International Scholarly Research Network
ISRN Biomathematics
Volume 2012, Article ID 726429, 11 pages
doi:10.5402/2012/726429

Research Article

Partitioning a PPI Network into Overlapping Modules Constrained by High-Density and Periphery Tracking

Md. Altaf-Ul-Amin, Masayoshi Wada, and Shigehiko Kanaya

Computational Systems Biology Lab, Nara Institute of Science and Technology, Ikoma, Nara 630-0101, Japan

Correspondence should be addressed to Md. Altaf-Ul-Amin, amin-m@is.naist.jp

Received 13 February 2012; Accepted 14 March 2012

Academic Editors: G. L. Borosky, J. Chow, and J. M. Peregrín-Alvarez

This paper presents an algorithm called DPClusO for partitioning simple graphs into overlapping modules, that is, clusters constrained by density and periphery tracking. The major advantages of DPClusO over the related and previously published algorithm DPClus are shorter running time and ensuring coverage, that is, each node goes to at least one module. DPClusO is a general-purpose clustering algorithm and useful for finding overlapping cohesive groups in a simple graph for any type of application. This work shows that the modules generated by DPClusO from several PPI networks of yeast with high-density constraint match with more known complexes compared to some other recently published complex generating algorithms. Furthermore, the biological significance of the high density modules has been demonstrated by comparing their P values in the context of Gene Ontology (GO) terms with those of the randomly generated modules having the same size, distribution, and zero density. As a consequence, it was also learnt that a PPI network is a combination of mainly high-density and star-like modules.

1. Introduction

The ability to generate comprehensive datasets incorporating all biological molecules and their interactions at different omics levels has led to the emergence of system biology. Cellular functions rely on the coordinated actions of multiple genes, RNAs, proteins, and metabolites. Therefore, organizing biological information in the context of networks is important for holistic understanding of biological systems. Some characteristics of the biological networks are that they are complex, autonomous, robust, function to execute physicochemical processes, and are conserved but can adjust [1].

Constructing biological networks by integrating information piece by piece has led to the discovery of emergent topological properties in terms of degree distribution, average path length, clustering coefficient, robustness, and so forth. These emergent properties of biological networks are quite distinct compared to those of the randomized networks of the same size and could not be understood by analyzing an individual or a small group of components. On the other hand networks having the same global topological property for example, degree distributionn may have different local topologies [2]. Analyzing these local topologies is useful to understand the design principles of biological networks and the functions of the individual components.

A feature for many networks including biological networks is their modular organization. Modularity is a property that allows a network to be divided into cohesive groups of entities. Biological systems can be regarded as modular because of the fact that in such systems different functions are performed by group of interacting molecules [3]. A functional module can be defined on the basis of spatial and/or temporal activities of a group of molecules. Modules can be realized differently with different objectives for different biological networks, for example, transcriptional networks [4], metabolic pathways [5], protein-protein interaction (PPI) networks [6, 7], and networks consisting of different types of molecules. Furthermore, it is well known that a molecule often takes part in multiple functions indicating that the modules in a biological network are supposed to be overlapping.

In the present work we focus on PPI networks. The active interactions among proteins might vary over time

and space a phenomenon that is referred to as network dynamics. But current experiments [8–10] used to determine PPI interactions cannot detect the time and location of the interactions within the cell very precisely. Therefore, most of the studies regarding PPI networks have been conducted on static networks, that is, by assembling all the known interactions concerning the proteins of an organism in a single network, and undoubtedly these studies have been able to produce valuable results despite the fact that many of the experimentally determined interactions are false positives [11].

In proteomics a group of proteins involved in a particular cellular function is called a protein complex. There is no final say about the exact topological properties, for example, density, diameter, and average path length of the complexes within a PPI network until and unless all the interactions are accurately known. However, it is assumed that high-density subgraphs in PPI networks are protein complexes [12] or at least their signatures. Gavin et al. proposed that a protein complex consists of high-density core proteins and some attachment proteins. Several complex detection algorithms have been developed based on this notion [13–15].

Most of the complex detection algorithms were applied to yeast PPI networks, and they were evaluated using two curated sets of known complexes referred to as MIPS complexes [16] and CYC2008 complexes [17]. A total of 1836 proteins are involved in all the complexes of these two sets. But yeast has roughly 6,000 proteins indicating that there might be other protein complexes. It is stated in [18] that edge density is the most telling statistical clustering metric, and it is easy to realize that the higher the density of a subgraph the more cohesive the nodes. It can be hypothesized that comprehensive determination of high-density subgraphs can give clues to comprehensive mapping of complexes and might be helpful for deciphering other biological information. It would be computationally very difficult to check every combination of protein groups of different sizes for the purpose of searching high-density subgraphs. The number of overlapping cliques in a network might be enormous [19] based on that it can be said that the number of subgraphs of a given density also will be too many. The algorithm of Georgii et al. [20] attempted to find all possible subgraphs of a given density, but the results indicate that the number of clusters as well as computational time would be very high and would not be feasible for big and dense networks. One limitation of density based clustering methods is the coverage [21]. In this work we give emphasis not only on density but also on periphery and propose an algorithm that heuristically try to assign each protein to clusters as large as possible and also avoid generating too many nearly identical overlapping clusters to narrow down the candidate clusters while at the same time cover all the proteins in the network. Though coverage can be achieved by assigning each node to a single cluster or by considering each interaction as a cluster, DPClusO does not ensure coverage in such trivial ways. DPClusO has been developed for overlapping clustering and has the following properties: each node goes to at least one cluster, no two clusters are completely the same, density of each cluster is more than or equal to the user given density, clusters are constrained by periphery if that exists. Each of the high-density clusters determined by the proposed method can be considered cohesive group of entities. We show that modules generated by DPClusO from PPI networks of yeast are closely relevant to known protein complexes and useful for determining functions and localizations of unknown proteins.

2. The DPClusO Algorithm

2.1. Relevant Definitions. First we mention definitions for some terms from our previously published paper which are also relevant to this work.

Density. The density d_k [12, 22] of any cluster k is the ratio of the number of edges present in the cluster ($|E_k|$) and the maximum possible number of edges in the cluster ($|E_k|_{\max}$) and is represented by

$$d_k = \frac{|E_k|}{|E_k|_{\max}} = \frac{2 \times |E_k||N_k|}{|N_k|(|N_k| - 1)}. \quad (1)$$

Here, $|N_k|$ is the size of the cluster, that is, the number of nodes in the cluster. The density of a cluster is a real number ranging from 0 to 1.

Cluster Property. The cluster property cp_{nk} [22] of any node n with respect to any cluster k of density d_k and size $|N_k|$ is defined by

$$\mathrm{cp}_{nk} = \frac{|E_{nk}|}{d_k \times |N_k|}. \quad (2)$$

Here, $|E_{nk}|$ is the total number of edges between the node n and each of the nodes of cluster k.

It is noteworthy that a higher value of cluster property of a neighbor indicates that it is part of the cluster while a lower value indicates that it is part of the periphery.

Weight of an Edge. In a graph $G(N, E)$ the weight w_{uv} of an edge $(u, v) \in E$ is the number of the common neighbors of the nodes u and v [22].

Weight of a Node. In a graph $G(N, E)$ the weight w_n of a node n is the sum of the weights of the edges connected to the node, that is, $w_n = \Sigma w_{nu}$ for all u such that $(n, u) \in E$ [22].

2.2. Flowchart of the Algorithm. The input to the DPClusO algorithm is the adjacency list of an undirected simple graph. The flowchart of Figure 1 shows the outline of the algorithm. We discuss the important terms in the flowchart in the following

Termination Check. When a cluster is generated, the edges of the cluster, that is, only the edges whose both nodes belong to the cluster, are removed from the graph, and the generation of the next cluster is then started in the remaining graph. The algorithm iterates until no edge is left in the remaining graph.

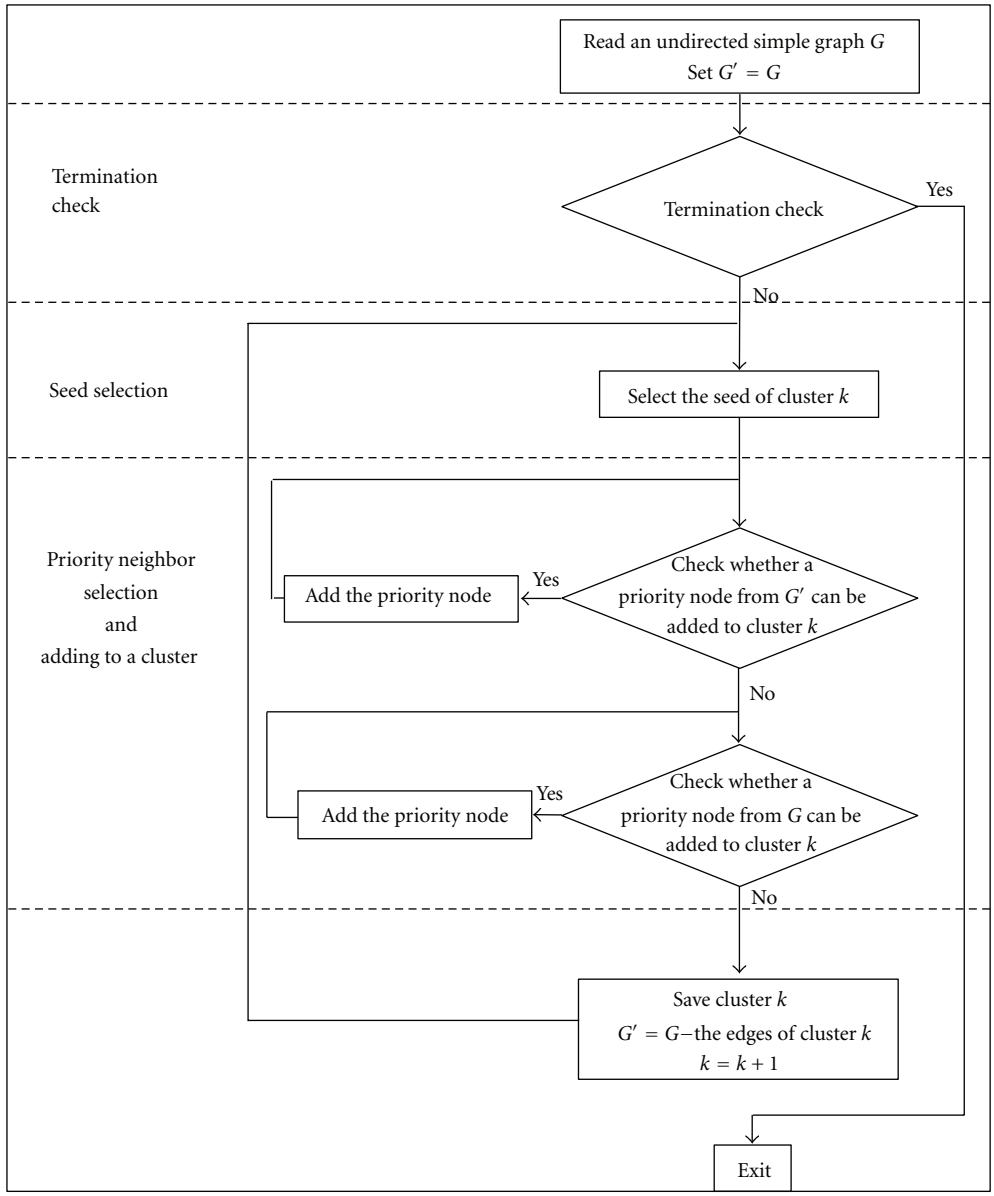

Figure 1: The flowchart of the DPClusO algorithm.

Seed Selection. A cluster is always started at a seed node. When a cluster is generated, its edges are removed from the graph and seed for the next cluster is determined in the remaining graph. Initially the seeds are selected based on the highest node weight. A node that is already part of a cluster is not allowed to be a seed node as long as the seeds are selected based on highest weight. When in the remaining graph the weight of each node is zero, the seeds are selected based on the highest degree. At this stage we allow the nodes to become seeds which are already part of clusters generated using weight-based seeding. However, when seeding based on degrees starts, again we do not allow a node to become a seed that is already included in a cluster generated using degree-based seeding. These restrictions for nodes to become seeds bar/limit the generation of completely/highly overlapping clusters.

Priority Neighbor Selection. First the neighboring node for which the sum of the weights of the edges between it and each of the nodes of the cluster is the highest is selected as the priority node. If such a node cannot be added to the cluster then the neighboring node having the highest number of links with the cluster is considered as the priority node. In both of the above cases, if a tie occurs, then any of the highest priority nodes is considered.

Adding a Neighbor to a Cluster. After selecting a seed a cluster is formed by recursively adding priority nodes from among its neighbors until none is available that satisfies two conditions as follows: (i) addition of the node does not cause the density of the cluster to be less than the given density and (ii) the cluster property of the node is more than or equal to the given cluster property.

When DPClusO is run with $d_{in} < 0.7$, it empirically replaces G by G′ at the point when weight-based seeding is over. Also this is done to limit the generation of highly overlapping clusters. We implemented the DPClusO algorithm in JAVA, and it is available for use at http://kanaya.naist.jp/DPClusO/.

2.3. Major Differences with DPClus. DPClusO has been developed for overlapping clustering while DPClus [22, 23] was mainly developed for nonoverlapping clustering with limited capability of overlapping clustering. The DPClusO algorithm guarantees coverage, that is, each node becomes part of at least one cluster, wheras the DPClus algorithm does not ensure coverage. Thus, DPClusO retrieves more information from the network. DPClus works on the adjacency matrix of the network, but DPClusO works on the adjacency list. Adjacency matrix consumes huge amount of memory (proportional to the square of the number of nodes) for storing big graphs whether it is dense or sparse. But when a graph is sparse, it requires relatively much less memory (proportional to the number of edges) to store it as an adjacency list format. Usually most practical big networks including PPI networks are sparse and, therefore, DPClusO is memory efficient compared to DPClus. Furthermore, DPClusO is faster compared to DPClus. In case of DPClus, after generation of a cluster it is removed from the graph, and then the generation of the next cluster is started in the remaining graph. Every time after removal of a cluster DPClus recalculates the weights of all edges and nodes in the context of the remaining graph for seed selection which increases the computational time very much. For DPClusO when a cluster is generated only its edges (not the nodes) are removed from the graph before generation of the next cluster. This change for DPClusO ensures that the generated modules cover all the nodes in the network. Furthermore, only the weights of those neighbors which are already not part of any previously generated cluster are updated because the weights of the nodes at a distance 2 or more from the nodes of the cluster (whose edges are removed) remain unaffected which is proved in the following theorem.

Theorem 1. *If the edges of a cluster (a group of densely connected nodes) are removed from a graph, only the weights of the nodes belong to the cluster and their neighbors may change.*

Proof. Let $G(N, E)$ be a graph and $G'(N', E')$ a subgraph of G such that E' includes all the edges (a, b) for which both $a, b \in N'$. If an edge connected to a node is removed, it is obvious that the weight of that node may change. Therefore, when the edges in E' are removed the weights of the nodes in N' may change. Furthermore, the weight of a node may change if the weight of an edge connected to it is changed, and based on this fact it can be said that the weights of the neighbors of N' may change when the edges of E' are removed from G. To change the weight of a node an edge connected to it or any of its neighbors must be removed. Based on this fact it can be said that the weights of the nodes that are at a distance 2 from all the nodes of N' do not change when edges of E' are removed from G. □

3. Results and Discussion

3.1. Measures Used for Evaluation. Initially we provide definitions of some measures which were previously used in other works for the evaluation of modules generated from PPI networks.

Overlapping Score. The overlapping score (OV) [12] between any two complexes, a and b, is calculated by using the following equation:

$$\mathrm{OV} = \frac{|i|^2}{|a| * |b|}. \tag{3}$$

Here, i is the intersection set of the two complexes a and b.

Precision, Recall, and F-Measure. Let P be a set of complexes predicted by an algorithm from a PPI network and B a set of benchmark, or known complexes. The precision, recall, and F-measure [13] of the algorithm in the context of the PPI network and an overlapping threshold OV_{th} are defined as follows:

$$\begin{aligned} \text{Precision} &= \frac{N_{cp}}{|P|}, \\ \text{Recall} &= \frac{N_{cb}}{|B|}, \\ F\text{-measure} &= \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \end{aligned} \tag{4}$$

Here

$$\begin{aligned} N_{cp} &= \left|\{p \mid p \in P, \exists b \in B, \mathrm{OV}(p, b) \geq \mathrm{OV}_{th}\}\right|, \\ N_{cb} &= \left|\{b \mid b \in B, \exists p \in P, \mathrm{OV}(p, b) \geq \mathrm{OV}_{th}\}\right|. \end{aligned} \tag{5}$$

Hypergeometric P Value. The hypergeometric P value of a module is defined as follows:

$$P = 1 - \sum_{i=0}^{k-1} \frac{\binom{F}{i}\binom{N-F}{C-i}}{\binom{N}{C}}. \tag{6}$$

Here N, C, and F are the sizes of the whole network, a module and a functional group in the network, respectively, and k is the number of proteins of the functional group in the module.

3.2. PPI Data Sets. We collected four sets of PPI data from Gavin et al. [7], Krogan et al. [24], DIP [25], and MIPS [26]. We also constructed a big PPI dataset which is the union of all four datasets and is referred to as "Union" network. The number of interactions and number of proteins of each dataset is shown in the first two columns of Table 1. The degree distributions of all the datasets roughly follow power law [27]. (See Supplementary Figure 1 in Supplementary Material available online at doi: 10.5402/2012/726429).

3.3. Overlapping among DPClusO Modules. To assess how overlapping are the modules generated by DPClusO, we used the measure defined by (3). The third and fourth

Table 1: Number of proteins and interactions in five different PPI data sets (netwrks) and the number of modules generated by DPClusO, COACH, and CORE algorithms in each of the PPI networks.

Dataset	Proteins	Interaction	DPClusO (All)	DPClusO (Size >3)	COACH (All)	CORE (All)
Gavin	1430	7592	1135	759	326	662
Krogan	2674	7079	2206	656	345	785
MIPS	4546	12319	4136	969	489	1678
DIP	4959	21752	4818	1582	896	1635
Union	5497	35898	5614	2434	1283	1613

columns of Table 1, respectively, show the number of all modules and the number of modules of size greater than 3 generated by DPClusO from all the five networks. The number of modules is much less than the number of interactions but comparable to the number of proteins in the network. Many modules are consisting of two proteins because clustering was performed using very high d_{in} value of 0.9. To assess how overlapping are the modules, we developed a filtering technique corresponding to a given maximum overlapping score OV_{max} as follows: sort the clusters in decreasing order according to their size, and then in each step a cluster is selected from the top if its overlapping with any of the previously selected cluster is less than or equal to OV_{max}. Figure 2 shows the plot of number of modules against maximum overlapping score corresponding to all five networks. When two modules of size 2 share one protein their overlapping score is 0.25, and when two modules of size 3 share one protein their overlapping score is 0.11. In Figure 2 the high slopes between 0.1 and 0.3 are caused by the fact that many pairs of generated modules of size 2 and 3 share one protein. The number of completely nonoverlapping modules for different data sets is indicated by the points corresponding to $\text{OV}_{\text{max}} = 0$. For example in case of the "Union" network, 303 modules are completely non-overlapping, however, 5,114 modules, that is, around 91% of all the modules are such that overlapping between any two of them is less than or equal to 0.5. Figure 2 implies that the trend is similar for other networks. Here it is noteworthy that the algorithm of [20] generated 24,803 modules using a high-density constraint from a network consisting of 3,559 proteins and 14,212 interactions out of which only 1,083 are such that overlapping between any two of them is less than or equal to 0.5, and in that work those 1,083 modules were considered as distinct clusters. It can be concluded that the modules generated by DPClusO are not too overlapping that is, not nearly identical and hence contain diversified information.

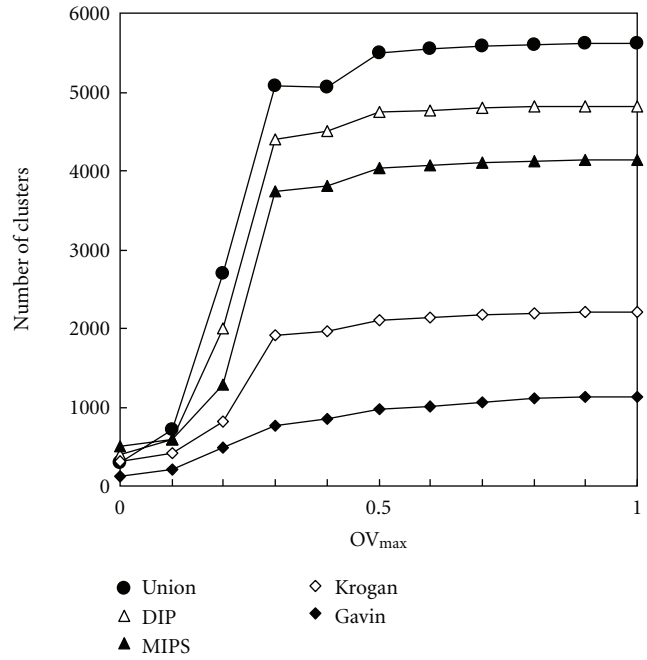

Figure 2: Plot of the number of clusters generated by DPClusO with respect to maximum overlapping. $\text{OV}_{\text{max}} = 0$ means all modules are completely nonoverlapping. For other points OV_{max} indicates the maximum overlapping score between any two modules.

3.4. Relation with Known Complexes. The modules generated by DPClusO highly match with known complexes of yeast. We downloaded a set of 408 known protein complexes of yeast from Wodak Lab [17]. Out of them 236 consist of 3 or more proteins. To measure the extent of matching of a module with a known complex, we used the overlapping score of (3). In this context we also compared DPClusO with two other complex generating algorithms called COACH [13] and CORE [14]. The number of modules generated by CORE and COACH in different networks are shown, respectively, in columns 6 and 7 of Table 1. The COACH complexes were generated using omega value 0.225, and the CORE complexes got generated by contacting with the group who developed the algorithm. Figure 3 shows that modules generated by DPClusO always matched with much more known complexes in case of all 5 datasets and across all overlapping scores. This implies that recall for DPClusO is always higher. Notice that DPClusO cover 100% of the proteins, but COACH and CORE complexes covered only 50.15% and 73.82%, respectively, in case of the "Union" network. For the sake of coverage DPClusO generated somewhat higher number of modules, and hence precision and F-measure are not higher if calculated in the context of all the generated modules. However, by simple filtering it is possible to obtain much better F-measure compared to COACH and CORE. Figure 4 shows the F-measure based on $\text{OV}_{\text{th}} = 0.6$ for DPClusO, COACH, and CORE calculated by filtering out the size 2 modules and some overlapping

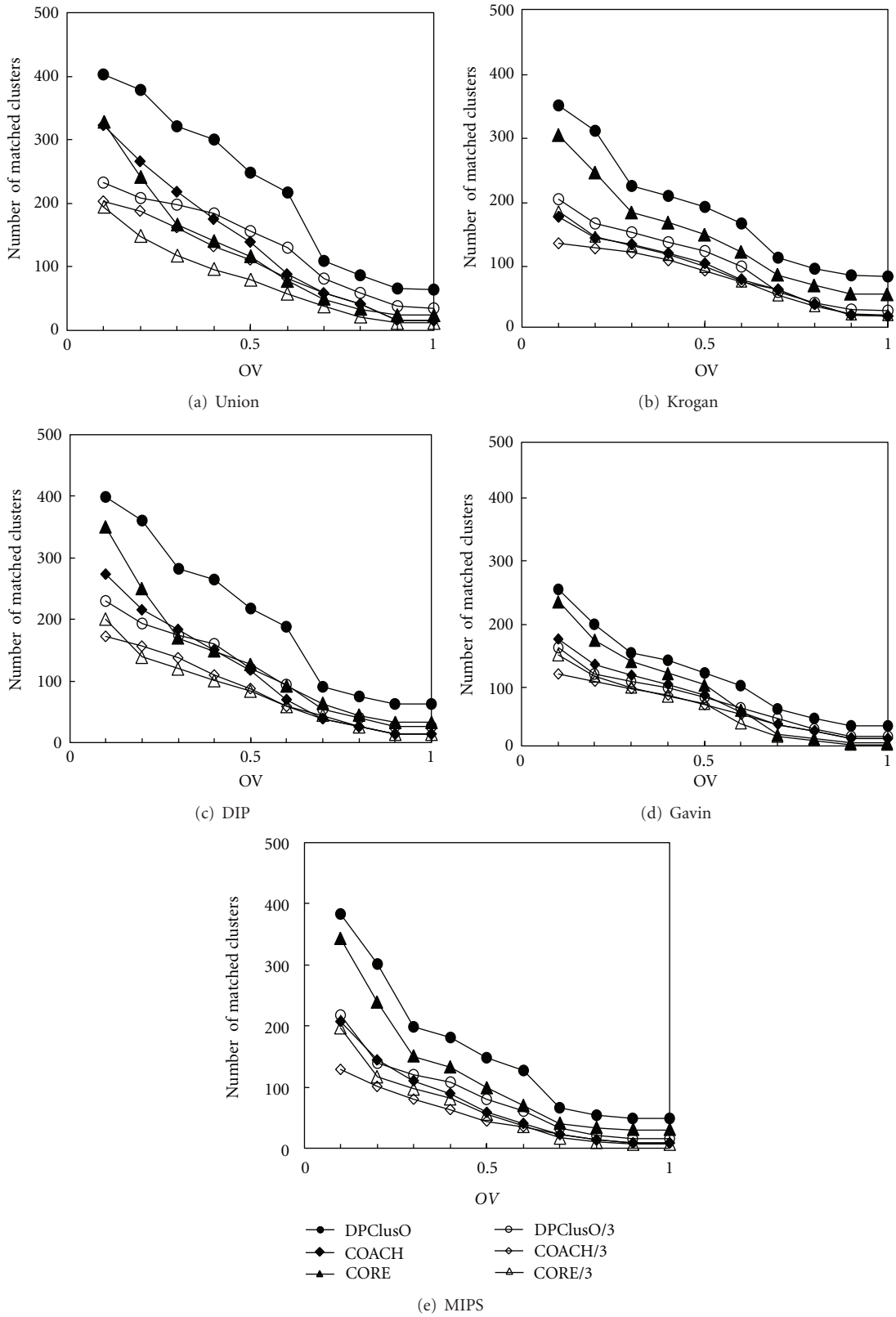

Figure 3: Plots showing how many and to what extent the known protein complexes (all complexes and size 3 or more complexes shown separately) of yeast matched with modules predicted by DPClusO, COACH, and CORE corresponding to five different datasets.

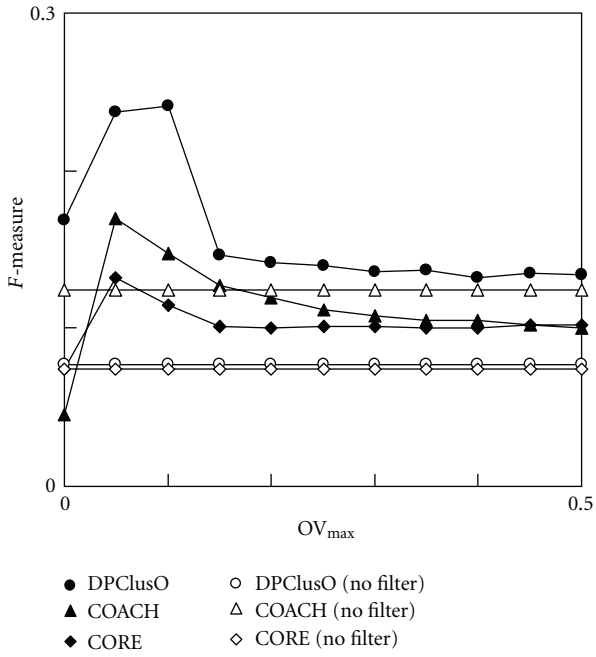

Figure 4: Variation of F-measure with maximum overlapping score (used as a filtering parameter) for modules of size 3 or more generated by DPClusO, COACH, and CORE. The marked horizontal lines indicate F-measures for three algorithms in case of no filtering.

modules. Three horizontal lines in Figure 4 show the F-measures for three methods in case of no filtering. The COACH algorithm achieved the highest F-measure in case of no filtering. However by simple filtering DPClusO can achieve higher F-measure compared to what COACH can achieve in case of both filtering and no-filtering. Also, it is noteworthy that the filtering process we adopted here does not need any information about the functions of proteins and can be easily added to the clustering process. However, we do not add the filtering permanently to DPClusO to keep it a general clustering algorithm instead of just suite it to predict the CYC2008 complexes.

3.5. DPClusO Is a Robust Algorithm. The results presented in the previous section imply that DPClusO performed well for all 5 datasets in the context of identifying known complexes which indicate its robustness. To further investigate its robustness, we constructed 6 networks by altering the "Union" network as follows: 2 networks by adding 5% and 10% random edges, 2 networks by removing 5% and 10% randomly selected edges, and 2 networks by randomly rearranging 5% and 10% edges. The random addition, removal, and rearrangement were done by using the "Random" class of JAVA. Modules were then generated using all 6 networks and matched with known complexes. Figure 5 shows the comparison among modules generated from the original and altered "Union" networks in the context of their matching with known complexes. The effects of addition, removal, and rearrangement of 5% edges are minimal. The effects of addition of 10% random edges are also minimal. However, removal and rearrangement of 10% edges caused some deviation but not very much (Figures 5(c) and 5(d)). On the basis of overall judgement, the deviation is minimal in cases of adding random edges which implies that DPClusO can perform more robustly against false positives.

3.6. Accumulation of Similar Proteins. If the interactions of a network are all true positives, then the high-density modules are obviously the cohesive groups of proteins. However, PPI data may contain false positives, but still the high-density modules are important and likely to contain similar function proteins. The richness of similar function proteins in a module can be estimated by calculating P values using (6) based on hypergeometric distribution. For determining P values using (6) we need to know the classification of the proteins. Gene Ontology classification terms are most comprehensive and popularly used [28]. The GO terms are primarily divided into 3 types: molecular function (MF), biological process (BP), and cellular compartment (CC). As size 2 complexes are mere interactions we calculated P values of 2,434 modules of size 3 or more using the R package GOStats [29]. We also calculated P values for 2434 random protein groups having the same size distribution and zero density to distinguish the overall statistical significance of the DPClusO generated modules. Figure 6 shows the distributions for both the high-density modules and the randomly selected protein groups with respect to $-\log(P \text{ value})$. The distributions of the high-density modules are quite different from the distributions of the randomly selected protein groups corresponding to all three types of GO terms. Clearly the overall statistical significance of the DPClusO-generated modules is much higher.

3.7. Star and Star-Like Modules. A graph is called a star when one node (the central node) is adjacent to all the other nodes (peripheral nodes) of the graph. For a perfect star there is no edge between the peripheral nodes. By star-like cluster we mean a star with a few edges among the peripheral nodes. DPClusO generates clusters first by weight-based seeding and then by degree-based seeding. When weight-based seeding is over the remaining graph is a tree. However, if the clustering is continued with density constraint >0.67 only two-protein modules are generated. In the "Union" network we obtained 3,180 two-protein modules which indicate the possibility of existence of many star-like modules in the network. We generate star-like clusters by selecting the seeds based on highest degrees and their first neighbors in the remaining graph at the point when weight-based seeding is over. In the "Union" network we found 1,216 star-like clusters many of which are perfectly stars. Finding the abundance of star-like clusters in the PPI network we assume that they also have biological significance like the high-density clusters and therefore we determined their P values the same way we did for the high-density clusters. Interestingly we found that the presence of similar function proteins in star-like clusters is statistically significant compared to that in randomly generated modules of the same size, distribution, and zero

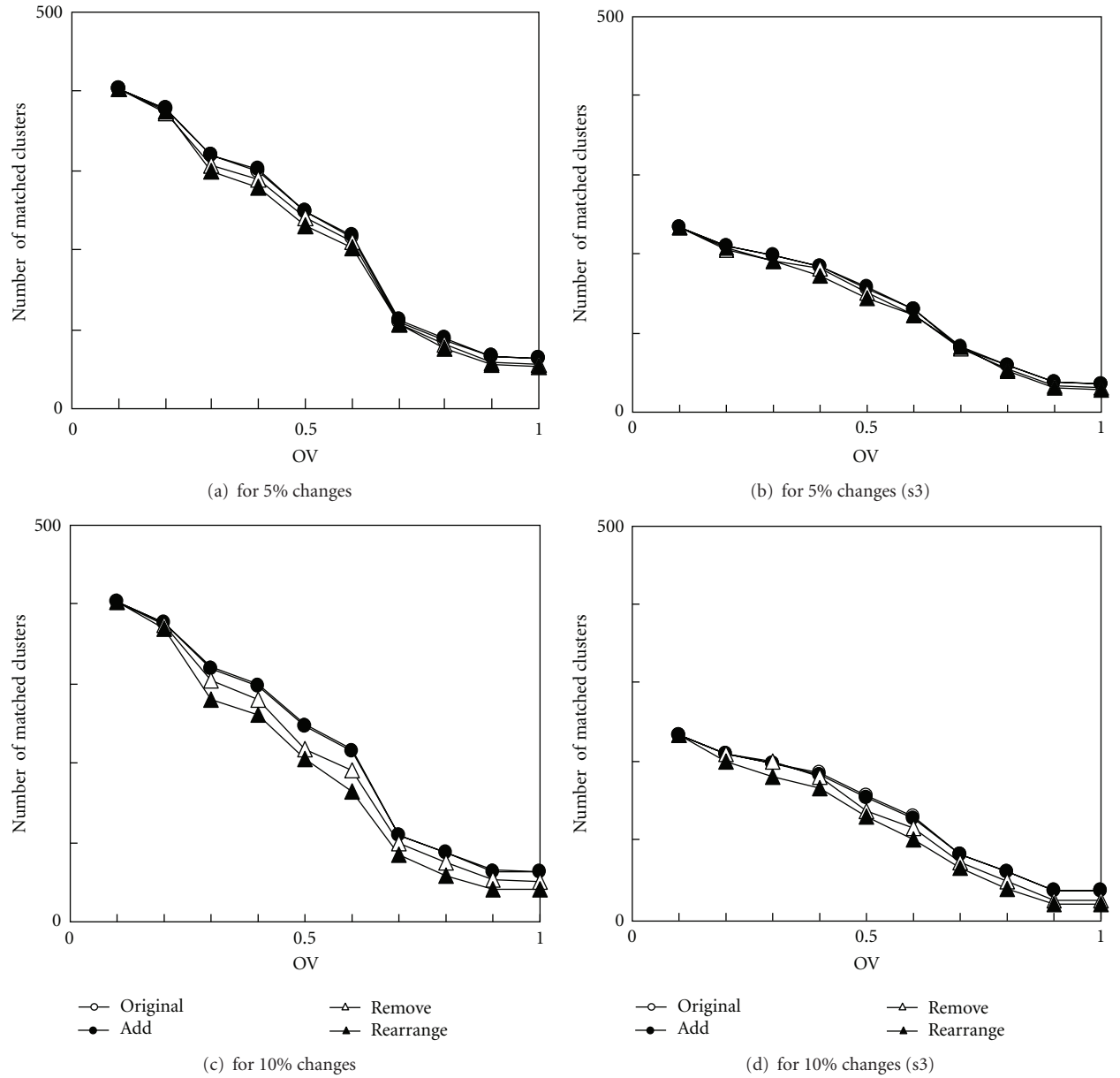

Figure 5: Verifying robustness of DPClusO by comparing generated modules from real and randomly altered PPI networks in the context of matching with known complexes. ((a) and (b)) In case of addition, removal, and rearrangement of 5% edges in the context of all and size 3 or more known complexes, respectively. ((c) and (d)) In case of addition, removal, and rearrangement of 10% edges in the context of all and size 3 or more known complexes, respectively.

density for all three types of GO terms (Figure 7). As the "Union" network is quite big in the context of yeast the trends in it are important. It can be concluded that a PPI network is a combination of overlapping high-density and star-like modules, and the richness of similar proteins in those modules indicates that they are supposed to contain clues to many known or unknown biological phenomena.

4. Conclusions

In this paper we presented an overlapping graph clustering algorithm called DPClusO. The merits of DPClusO are ensuring coverage, that is, each node goes to at least one module and at the same time the generated clusters are not too overlapping. DPClusO is a general purpose clustering algorithm and applicable to simple graphs for different types of applications. Density is an easy to understand property of a cluster, and a user can select the minimum threshold density for the generated clusters depending on the purpose of analysis and sparseness of the network and so forth. In the present work we generated high-density modules using DPClusO from several PPI networks of yeast. The high-density modules were shown to match with more known protein complexes of yeast compared to some

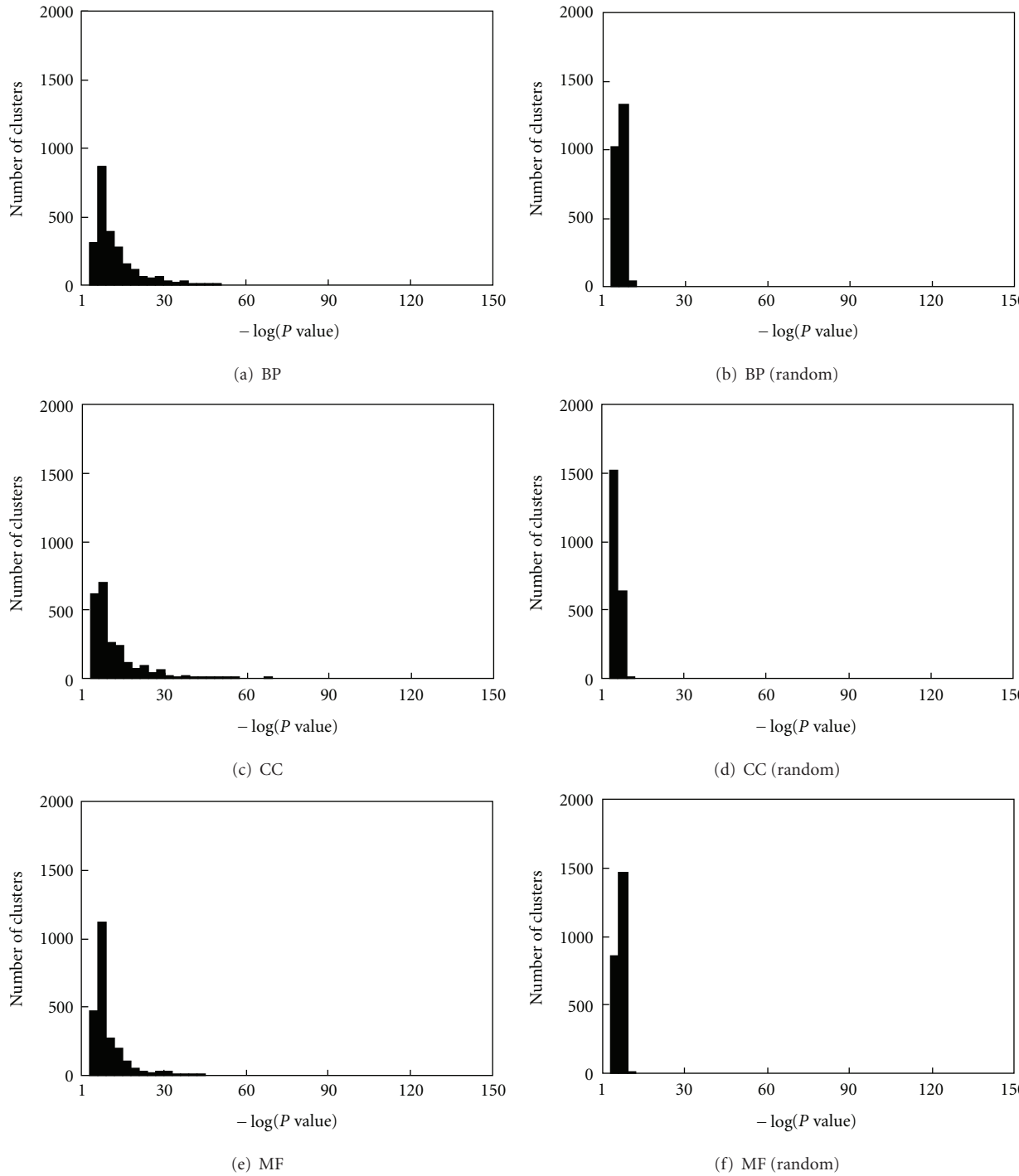

Figure 6: Comparison between the distributions of the high-density modules and randomly selected protein groups with respect to $-\log(P \text{ value})$ in the contexts of three types of gene ontology terms: ((a) and (b)) biological process (BP), ((c) and (d)) cellular compartment (CC), and ((e) and (f)) molecular function (MF).

other recently developed complex finding algorithms. Also, DPClusO performs robustly in cases of random addition, removal, and rearrangement of edges in PPI networks. It was also verified based on hypergeometric P values that the generated modules contain statistically significant proportion of similar function proteins. Hence, these modules can be used to predict function of function-unknown proteins based on guilt by association and to get other insights about function-known proteins. As DPClusO ensures coverage, it can be utilized to investigate to how many and to what types of modules any protein of interest belongs. Partitioning PPI networks using very high-density constraints resulted in many interesting modules of larger size together with many two-protein modules which tempted to believe that there

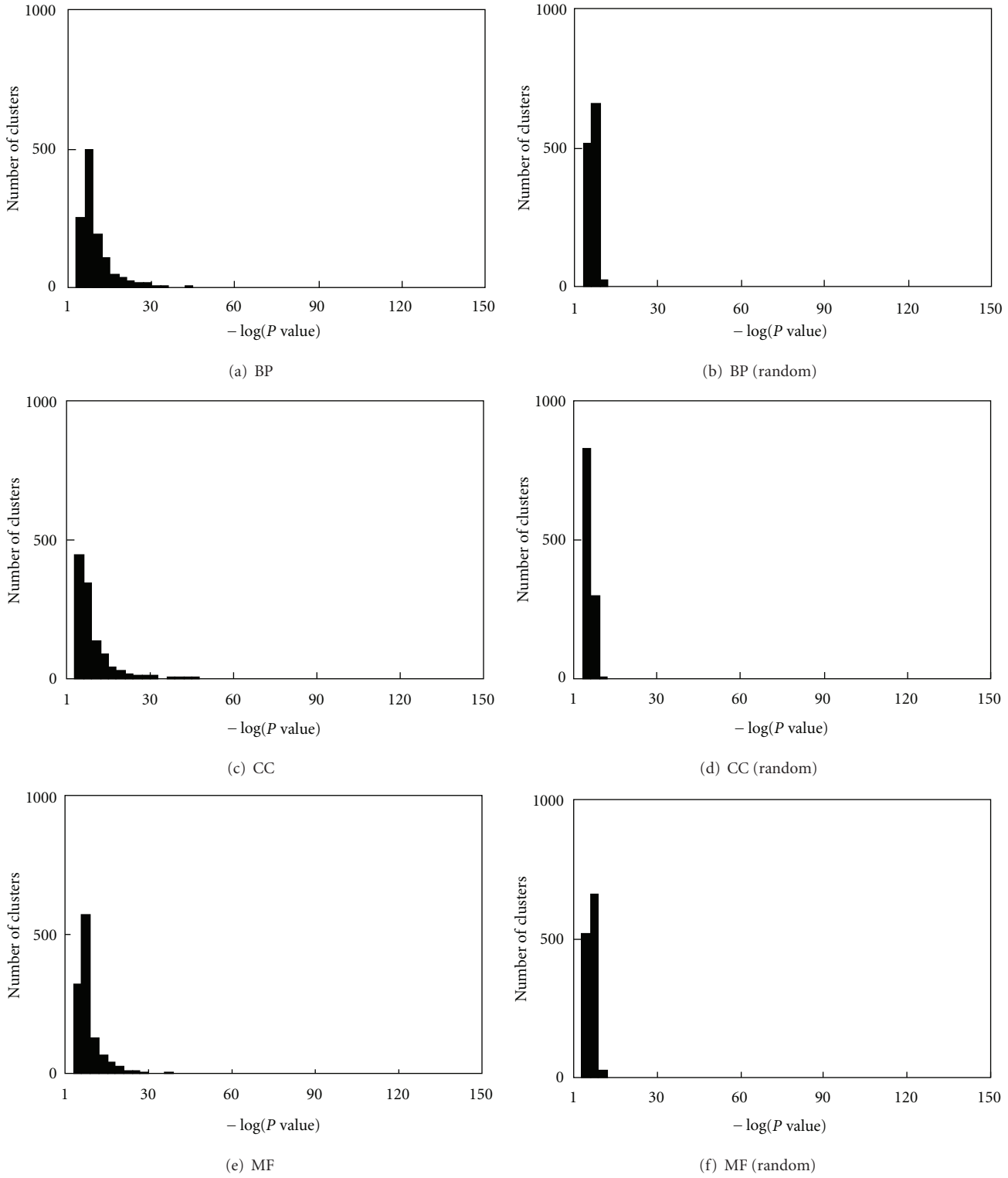

Figure 7: Comparison between the distributions of the star and star like modules and randomly selected protein groups with respect to – log(P value) in the contexts of three types of gene ontology terms: ((a) and (b)) biological process (BP), ((c) and (d)) cellular compartment (CC), and ((e) and (f)) molecular function (MF).

might be many star-like modules in a PPI network. Finally we found many star-like modules in the PPI network and came to the conclusion that a PPI network is a combination of both high-density and star-like modules. Also we verified that the star-like modules in a PPI network are rich with similar function proteins.

Acknowledgment

This paper was partially supported by National Bioscience Database Center (JST-NBDC), Japan.

References

[1] B. O. Palsson, *Systems Biology: Properties of Reconstructed Networks*, Cambridge University Press, Cambridge, UK, 2007.

[2] J. D. J. Han, "Understanding biological functions through molecular networks," *Cell Research*, vol. 18, no. 2, pp. 224–237, 2008.

[3] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, no. 6761, pp. C47–C52, 1999.

[4] N. Guelzim, S. Bottani, P. Bourgine, and F. Képès, "Topological and causal structure of the yeast transcriptional regulatory network," *Nature Genetics*, vol. 31, no. 1, pp. 60–63, 2002.

[5] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabási, "Hierarchical organization of modularity in metabolic networks," *Science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[6] J. B. Pereira-Leal, A. J. Enright, and C. A. Ouzounis, "Detection of Functional Modules from Protein Interaction Networks," *Proteins: Structure, Function and Genetics*, vol. 54, no. 1, pp. 49–57, 2004.

[7] A. C. Gavin, P. Aloy, P. Grandi et al., "Proteome survey reveals modularity of the yeast cell machinery," *Nature*, vol. 440, no. 7084, pp. 631–636, 2006.

[8] B. A. Shoemaker and A. R. Panchenko, "Deciphering protein-protein interactions. Part I. Experimental techniques and databases," *PLoS Computational Biology*, vol. 3, no. 3, article e42, pp. 0337–0344, 2007.

[9] Y. Ho, A. Gruhler, A. Heilbut et al., "Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry," *Nature*, vol. 415, no. 6868, pp. 180–183, 2002.

[10] M. Arifuzzaman, M. Maeda, A. Itoh et al., "Large-scale identification of protein-protein interaction of Escherichia coli K-12," *Genome Research*, vol. 16, no. 5, pp. 686–691, 2006.

[11] B. Schwikowski, P. Uetz, and S. Fields, "A network of protein-protein interactions in yeast," *Nature Biotechnology*, vol. 18, no. 12, pp. 1257–1261, 2000.

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

[13] M. Wu, X. Li, C. K. Kwoh, and S. K. Ng, "A core-attachment based method to detect protein complexes in PPI networks," *BMC Bioinformatics*, vol. 10, article 169, 2009.

[14] H. C. M. Leung, Q. Xiang, S. M. Yiu, and F. Y. L. Chin, "Predicting protein complexes from PPI data: a core-attachment approach," *Journal of Computational Biology*, vol. 16, no. 2, pp. 133–144, 2009.

[15] S. Srihari, K. Ning, and H. W. Leong, "Refining Markov Clustering for protein complex prediction by incorporating core-attachment structure," *Genome Informatics*, vol. 23, no. 1, pp. 159–168, 2009.

[16] ftp://ftpmips.gsf.de/yeast/catalogues/complexcat/.

[17] S. Pu, J. Wong, B. Turner, E. Cho, and S. J. Wodak, "Up-to-date catalogues of yeast protein complexes," *Nucleic Acids Research*, vol. 37, no. 3, pp. 825–831, 2009.

[18] J. D. Eblen, I. C. Gerling,, A. M. Saxton, J. Wu, J. R. Snoddy, and M. A. Langston, "Graph algorithms for integrated biological analysis, with application to type 1 diabetes data," in *Clustering Challenges in Biological Networks*, S. Butenko, W. A. Chaovalitwongse, and P. M. Pardalos, Eds., pp. 207–222, World Scientific, 2009.

[19] M. A. Langston, L. Lin, X. Peng et al., "A combinatorial approach to the analysis of differential gene expression data: the use of graph algorithms for disease prediction and screening," in *Methods of Microarray Data Analysis IV*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2005.

[20] E. Georgii, S. Dietmann, T. Uno, P. Pagel, and K. Tsuda, "Enumeration of condition-dependent dense modules in protein interaction networks," *Bioinformatics*, vol. 25, no. 7, pp. 933–940, 2009.

[21] M. Zhang, J. Deng, C. V. Fang, X. Zhang, and L. J. Lu, "Molecular network analysis and applications," in *Knowledge-Based Bioinformatics: From Analysis to Interpretation*, G. Alterovitz and M. Romani, Eds., pp. 253–287, John Wiley & sons, New York, NY, USA, 2010.

[22] M. Altaf-Ul-Amin, Y. Shinbo, K. Mihara, K. Kurokawa, and S. Kanaya, "Development and implementation of an algorithm for detection of protein complexes in large interaction networks," *BMC Bioinformatics*, vol. 7, article 207, 2006.

[23] M. Altaf-Ul-Amin, H. Tsuji, K. Kurokawa et al., "A density-periphery based graph clustering software mainly focused on detection of protein complexes in interaction networks," *Journal of Computer Aided Chemistry*, vol. 7, pp. 150–156, 2006.

[24] N. J. Krogan, G. Cagney, H. Yu et al., "Global landscape of protein complexes in the yeast Saccharomyces cerevisiae," *Nature*, vol. 440, no. 7084, pp. 637–643, 2006.

[25] http://dip.mbi.ucla.edu/.

[26] ftp://ftpmips.gsf.de/yeast/PPI/.

[27] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, no. 6833, pp. 41–42, 2001.

[28] http://geneontology.org/.

[29] S. Falcon and R. Gentleman, "Using GOstats to test gene lists for GO term association," *Bioinformatics*, vol. 23, no. 2, pp. 257–258, 2007.