Protein Complex Prediction via Bottleneck-Based Graph Partitioning

Jaegyoon Ahn[1]
ajk@cs.yonsei.ac.kr

Dae Hyun Lee[1]
mudjjang@yonsei.ac.kr

Youngmi Yoon[2]
ymyoon@gachon.ac.kr

Yunku Yeu[1]
yyk@cs.yonsei.ac.kr

Sanghyun Park[1,*]
sanghyun@cs.yonsei.ac.kr

[1] Department of Computer Science, Yonsei Univ., 3rd Engineering Bldg. 533-1, Shinchon-dong, Seodaemun-gu, Seoul, Korea, 0082-2-2123-7757

[2] Department of Computer Engineering, Gachon Univ., 1342 Seongnamdaero, Sujeong-gu, Seongnam-si, Gyeonggi-do, Korea, 0082-32-820-4393

ABSTRACT
Detecting protein complexes is one of essential and fundamental tasks in understanding various biological functions or processes. Therefore, precise identification of protein complexes is indispensible. For more precise detection of protein complexes, we propose a novel data structure which employs bottleneck proteins as partitioning points for detecting the protein complexes. The partitioning process allows overlapping between resulting protein complexes. We applied our algorithm to several PPI (Protein-Protein Interaction) networks of Saccharomyces cerevisiae and Homo sapiens, and validated our results using public databases of protein complexes. Our algorithm resulted in overlapping protein complexes with significantly improved F1 score, which comes from higher precision.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Data mining; J.3 [**Life and Medical Sciences**]: Biology and Genetics

General Terms
Algorithms

Keywords
Network clustering, Protein complex detection, Protein-protein interaction, Bottleneck protein

1. INTRODUCTION
Most proteins are known to be involved in complex biological processes or functions in a cell, forming a protein complex with other proteins [1]. Therefore, detecting protein complexes is one of essential and fundamental tasks in understanding various biological functions or processes. A protein complex can be modeled as an undirected graph of which node is a protein and edge is a physical interaction between two protein nodes. This physical interaction of two proteins is called PPI (Protein-Protein Interaction). Representative methods to find those interactions are two-hybrid system [2] and Mass Spectrometry [3]. Recent development of those high-throughput methods has resulted in abundant PPI network.

A protein complex is a set of proteins that interact with each other, so it is frequently assumed that distances between its member proteins are short, and its members tend to form clique-like structure in the PPI network. Accordingly, a protein complex is often assumed as a dense sub-graph in the PPI network. There have been active researches to develop algorithms for detecting protein complexes, and many of them are based on searching dense sub-graph in the PPI network. MCODE [4] gives high weight to nodes of which degree is high, and searches the network using those nodes as seeds. It enforces local search on the network, and finds sub-network whose nodes are highly interconnected. CMC [5] gives weight to PPIs using an iterative scoring method to assess the reliability of PPI, finds maximal cliques from the weighted PPI network, and then removes or merges overlapping maximal cliques based on their interconnectivity. MCL [6] detects clusters by distinguishing the strong and weak connections in the network and partitioning the network, based on manipulation of transition probabilities or stochastic flows between vertices of the graph. MCL has been reported to have good performance, and many variations of it have been proposed [7, 8, 9]. However, they are known to suffer from imbalance of resulting clusters [9].

These network clustering algorithms commonly do not allow overlapping between identified protein complexes. In other words, a protein can be involved in only one protein complex. Recently, algorithms that allow overlapping have been extensively studied. DPClus [10] detects initial protein complexes starting from the seeds and then including neighbors so as to maintain the edge's density of the sub-network above the threshold. Then it finds overlapped protein complexes extending the initial protein complexes. CFinder [11] is based on Clique Percolation Method

* To whom correspondence should be addressed.

DTMBIO'12, October 29, 2012, Maui, Hawaii, USA.

(CPM) [12], which defines a protein complex as a union of *k*-cliques that share (*k*-1) vertices. The result of CFinder is sensitive to the value of k. As *k* increases, it tends to find smaller, but highly denser sub-network. Link Cluster [13] firstly substitute edges to virtual nodes, and then make edge between those virtual nodes (edges) that share nodes. Virtual nodes of the substituted network are closer as their connectivity increase. Hierarchical clustering of those virtual nodes results in the clusters of the edges, and as a result, those clusters can share nodes. Allowing the overlaps between resulting protein complexes obviously leads to higher recall and precision, because a protein is frequently involved in several protein complexes [10].

Precise prediction of protein complexes is important since they are likely to be fundamental units for various biological functions or processes. Also, the validation cost of predicted protein complexes is high. For more precise detection of protein complexes, we used the characteristics of bottlenecks in the network. A bottleneck of a network is a node that the information of the network is concentrated. The bottleneckness of a node can be calculated using betweenness centrality, which is a measure of a node's centrality in a network, and equal to the number of shortest paths going through it. Yu *et al.* [14] revealed that bottleneck proteins tend to be essential proteins and correspond to the dynamic component of the PPI network. Moreover, they can be global connectors between functional modules of the PPI network. Therefore, sub-graphs of which boundary proteins are bottleneck proteins have higher chance to be functional modules. We expected that finding these sub-graphs as candidate protein complexes will efficiently filter the possible false predictions out.

We designed the algorithm that exploits the bottleneck proteins as partitioning points for detecting the protein complexes. Our algorithm iteratively constructs directed acyclic graphs of which starting node is bottlenecks in the PPI network. The search ends at nodes where flows from the starting node are concentrated. This graph is called DG (Distance Graph), and terminal nodes of DG tend to be bottlenecks of the PPI network. Established DGs are used to identify sub-graphs that may be overlapped with each other. The sub-graphs having enough edge-density are reported as protein complexes.

We applied our algorithm to several PPI networks of Saccharomyces cerevisiae and Homo sapiens, and validated our results using public databases of protein complexes. Our algorithm resulted in significantly improved F1 score, which comes from higher precision. This result supports our expectation that use of bottleneck proteins brings in precise prediction.

2. ALGORITHM

2.1 Overview

The protein complex detection method proposed in this study is composed of four parts. First, we calculate betweenness centrality of all the nodes in the PPI network. Second, for each node of which centrality betweenness is greater than given threshold (i. e. bottleneck), we build DG. Section 2.2 describes the details of DG. Third, candidate protein complexes are detected using DGs built in the second part. Among those candidates, ones that have high edge-density are reported as protein complexes. Details of aforementioned process are described in Section 2.3.

2.2 Distance Graph

DG of bottleneck node *A* is a directed acyclic graph of which starting (root) node is *A*. DG resembles a level-order tree, except that two or more nodes can be parent of a node. Nodes that are directly connected to *A* become child nodes of *A*. Levels of the root node *A* and its child nodes are 0 and 1, respectively. For all the level (*n* - 1) nodes, directly connected nodes become their child nodes of which level is *n*. When the level of node *B* is *n*, the nodes of which level is less than or equal to *n* are not allowed to be child nodes of *B*. We do not include nodes that have been already included in the tree. If a child node has two or more parent nodes, no more expanding is allowed from this node. When there are no more nodes to be expanded, the building process will be terminated.

If the process encounters a node that has two or more parent nodes, expanding is not allowed. That means these nodes tend to be bottlenecks of the network. Because the root node is also bottleneck, DG can be defined as a sub-graph of which boundaries are bottleneck nodes.

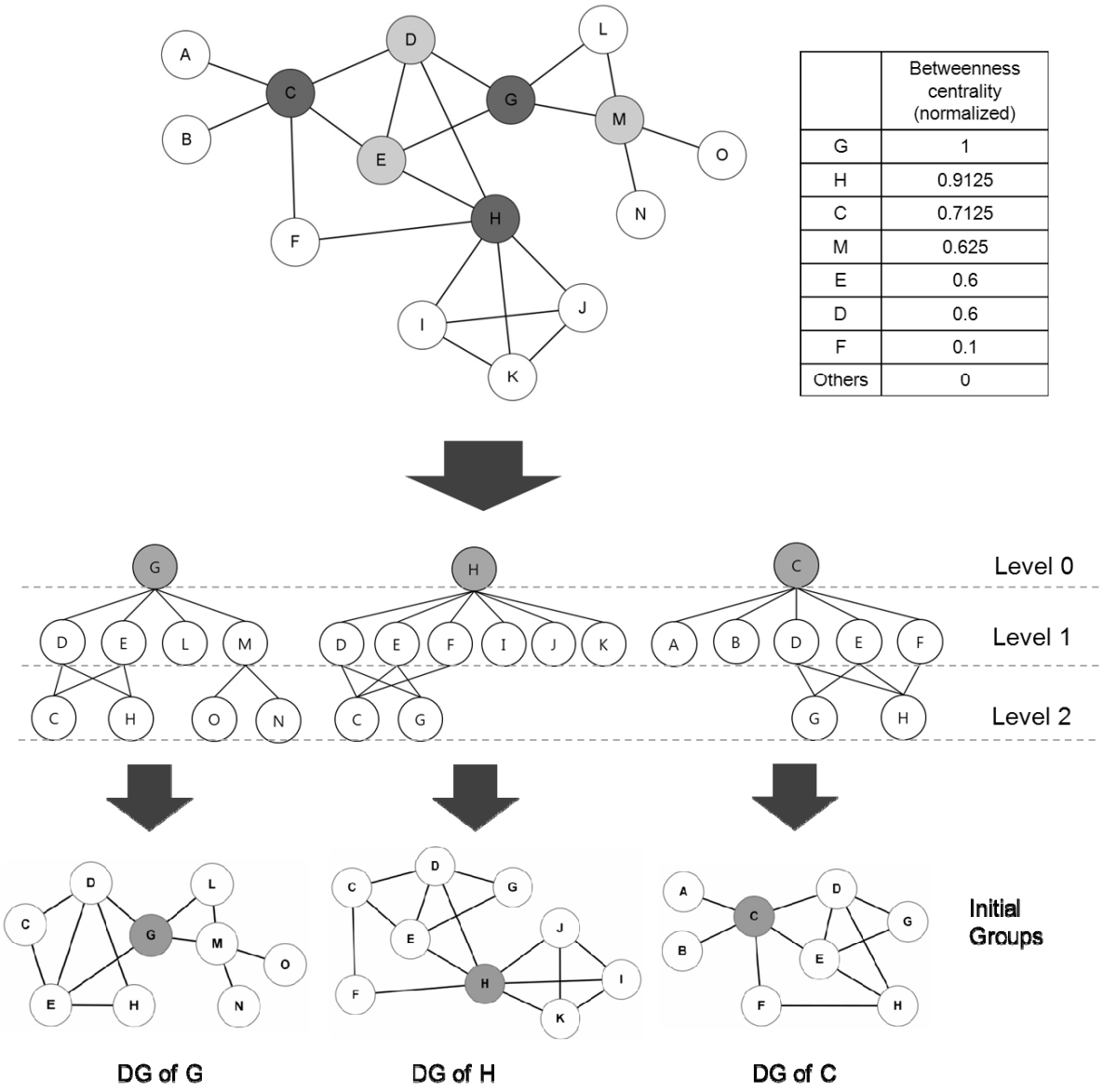
As we have described in Section 2.1, betweenness centrality of each node in the network is calculated, then the DG is built for top *BC* percent nodes when sorted by its betweenness centrality score in descending orders. *BC* is a user parameter for betweenness centrality threshold. In Figure 1, DGs of bottleneck nodes G, H and C are built in turn. For bottleneck node G, directly connected nodes D, E, L and M become child nodes of G, and are expanded in turn. First, C and H are visited by D and included. Then, they are visited again by E, and included. Because node C and H have more than one parent node, they are no longer expandable. Note that terminal nodes of DG of C, G and H are all bottlenecks, A, B, O, and N, however, are nodes without any child nodes. Also, note that there is no parent-child relationship between siblings D and E. Since E is child node of G, it is not expandable from D.

2.3 Detecting Protein Complexes

All nodes in the DG form some initial groups to detect protein complexes. In Figure 1, we can see resulting three initial groups.

Simply, the first initial group divides the whole PPI network into two or more parts. Divided parts are further divided by the second initial group, and so on. Dividing process ends when there is no initial group left to divide parts, and reports the remaining parts as candidate protein complexes.

When the part of network is divided by the initial group into part *A* and part *B*, these two parts share nodes that are marked with bottleneck of the initial group (DG). This process is illustrated in Figure 2. Initial PPI network (part 1 in Figure 2) is divided into parts 2, 3, 4 and 5 by DG of G. Node C and H are bottlenecks and terminal nodes of DG of G. Therefore parts 2 and 5 share node C, parts 3 and 5 share node C and H, and parts 4 and 5 share node H. This sharing enables us to detect overlapping protein complexes.



	Betweenness centrality (normalized)
G	1
H	0.9125
C	0.7125
M	0.625
E	0.6
D	0.6
F	0.1
Others	0

Figure 1. Example for building DGs. First, Betweenness centrality of each node in the PPI network is calculated. Then DGs of nodes that have high betweenness centrality (bottlenecks, gray nodes in the example graph) are built. In this example, DGs of C, G, and H are built. Each DG forms an initial group to detect protein complexes.

Parts 2, 3, 4 and 5 are examined whether they can be further divided, by remaining initial groups. In the example of Figure 2, part 5 is divided into 2 parts (parts 6 and 7) by DG of H. Likewise, parts 6 and 7 share node G. Lastly, parts 2, 3, 4, 6 and 7 are examined whether they can be divided. Since none of above can be divided by remaining initial group further, DG of C, it halts the process and reports them as candidate protein complexes.

Finally, clustering coefficient of each candidate protein complexes is calculated, and candidates of which clustering coefficient $\geq CC$ are reported as protein complexes. Clustering coefficient is widely used measure for calculating edge density of the graph, and has been adopted by many protein complex detection algorithms including MCODE [4]. *CC* is a user parameter for clustering coefficient threshold.

3. RESULT AND DISCUSSION

3.1 EXPERIMENTAL ENVIRONMENT

We downloaded two PPI networks of Saccharomyces cerevisiae (yeast) from DIP [15] and BioGRID [16] database. Also, 109,086 human PPIs were downloaded from the I2D database [19]. PPIs from DIP are biologically validated, thus the number of PPIs is relatively small, but they tend to be more accurate. Meanwhile, BioGRID has about ten times more PPIs than DIP. BioGRID has many predicted PPIs, which result in much higher false positive error rate. Table 1 shows the information of the PPI network datasets.

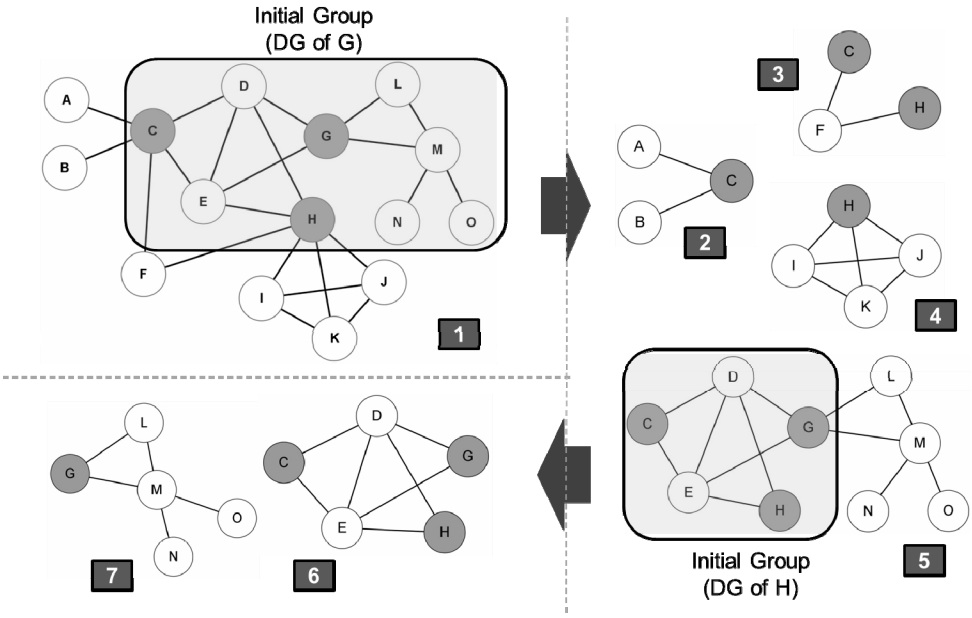


Figure 2. Detecting protein complexes. Small gray squares indicate the number of the part. Initial PPI network (part 1) is divided into four parts (2, 3, 4 and 5) by DG of G. Among these, part 5 is further divided into 2 parts (parts 6 and 7) by DG of G. Because parts 2, 3, 4, 6 and 7 are not divisible by DG of C, they are reported as candidate protein complexes.

Table 1. PPI network datasets

Database (version)	Species	Number of proteins	Number of PPIs
DIP (20071007)	Saccharomyces cerevisiae	4,823	16,914
BioGRID (3.1.69)	Saccharomyces cerevisiae	5,920	162,378
I2D (1.95)	Homo Sapiens	14,610	209,440

Table 2. Reference datasets

Database (version)	Species	Number of protein complexes	Number of proteins	Average number of proteins in protein complexes
MIPS	Saccharomyces cerevisiae	81	885	12.358
CYC2008 (2.0)	Saccharomyces cerevisiae	236	1,627	6.678
CORUM (17.02.2012)	Homo Sapiens	1,942	4,394	5.789

We also collected known protein complexes (reference) to validate the results of our algorithm. Two reference datasets of Saccharomyces cerevisiae were downloaded from MIPS [17] and CYC2008 [18] database. One reference dataset of Homo sapiens was downloaded from CORUM database [20]. For both reference datasets and identified protein complex sets, we used complexes of which size is more than or equal to three. Table 2 shows the information of collected reference datasets.

3.2 PERFORMANCE TEST

To see whether a complex identified by an algorithm is matched with protein complexes in the reference datasets, we used affinity score. Given set of proteins in a protein complex in a reference dataset and set of proteins in an identified protein complex, which we call A and B respectively, affinity score between A and B can be calculated by the following formula.

$$\mathrm{aff}(A, B) = n\,(A \cap B)^2 / (n(A) \times n(B))$$

The searching is successful if a protein complex is identified with affinity score ≥ 0.2 for any protein complex in a reference datasets. If this threshold is too big or small, the affinity score loses its assessment function. Through iterative experiments, we set the affinity score threshold as 0.2, which makes the difference between results of various algorithms.

The performance of a clustering algorithm can be measured using recall, precision and F1 score, which are calculated as follows:

$$\mathrm{Recall} = |R_{hit}| / |R|,\ \mathrm{Precision} = |C_{hit}| / |C|,$$

$$\text{F1 score} = \text{harmonic mean of Recall and Precision},$$

$$R_{hit} = \{\, R_i \in R \mid \mathrm{aff}(R_i, C_j) \geq 0.2,\ C_j \in C \,\},$$

$$C_{hit} = \{\, C_i \in C \mid \mathrm{aff}(C_i, R_j) \geq 0.2,\ R_j \in R \,\},$$

where C is a set of protein complexes found by a clustering algorithm, and R is a set of protein complexes in a reference dataset. Recall means a rate of protein complexes in the reference datasets that were successfully found, precision means a rate of protein complexes identified by an algorithm that are matched

with the protein complexes in the reference datasets, and F1 score means an overall accuracy of the test.

First, we tested the performance of proposed algorithm varying two user parameters, *BC* and *CC*. We can confirm that the optimal *BC* is around 20% to 30% in all the graphs in Figure 3.

Three graphs using DIP and I2D datasets (DIP-MIPS, DIP-CYC and I2D-CORUM) shows optimal *CC* range from 0.35 to 0.5. However, two graphs using BioGRID dataset show different tendency than other three graphs. The supposed reason is that BioGRID has large number of predicted PPI, which leads to higher false positive complex predictions. Therefore, the precision would decrease unless *CC* is high enough, as shown in these two graphs.

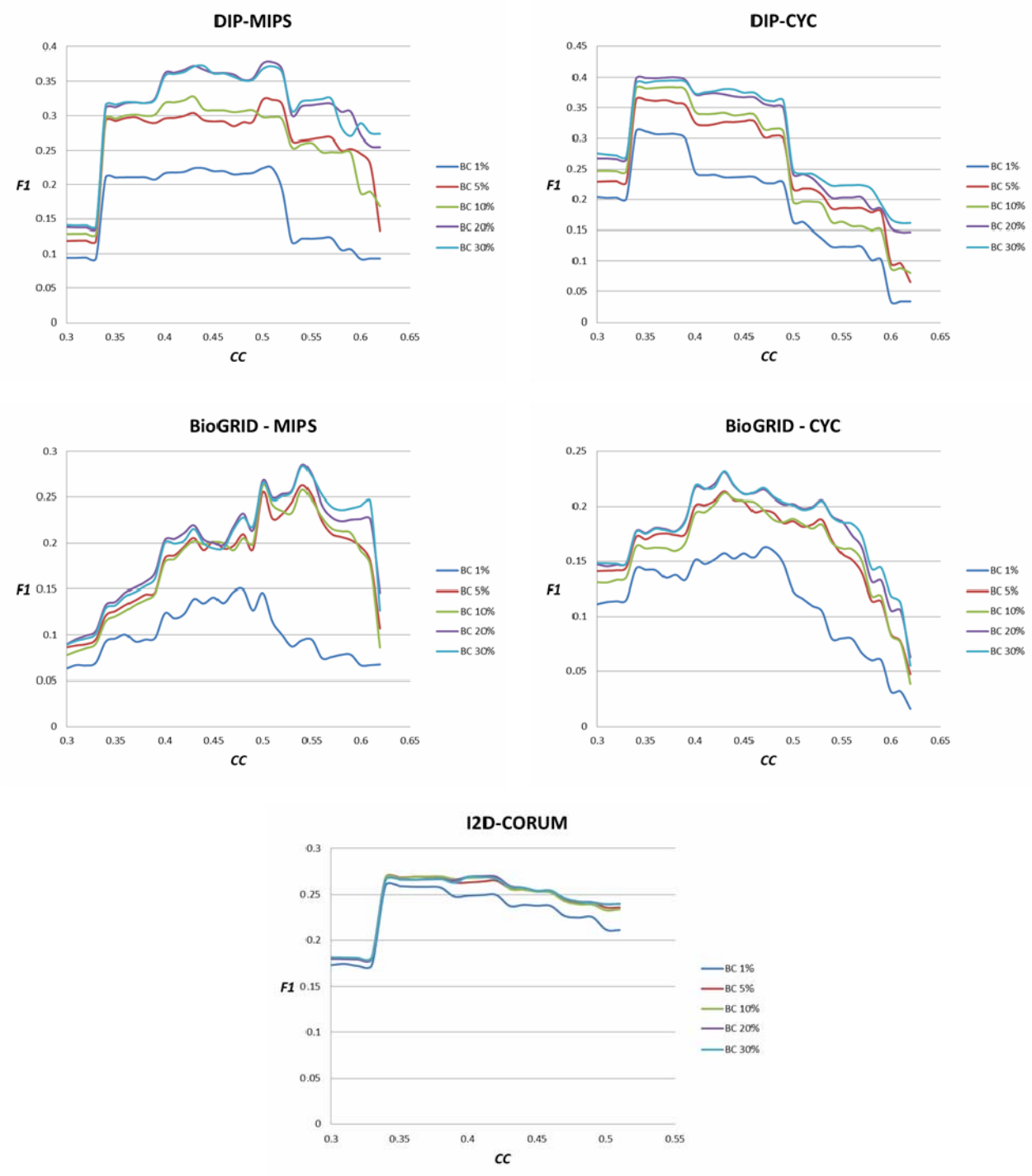


Figure 3. Experimental results for obtaining optimal user parameters. Each title of the graph indicates "PPI network dataset – reference dataset"

Table 3. Result of comparison test

Protein interaction network dataset	Reference dataset	Algorithm	Optimal parameters	Number of protein complexes	Recall	Precision	F1 score
DIP	MIPS	Proposed	*CC* = 0.51, *BC* = 20%	76	0.3210	0.4605	0.3783
		Link Cluster	Partition_density = 0.30	1,177	0.7037	0.1427	0.2373
		MCL	Granularity = 2.00	614	0.5679	0.0739	0.1298
		MCODE	Node_score = 0.10	83	0.2930	0.2530	0.2729
	CYC	Proposed	*CC* = 0.38, *BC* = 20%	333	0.3898	0.4114	0.4003
		Link Cluster	Partition_density = 0.29	1,179	0.5932	0.2858	0.3857
		MCL	Granularity = 2.40	639	0.4746	0.1690	0.2493
		MCODE	Node_score = 0.10	83	0.2119	0.5542	0.3065
BioGRID	MIPS	Proposed.	*CC* = 0.54, *BC* = 20%	69	0.2346	0.3623	0.2848
		Link Cluster	Partition_density = 0.30	10,463	0.5926	0.0893	0.1552
		MCL	Granularity = 3.60	216	0.2099	0.0556	0.0879
		MCODE	Node_score = 0.10	120	0.086	0.0500	0.0633
	CYC	Proposed	*CC* = 0.43, *BC* = 30%	324	0.2500	0.2160	0.2318
		Link Cluster	Partition_density = 0.28	10,915	0.5297	0.2802	0.3697
		MCL	Granularity = 3.00	225	0.1144	0.1111	0.1127
		MCODE	Node_score = 0.10	120	0.0593	0.1167	0.0787
I2D	CORUM	Proposed	*CC* = 0.41, *BC* = 20%	1,132	0.2961	0.2491	0.2706
		Link Cluster	Partition_density = 0.21	8,033	0.4576	0.1595	0.2378
		MCL	Granularity = 1.60	750	0.0623	0.0587	0.0604
		MCODE	Node_score = 0.10	251	0.0469	0.1076	0.0652

The implication of two parameters used in the proposed method is simple and straightforward. As *BC* gets bigger, the algorithm will create more DGs, thereby creating more complexes after division. Higher *CC* results in denser protein complexes, and the number of complex is decreased.

We then measured the prediction performance of proposed algorithm, and compared the results with representative network clustering algorithms, MCODE [4], MCL [5] and Link Cluster [12]. We applied each algorithm including proposed algorithm to PPI networks and two reference datasets. For each algorithm, we found optimal parameters that result in best F1 score.

In Table 3, we can see that proposed algorithm has overall high F1 score due to high precision. This means exploiting the bottleneck proteins as partitioning points of the PPI network efficiently filter the possible false predictions out. Generally, recall and precision tends to increase and decrease, as the number of detected protein complexes increases, respectively. However, in most cases, the proposed algorithm shows higher precision than MCL and MCODE, while the number of detected protein complexes is greater than theirs. Because biological validation of protein complex is expensive, it is indispensable that algorithm has high precision score, and predicts accurate protein complex.

Not only proposed algorithm shows highest precision in most of cases, but also it represents relatively high recall. In case of BioGRID-MIPS experiment in Table 3, the proposed algorithm finds one-thirds of protein complexes compared to the MCL. However, recall of our method is higher than those of MCL, and MCODE.

We can see that proposed algorithm shows lower F1 score than Link Cluster in BioGRID-CYC experiment. The supposed reasons are that the average number of proteins in the protein complexes of CYC is smaller than MIPS, and Link Cluster seems to be good at detecting small sized complexes. These explanations can be also supported by DIP-CYC and I2D-CORUM experiments (The average size of complexes of CORUM is also smaller than that of MIPS), where improvement of F1 over Link Cluster is not significant. Therefore, it is likely that proposed algorithm is more suited for detecting higher sized protein complexes.

Another strength of the proposed algorithm is that it allows overlapping between resulting protein complexes. In Figure 4, YBR081C and YDR448W are shared among two protein complexes, SILK (SAGA-like) complex and Ada2/Gcn5/Ada3 Transcription activator Complex. In our approach, the overlapped portion may include interactions as well as single node (for example, YBR081C-YDR448W), while Link Cluster allows

overlapping of only single node. In addition, we can confirm that the YBR081C and YDR448W are both bottleneck proteins of the network, as clearly shown in Figure 4.

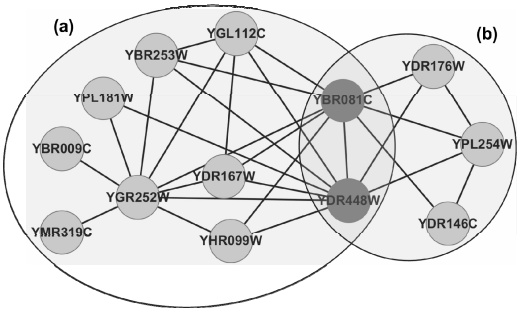


Figure 4. Example protein complexes, (a) SILK (SAGA-like) complex, (b) Ada2/Gcn5/Ada3 Transcription activator Complex. Both complexes were predicted using DIP dataset, and annotated using GO database (p-value < 0.01).

4. CONCLUSION

We proposed the novel network clustering algorithm for search protein complexes from the protein-protein interaction network. Also, our algorithm is capable of finding protein complexes which allow overlapping with each other, based on the fact that one protein can be involved in many biological functions or processes. As a result, the proposed method exhibited a significantly improved F1 score, which comes from higher precision.

As future works, we extend our algorithm to detect the hierarchical relationship between sub-networks identified. This algorithm would help us to elucidate hierarchical structure of various protein complexes or functional modules in a cell, and to infer a function of them in conjunction with various biology databases such as Gene Ontology database.

5. ACKNOWLEDGMENTS

This work was supported by National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012010775).

6. REFERENCES

[1] Kumar A. and Snyder M. 2002. Protein complexes take the bait. *Nature* 415 123-124.

[2] Fields S. and Song O. 1989. A novel genetic system to detect protein–protein interactions. *Nature* 340 245-245.

[3] Ho Y., Gruhler A., Bader G. D., Moore L., Adams S. L., Miller A. *et al.* 2002. Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry, *Nature* 415 180-183.

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

[5] Liu G., Wong. L. and Chua H. N. 2009. Complex discovery from weighted PPI networks. *Bioinformatics* 25(15) 1891-1897.

[6] Dongen S. V. 2000. Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht.

[7] Brohee S. and van Helden J. 2006. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics* 7:488.

[8] Vlasblom J. and Wodak S. 2009. Markov clustering versus affinity propagation for the partitioning of protein interaction graphs. *BMC bioinformatics* 10:99.

[9] Satuluri V., Parthasarathy S. and Ucar D. 2010. Markov Clustering of Protein Interaction Networks with Improved Balance and Scalability. *ACM-BCB 2010*, 247-256.

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

[11] Adamcsek B., Palla G., Farkas I., Derenyi I. and Vicsek T. 2006. CFinder:locating cliques and overlapping modules in biological networks. *Bioinformatics* 22(8) 1021-1023.

[12] Palla G., Derenyi I., Farkas I. and Vicsek T. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435 814-818.

[13] Ahn Y., Bagrow J. P. and Lehmann S. 2010. Link communities reveal multiscale complexity in networks. *Nature* 466 761-765.

[14] Yu H., Kim P. M., Sprecher E., Trifonov V. and Gerstein M. 2007. The Importance of Bottlenecks in Protein Networks: Correlation with Gene Essentiality and Expression Dynamics. *PLoS Comp. Biol.* 3(4): e59. doi:10.1371/journal.pcbi.0030059.

[15] Salwinski L., Miller C. S., Smith A. J., Pettit F. K., Bowie J. U., and Eisenberg D. 2004. The database of interacting proteins: 2004 update. *Nucleic Acids Research* 32(Database issue) D449-D451.

[16] Stark C., Breitkreutz B. J., Reguly T., Boucher L., Breitkreutz A., and Tyers M. 2006. BioGRID: a. general repository for interaction datasets. *Nucleic Acids Research* 34(Database issue) D535-D539.

[17] Güldener U., Münsterkötter M., Kastenmüller G., Strack N., van Helden J., Lemer C. *et al.* 2005. CYGD: the comprehensive yeast genome database. *Nucleic Acids Research* 33(Database issue) D364-D368.

[18] Pu S., Wong J., Turner B., Cho E. and Wodak S. 2009. Up-to-date catalogues of yeast protein complexes. *Nucleic acids research* 37(3) 825-831.

[19] Brown K. R. and Jurisica I. 2007. Unequal evolutionary conservation of human protein interactions in interologous networks. *Genome Biol.* 8 R95.

[20] Ruepp A., Waegele B., Lechner M., Brauner B., Dunger-Kaltenbach I., Fobo G. *et al.* 2010. CORUM: the comprehensive resource of mammalian protein complexes-2009. *Nucleic Acids Research* 38(Database issue) D497-501.