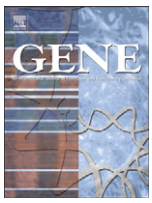
Gene 518 (2013) 42–51

Contents lists available at SciVerse ScienceDirect

Gene



Spotlight: Assembly of protein complexes by integrating graph clustering methods

Chia-Hao Chin [a,1], Shu-Hwa Chen [a,b,1], Chun-Yu Chen [c], Chao A. Hsiung [c], Chin-Wen Ho [d], Ming-Tat Ko [a,*], Chung-Yen Lin [a,c,e,f,**]

[a] Institute of Information Science, Academia Sinica, No. 128 Yan-Chiu-Yuan Rd., Sec. 2, Taipei 115, Taiwan
[b] Computational Biology Research Center, National Institute of Advanced Industrial Science and Technology (AIST), 2-41-6, Aomi, Koto-ku, Tokyo 135-0064, Japan
[c] Division of Biostatistics and Bioinformatics, Institute of Population Health Sciences, National Health Research Institutes. No. 35 Keyan Rd. Zhunan, Miaoli County 350, Taiwan
[d] Department of Computer Science and Information Engineering, National Central University, No.300, Jung-da Rd, Chung-li, Tao-yuan 320, Taiwan
[e] Institute of Fisheries Science, College of Life Science, National Taiwan University, No. 1, Roosevelt Rd. Sec 4, Taipei, Taiwan
[f] Research Center of Information Technology Innovation, Academia Sinica, No. 128 Yan-Chiu-Yuan Rd., Sec. 2, Taipei 115, Taiwan

ARTICLE INFO

Available online 26 December 2012

Keywords:
Network biology
Topology
Protein complex
Algorithm

ABSTRACT

As is generally assumed, clusters in protein–protein interaction (PPI) networks perform specific, crucial functions in biological systems. Various network community detection methods have been developed to exploit PPI networks in order to identify protein complexes and functional modules. Due to the potential role of various regulatory modes in biological networks, a single method may just apply a single graph property and neglect communities highlighted by other network properties.

This work presents a novel integration method to capture protein modules/protein complexes by multiple network features detected by different algorithms. The integration method is further implemented in a web-based platform with a highly effective interactive network analyzer. Conventionally adopted methods with different perspectives on network community detection (e.g., CPM, FastGreedy, HUNTER, MCL, LE, SpinGlass, and WalkTrap) are also executed simultaneously.

Analytical results indicate that the proposed method performs better than the conventional ones. The proposed approach can capture the transcription and RNA splicing machineries from the yeast protein network. Meanwhile, proteins that are highly associated with each other, yet not described in both machineries are also identified. In sum, a protein that is closely connected to components of a known module or a complex in the network view implies the functional association among them. Importantly, our method can detect these unique network features, thus facilitating efforts to discover unknown components of functional modules/protein complexes.

Availability: *Spotlight* is freely accessible at http://hub.iis.sinica.edu.tw/spotlight. Video clips for a quick view of usage are available in the website online help page.

1. Introduction

Elucidating protein complexes and functional modules is essential for understanding genome functions. A protein complex comprises a small set of proteins that are closely associated with each other, and also present in the same scenario. Meanwhile, as a group of proteins, a functional module participates in a specific process, while each binding event may occur in the same or different time and place (Spirin and Mirny, 2003). Several protein–protein interaction databases have emerged with the advent of high-throughput technologies such as yeast two-hybrid assays and affinity purification along with tandem mass spectrometry. Previous studies analyzed the graph topology of protein–protein interaction (PPI) networks, in which the proteins are denoted as nodes and pairwise interactions are denoted as linking edges. According to their results, protein complexes and functional modules tend to be densely connected to each other while having fewer connections to the other proteins in a network (Barabási and Oltvai, 2004; Rives and Galitski, 2003; Spirin and Mirny, 2003). Above observations also imply the rationale to identify protein complexes and functional modules by detecting communities/clusters from a high coverage PPI network.

Among the various community structure detection (or graph clustering) methods applied to the PPI network to detect protein complexes and functional modules include random walk based

Abbreviations: CPM, Clique percolation method; *CS*, community score function; *CSS*, Consensus method; DAVID, Database for Annotation, Visualization and Integrated Discovery (DAVID); *E*, edge set; FN, False negative; FP, False positives; *G*, giving graph; GO, gene ontology; GUI, Graphical User Interface; *IC*, integrated clusters; *KC*, Known complex; LE, Leading eigenvector; Lsm, Like Sm; MCL, Markov cluster; MIPS, Munich Information Center for Protein Sequences; *PC*, Predicted complex; PPI, Protein–protein Interaction; PSI-MI, Proteomics Standards Initiative, Molecular Interaction; *S*, cluster; SGD, Saccharomyces Genome Database; Sm, a family of RNA-binding proteins; snRPNs, Small nuclear ribonucleoproteins; *TC*, total clusters; TP, true positives; *V*, vertex; *W*, edge weight.

* Corresponding author.
** Correspondence to: C.Y. Lin, Institute of Information Science, Academia Sinica, No. 128 Yan-Chiu-Yuan Rd., Sec. 2, Taipei 115, Taiwan.
E-mail addresses: mtko@iis.sinica.edu.tw (M.-T. Ko), cylin@iis.sinica.edu.tw (C.-Y. Lin).
[1] These authors contributed equally to this work.

0378-1119/$ – see front matter.
http://dx.doi.org/10.1016/j.gene.2012.11.087

methods (Enright et al., 2002; Pons and Latapy, 2005; van Dongen, 2000), edge betweenness-based methods (Dunn et al., 2005; Girvan and Newman, 2002; Luo et al., 2007), clique percolation methods (Adamcsek et al., 2006; Zhang et al., 2006), and core-attachment based methods (Chin et al., 2010; Leung et al., 2009; Liu et al., 2009; Wu et al., 2009). While relying on widely divergent approaches, these methods have their own unique strengths and limitations. Additionally, while various regulatory modes are presented in biological networks, a single method may just encompass a single graph property and disregards communities that may be highlighted by other network properties. Bench researchers have difficulty in justifying the applicability of various algorithms on their interesting targets. Despite the development of some consensus clustering methods to solve this problem (Asur et al., 2007; Lancichinetti and Fortunato, 2012; Zhang et al., 2009), such approaches failed to include overlapping community detection methods while attempting to integrate the partition methods that divide the entire graph into smaller subgraphs and assign each node to one cluster. Therefore, this work presents a novel integration method, capable of grabbing network community structures from the input protein network. The proposed clustering approach, in which graphs are integrated, performs superior to other conventionally adopted methods in terms of protein complex harvesting and gene ontology (GO) term enrichment. Moreover, the proposed integration method allows for the successful retrieval of the transcription machineries from the yeast protein network, as well as those proteins that are closely related to the transcription process yet are not included in the complex.

The proposed integrated graph clustering method is implemented into a web-based protein complex detection scheme with an interactive network analyzer called Spotlight. With an intuitive, zoomable graphical interface, Spotlight displays the PPI network clustering results with rich and updated annotations of proteins and their linking edges (i.e. the interactions) if the input PPIs are described by standard UniProt ID or yeast SGD IDs. For user convenience, the proposed approach includes other conventionally adopted network clustering methods (e.g., CPM (Palla et al., 2005), FastGreedy (Clauset et al., 2004), HUNTER (Chin et al., 2010), LE (Newman, 2006), MCL (van Dongen, 2000), SpinGlass (Reichardt and Bornholdt, 2006), and WalkTrap (Pons and Latapy, 2005)) in Spotlight platform to easily perform network analysis, as well as view/export, and link results to further functional analysis processes. The Spotlight-based integrating graph clustering method outperforms other network clustering methods by exploiting unique network properties that imply the functional association among proteins. Importantly, the proposed method facilitates research efforts to discover unknown functional modules/protein complex structures as well as novel complex components/regulators components from a PPI network.

As graph clustering is a variant of data clustering, related methods differed mainly in the similarity of the objects handled. Restated, in data clustering, the similarity of any two objects of the input data is well defined; meanwhile, in graph clustering, the similarity of objects is expressed by edges of an input graph. Data clustering can be classified into hard data clustering and soft data clustering. In hard clustering, an object belongs to exactly one cluster; meanwhile, in soft clustering, an object is assigned to multiple clusters with membership weights that are equivalent to one. In contrast to data clustering, graph clustering is classified into overlapping graph and non-overlapping graph. Similar to hard data clustering, an object in a non-overlapping graph clustering outcome belongs to exactly one cluster. Unlike soft data clustering, an object in an overlapping clustering result is assigned to multiple clusters with weighted ones respectively. In contrast to conventionally adopted data clustering methods, consensus clustering (also called ensemble clustering or median partitioning) attempts to integrate multiple data clustering results in order to obtain better results. Consensus clustering is based on the premise of majority rule. Restated, a consensus clustering outcome is, on average, most similar to all of the input clustering results. Therefore, consensus clustering performs poorly when the integrated clustering results significantly differ from each other. To avoid this problem, the proposed method adopts the elitist strategy, in which good clusters are chosen from all clustering results and merged together.

2. Methods

The proposed integrating graph clustering approach attempts to identify good clusters with multiple network features of a PPI network. Therefore, a measure must be designed for qualifying the clustering results concluded from different methods. This section describes the community score function for evaluating the cluster quality. The integration method is introduced as well.

2.1. Community score function

Based on the definition of weak community (Radicchi et al., 2004), Lázár et. al. proposed a measure shown as Formula (1) to judge the quality of a cluster S in a graph G (Lázár et al., 2010).

$$L(S) = \frac{1}{|S|} \times \frac{|E(G[S])|}{\binom{|S|}{2}} \sum_{i \in S} \frac{k_i^{in}(S) - k_i^{out}(S)}{d_i \times s_i}, \quad (1)$$

where $|S|$ denotes the cardinality of S; $|E(G[S])|$ represents the number of edges in the subgraph induced by S; $k_i^{in}(S)$ is the number of neighbors of a vertex i, which are also in S; in contrast, $k_i^{out}(S)$ denotes the number of neighbors of i, which are not in S; d_i represents the number of neighbors of i and s_i is the number of clusters containing i; and the ratio of $|E(G[S])|$ and $\binom{|S|}{2}$ represents the edge density of the cluster. Therefore, the fact that $L(S)$ is higher suggests that the quality of cluster S should be better. Another assumption of this measure is that the number of inward going edges (i.e. $k_i^{in}(S)$) should be greater than that of outward going edges. However, this measure is inappropriate for those clusters containing vertices in high degree. To explain this situation, Fig. 1 describes a simple example. The network shown in Fig. 1 contains four clusters. For cluster A, although the number of outward going edges of vertex v is significantly greater than that of inward going edges of vertex v, cluster A is viewed here as a cluster from this network because the outward going edges are not across different clusters. We believe that the quality of cluster A should be better than that of cluster D because the members of cluster A are more closely associated with each other than that of cluster D. However, according to Formula (1), $L(A)$ 0.6$<$$L(D)$ 0.73, which contradicts our intuition. To solve this problem, this work proposes the measure (i.e. the community score function) to help us select good clusters from the integrated clustering results.

Our measure is based on two observations of Watts and Strogatz: (1) PPI networks have a small average shortest path between two proteins; (2) the clustering coefficient is significantly higher than would be expected under a random selection (Watts and Strogatz, 1998). Our measure is introduced formally by using graph terms hereinafter to describe it. A vertex and an edge denote a protein and an interaction between two proteins of a PPI network. For an undirected graph G, let $G = (V, E, w)$, where V is a vertex set; E represents an edge set; and w refers to an edge weight function. For a cluster $S \subset V$, a vertex-induced subgraph $G[S]$ is S together with any edge whose endpoints are both in S. Here, the number of closed, three-step walk paths is used to describe neighboring condition of a cluster, along with the average length of shortest paths used to describe the compactness of a cluster. The community score function $CS(S)$ is defined as

$$CS(S) = \frac{\text{the number of closed walks whose step is three in G[S]}}{\text{the average shortest path lenght in G[S]}}.$$

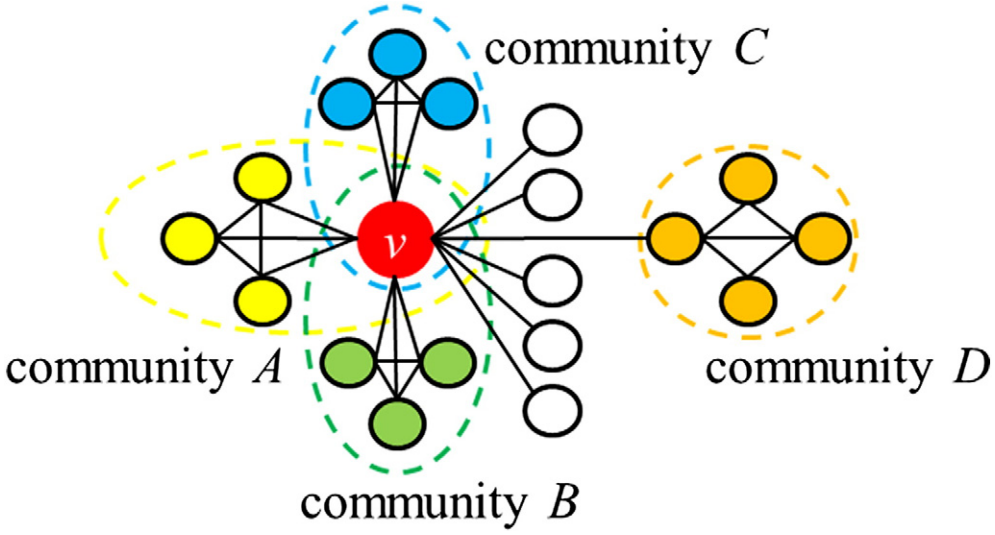


Fig. 1. Toy example that illustrates the weakness of previous cluster measures. Although this network contains four clusters, most cluster measures them as good clusters.

As is expected, proteins in a cluster have more neighbors. Also, a cluster should be compact, that means a shorter distance between two proteins in a cluster implies a higher community score.

2.2. An integration framework for graph clustering

The illustration and pseudo code in Fig. 2 provide an overview of the proposed integration method. Briefly, all clusters produced by various clustering methods are accumulated in the list total clusters (*TC*) and sorted in a descending order with the community score. For each cluster *S* in *TC*, it is relocated to the set integrated clusters (*IC*) when the cardinality of *S* does not exceed threshold α, $CS(S)>0$ (i.e. a protein with at least two neighbors that are connected directly). And the overlapping of *S* to any present clusters in the *IC* does not exceed threshold β.

3. The implementation and usage of the spotlight web platform

3.1. Clustering methods in spotlight

Seven other methods are used in Spotlight, as described in the following. Clique percolation method (CPM) is a conventional means of detecting overlapping community structures in networks. A cluster computed by CPM is the maximal union of *k*-cliques that can be linked through a series of adjacent *k*-cliques, in which a *k*-clique is a fully connected sub-graph of *k* nodes. Notably, two *k*-cliques are considered adjacent if they share *k*-1 nodes. Hence, when the value of parameter *k* of CPM increases, although clusters become denser, coverage of clusters may decrease. A previous study suggested a *k* value between 4 and 6 (Palla et al., 2005); the value of *k* for CPM in Spotlight is 4 if it is applicable.

The Markov Cluster (MCL) algorithm calculates the successive powers of the associated Markov matrix to simulate the flow (i.e. random walks) within a graph. The expansion and inflation of a flow are alternately simulated until an equilibrium state is reached. MCL is an efficient and scalable clustering method. However, the value of the inflation parameter significantly influences the number of clusters (Brohée and van Helden, 2006). The default inflation parameter (i.e. 2) is used in Spotlight. Meanwhile, the subgraphs induced by a cluster computed by MCL may not be connected.

As a parameter-free clustering method (Chin et al., 2010), HUNTER generates a module seed from a vertex and, then, the seed grows gradually by adding vertices that are strongly connected to it. The method further merges any two grown modules with common vertices above a threshold iteratively, and finally determines the output clusters.

Additionally, this work implemented FastGreedy, leading eigenvector (LE), SpinGlass and WalkTrap based on the Igraph package (Csárdi and Nepusz, 2006). FastGreedy utilizes a benefit function to detect communities in the network by Newman's modularity (Newman, 2004); LE detects a community's structure by estimating the leading non-negative eigenvector of the modularity matrix. SpinGlass detects communities via a spin-glass model with simulated annealing. WalkTrap is an agglomerative clustering method that uses a random walk to detect the dense part a network. This work performs these methods in Spotlight with their own default setting.

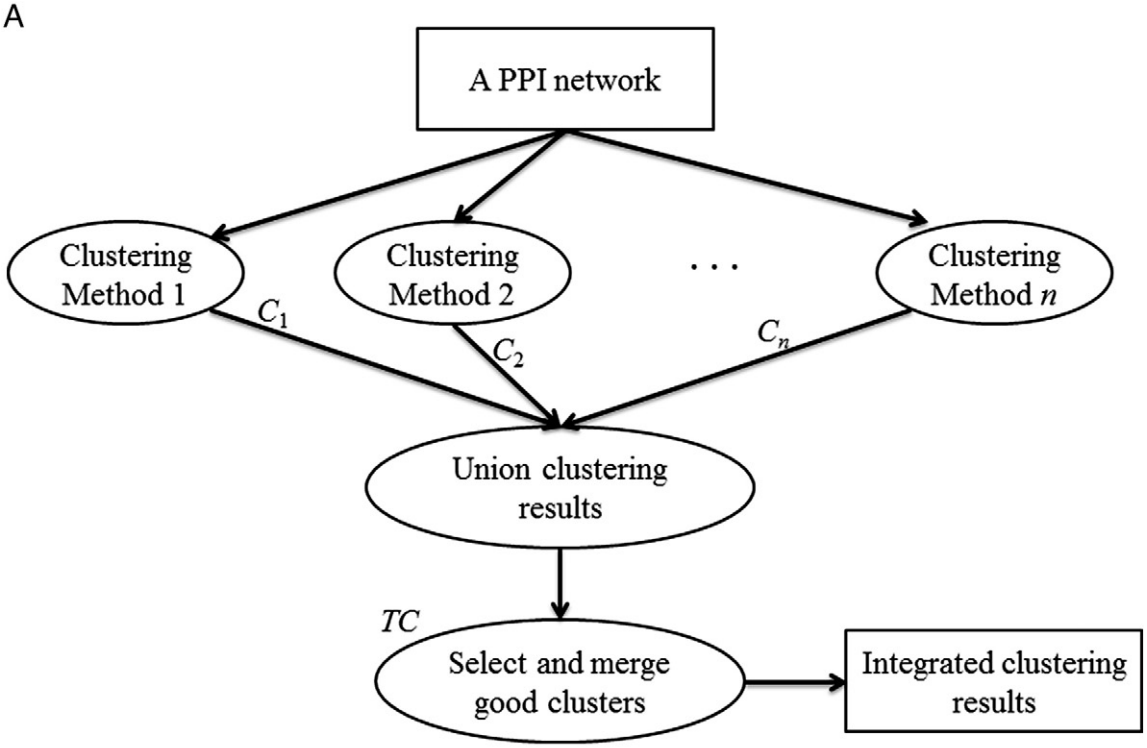
3.2. Implementation of spotlight

Spotlight (http://hub.iis.sinica.edu.tw/spotlight) platform's architecture is in three tiers: client/server/databases. In the client part, the visualization interface of the network clustering results is a Java web applet based on Piccolo (Bederson et al., 2004) and Jung 2.0 (http://jung.sourceforge.net). The server tier is constructed on an open-source structure of Linux Ubuntu (ver.9.10), Apache-Tomcat (web server), PHP and JSP (html-embedded scripting languages), PostgreSQL (a relational database), GO-TermFinder (GO enrichment analysis) (Boyle et al., 2004), and XMLMakerFlattener (converting data format) (Hermjakob et al., 2004). The database tier provides the annotation of nodes and edges on Spotlight visualization interface, as well as the retrieval demands of clustering results. A relational database is constructed to incorporate data sources from several generalized and specialized databases, including UniProt, KEGG, GO, and InterPro (Hunter et al., 2012), when they are routinely updated. The database (serves OR functions) is the annotation pool for the input network if the input PPI sets are in standard Uniprot IDs or yeast SGD IDs. Fig. 2B describes the algorithm pseudo code of integrating graph clustering method. The thresholds α and β in our method are 150 and 0.5, respectively.

3.3. The usage of the spotlight web platform

Spotlight takes three protein interaction data formats: the standard PSI-MI XML format, tab-delimited text file weight values, and

A



B

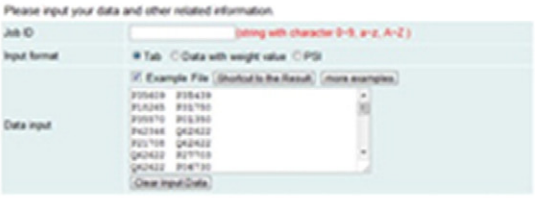
Integration Graph Clustering Method

Input:
$G=(V, E, w)$ is an undirected PPI network;
k = the number of clustering methods;
C_i ={S: S is a cluster computed by the i-th clustering method}
α= cluster size threshold (α is an integer);
β= cluster merge threshold (0<β<1).
Output:
IC is an integrated clustering result;

Description:
$IC = \varnothing$, and $TC = C_1 \cup C_2 \cup \ldots \cup C_k$.
Sort clusters in TC descending by community scores
for all $S \in TC$ **do**
  if $|S| < \alpha$ **and** $CS(S) > 0$ **then**
    if $\forall T \in IC$ *s.t.* $|S \cap T| \leq \beta \times \min(|S|, |T|)$ **then**
      $IC = IC \cup \{S\}$
Output IC

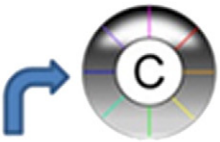
Fig. 2. (A) Overview of the integration graph clustering method. Notably, the clusters in *TC* are sorted in a descending order according to their community score. (B) The pseudocode of method.

tab-delimited text file without weight value. A notification mail with clustering result retrieval link is sent to a job submitter when the calculation is completed. Fig. 3 illustrates an example of the Spotlight result. In default, the top ten clusters from the integration method appear in the GUI java applet, followed by a summary of clustering results. The right panel of the GUI provides various information tabs, including a summary of the mouse-click selected cluster, annotations of the selected proteins in the expended cluster, or the information of a selected linking edge in the canvas panel. Double clicks on a cluster node expand the cluster to view the composite PPI subnetwork. The menu bar and tool bar inlet provide advance functions such as filtering out low *k*-value nodes, adjusting the graph layout

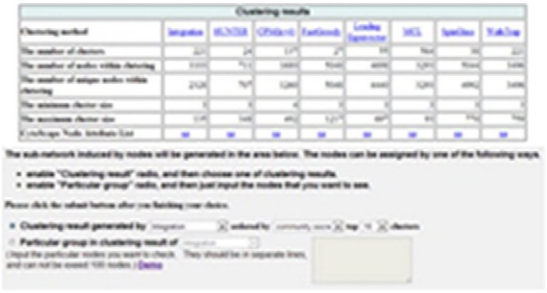


Input Interactome in PSI/ Tab / Weighted Data

Default



Refine

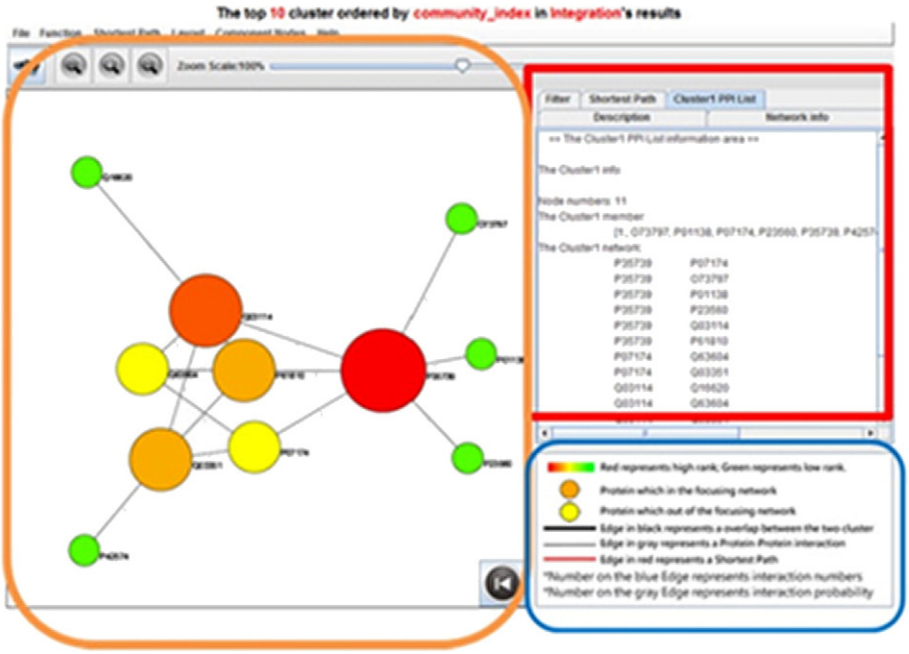


Top 10 predicted complexes

Zoom in a complex by mouse double-click

Select result by various topological methods. Or submit a list of nodes to find those complexes they involved. All the output can be opened in Cytoscape by submitted network and "CytoScape Node Attribute List" in the clustering result panel

Interactive and scalable network analyzer with various layouts



Annotation/ Functional Enrichment for predicted complex and each node

Legends for those symbols in network

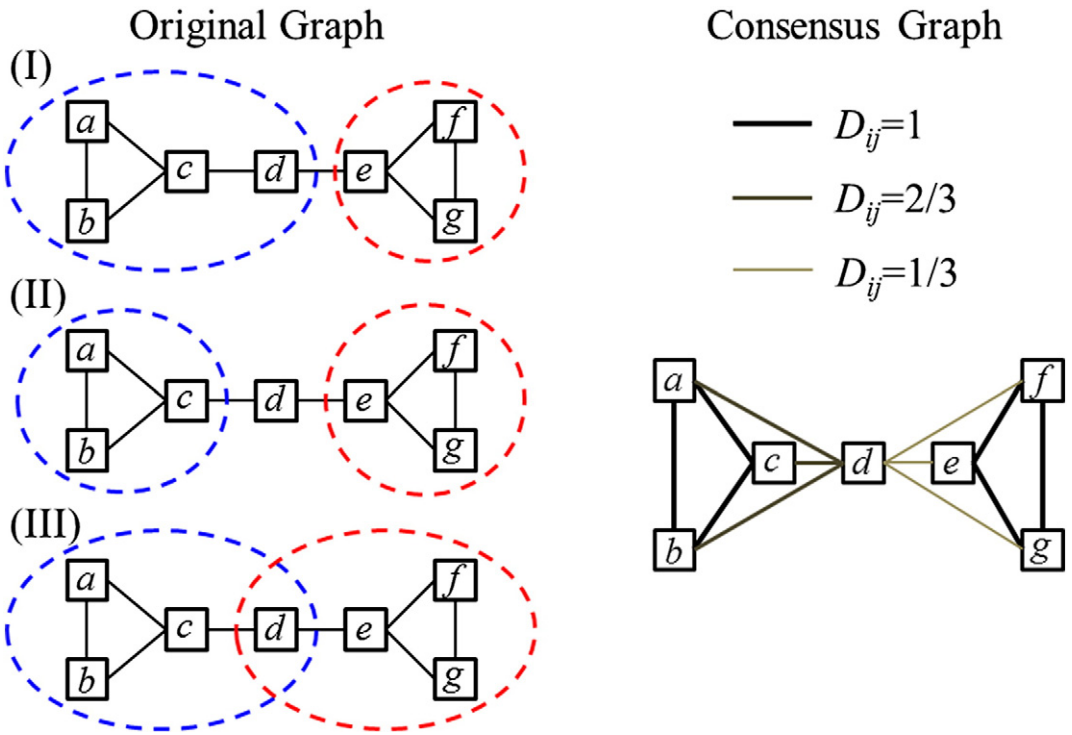


Fig. 4. Schematic illustration of the consensus clustering method. Three clustering results are shown as (I), (II) and (III) on the left side of the figure. The dash lines represent clusters. For example, in (I), there are two clusters: {a, b, c, d} and {e, f, g}. Based on votes of different clustering results, the combination of three clustering results yields the consensus graph shown on the right hand side. The thickness of each edge is proportional to its weight. If edges whose weights are less than 0.7 are removed, three clusters {a, b, c}, {d} and {e, f, g} emerge.

style, and saving the graph or the PPIs involved in the displayed graph. A shortcut is also available to submit the cluster to DAVID Bioinformatics Analysis database (Dennis et al., 2003) for further functional enrichment analysis.

The summary table describes the clustering results from all eight methods available in Spotlight. The clustering result is sent to GUI through a query form for tuning parameters of the clustering method, cluster ranking criteria and the number of clusters to display. The clustering results can be exported as node attributes and utilized with other plug-ins running in Cytoscape. Clusters containing particular protein nodes are retrieved by a query form beneath the result summary table. Detailed Spotlight usages can be found in the website help page.

4. Experiments

4.1. Datasets

The yeast PPI network was downloaded from DIP (Scere20111027) (Salwinski et al., 2004) as the test network. Briefly, this PPI network consists of 5095 proteins and 24,700 interactions. The maximal connected component set, including 4894 proteins and 21,720 interactions, were used as the input set. Clusters derived from the input set were then compared with gene ontology (GO) annotations and known protein complexes to evaluate the predicting power of functional modules and protein complexes. Next, GO database were downloaded from Gene Ontology Consortium Online Database (released date 04/08/2011) (Ashburner et al., 2000) and GO annotations of yeast proteins were obtained from *Saccharomyces* Genome Database (release date, 04/09/2011) (Cherry et al., 2012). Finally, information about yeast known protein complexes and their components list were obtained from MIPS (Mewes et al., 2006) and Aloy et al. (Aloy et al., 2004).

4.2. Method comparison

In contrast to previous methods (Asur et al., 2007; Lancichinetti and Fortunato, 2012; Zhang et al., 2009) which can only integrate partitions computed with different methods, our algorithm incorporates various partitions and overlapping graph clustering results. This work also implemented a consensus method (referred to as *CSS*) according the work of Lancichinetti and Fortunato (2012), which can integrate overlapping graph clustering.

Fig. 4 describes a simple example to illustrate our proposed method. Applying three different clustering methods on a graph allows us to obtain three results, as shown in the remaining three graphs. Among these results, (I) is produced by a partition method and the remaining results are computed by two overlapping clustering methods respectively. As for the results of a partition method, a vertex belongs to exact a cluster. Therefore, despite the difficulty of assigning vertex *d* in this figure, this vertex must be classified into a cluster. In contrast to the partitioning method, a vertex may belong to more than one cluster (e.g., vertex *d* in (III)) or not belong to any cluster (e.g., vertex *d* in (II)). To construct

Fig. 3. Snapshots of *Spotlight* workflow and result displaying. PPI set is submitted to Spotlight through a succinct data-uploading interface. The results of clustering are shown in the graph applet and listed in a summary table. Each cluster in the graph can be expanded for its node/edge components. Details for viewing Spotlight results are described in the manuscript and website help page.

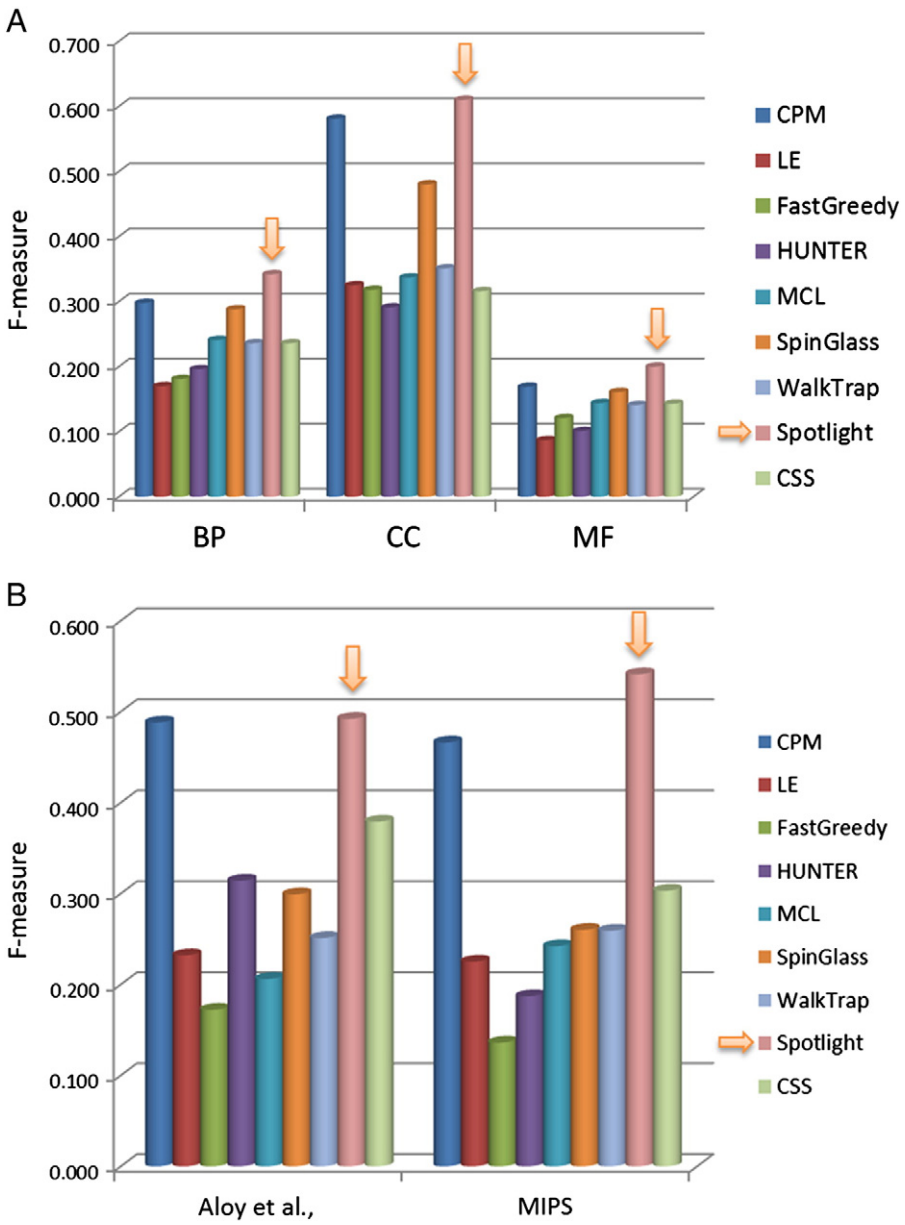


Fig. 5. Comparison of the performance of the integrating graph clustering method and that of the other clustering methods: (A) F-Measure with GO term enrichment on the test data (DIP, Scere20111027); (B) F-Score on two sets of experimental protein complexes (Aloy et al., and MIPS).

consensus matrix **D**, this work computes the similarity between two vertices. For instance, the similarity between vertices *c* and *d* is 2/3 because they are the same clustering in 2 out 3 times. According to this matrix, the corresponding consensus graph appears in the right down graph in this figure. The thickness of each edge is proportional to its weight. Three components are found if edges whose weights are less than 0.7 are removed. Additionally, *CSS* with thresholds ranging from 0.01 to 0.99 is performed in steps 0.01 to integrate the different clustering results, in which the outcome is chosen as its result.

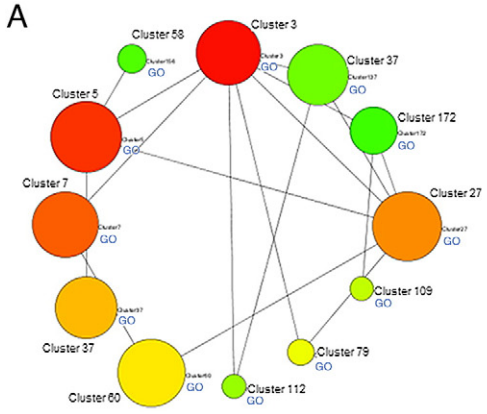
4.3. Evaluation on functional modules

The annotations of Biological Process of gene ontology (GO) are chosen here as the known functional modules. This subsection first describes the *p*-value used in the evaluation method. For a given GO ontology, *N* denotes the total number of proteins annotated in the ontology. Additionally, for a given term in the ontology and a given cluster, *M* is the total number of proteins with this annotation term; *n* represents the number of proteins in a cluster; and *x* refers to the number of proteins in the cluster with annotations containing that term. In the ontology, the *p*-value defined in Formula (2) is the probability of observing *x* or more proteins in a given cluster:

$$p-value = \sum_{i=x}^{n} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}}. \quad (2)$$

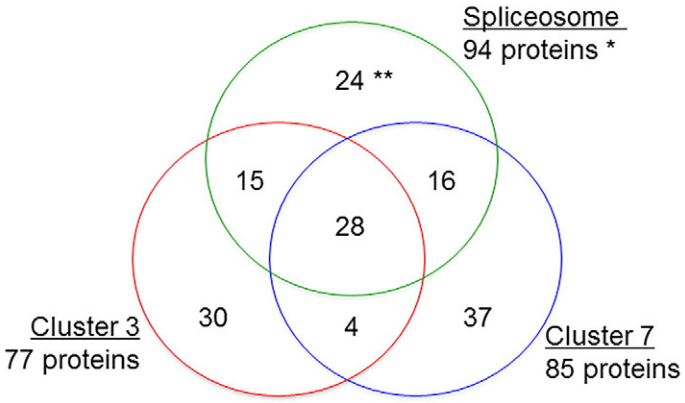
After the *p*-value is estimated, the value is modified by the *Bonferroni* correction and generated *E*-value (Boyle et al., 2004).

A



ClusterID	Cluster size	spliceosomal proteins in this cluster
Cluster3	77	42
Cluster5	122	1
Cluster7	85	45
Cluster27	107	1
Cluster37	66	1
Cluster60	97	3
Cluster79	5	1
Cluster109	4	1
Cluster112	4	1
Cluster137	59	2
Cluster156	6	2
Cluster172	24	3

B



* : three spliceosomal proteins, YSF3, SAD1, CWC15, are not present in the original yeast PPI dataset.
**: 18 out of the 24 proteins were not assigned into clusters of Integration method.

C

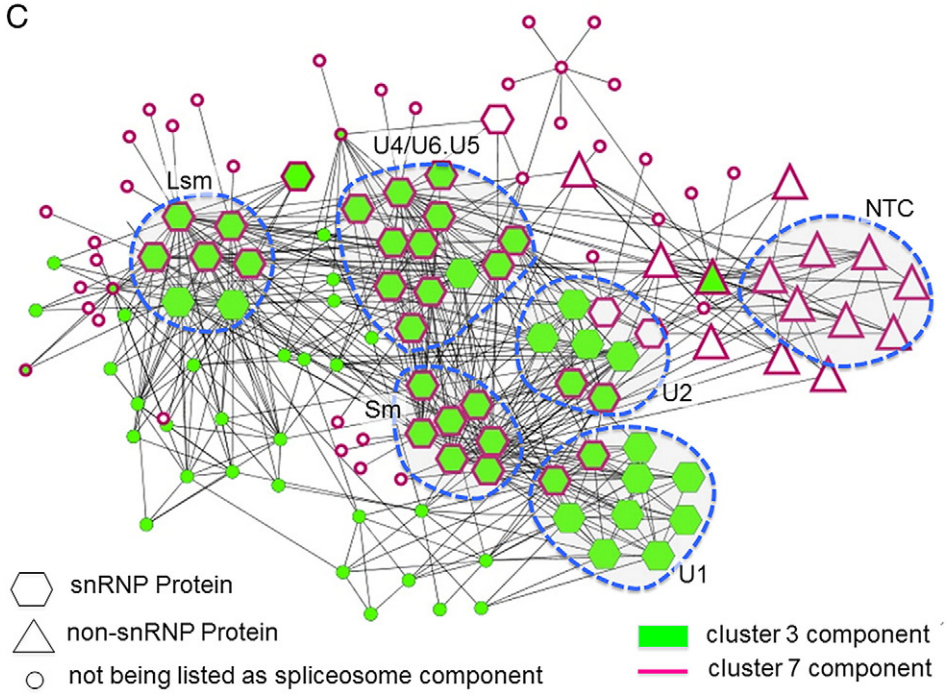


Fig. 6. Subnetwork of yeast spliceosome analyzed by Spotlight. (A) The top clusters ranked by community score and the overlaps among clusters and known components in splicesome. (B) Sharing of protein components among cluster 3, cluster 7, and spliceosome. (C) Subnetworks of the union of cluster 3 and cluster 7. Subunits of spliceosome are denoted by a blue dash line.

Here, based on the F-measure defined in Formula (3), the clustering performance is evaluated by annotations of biological process sub-ontology. In Formula (3), *sensitivity* is defined as the fraction of annotations enriched in at least one cluster with an *E*-value$<10^{-4}$; in addition, *specificity* refers to the fraction of clusters enriched by at least one annotation with an *E*-value$<10^{-4}$ (Ulitsky and Shamir, 2009).

$$\text{F}-\text{measure} = \frac{2 \times \text{Sensitivity} \times \text{Specificity}}{\text{Sensitivity} + \text{Specificity}} \quad (3)$$

Fig. 5A summarizes the performance of functional modules obtained from GO, as predicted by various methods. This figure reveals that Spotlight and CPM are superior to the other methods.

4.4. Evaluation on protein complexes

The overlapping score, shown as Formula (4), was used to determine how effective a predicted complex *PC* could match a known protein complex *KC* from the benchmark set of complexes. As is assumed here, *PC* matches *KC* if $OS(PC, KC) > 0.2$ (Bader and Hogue, 2003).

$$OS(PC, KC) = \frac{|PC \cap KC|^2}{|PC| \times |KC|} \quad (4)$$

Additionally, the performance of algorithms for clustering was evaluated by performing specificity versus sensitivity analysis again. The number of true positives (TP) is defined as the number of predicted complexes matching at least one known protein complex; in addition, the number of false positives (FP) represents all predicted complexes minus TP. Moreover, the number of false negatives (FN) equals the number of known complexes not matched by predicted complexes. Sensitivity is defined as TP/(TP + FN), while specificity represents TP/(TP + FP). In this work, the union of protein complexes from MIPS and Aloy et al. (2004) was used as the benchmark. Fig. 6 shows the performance of various methods. The same evaluating procedure was applied to GO term (Fig. 5A), indicating that the proposed approach (i.e. Spotlight) is still superior to other methods (Fig. 5B).

4.5. Recovery of essential complexes in spotlight clusters

Supplementary Table S1 lists the top 10 clusters of yeast network deciphered by Spotlight, as ranked by the community score. This table lists the MIPS complexes matched to the cluster components (*E*-value<0.001) as well. Their biological relevance is also more closely examined with respect to GO term enrichment and protein complex recovery. Notably, the recovery of MIPS complexes by a cluster is defined by the enrichment of the complex components in the cluster (*E*-value<0.001). This table reveals that each of the top 10 clusters is significantly related to particular GO terms which can be available at the Spotlight online help validation section, http://hub.iis.sinica.edu.tw/spotlight/Help/main.htm#validation.

An important feature of the spotlight algorithm is that the protein component appears in more than one cluster. This algorithm reflects the biochemical properties of some multi-talent proteins. For this situation, consider the spliceosome as an example. Most of the eukaryotic genes contain introns. These non-protein-coding segments are transcripted and spliced from pre-mRNA, and coding segments (exons) are ligated before translation begins. This RNA splicing is accomplished by enormous cellular machinery, the spliceosome (Stevens et al., 2002; Will and Luhrmann, 2011). The components and dynamics of splicesome are overall conserved among metazoans. The size of spliceosome is in a multi-megadalton level, as composed by five snRNPs and many non-snRNP proteins. The spliceosome is highly dynamic with respect to its components. It is flexible yet has a high fidelity, which is assumed to be involved in alternative splicing that is beneficial in diversifying the gene product.

Protein components of spliceosome and their subunit assignment are based on the findings of Chen and Cheng (2012). Briefly, in this work, the 94 known spliceosomal proteins were mapped to the clusters identified by Spotlight Integration method. Except for those proteins that are not found in the yeast PPI dataset (Scere20111027) or not assigned to clusters derived from the integration method. The mapping results of the remaining 73 spliceosomal proteins indicate that two clusters (i.e. clusters 3 and 7) attract most of the spliceosome components (Fig. 6A). Next, the function and network structure of clusters 3 and 7 components are more closely examined. According to Figs. 6B and C, most of the cluster 3/cluster 7 common components are found in snRNPs Sm, Like Sm (Lsm), and tri-snRNP U4/U6.U5. The major contributor of U1 is cluster 3, while most of the non-snRNP proteins belong to cluster 7.

As mentioned earlier, spliceosome is not a pure protein complex. The structure and function heavily rely on the RNA component, explaining why the protein network alone may not reveal the full spliceosome. Moreover, the dynamic nature and flexibility on composition make it challenging to identify its structural information. Based on use of the clustering approach, our results indicate that about 60 proteins in these two clusters are highly associated with spliceosome. These proteins may serve as candidates of intermediate components as well as regulators for spliceosome and RNA editing process.

Moreover, this work presents an example of the complexes involved in yeast transcription machinery to more fully utilize the spotlight clustering results (Supplementary S1, figure S1). In the proposed approach, spotlight can detect these unique network features, subsequently contributing to efforts to discover unknown components of functional modules/protein complexes.

5. Discussion and conclusion

This work develops an integrating graph clustering method to capture protein modules/protein complexes by multiple network features in different algorithms. To streamline the use of the clustering method in the PPI sets of research interest, this work further implements the integration method into a web-based protein network topology analysis scheme, Spotlight. The integration method performs better than other available network clustering methods in terms of the functional association among clustered members and the precision of obtaining protein complexes. This method can obtain the transcription machineries and RNA splicing machinery from the yeast protein network successfully with some proteins that are highly associated yet not described in a complex. Importantly, clusters identified by Spotlight provide further insight into protein modules and complexes by inspecting the PPI network community structures. In this way, novel complex components or regulators can be highlighted by the close agglomeration to known biological complexes.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

CHC and CYL conceptualized the algorithm, designed the method, and drafted the manuscript together with SHC. CHC and JYC were responsible for the construction of web database and network visualization. CAH, CWH and MTK participated in discussion and conceptualization as well as revising the draft. All the authors read and approved the manuscript.

Acknowledgments

This work was supported by the National Science Council (NSC) of Taiwan, Grant NSC 99-2221-E-008-045 to CWH, Grant NSC 100-2628-E-001-007-MY3 to CYL. Ted Knoy is appreciated for his editorial assistance.

Appendix A. Supplementary data

Supplementary data to this article can be found online at http://dx.doi.org/10.1016/j.gene.2012.11.087.

References

Adamcsek, B., Palla, G., Farkas, I.J., Derenyi, I., Vicsek, T., 2006. CFinder: locating cliques and overlapping modules in biological networks. Bioinformatics 22, 1021–1023.
Aloy, P., et al., 2004. Structure-based assembly of protein complexes in yeast. Science 303, 2026–2029.
Ashburner, M., et al., 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. 25, 25–29.
Asur, S., Ucar, D., Parthasarathy, S., 2007. An ensemble framework for clustering protein–protein interaction networks. Bioinformatics 23, I29–I40.
Bader, G.D., Hogue, C.W., 2003. An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinforma. 4, 2.
Barabási, A.-L., Oltvai, Z.N., 2004. Network biology: understanding the cell's functional organization. Nat. Rev. Genet. 5, 101–113.
Bederson, B.B., Grosjean, J., Meyer, J., 2004. Toolkit design for interactive structured graphics. IEEE Trans. Softw. Eng. 30, 535–546.
Boyle, E.I., et al., 2004. GO::TermFinder — open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. Bioinformatics 20, 3710–3715.
Brohée, S., van Helden, J., 2006. Evaluation of clustering algorithms for protein–protein interaction networks. BMC Bioinforma. 7.
Chen, H.C., Cheng, S.C., 2012. Functional roles of protein splicing factors. Biosci. Rep. 32, 345–359.
Cherry, J.M., et al., 2012. Saccharomyces genome database: the genomics resource of budding yeast. Nucleic Acids Res. 40, D700–D705.
Chin, C.H., Chen, S.H., Ho, C.W., Ko, M.T., Lin, C.Y., 2010. A hub-attachment based method to detect functional modules from confidence-scored protein interactions and expression profiles. BMC Bioinforma. 11 (Suppl. 1), S25.
Clauset, A., Newman, M.E.J., Moore, C., 2004. Finding community structure in very large networks. Phys. Rev. E 70.
Csárdi, G., Nepusz, T., 2006. The igraph software package for complex network research. InterJ. Complex Syst. (http://www.interjournal.org/).
Dennis, G., et al., 2003. DAVID: database for annotation, visualization, and integrated discovery. Genome Biol. 4, R60.
Dunn, R., Dudbridge, F., Sanderson, C.M., 2005. The use of edge-betweenness clustering to investigate biological function in protein interaction networks. BMC Bioinforma. 6.
Enright, A.J., Van Dongen, S., Ouzounis, C.A., 2002. An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res. 30, 1575–1584.
Girvan, M., Newman, M.E.J., 2002. Community structure in social and biological networks. Proc. Natl. Acad. Sci. U. S. A. 99, 7821–7826.
Hermjakob, H., et al., 2004. The HUPO PSI's molecular interaction format—a community standard for the representation of protein interaction data. Nat. Biotechnol. 22, 177–183.
Hunter, S., et al., 2012. InterPro in 2011: new developments in the family and domain prediction database. Nucleic Acids Res. 40, D306–D312.
Lancichinetti, A., Fortunato, S., 2012. Consensus clustering in complex networks. Sci. Rep. 2.
Lázár, A., Ábel, D., Vicsek, T., 2010. Modularity measure of networks with overlapping communities. Europhys. Lett. 90, 18001.
Leung, H.C., Xiang, Q., Yiu, S.M., Chin, F.Y., 2009. Predicting protein complexes from PPI data: a core-attachment approach. J. Comput. Biol. 16, 133–144.
Liu, G., Wong, L., Chua, H.N., 2009. Complex discovery from weighted PPI networks. Bioinformatics 25, 1891–1897.
Luo, F., Yang, Y., Chen, C.F., Chang, R., Zhou, J., Scheuermann, R.H., 2007. Modular organization of protein interaction networks. Bioinformatics 23, 207–214.
Mewes, H.W., et al., 2006. MIPS: analysis and annotation of proteins from whole genomes in 2005. Nucleic Acids Res. 34, D169–D172.
Newman, M.E.J., 2004. Fast algorithm for detecting community structure in networks. Phys. Rev. E 69.
Newman, M.E.J., 2006. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74.
Palla, G., Derényi, I., Farkas, I., Vicsek, T., 2005. Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818.
Pons, P., Latapy, M., 2005. Computing communities in large networks using random walks: Computer and Information Sciences — Iscis 2005, Proceedings, 3733, pp. 284–293.
Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D., 2004. Defining and identifying communities in networks. Proc. Natl. Acad. Sci. U. S. A. 101, 2658–2663.
Reichardt, J., Bornholdt, S., 2006. Statistical mechanics of community detection. Phys. Rev. E 74.
Rives, A.W., Galitski, T., 2003. Modular organization of cellular networks. Proc. Natl. Acad. Sci. U. S. A. 100, 1128–1133.
Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U., Eisenberg, D., 2004. The Database of Interacting Proteins: 2004 update. Nucleic Acids Res. 32, D449–D451.
Spirin, V., Mirny, L.A., 2003. Protein complexes and functional modules in molecular networks. Proc. Natl. Acad. Sci. U. S. A. 100, 12123–12128.
Stevens, S.W., et al., 2002. Composition and functional characterization of the yeast spliceosomal penta-snRNP. Mol. Cell 9, 31–44.
Ulitsky, I., Shamir, R., 2009. Identifying functional modules using expression profiles and confidence-scored protein interactions. Bioinformatics 25, 1158–1164.
van Dongen, S., 2000. Graph clustering by flow simulation. Ph.D. Thesis, University of Utrecht, The Netherlands.
Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of 'small-world' networks. Nature 393, 440–442.
Will, C.L., Luhrmann, R., 2011. Spliceosome structure and function. Cold Spring Harb. Perspect. Biol. 3.
Wu, M., Li, X., Kwoh, C.K., Ng, S.K., 2009. A core-attachment based method to detect protein complexes in PPI networks. BMC Bioinforma. 10, 169.
Zhang, S., Ning, X., Zhang, X.S., 2006. Identification of functional modules in a PPI network by clique percolation clustering. Comput. Biol. Chem. 30, 445–451.
Zhang, Y., Zeng, E.L., Li, T., Narasimhan, G., 2009. Weighted consensus clustering for identifying functional modules in protein–protein interaction networks. Eighth International Conference on Machine Learning and Applications, Proceedings, pp. 539–544.