally and experimentally as described in a previous section. In most of the cases, PPI networks follow the laws of scale-free networks [93]. In such networks there are always proteins with higher degree of connectivity that appear to be of higher biological significance. Such proteins are the most important for the survival of the cell [93]. Large-scale maps of protein interaction networks have been constructed recently using high-throughput approaches to identify protein interactions [151-155]. It has been shown that these networks are highly dynamic, both during common cellular processes and on the evolutionary scale [109]. Further details on PPI network construction and analysis are given in [156].

Regulatory networks (GRNs) are usually sparsely connected. More specifically, the average number of upstream-regulators per gene is less than two [64]. Theoretical results show that the selection for robust gene networks will form minimal complexes even more sparsely connected [64], thus a fundamental design constraint could shape the evolution of gene network complexity. Network maps have been constructed for the transcriptional regulatory networks of *E. coli* and *S. cerevisiae* and are maintained in databases [26,157,158]. They are very sensitive and flexible to evolution [159] since their dynamics changes continuously over time and since transcription factors evolve faster than their target genes [160]. The number of regulators N_{reg} grows faster than the number of genes N_{tot} they regulate and it has been shown that $\frac{\mathrm{N_{reg}}}{N_{\mathrm{tot}}} \approx N$ for prokaryotes and $\frac{\mathrm{N_{reg}}}{\mathrm{N_{tot}}} \approx N^{0.3}$ for eukaryotes, where N is the network size [161,162]. Mostly they follow the power-law distributions and thus belong to the scale-free network category, even though some of them, like the transcriptional regulatory networks of *E. coli* and *S. cerevisiae* have been shown to possess mixed scale-free and exponential properties [75].

Signal transduction networks are characterized by several patterns and motifs like self-sustaining feedback loops. These patterns appear at every time point during the signal transduction in the network and they reveal information about the topology of the network, therefore are important for biological functionality [163]. The nodes with the highest centralities in such networks correspond to domains involved in signal transduction and cell-cell contacts [164]. Signal transduction networks are sparse and they follow the scale-free properties. In *E. coli* and *S. sereviase*, the degree distribution is $P(k) = k^{-\gamma}$, $\gamma \approx 2$ and most of the molecules are involved into few interactions and only few of them have higher connectivity [8,165].

Metabolic and biochemical networks are scale-free networks indicating a small-world structure considering the topology of the network based on its metabolites [166,167], where all of the nodes in such networks are connected through a short path to any other. One example is presented in [167] for *E. coli*. The probability that a substrate participates as input in k metabolic reactions follows the power-law distribution $P(k) = k^{-\gamma_{in}}$, $\gamma_{in} \approx 2.2$ whereas the probability of a substrate to be produced by k metabolic reactions equals similarly to $P(k) = k^{-\gamma_{out}}$, $\gamma_{out} \approx 2.2$. Metabolic networks are extremely heterogeneous and vary from organism to organism. The scale-free structure remains robust even after removal of some central nodes [166] and despite the fact that the architecture of the metabolic networks rests on highly connected substrates [167]. A characteristic feature of these networks is the apparent conservation of network diameter even in distantly related organisms [167]. It has been shown that metabolic networks can form hierarchical structures [168,169] where specific patterns and motifs are overrepresented. Methods to detect such motifs have been applied on network pathways analysis [44,45,47,48], one example being flux mode analysis [48].

Conclusions

The mathematical discipline which underpins the study of complex networks in biological and other applications is graph theory. It has been successfully applied to the study of biological network topology, from the global perspective of their scale-free, small world, hierarchical nature, to the zoomed-in view of interaction motifs, clusters and modules and the specific interactions between different biomolecules. The structure of biological networks proves to be far away from randomness but rather linked to function. Furthermore, the power of network topology analysis is limited, as it provides a static perspective of what is otherwise a highly dynamic system, such that additional tools should be combined with this approach in order to obtain a deeper understanding of cellular processes.

The complexity of biological networks increases as data are accumulated. The inherent variability of biological data, data inaccuracy and noise, the overload of information and the need to study the dynamics and network topology over time, are currently the bottlenecks in systems biology. Improved techniques for integration of data arising from different sources, as well as for visualization, will be crucial for understanding the functionality of complex networks. Moreover, new mathematical developments in the field and discovery of new areas of applications should be pursued in the near future.

Acknowledgements

GAP and MS were financially supported by the EMBL PhD Programme. GAP was financially supported as a postdoctoral fellow from the Greek State Scholarship Foundation (I.K.Y - http://www.iky.gr/IKY/portal/en).

Author details

[1]Department of Computer Science and Biomedical Informatics, University of Central Greece, Lamia, 35100, Greece. [2]Faculty of Engineering - ESAT/SCD, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001, Leuven-Heverlee, Belgium. [3]Structural and Computational Biology Unit, EMBL, Meyerhofstrasse 1, 69117, Heidelberg, Germany. [4]Department of Computer Engineering & Informatics, University of Patras, Rio, 6500, Patras, Greece. [5]Bioinformatics & Medical Informatics Team, Biomedical Research Foundation, Academy of Athens, Soranou Efessiou 4, 11527, Athens, Greece. [6]Life Biosystems GmbH, Belfortstrasse 2, 69117, Heidelberg, Germany. [7]Luxembourg Centre for Systems Biomedicine (LCSB), University of Luxembourg, Campus Limpertsberg, 162 A, avenue de la Faïencerie, L-1511 Luxembourg.

Authors' contributions

MS was financially supported by the EMBL PhD Programme. PGB and RS supervised the study. CNM, TGS and SK wrote parts of the manuscript. JA input was crucial for the article. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Received: 2 November 2010 Accepted: 28 April 2011 Published: 28 April 2011

References

1. Pellegrini Matteo, Haynor David, Johnson JM: **Protein interaction networks.** *Expert Rev Proteomics* 2004, **1(2)**.
2. Vikis HG, Guan KL: **Glutathione-S-transferase-fusion based assays for studying protein-protein interactions.** *Methods Mol Biol* 2004, **261**:175-186.
3. Puig O, Caspary F, Rigaut G, Rutz B, Bouveret E, Bragado-Nilsson E, Wilm M, Seraphin B: **The tandem affinity purification (TAP) method: a general procedure of protein complex purification.** *Methods* 2001, **24(3)**:218-229.
4. Ito T, Chiba T, Ozawa R, Yoshida M, Hattori M, Sakaki Y: **A comprehensive two-hybrid analysis to explore the yeast protein interactome.** *Proc Natl Acad Sci USA* 2001, **98(8)**:4569-4574.
5. Gavin AC, Bosche M, Krause R, Grandi P, Marzioch M, Bauer A, Schultz J, Rick JM, Michon AM, Cruciat CM, *et al*: **Functional organization of the yeast proteome by systematic analysis of protein complexes.** *Nature* 2002, **415(6868)**:141-147.
6. Stoll D, Templin MF, Bachmann J, Joos TO: **Protein microarrays: applications and future challenges.** *Curr Opin Drug Discov Devel* 2005, **8(2)**:239-252.
7. Willats WG: **Phage display: practicalities and prospects.** *Plant Mol Biol* 2002, **50(6)**:837-854.
8. Tong AH, Lesage G, Bader GD, Ding H, Xu H, Xin X, Young J, Berriz GF, Brost RL, Chang M, *et al*: **Global mapping of the yeast genetic interaction network.** *Science* 2004, **303(5659)**:808-813.
9. Krogan NJ, Cagney G, Yu H, Zhong G, Guo X, Ignatchenko A, Li J, Pu S, Datta N, Tikuisis AP, *et al*: **Global landscape of protein complexes in the yeast Saccharomyces cerevisiae.** *Nature* 2006, **440(7084)**:637-643.

10. Xenarios I, Rice DW, Salwinski L, Baron MK, Marcotte EM, Eisenberg D: **DIP: the database of interacting proteins.** *Nucleic Acids Res* 2000, **28(1)**:289-291.
11. Mewes HW, Frishman D, Mayer KF, Munsterkotter M, Noubibou O, Pagel P, Rattei T, Oesterheld M, Ruepp A, Stumpflen V: **MIPS: analysis and annotation of proteins from whole genomes in 2005.** *Nucleic Acids Res* 2006, , **34 Database:** D169-172.
12. Gavin AC, Aloy P, Grandi P, Krause R, Boesche M, Marzioch M, Rau C, Jensen LJ, Bastuck S, Dumpelfeld B, *et al*: **Proteome survey reveals modularity of the yeast cell machinery.** *Nature* 2006, **440(7084)**:631-636.
13. Hodges PE, McKee AH, Davis BP, Payne WE, Garrels JI: **The Yeast Proteome Database (YPD): a model for the organization and presentation of genome-wide functional data.** *Nucleic Acids Res* 1999, **27(1)**:69-73.
14. Mewes HW, Amid C, Arnold R, Frishman D, Guldener U, Mannhaupt G, Munsterkotter M, Pagel P, Strack N, Stumpflen V, *et al*: **MIPS: analysis and annotation of proteins from whole genomes.** *Nucleic Acids Res* 2004, , **32 Database:** D41-44.
15. Zanzoni A, Montecchi-Palazzi L, Quondam M, Ausiello G, Helmer-Citterich M, Cesareni G: **MINT: a Molecular INTeraction database.** *FEBS Lett* 2002, **513(1)**:135-140.
16. Kerrien S, Alam-Faruque Y, Aranda B, Bancarz I, Bridge A, Derow C, Dimmer E, Feuermann M, Friedrichsen A, Huntley R, *et al*: **IntAct–open source resource for molecular interaction data.** *Nucleic Acids Res* 2007, , **35 Database:** D561-565.
17. Bader GD, Donaldson I, Wolting C, Ouellette BF, Pawson T, Hogue CW: **BIND–The Biomolecular Interaction Network Database.** *Nucleic Acids Res* 2001, **29(1)**:242-245.
18. Stark C, Breitkreutz BJ, Reguly T, Boucher L, Breitkreutz A, Tyers M: **BioGRID: a general repository for interaction datasets.** *Nucleic Acids Res* 2006, , **34 Database:** D535-539.
19. Keshava Prasad TS, Goel R, Kandasamy K, Keerthikumar S, Kumar S, Mathivanan S, Telikicherla D, Raju R, Shafreen B, Venugopal A, *et al*: **Human Protein Reference Database–2009 update.** *Nucleic Acids Res* 2009, , **37 Database:** D767-772.
20. Han K, Park B, Kim H, Hong J, Park J: **HPID: the Human Protein Interaction Database.** *Bioinformatics* 2004, **20(15)**:2466-2470.
21. Yu J, Pacifico S, Liu G, Finley RL Jr: **DroID: the Drosophila Interactions Database, a comprehensive resource for annotated gene and protein interactions.** *BMC Genomics* 2008, **9**:461.
22. Kuhn Michael, Szklarczyk Damian, Franceschini Andrea, Campillos Monica, von Mering Christian, Lars Juhl Jensen AB, Bork P: **STITCH 2: an interaction network database for small molecules and proteins.** *Nucleic Acids Res* 2010, , **38:** D552-D556.
23. Jensen LJ, Kuhn M, Stark M, Chaffron S, Creevey C, Muller J, Doerks T, Julien P, Roth A, Simonovic M, *et al*: **STRING 8–a global view on proteins and their functional interactions in 630 organisms.** *Nucleic Acids Res* 2009, , **37 Database:** D412-416.
24. Pea Carninci: **The transcriptional landscape of the mammalian genome.** *Science* 2005, **309**:1559-1563.
25. Rea Linding: **NetworKIN: a resource for exploring cellular phosphorylation networks.** *Nucleid Acids Res* 2008, **36**: D695-D699.
26. Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, Gerber GK, Hannett NM, Harbison CT, Thompson CM, Simon I, *et al*: **Transcriptional regulatory networks in Saccharomyces cerevisiae.** *Science* 2002, **298(5594)**:799-804.
27. Sandelin A, Alkema W, Engström P, Wasserman WW, Lenhard B: **JASPAR: an open-access database for eukaryotic transcription factor binding profiles.** *Nucleic Acids Res* 2004, **32**:D91-94.
28. Wingender E, Dietze P, Karas H, Knuppel R: **TRANSFAC: a database on transcription factors and their DNA binding sites.** *Nucleic Acids Res* 1996, **24(1)**:238-241.
29. Matys V, Fricke E, Geffers R, Gossling E, Haubrock M, Hehl R, Hornischer K, Karas D, Kel AE, Kel-Margoulis OV, *et al*: **TRANSFAC: transcriptional regulation, from patterns to profiles.** *Nucleic Acids Res* 2003, **31(1)**:374-378.
30. Lefebvre C, Lim WK, Basso K, Dalla Favera R, Califano A: **A context-specific network of protein-DNA and protein-protein interactions reveals new regulatory motifs in human B cells.** *Lecture Notes in Bioinformatics (LNCS)* 2007, **4532**:42-56.
31. Diella FCS, Gemünd C, Linding R, Via A, Kuster B, Sicheritz-Pontén T, Blom N, Gibson TJ: **Phospho.ELM: a database of experimentally verified phosphorylation sites in eukaryotic proteins.** *BMC Bioinformatics* 2004, **5**.
32. Miller ML, *et al*: **Linear motif atlas for phosphorylation-dependent signaling.** *Sci Signal* 2008, **1(35)**.
33. Gnad F, Ren S, Cox J, Olsen JV, Macek B, Oroshi M, Mann M: **PHOSIDA (phosphorylation site database): management, structural and evolutionary investigation, and prediction of phosphosites.** *Genome Biol* 2007, **8(11)**.
34. Kholodenko BN, Hancock JF, Koch W: **Signalling ballet in space and time.** *Nature Rev Molecular Cell Biology* 2010, **11**:414-426.
35. Ulrich LE, Z IB: **MiST: a microbial signal transduction database.** *Nucleic Acids Res* 2007, **35**:D386-390.
36. Krull M, Voss N, Choi C, Pistor S, Potapov A, Wingender E: **TRANSPATH: an integrated database on signal transduction and a tool for array analysis.** *Nucleic Acids Res* 2003, **31(1)**:97-100.
37. Jeong H, Tombor B, Albert R, Oltvai ZN, AL: **The large-scale organization of metabolic networks.** *Nature* 2000, **407**:651-654.
38. Feist AM, Herrgård MJ, Thiele I, Reed JL, Palsson B: **Reconstruction of biochemical networks in microorganisms.** *Nature Rev Microbiology* 2009, **7**:129-143.
39. Ma H, Mazein A, Selkov A, Selkov E, Demin O, Goryanin I: **The Edinburgh human metabolic network reconstruction and its functional analysis.** *Mol Syst Biol* 2007, **3(135)**.
40. Kanehisa M, Goto S, Furumichi M, Tanabe M, Hirakawa M: **KEGG for representation and analysis of molecular networks involving diseases and drugs.** *Nucleic Acids Res* , **38 Database:** D355-360.
41. Keseler IM, Bonavides-Martinez C, Collado-Vides J, Gama-Castro S, Gunsalus RP, Johnson DA, Krummenacker M, Nolan LM, Paley S, Paulsen IT, *et al*: **EcoCyc: a comprehensive view of Escherichia coli biology.** *Nucleic Acids Res* 2009, , **37 Database:** D464-470.
42. Karp PD, Ouzounis CA, Moore-Kochlacs C, Goldovsky L, Kaipa P, Ahren D, Tsoka S, Darzentas N, Kunin V, Lopez-Bigas N: **Expansion of the BioCyc collection of pathway/genome databases to 160 genomes.** *Nucleic Acids Res* 2005, **33(19)**:6083-6089.
43. Whitaker JW, Letunic I, McConkey GA, Westhead DR: **metaTIGER: a metabolic evolution resource.** *Nucleic Acids Res* 2009, , **37 Database:** D531-538.

44. Schilling CH, Letscher D, Palsson BO: Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *J Theor Biol* 2000, 203(3):229-248.
45. Schilling CH, Palsson BO: Assessment of the metabolic capabilities of Haemophilus influenzae Rd through a genome-scale pathway analysis. *J Theor Biol* 2000, 203(3):249-283.
46. Schilling CH, Schuster S, Palsson BO, Heinrich R: Metabolic pathway analysis: basic concepts and scientific applications in the post-genomic era. *Biotechnol Prog* 1999, 15(3):296-303.
47. Schuster S, Fell DA, Dandekar T: A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nat Biotechnol* 2000, 18(3):326-332.
48. Schuster S, Dandekar T, Fell DA: Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends Biotechnol* 1999, 17(2):53-60.
49. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, *et al*: The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 2003, 19(4):524-531.
50. Finney A, Hucka M: Systems biology markup language: Level 2 and beyond. *Biochemical Society transactions* 2003, 31(Pt 6):1472-1473.
51. Hermjakob H, Montecchi-Palazzi L, Bader G, Wojcik J, Salwinski L, Ceol A, Moore S, Orchard S, Sarkans U, von Mering C, *et al*: The HUPO PSI's molecular interaction format - a community standard for the representation of protein interaction data. *Nat Biotechnol* 2004, 22(2):177-183.
52. Murray RP, S RH: Chemical Markup, XML, and the Worldwide Web. 1. Basic Principles. *Chem Inf Comput Sci* 1999, 39:928-942.
53. Murray-Rust P, Rzepa HS, Wright M: Development of Chemical Markup Language (CML) as a System for Handling Complex Chemical Content. *New J Chem* 2001, 618-634.
54. BioPAX Working group: BioPAX-biological pathways exchange language. *Version 10 Documentation* 2004.
55. Lloyd CM, Halstead MD, Nielsen PF: CellML: its future, present and past. *Progress in biophysics and molecular biology* 2004, 85(2-3):433-450.
56. Lassila O, Swick R: Resource Description Framework (RDF) Model and Syntax Specification. *The World Wide Web Consortium (W3C) MIT, INRIA* 1999.
57. RDF vocabulary description language 1.0: RDF Schema. [http://www.w3.org/tr/2002/wd-rdf-schema-20020430/].
58. Cormen TH, Leiserson CE, Rivest Ronald L, Stein C: Introduction to algorithms. Cambridge, Massachusetts 02142: The MIT Press; 2002.
59. Huber W, Carey VJ, Long L, Falcon S, Gentleman R: Graphs in molecular biology. *BMC Bioinformatics* 2007, 8(Suppl 6):S8.
60. Lee HK, Hsu AK, Sajdak J, Qin J, Pavlidis P: Coexpression analysis of human genes across many microarray data sets. *Genome Res* 2004, 14(6):1085-1094.
61. Schulz H-J, John M, Unger A, Schumann H: Visual analysis of bipartite biological networks. *Eurographics Workshop on Visual Computing for Biomedicine* 2008.
62. Burgos E, Ceva H, Hernández L, Perazzo RPJ, Devoto M, Medan D: Two classes of bipartite networks: nested biological and social systems. *Phys Rev* 2008, 78.
63. Picard F, Miele V, Daudin J-J, Cottret L, Robin S: Deciphering the connectivity structure of biological networks using MixNet. *BMC Bioinformatics* 2009, 10.
64. Leclerc RD: Survival of the sparsest: robust gene networks are parsimonious. *Mol Syst Biol* 2008, 4:213.
65. Dijkstra EW: A note on two problems in connexion with graphs. *Numerische Mathematik* 1959, 1:269-271.
66. Floyd RW: Algorithm 97. *Comm ACM* 1962, 5-6:345.
67. Bron C, Kerbosch J: Algorithm 457: finding all cliques of an undirected graph. *Commun ACM (ACM)* 1973, 16(9):575-577.
68. Zhang H, Song X, Wang H, Zhang X: MIClique: An Algorithm to Identify Differentially Coexpressed Disease Gene Subset from Microarray Data. *Journal of Biomedicine and Biotechnology* 2009.
69. Voy BH, Scharff JA, Perkins AD, Saxton AM, Borate B, Chesler EJ, Branstetter LK, Langston MA: Extracting Gene Networks for Low-Dose Radiation Using Graph Theoretical Algorithms. *PLoS Comput Biol* 2006, 2(7).
70. Manfield IW, Jen CH, Pinney JW, Michalopoulos I, Bradford JR, Gilmartin PM, Westhead DR: Arabidopsis Co-expression Tool (ACT): web server tools for microarray-based gene expression analysis. *Nucleic Acids Res* 2006, , 34 Web Server: W504-509.
71. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, *et al*: Bioconductor: open software development for computational biology and bioinformatics. *Genome biology* 2004, 5(10):R80.
72. Ravasz E, Somera A, Mongru D, Oltvai Z, Barabási A-L: Hierarchical organization of modularity in metabolic networks. *Science* 2002, 297:1551-1555.
73. Barabási A-L, Gulbahce N, Loscalzo J: Network medicine: a network-based approach to human disease. *Nature Reviews Genetics* 2011, 12:56-68.
74. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U: Network motifs: simple building blocks of complex networks. *Science* 2002, 298(5594):824-827.
75. Shen-Orr S, Milo R, Mangan S, Alon U: Network motifs in the transcriptional regulation network of Escherichia coli. *Nat Genet* 2002, 31:64-68.
76. Ingram PJ, Stumpf MP, Stark J: Network motifs: structure does not determine function. *BMC Genomics* 2006, 7:108.
77. Zotenko E, Mestre J, O'Leary DP, Przytycka TM: Why do hubs in the yeast protein interaction network tend to be essential: re-examining the connection between the network topology and essentiality. *PLoS Comput Biol* 2008, 4:1-16.
78. Levy SF, S ML: Network hubs buffer environmental variation in Saccharomyces cerevisiae. *PLoS Biol* 2008, 6(11).
79. Ma H-W, Z A-P: The connectivity structure, giant strong component and centrality of metabolic networks. *Bioinformatics* 2003, 19(11).
80. Mazurie A, Bonchev D, Schwikowski B, Buck GA: Evolution of metabolic network organization. *BMC Syst Bio* 2010, 4.

81. da Silva MR, Ma H, Zeng A-P: **Centrality, Network Capacity, and Modularity as Parameters to Analyze the Core-Periphery Structure in Metabolic Networks.** *Proceedings of the IEEE* 2008, **96(8)**:1411-1420.
82. Rong ZHL, X Lu, W L: **Pinning a complex network through the betweenness centrality strategy.** *Circuits and Systems IEEE International Symposium* 2009, 1689-1692.
83. Kitsak M, Havlin S, Paul G, Riccaboni M, Pammolli F, Stanley HE: **Betweenness centrality of fractal and nonfractal scale-free model networks and tests on real networks.** *Phys Rev E* 2007, **75.**
84. Joy MP, Brock A, Ingber DE, Huang S: **High-Betweenness Proteins in the Yeast Protein Interaction Network.** *J Biomed Biotechnol* 2005, **2**:96-103.
85. Paladugu SR, Zhao S, Ray A, Raval A: **Mining protein networks for synthetic genetic interactions.** *BMC Bioinformatics* 2008, **9.**
86. Özgür A, Vu T, Erkan G, Radev DR: **Identifying gene-disease associations using centrality on a literature mined gene-interaction network.** *Bioinformatics* 2008, **24(13)**:i277-i285.
87. Chavali S, Barrenas F, Kanduri K, Benson M: **Network properties of human disease genes with pleiotropic effects.** *BMC Syst Bio* 2010, **4.**
88. Estrada E: **Characterization of the folding degree of proteins.** *Bioinformatics* 2002, **18**:697-704.
89. Estrada E, Uriarte E: **Recent advances on the role of topological indices in drug discovery research.** *Curr Med Chem* 2001, **8**:1699-1714.
90. Estrada E: **Generalized walks-based centrality measures for complex biological networks.** *J Theor Biol* 2010, **263(4)**:556-565.
91. Nisbach F, K M: **Developmental time windows for spatial growth generate multiple-cluster small-world networks.** *Eur Phys J B* 2007, **58**:185-191.
92. Costa LdF, Kaiser M, Hilgetag CC: **Predicting the connectivity of primate cortical networks from topological and spatial node properties.** *BMC Syst Bio* 2007, **1.**
93. Jeong H, Mason SP, Barabasi A-L, Oltvai ZN: **Lethality and centrality in protein networks.** *Nature* 2001, **411(6833)**:41-42.
94. Hahn M, K A: **Comparative genomics of centrality and essentiality in three eukaryotic protein-protein interaction networks.** *Mol Biol Evol* 2005, **22**:803-806.
95. Koschützki D, S F: **Comparison of Centralities for Biological Networks.** *Proc German Conf Bioinformatics (GCB'04)* 2004, **P-53 of LNI.**
96. Junker BH, Koschützki D, Schreiber F: **Exploration of biological network centralities with CentiBiN.** *BMC Bioinformatics* 2006, **7.**
97. Baur M, Benkert M, Brandes U, Cornelsen S, Gaertler M, Köpf B, Lerner J, Wagner D: **visone - Software for Visual Social Network Analysis.** *Proc 9th Intl Symp Graph Drawing (GD '01), LNCS* 2002, **2265**:463-464.
98. Batagelj V, Mrvar A: **Pajek - Program for Large Network Analysis.** *Connections* 1998, **21**:47-57.
99. Hu Z, Mellor J, Wu J, Yamada T, Holloway D, DeLisi C: **VisANT: data-integrating visual framework for biological networks and modules.** *Nucleic Acids Res* 2005, **33**:W352-W357.
100. Albert R: **Scale-free networks in cell biology.** *Journal of Cell Science* 2005, **118.**
101. Lima-Mendez G, van Helden J: **The powerful law of the power law and other myths in network biology.** *Mol Biosyst* 2009, **5(12)**:1482-1493.
102. Newman MEJ: **Assortative Mixing in Networks.** *Phys Rev Lett* 2002, **89(208701)**.
103. Newman MEJ: **Mixing patterns in networks.** *Phys Rev* 2003, **67.**
104. Redner S: **Networks: teasing out the missing links.** *Nature* 2008, **453**:47-48.
105. Erdös P, R A: **On the strength of connectedness of a random graph.** *Acta Math Acad Sci Hungar* 1961, **12**:261-267.
106. Watts DJ, S SH: **Collective dynamics of 'small-world' networks.** *Nature* 1998, **393**:440-442.
107. Barabási A-L, A R: **Emergence of scaling in random networks.** *Science* 1999, **286**:509-512.
108. Berg J, Lassig M, Wagner A: **Structure and evolution of protein interaction networks: a statistical model for link dynamics and gene duplications.** *BMC Evol Biol* 2004, **4(1)**:51.
109. Yamada T, B P: **Evolution of biomolecular networks - lessons from metabolic and protein interactions.** *Nature Rev Molecular Cell Biology* 2009, **10**:791-803.
110. Jain AK, Murty MN, Flynn PJ: **Data Clustering: A Review.** *ACM Computing Surveys (CSUR)* 1999, **31(3)**:264-323.
111. Duda RO, Hart PE, Stork DG: **Pattern Classification, ch.10: Unsupervised learning and clustering.** *Wiley, New York* 2001, 571.
112. Saitou N, Nei M: **The neighbor-joining method: a new method for reconstructing phylogenetic trees.** *Mol Biol Evol* 1987, **4(4)**:406-425.
113. Borate BR, Chesler EJ, Langston MA, Saxton AM, Voy BH: **Comparison of threshold selection methods for microarray gene co-expression matrices.** *BMC Res Notes* 2009, **2(240)**.
114. Perkins AD, L MA: **Threshold selection in gene co-expression networks using spectral graph theory techniques.** *BMC Bioinformatics* 2009, **10.**
115. Quackenbush J: **Computational genetics: Computational analysis of microarray data.** *Nat Rev Genetics* 2001, **2**:418-427.
116. Milligan WGlenn, Cooper MC: **Methodology Review: Clustering Methods.** *Applied Psychological Measurement* 1987, **11(4)**:329-354.
117. Sneath PHA, Sokal RR: **Unweighted Pair Group Method with Arithmetic Mean.** *Numerical Taxonomy* San Francisco: Freeman; 1973, 230-234.
118. Michener CD, Sokal RR: **A Quantitative Approach to a Problem in Classification.** *Evolution* 1957, **11(2)**:130-162.
119. Gascuel O, Steel M: **Neighbor-joining revealed.** *Mol Biol Evol* 2006, **23(11)**:1997-2000.
120. D'andrade R: **U-Statistic Hierarchical Clustering.** *Psychometrika* 1978, **4**:58-67.
121. Johnson SC: **Hierarchical Clustering Schemes.** *Psychometrika* 1967, **2**:241-254.
122. Seo J, Shneiderman B: **Interactively Exploring Hierarchical Clustering Results.** *Computer* 2002, **35(7)**:80-86.
123. Seo J, Gordish-Dressman H, Hoffman EP: **An interactive power analysis tool for microarray hypothesis testing and generation.** *Bioinformatics* 2006, **22(7)**:808-814.

124. Kumar S, Tamura K, Nei M: **MEGA3: Integrated software for Molecular Evolutionary Genetics Analysis and sequence alignment.** *Brief Bioinform* 2004, **5(2)**:150-163.
125. Tamura K, J D, Nei M, S K: **MEGA4: Molecular Evolutionary Genetics Analysis (MEGA) software version 4.0.** *Molecular Biology and Evolution* 2007, **24**:1596-1599.
126. Kumar S, Tamura K, Jakobsen I, Nei M: **MEGA2: molecular evolutionary genetics analysis software.** *Bioinformatics* 2001, **17(12)**:1244-1245.
127. Kumar S, Tamura K, Nei M: **MEGA: Molecular Evolutionary Genetics Analysis software for microcomputers.** *Comput Appl Biosci* 1994, **10(2)**:189-191.
128. Saeed AI, Sharov V, White J, Li J, Liang W, Bhagabati N, Braisted J, Klapa M, Currier T, Thiagarajan M, *et al*: **TM4: a free, open-source system for microarray data management and analysis.** *BioTechniques* 2003, **34(2)**:374-378.
129. Pavlopoulos GA, Soldatos TG, Barbosa-Silva A, Schneider R: **A reference guide for tree analysis and visualization.** *BioData Min* 2010, **3(1)**:1.
130. Enright AJ, Van Dongen S, Ouzounis CA: **An efficient algorithm for large-scale detection of protein families.** *Nucleic Acids Res* 2002, **30(7)**:1575-1584.
131. Moschopoulos CN, Pavlopoulos GA, Schneider R, Likothanassis SD, Kossida S: **GIBA: a clustering tool for detecting protein complexes.** *BMC Bioinformatics* 2009, **10(Suppl 6)**:S11.
132. Gao L, Sun PG, Song J: **Clustering algorithms for detecting functional modules in protein interaction networks.** *J Bioinform Comput Biol* 2009, **7(1)**:217-242.
133. Zhong W, Altun G, Harrison R, Tai PC, Pan Y: **Improved K-means clustering algorithm for exploring local protein sequence motifs representing common structural property.** *IEEE Trans Nanobioscience* 2005, **4(3)**:255-265.
134. van Dogen S: **Graph Clustering by Flow Simulation.** *PhD thesis* University of Utrecht; 2000.
135. Vlasblom J, Wodak SJ: **Markov clustering versus affinity propagation for the partitioning of protein interaction graphs.** *BMC Bioinformatics* 2009, **10**:99.
136. Enright AJ, Van Dongen S, Ouzounis CA: **An efficient algorithm for large-scale detection of protein families.** *Nucleic Acids Res* 2002, **30(7)**:1575-1584.
137. MacQueen B: **Some Methods for classification and Analysis of Multivariate Observations.** In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. Volume 1.* Berkeley, University of California Press; 1967:281-297.
138. Lu Y, Lu S, Fotouhi F, Deng Y, Brown SJ: **Incremental genetic K-means algorithm and its application in gene expression data analysis.** *BMC Bioinformatics* 2004, **5**:172.
139. Frey BJ, Dueck D: **Clustering by passing messages between data points.** *Science* 2007, **315(5814)**:972-976.
140. King AD, Przulj N, Jurisica I: **Protein complex prediction via cost-based clustering.** *Bioinformatics* 2004, **20(17)**:3013-3020.
141. Paccanaro A, Casbon JA, Saqi MA: **Spectral clustering of protein sequences.** *Nucleic Acids Res* 2006, **34(5)**:1571-1580.
142. Li X, Wu M, Kwoh CK, Ng SK: **Computational approaches for detecting protein complexes from protein interaction networks: a survey.** *BMC Genomics* 2010, **11(Suppl 1)**:S3.
143. Brohee S, van Helden J: **Evaluation of clustering algorithms for protein-protein interaction networks.** *BMC Bioinformatics* 2006, **7**:488.
144. Pavlopoulos GA, Wegener AL, Schneider R: **A survey of visualization tools for biological network analysis.** *BioData Min* 2008, **1**:12.
145. Brohee S, Faust K, Lima-Mendez G, Sand O, Janky R, Vanderstocken G, Deville Y, van Helden J: **NeAT: a toolbox for the analysis of biological networks, clusters, classes and pathways.** *Nucleic Acids Res* 2008, , **36 Web Server:** W444-451.
146. Pavlopoulos GA, Moschopoulos CN, Hooper SD, Schneider R, Kossida S: **jClust: a clustering and visualization toolbox.** *Bioinformatics* 2009, **25(15)**:1994-1996.
147. Yoshida R, Higuchi T, Imoto S, Miyano S: **ArrayCluster: an analytic tool for clustering, data visualization and module finder on gene expression profiles.** *Bioinformatics* 2006, **22**:1538-1539.
148. Hooper SD, Bork P: **Medusa: a simple tool for interaction graph analysis.** *Bioinformatics* 2005, **21(24)**:4432-4433.
149. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T: **Cytoscape: a software environment for integrated models of biomolecular interaction networks.** *Genome Res* 2003, **13(11)**:2498-2504.
150. Pavlopoulos GA, O'Donoghue SI, Satagopam VP, Soldatos TG, Pafilis E, Schneider R: **Arena3D: visualization of biological networks in 3D.** *BMC systems biology* 2008, **2**:104.
151. Uetz P, Giot L, Cagney G, Mansfield TA, Judson RS, Knight JR, Lockshon D, Narayan V, Srinivasan M, Pochart P, *et al*: **A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae.** *Nature* 2000, **403(6770)**:623-627.
152. Rain JC, Selig L, De Reuse H, Battaglia V, Reverdy C, Simon S, Lenzen G, Petel F, Wojcik J, Schachter V, *et al*: **The protein-protein interaction map of Helicobacter pylori.** *Nature* 2001, **409(6817)**:211-215.
153. Giot L, Bader JS, Brouwer C, Chaudhuri A, Kuang B, Li Y, Hao YL, Ooi CE, Godwin B, Vitols E, *et al*: **A protein interaction map of Drosophila melanogaster.** *Science* 2003, **302(5651)**:1727-1736.
154. Li S, Armstrong CM, Bertin N, Ge H, Milstein S, Boxem M, Vidalain PO, Han JD, Chesneau A, Hao T, *et al*: **A map of the interactome network of the metazoan C. elegans.** *Science* 2004, **303(5657)**:540-543.
155. von Mering C, Krause R, Snel B, Cornell M, Oliver SG, Fields S, Bork P: **Comparative assessment of large-scale data sets of protein-protein interactions.** *Nature* 2002, **417(6887)**:399-403.
156. Raman K: **Construction and analysis of protein-protein interaction networks.** *Autom Exp* 2010, **2(1)**:2.
157. Salgado H, Santos-Zavaleta A, Gama-Castro S, Peralta-Gil M, Penaloza-Spinola MI, Martinez-Antonio A, Karp PD, Collado-Vides J: **The comprehensive updated regulatory network of Escherichia coli K-12.** *BMC Bioinformatics* 2006, **7**:5.
158. Salgado H, Gama-Castro S, Peralta-Gil M, Diaz-Peredo E, Sanchez-Solano F, Santos-Zavaleta A, Martinez-Flores I, Jimenez-Jacinto V, Bonavides-Martinez C, Segura-Salazar J, *et al*: **RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions.** *Nucleic Acids Res* 2006, , **34 Database:** D394-397.
159. Lozada-Chavez I, Janga SC, Collado-Vides J: **Bacterial regulatory networks are extremely flexible in evolution.** *Nucleic Acids Res* 2006, **34(12)**:3434-3445.
160. Madan Babu M, Teichmann SA, Aravind L: **Evolutionary dynamics of prokaryotic transcriptional regulatory networks.** *J Mol Biol* 2006, **358(2)**:614-633.

161. Sneppen Kim, Zocchi G: **Physics in Molecular Biology.** Giovanni Zocchi; 2005.
162. van Nimwegen E: **Scaling laws in the functional content of genomes.** *Trends Genet* 2003, **19(9)**:479-484.
163. Bhalla US, Iyengar R: **Emergent properties of networks of biological signaling pathways.** *Science* 1999, **283(5400)**:381-387.
164. Junker HBjörn, Schreiber F: **Analysis of Biological Networks.** 2008.
165. Guelzim N, Bottani S, Bourgine P, Kepes F: **Topological and causal structure of the yeast transcriptional regulatory network.** *Nat Genet* 2002, **31(1)**:60-63.
166. Ma H, Zeng AP: **Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms.** *Bioinformatics* 2003, **19(2)**:270-277.
167. Jeong H, Tombor B, Albert R, Oltvai ZN, Barabási A-L: **The large-scale organization of metabolic networks.** *Nature* 2000, **407(6804)**:651-654.
168. Gagneur J, Jackson DB, Casari G: **Hierarchical analysis of dependency in metabolic networks.** *Bioinformatics* 2003, **19(8)**:1027-1034.
169. Holme P, Huss M, Jeong H: **Subnetwork hierarchies of biochemical pathways.** *Bioinformatics* 2003, **19(4)**:532-538.

doi:10.1186/1756-0381-4-10
Cite this article as: Pavlopoulos *et al.*: Using graph theory to analyze biological networks. *BioData Mining* 2011 **4**:10.