An Edge Based Core-Attachment Method to Detect Protein Complexes in PPI Networks

Yu Wang, Lin Gao*, Zhe Chen
School of Computer Science and Technology, Xidian University
Xi'an,710071, China
cheerwangyu@163.com; lgao@mail.xidian.edu.cn; xschenzhezhtx@yahoo.com.cn

Abstract— **Characterization and identification of protein complexes in protein-protein interaction (PPI) networks is important in understanding cellular processes. With the core-attachment concept, a novel core-attachment algorithm is proposed by characterizing the protein complex core from the perspective of edges. We reinvite a protein complex core to be a set of closely interrelated edges rather than a set of interrelated proteins. We first identify the edges must belong to a core, and then partition these edges to extract cores. After that, we select the attachments for each complex core to form a protein complex. Finally, we evaluate the performance of our algorithm by applying it on two different yeast PPI networks. The experimental results show that our algorithm outperforms the MCL, CPM, CoAch in terms of number of precisely predicted protein complexes, localization as well as GO semantic similarity. Our proposed method is validated as an effective algorithm in identifying protein complexes and can provide more insights for future biological study. It proves that edge community is a better topological characterization of protein complex.**

Keywords-protein-protein interaction networks; protein complex; edge community; core-attachment

I. INTRODUCTION

Protein complexes encompass groups of genes or proteins involved in common elementary biological processes [1]. They play a critical role in integrating multiple gene products to perform useful cellular functions. Identifying protein complexes is an important and challenging task in post genomic era.

Many computational methods have been proposed to predict protein complexes in PPI networks, such as MCL [2], MCODE [3], RNSC [4] and CPM [5]. Most of the existing methods relied on the assumption that proteins within the same complex would have relatively more interactions [6]. So the problem of detecting protein complexes is translated into finding dense sub-graphs in PPI networks.

Gavin et al. [7] took a further study on the organization of protein complexes, demonstrating that a protein complex should generally contain a core and attachments. Core proteins have relatively more interactions among themselves and each protein complex has a unique set of core proteins. Each attachment protein usually binds to two or more core proteins depending on the size of core protein set. Based on this core-attachment concept, Wu et al. [8] presented the CoAch algorithm, which extracted the protein complex cores from each vertex's neighbor hood graph. Leung et al. [9] proposed the CORE algorithm, a statistical framework to identify protein complex cores. Both approaches outperform the existing non core-attachment based computational methods dramatically, demonstrating the significance of the core attachment structure.

Recently, Ahn et al. [10] suggested an unorthodox approach which reinvented communities as groups of edges rather than vertices. They defined the similarity between each pair of adjacent edges. After that, they employed an agglomerative hierarchical clustering technique to build a dendrogram where each leaf is an edge from the original network and branches represent edge communities. Since all the existing core-attachment approaches focus their attention on grouping vertices, it is natural to consider that whether we can characterize protein-complex cores and attachments based on edges or not.

Inspired by this insight, in this paper, we develop an algorithm to detect protein-complex cores and attachment proteins in PPI networks. Quite different from all the existing core-attachment approaches, our algorithm characterizes and identifies protein-complexes based on edges. The key idea of our algorithm consists of two main stages: (1) detect all the complex cores. We first identify the edges must belong to a core. The bridgeness [11] and clique size [12] of an edge are employed with an immediate purpose to differentiate the roles of edges: edges in cores, edges out of cores. After collecting all the in-core edges, we cluster these edges by the edge similarity to obtain the edge communities. The induced vertices of an edge community are considered as proteins in the protein-complex core; (2) identify attachments for each core to form protein complexes. We apply our algorithm on two different yeast PPI networks and the experimental results show that compared with MCL, CPM, and CoAch, our algorithm can discover protein-complex precisely and get a better score in localization as well as in GO semantic similarity.

II. PRELIMINARIES

Prior to the detail description of our concrete algorithm, let us introduce some conceptions widely used in the forthcoming sections.

978-1-4577-1666-9/11/$26.00 ©2011 IEEE

A. *Bridgeness*

Edges in a network can be divided into two kinds according to their different roles: some enhance the locality like the ones inside a cluster, others contribute to the global connectivity like the ones connecting various clusters. In order to differentiate which kind of roles an edge plays, Cheng et al. propose an index called bridgeness [11] to quantify the edge significance in maintaining connectivity. The bridgeness of an edge E is defined as

$$B_E = \frac{\sqrt{S_x S_y}}{S_E}, \qquad (1)$$

where x and y are the two vertices of the edge E. S_x, S_y and S_E are the clique sizes of x, y and E. The clique size of a vertex or an edge is defined as the size of the maximum clique that contains this vertex or this edge [12].

B. *Edge similarity*

Ahn et. al [10] defined the inclusive neighbors of a vertex i as:

$$n_+(i) \equiv \{x \,|\, d(i,x) \le 1\}, \quad (2)$$

where $d(i,x)$ is the length of the shortest path between vertices i and x. The set simply contains the vertex itself and its neighbors. From this, the similarity between edges e_{ik} and e_{jk} is

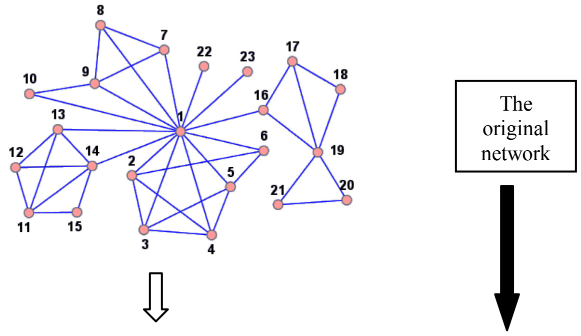
$$S(e_{ik}, e_{jk}) = \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cup n_+(j)|} \qquad (3)$$

C. *Induced vertices*

For a graph $G=(V,E)$, P is a set of edges and $P \subset E$. The induced vertices set of P is defined as $\cup_{e_{ij}\in P}\{i,j\}$.

III. Method

In this section we discuss the steps of our algorithm in turns. To explain more intuitively our algorithm, an example is presented in Fig.1.



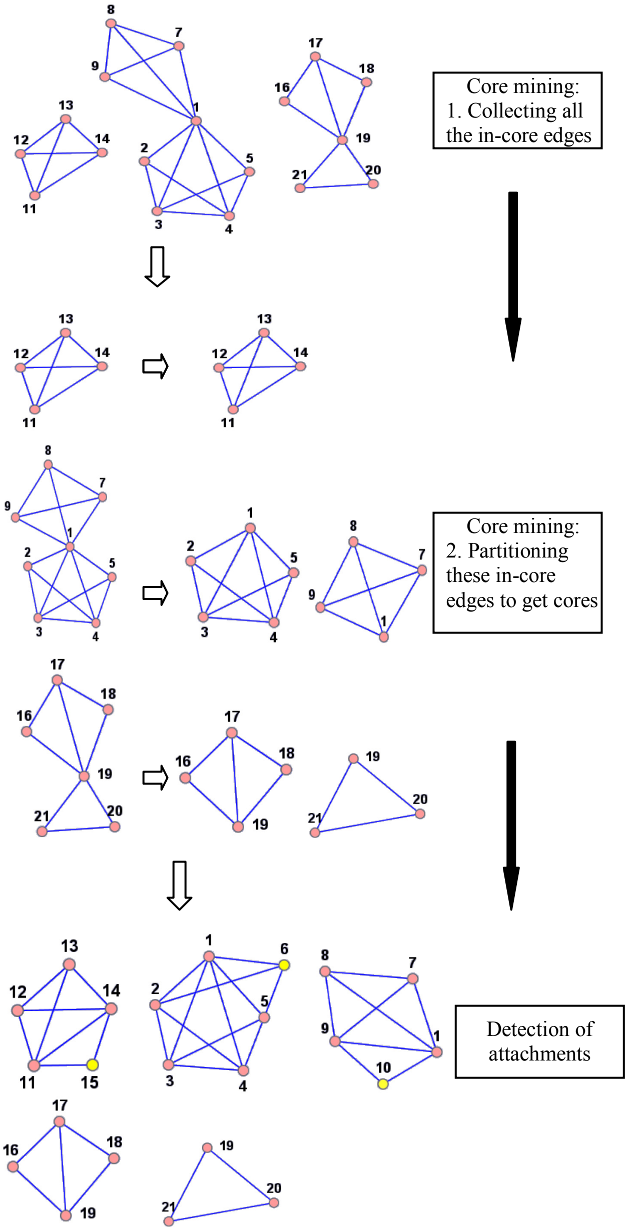

Core mining:
1. Collecting all the in-core edges
Core mining:
2. Partitioning these in-core edges to get cores
Detection of attachments

Fig. 1. The diagram of our method. The procedure of our algorithm mainly consists of two stages: detection of protein complex cores and identification of attachments. The vertices in red color are cores and the vertices in yellow color are attachments.

A. *Characterizing the protein complex core via edges*

A protein complex core is a small group of proteins which show a high mRNA co-expression patterns and share high degree of functional similarity. It is the key functional unit of the complex and largely determines the cellular role and essentiality of the complex. Although no common definition has been agreed upon, it is usually accepted that protein-complex cores correspond to small, dense and reliable sub-graphs in PPI networks [7].

In this section, we try to characterize protein complex cores via edges. Since protein complex cores are dense and small, it is very plausible to take clique whose size is larger than 3 as a core. Because the proteins in the same core share

978-1-4577-1666-9/11/$26.00 ©2011 IEEE

high degree of functional similarity, which implies that the proteins in the same core are similar in topology. Therefore, we have enough reasons to believe that the bridgeness of an edge in a protein complex core should be as low as possible. Likewise, the edges in the same core share high degree of similarity.

Based on the analysis above, a protein complex core we defined should satisfy following three constraints:

(1) The clique size values of edges in a core are large enough, say 4;

(2) The bridgeness values of edges in a core are relatively low, implying that they function as an intra-edges contributing to the local connectivity of networks involved;

(3) The topological similarity between any edge pairs should be closely interrelated.

A subgraph consists of 3 clique or a combination of 3 cliques could also be regarded as a core if the bridgeness of all the edges in this sub-graph equals to 1 and the similarity between any edge pairs is relatively high.

B. Extracting the protein complex core

Based on the definition of protein complex cores, we are now ready to describe our proposed algorithm to extract them. In our algorithm, we first pick up the edges that must be in cores. Although these edges are in-core edges, they are in different cores. In order to decide which edges belong to the same core, we group them by the similarities between them. In this way we get some edge community and consider the edges in the same community are the edges in the same core. The induced subgraphs of the edge communities are the cores we want to extract.

Algorithm 1 illustrates the overall framework to extract protein-complex cores. For each edge e in the PPI network $G=(V,E)$, we calculate its clique size and bridgeness value in line 3-4. Then we construct a virtual graph $G^{'}$ in line 6. The vertices in $G^{'}$ are the same vertices in G and there is no edge in $G^{'}$ initially. In line 7-13, we check every edge in graph G and add to graph $G^{'}$ the edges that satisfy the constraints in line 8 or line 10; therefore, the graph $G^{'}$ only contains the edges belonging to cores and we can regard graph $G^{'}$ as a combination of protein complex cores. After that, in line 16, we use edge-partition algorithm to extract protein complex cores in each connected component in $G^{'}$, and the details of edge-partition algorithm are described in Algorithm 2.

Algorithm 1. Protein-complex core extracting algorithm

```
Input: The PPI network $G=(V,E)$;
       Bridgeness threshold $\omega$;
       Similarity threshold $\tau$;
Output: The set of protein-complex cores, SC.
1: $SC=\Phi$; //initialization
2: for each edge $e \in E$ do
3:     calculate the clique size $S_e$ of $e$;
4:     calculate the bridgeness $B_e$ of $e$;
5: endfor
6: construct a virtual graph $G^{'}=(V^{'},E^{'})$, satisfying
   $V^{'}=V$ and $E^{'}=\Phi$;
7: for each edge $e \in E$ do
8:     if $S_e \geq 4$ and $B_e < \omega$ do
9:         add the same edge to $G^{'}$;
10:    elseif $S_e = 3$ and $B_e = 1$ do
11:        add the same edge to $G^{'}$;
12:    endif
13: endfor
14: obtain a set of connected components in $G^{'}$;
15: for each connected component comp do
16:    Cores=edge-partition(comp);
       // Cores is a set storing the protein complex cores
       // identified in the connected component comp
17:    insert all the elements in set Cores to SC;
18: endfor
```

Algorithm 2 is used to cluster those in-core edges. It needs a parameter, called similarity threshold τ, to decide the cluster granularity. For each edges in the component $comp=(V_{cp},E_{cp})$, if it isn't in a known community, we will insert it into an empty set named by $edges\_core$ in line 7. Then we find the edges similar with any of the edge in set $edges\_core$ in line 9, and insert these edges into set $edges\_core$ in line 10 (When we talk about two edges are similar, we mean the similarity between them is bigger than the parameter τ. In our experiments, τ is set to 0.6). This process will stop until we couldn't insert any edge into set $edges\_core$. The edges in $edges\_core$ constitute a new edge community. If there are more than one element in set $edges\_core$, the induced vertices of the $edges\_core$ will be regarded as a protein-complex core and inserted into set *Cores* in line 15-17.

Algorithm 2 edge-partition (*comp*)

```
Input: The connected component $comp=(V_{cp},E_{cp})$;
       Similarity threshold $\tau$;
Output: The set of protein-complex cores, Cores
1: $Cores=\Phi$;
2: $Searched\_Edges=\Phi$;
   // Set $Searched\_Edges$ stores the edges in known
   // communities.
3: for each edge $i \in E_{cp}$ do
4:   if $i \notin Searched\_Edges$ do
5:     insert $i$ into $Searched\_Edges$;
6:     $edges\_core=\Phi$;
7:     insert $i$ into $edges\_core$;
8:     for each edge $j \in edges\_core$ do
9:       if $\exists$ edge $k \notin Searched\_Edges$
         and $S(k,j) > \tau$ do
```

978-1-4577-1666-9/11/$26.00 ©2011 IEEE

```
        insert $k$ into $edges\_core$;
        insert $k$ into $Searched\_Edges$;
      endif
    endfor
  endif
  if $size(edges\_core) > 1$ do
    // $size(edges\_core)$ is the number of edges in
    // set $edges\_core$
    insert $vertices\_core$ into $Cores$
    // $vertices\_core$ is the induced vertices of $edges\_core$
  endif
endfor
return $Cores$
```

C. *Detecting attachment proteins*

After obtaining all the protein complex cores, what remains to do is to extract the peripheral information of each core and select the reliable attachments cooperating with them to form a protein complex.

We still detect attachment proteins via edges. Given a PPI network $G=(V,E)$, the set of all the in-core edges in G is labeled as E_{core}^{in}. For a complex core $C=(V_C,E_C)$, the attachment edge connecting a core protein in C and an attachment protein of C is defined as:

$$attach(C)=\{p \mid p\in E, p\notin E_{core}^{in}, \exists q\in E_C\ s.t. S(p,q)<\omega\} \quad ,$$

where ω is a closeness parameter which is used to control the closeness of the attachment proteins and the core. In our experiments, ω is set to 0.4. In this way, the attachments are closely associated with the complex core, showing that these attachments are in stable and reliable cooperation with the core.

The steps of the Finding Attachment Proteins procedure are described in Algorithm 3.

Algorithm 3 Finding Attachment Proteins (SC)

```
Input: The set of detected cores $SC$;
       The PPI network $G=(V,E)$;
       The closeness parameter $\omega$;
       The set of all the in-core edges $E_{core}^{in}$;
Output: The protein complexes set $PC$
$PC=\Phi$;
for each core $C\in SC$ do   // $C=(V_C,E_C)$
  for each edge $p\notin E_{core}^{in}$ do
    if $\exists edge\ q\in E_C\ and\ S(p,q)>\omega$ do
      insert the induced vertices of $q$ into core $C$
    endif
  endfor
  insert $C$ into $PC$
endfor
return $PC$
```

IV. RESULTS AND DISCUSSIONS

A. *Datasets*

We use two different yeast PPI networks to validate our algorithm, including the Database of Interaction Protein (DIP) [13] and Gavin [7] obtained from high-throughput technology. The DIP dataset consists of 4928 proteins and 17,201 interactions and the Gavin dataset consists of 1430 proteins and 6531 interactions.

To evaluate the predicted protein complexes, a benchmark set is constructed from the MIPS [14], Aloy et al. [15] and SGD database [16] based on the Gene Ontology (GO) notations. This benchmark set consists of 428 protein complexes firstly presented in [17].

B. *Evaluation criteria*

1) Precision, Recall and F-measure

Before giving the definition of precision, recall and F-measure, we should define some other concepts first. The neighborhood affinity score between a real complex b in the benchmark and a predicted complex p is used to determine whether they match with each other or not, and is defined as

$$NA(p,b)=\frac{|V_b\cap V_p|^2}{|V_b|\times|V_p|}.$$

If $NA(p,b)\geq\omega$, then they are considered to be matching, and ω is usually set as 0.2. Let B be the set of real complexes, it is the benchmark, and P be the set of clustering results. N_{cp} is the number of predicted complexes which match at least a real complex, the mathematic expression is $N_{cp}=|\{p \mid p\in P, \exists b\in B, NA(p,b)\geq\omega\}|$. N_{cb} is the number of real complexes which match at least a predicted one, the mathematic expression is $N_{cb}=|\{b \mid b\in B, \exists p\in P, NA(p,b)\geq\omega\}|$. Precision and recall are defined as [18]:

$$Precision=\frac{N_{cp}}{|P|} \qquad Recall=\frac{N_{cb}}{|B|}$$

F-measure which is the harmonic mean of precision and recall, is defined as:

$$F=2\times Precision\times Recall/(Precision+Recall).$$

It is used to evaluate the overall performance of the different techniques.

2) Co-localization

Since protein complexes are formed to perform a specific cellular function, proteins within the same complex tend to share common functions and be co-localized. Generally, higher co-localization scores [17] show that proteins within the same protein complexes tend to share higher functional similarity, and hence they can be used to evaluate the overall quality of predicted protein complexes[8].

The co-localization score is defined as the maximal fraction of proteins in a complex found at the same localization [17].

3) GO semantic similarity

978-1-4577-1666-9/11/$26.00 ©2011 IEEE

Semantic similarity is another way to evaluate the quality of predicted protein complexes. It is the comparison of Gene Ontology (GO) terms associated with the proteins within a complex. In 2006, Schlicker et al. [19] proposed a new scoring method to calculate the semantic similarity. This scoring method basically generates another protein-protein interaction scores network containing functional relationships based on the GO annotations. The network is then used to calculate average inner-complex scores for the predicted complex set. The higher this score, the better our prediction is.

C. *Complex set comparative evaluation*

In this section, we compare the performance of our algorithm with MCL, CPM and CoAch.

1) Precision, Recall and F-measure Comparison

Table 1 shows the comparison results in Gavin network. In table 1, our method precisely predicts 237 complexes. It demonstrates that our method can detect more complexes than the other three algorithms. Because we identify much more complexes, the precision of our method is lower than the other three algorithms. Correspondingly, we get the highest recall, which is 28.0%, 90.8% and 14.9% higher than MCL, CPM and CoAch respectively. The F-measure of our method on Gavin network is the highest of the four algorithms.

We also compare performance of the four algorithms on DIP dataset. The comparison results are shown in table 2. On DIP dataset, our method correctly predicts more complexes than the other three algorithms. The precision of our algorithm is 121.4% and 11.0% higher than MCL and CPM. The precision of CoAch is only 1.88% higher than ours. We get a pretty good F-measure, which is 78.9% and 51.2% higher than MCL and CPM, and only 3.9% lower than CoAch.

TABLE I.
THE COMPARISON OF VARIOUS ALGORITHMS USING GAVIN NETWORK

Algorithm	N_{cp}	$\lvert P \rvert$	Precision (%)	Recall (%)	F-measure (%)
MCL	112	232	48.3	30.7	37.5
CPM	54	98	55.1	20.6	30.0
CoAch	164	325	50.5	34.2	40.7
Our method	237	556	42.6	39.3	40.9

TABLE II.
THE COMPARISON OF VARIOUS ALGORITHMS USING DIP NETWORK

Algorithm	N_{cp}	$\lvert P \rvert$	Precision (%)	Recall (%)	F-measure (%)
MCL	140	835	16.8	46.3	24.6
CPM	82	245	33.5	25.7	29.1
CoAch	283	746	37.9	57.7	45.8
Our method	328	881	37.2	53.7	44.0

2) Colocalization similarity

To evaluate the localization consistency of our predicted complexes, we use two localization benchmark datasets. One is published by Kumar et al. [20] and the other one is published by Huh et al. [21]. The final localization score is calculated as the geometric mean of the co-localization scores based on the "Kumar" and "Huh" datasets. We used the ProCope[19] tool to calculate the co-localization.

Figure 2 shows the co-localization scores of complexes detected by various approaches on Gavin dataset and DIP dataset. The average co-localization score of our method on Gavin network is 0.71, which is 26.0%, 29.5% and 44.0% higher than CoAch, MCL and CPM respectively. The average co-localization score of our method on DIP network is 0.71, which is 14.2%, 29.7% and 59.5% higher than CoAch, MCL and CPM.

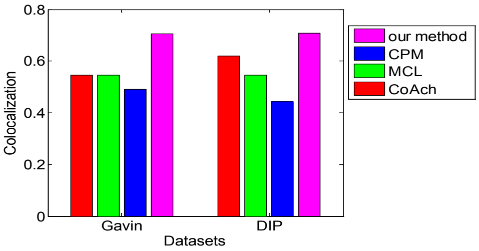


Fig. 2. The localization similarity comparisons on two datasets

3) GO semantic similarity

The GO semantic similarity score of a complex is the average relevance similarity of all protein pairs in this complex. The GO semantic similarity score of a set of complexes is the weighted mean over all complex GO scores and calculated separately for the "biological process", "cellular component" and "molecular function" ontologies. Then the final GO score of a set of complexes is calculated as the geometric mean of the three ontologies scores.

Figure 3 shows the results for the comparison of GO semantic similarity scores on two datasets. These scores are calculated by ProCope tool. The GO semantic similarity score of our method on Gavin network is 0.87, which is 6.1%, 10.5% and 18.7% higher than CoAch, MCL and CPM respectively. The GO semantic similarity score of our method on DIP network is 0.84. which is 7.7%, 31.2% and 30.2% higher than CoAch, MCL and CPM.

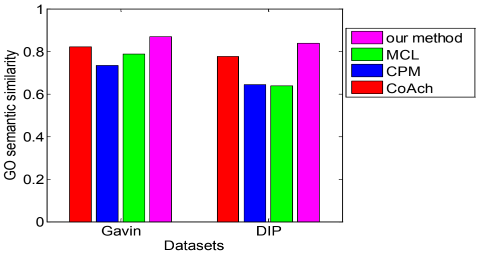


Fig. 3. The GO semantic similarity comparisons on two datasets.

V. CONCLUSIONS

In this paper, we propose a novel core-attachment algorithm by characterizing the protein complex core from the perspective of edges. First, we characterize protein complex cores via edges. Then we propose a method to extract the protein complex cores we defined. After that, we select the attachments for each complex core to form a protein complex. Finally, we estimate the performance of our algorithm by comparing it with MCL, CPM and CoAch algorithms. The experimental results show that our method and CoAch are better than MCL and CPM, and our algorithm is better than CoAch in many respects. These demonstrate that core-attachment is a better way to characterize protein complexes and edge communities is a better way to characterize protein complex cores. Our proposed method is validated as an effective algorithm in identifying protein complexes and can provide more insights for future biological study.

ACKNOWLEDGMENT

This work was supported by the National Key NSFC (Grant No. 60933009), NSFC (Grant No. 61072103), SRFDPHE (Grant No. 200807010013) and FRFCU(Grant No. K50510030006).

REFERENCES

[1] L. H. Hartwell, J. J. Hopfield, S. Leibler and A. W. Murray, “From molecular to modular cell biology”, Nature 1999, 402:C47-C52.

[2] S. V. Dongen, “Graph Clustering by Flow Simulation”, University of Utrecht, Netherlands, 2000.

[3] G. D. Bader and C. W. Hogue, “An automated method for finding molecular complexes in large protein interaction networks”, BMC Bioinformatics 2003, Vol. 4, No. 1, p.2.

[4] A. D. King, N. Przulj and I. Jurisica, “Protein complex prediction via cost-based clustering”, Bioinformatics 2004, 20(17):3013-20.

[5] G. Palla, I. Derenyi, I. Farkas and T. Vicsek “Uncovering the overlapping community structure of complex networks in nature and society”, Nature 2005, Vol. 435, No. 7043, 814-818

[6] A. H. Tong, et al, “A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules”, Science 2002, Vol. 295, No. 5553, 321–324.

[7] A.C. Gavin, P. Aloy, P. Grandi, et al., “Proteome survey reveals modularity of the yeast cell machinery”, Nature 2006, 440(7084):631-636.

[8] M. Wu, X. Li, C.K. Kwoh, et al., “A Core-Attachment based Method to Detect Protein Complexes in PPI Networks”, BMC Bioinformatics 2009, 10:169.

[9] H.C. Leung, Q. Xiang, S.M. Yiu, et al., “Predicting Protein Complexes from PPI Data: A Core-Attachment Approach”, Journal of Computational Biology 2009, 16(2): 133-144.

[10] Y. Y. Ahn, J. P. Bagrow and S. Lehmann, “Link communities reveal multiscale complexity in networks”, Nature 2010 466:761

[11] X. Q. Cheng, et al., “Bridgeness: a local index on edge significance in maintaining global connectivity”, Journal of Statistical Mechanics: Theory and Experiment 2010 P10011

[12] H. W. Shen, et al, “Quantifying and identifying the overlapping community structure in networks”, Journal of Statistical Mechanics: Theory and Experiment 2009 P07042

[13] I. Xenarios, D. W. Rice, et al., “Dip: the database of interacting proteins”, Nucleic Acids Research 2000, 28:289-291.

[14] H. W. Mewes, C. Amid,et al., “Mips: analysis and annotation of proteins from whole genomes”, Nucleic Acids Research 2004, 32:41-44.

[15] P Aloy, et al., “Structure-based assembly of protein complexes in yeast”, Science 2004, 303(5666): 2026-2029, 2004.

[16] S. S. Dwight,et al., “Saccharomyces genome database provides secondary gene annotation using the gene ontology” Nucleic Acids Research 2002, 30(1):69-72.

[17] C. C. Friedel, J. Krumsiek, and R. Zimmer. “Boostrapping the interactome: unsupervised identification of protein complexes in yeast”, In Proceedings of the 12th Annual Conference on Research in computational Molecular Biology (RECOMB), pages 3-16, 2008.

[18] H. N. Chua, W. K. Sung, and L. Wong, “Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions”, Bioinformatics 2006, 22(13):1623-1630.

[19] A. Schlicker, F. Domingues, J. Rahnenfuhrer, et al., "A new measure for functional similarity of gene products based on gene ontology," BMC Bioinformatics 2006, vol. 7, no. 1, pp. 302.

[20] L.F. Wu, T.R. Hughes, A.P. Davierwala, et al., “Large-scale prediction of Saccharomyces cerevisiae gene function using overlapping transcriptional clusters”, Nat Genet 2002, 31:255-265.

[21] S.H. Yook, Z.N. Oltvai, A.L. Barabàsi, “Functional and topological characterization of protein interaction networks”, Proteomics 2004, 4(4):928-942.