Ontology integration to identify protein complex in protein interaction networks

Bo Xu, Hongfei Lin, and Zhihao Yang
Department of Computer Science and Engineering
Dalian University of Technology
Dalian, China
xubofm@gmail.com, hflin@dlut.edu.cn, yangzh@dlut.edu.cn

Abstract—Protein complexes can be identified from the protein interaction networks derived from experimental data sets. However, these analyses are challenging because of the presence of unreliable interactions and the complex connectivity of the network. The integration of protein-protein interactions with the data from other sources can be leveraged for improving the effectiveness of protein complexes detection algorithms. We have developed novel semantic similarity method, which use Gene Ontology (GO) annotations to measure the reliability of protein-protein interactions. The protein interaction networks can be converted into a weighted graph representation by assigning the reliability values to each interaction as a weight. Following the approach of previously proposed clustering algorithm IPCA which expands clusters starting from seeded vertices, we present a clustering algorithm OIIP based on the new weighted Protein-Protein interaction networks for identifying protein complexes. The algorithm OIIP is applied to the protein interaction network of Saccharomyces cerevisiae and identifies many well known complexes. Experimental results show that the algorithm OIIP has higher F-measure and accuracy compared to other competing approaches.

Keywords—Protein complex; PPI networks; Gene Ontology;

I. INTRODUCTION

In the post-genomic era, one of the most important issues is to systematically analyze and comprehensively understand the topology of biological networks and biochemical progress in cells [1]. Protein complexes can help us to understand certain biological progress and to predict the functions of proteins. A wide range of graphtheoretic approaches have been employed for detecting protein complexes from protein interaction networks. However, they have been limited in accuracy due to the presence of unreliable interactions and the complex connectivity patterns of the networks [2].

To resolve the inaccuracy resulting from false connections, other functional knowledge can be integrated into the protein interaction networks. For example, many groups [3, 4] have investigated the integration of gene expression data from microarray experiments to improve protein complexes identification. However, gene expression data are also susceptible to experimental sources of bias and noise. The correlations of mRNA levels with even cognate protein expression may be modest at best. These factors limit the usefulness of microarray data for assessing the reliability of protein-protein interactions. Gene Ontology (GO) is another useful data source to combine with the protein interaction networks. It provides a collection of well-defined biological terms, called GO terms, spanning biological processes, molecular functions and cellular components.

In this work, we integrate protein-protein interactions with the information content in the GO annotation database and topology weights to enhance the modularization of interaction networks. An unweighted protein interaction network can be converted into a weighted graph representation by assigning a weight to each interaction. The weight of each interaction is interpreted as its reliability, i.e., the probability of the interaction being a true positive. We propose a novel method to measure the reliability of protein-protein interactions using GO annotation data and topology weights. Following the approach of that of the previously proposed clustering algorithm IPCA[5] which expands clusters starting from seeded vertices, we present a clustering algorithm OIIP based on the new large weighted protein interaction networks.

II. METHODS

A. Weighted network

Weights quantify the likelihood of the interaction between every pair of proteins, and they can be estimated by encoding the proteins using gene ontology (GO) consortium. It is very common for the same protein or proteins in the same subfamily to form protein complexes, for example, protein Ste2p and Ste3p from a complex that is among activated G protein-coupled receptors in yeast cellular mating [6]. It is also common for proteins in heterofamilies to form protein complexes if they share a conservative motif [6]. Complicated protein complexes may be formed by multiple proteins, some of which share same biological processes and some are from the same subfamily [6]. Thus GO consortium is considered to be a very helpful vehicle for investigating protein-protein interactions.

Semantic similarity has been used in Information Science to evaluate the similarity between two concepts in a taxonomy [7], and we applied it to protein-protein interactions to estimate the similarity between two proteins. We define an annotation size of a GO term as the number of annotated proteins on the GO term. The semantic similarity between two proteins is then calculated based on the annotation size of the GO term, on which both proteins are annotated. According to the transitivity property of GO annotation, if a protein x is annotated on a GO term gi, it is
also annotated on the GO terms on the path from gi to the root GO term in the GO structure. Thus, the proportion of the annotation size of a GO term to the total number of annotated proteins can quantify the specificity of the GO term. If two proteins are annotated on a more specific GO term and have more common GO terms, then they are functionally more similar.

Suppose a protein x is annotated on m different GO terms. S(x) denotes a set of annotated proteins on the GO term gi, whose annotation includes x, where 1 ≤ i ≤ m. In the same way, suppose both x and y are annotated on n different GO terms, where n ≤ m. S(x, y) denotes a set of annotated proteins on the GO term gj, whose annotation includes x and y, where 1 ≤ j ≤ n. Then, the minimum size of S(x), \( \min |S(x)| \), is less than or equal to \( \min |S(x, y)| \). C(x, y) denotes the sets of GO terms, whose annotation includes x and y. \( |C(x, y)| \) is the number of common GO terms which x and y both have.

Suppose the size of annotation represents the number of annotated proteins on a GO term. Using the annotation size of the most specific GO term, on which two proteins x and y are annotated, we define semantic similarity \( S_{\text{sem}}(x, y) \) between x and y as follows:

\[
S_{\text{sem}}(x, y) = \frac{-|C(x, y)| \times \log \left( \frac{\min |S(x, y)|}{S_{\text{max}}} \right)}{\log |V|}
\]  

(1)

\( S_{\text{max}} \) is the maximum size of annotation among all GO terms in a DAG structure. If two proteins x and y are annotated on a more specific GO term and more common GO terms than x and z, then x is semantically more similar to y than z.

Considering the graph topology, we also involve the topology weight. For an input graph \( G = (V, E) \), we assign the topology weight of an edge \( [u, v] \) to be the number of neighbors shared by the vertices u and v. Then we assigned the sum of \( S_{\text{sem}}(u, v) \) and topology weight to the edge between u and v as a weight.

B. Weighted vertex and selecting seed

We define the weight of each vertex to be the sum of the weights of its incident edges. After all vertices are assigned weights, we also sort in non-increasing order the vertices by their weights and store them in a queue \( S_q \) (vertices of the same weight are ordered in terms of their degrees). The complexity of calculating edge weights and vertex weights is \( O(|V||E|) \), and the complexity of sorting all vertices by their weights is \( O(|V| \log |V|) \).

The notion that vertex weight is a good measure for selecting seeds has been adopted by DPClus [8] and MCODE [9]. Here, we also pick the highest weighted vertices as the seeds. Our procedure proceeds as follows. We pick the first vertex in the queue \( S_q \) and use it as a seed to grow a new cluster. Once the cluster is completed, all vertices in the cluster are removed from the queue \( S_q \) and we pick the first vertex remaining in the queue \( S_q \) as the seed for the next cluster. There is an important difference between this seed selection procedure and the one used in the IPCA algorithm [5]. Our procedure computes the vertex weight for each vertex based on the weighted networks; while the IPCA algorithm computes the vertex weight based on the original networks. We feel that our approach is biologically more meaningful because a complex is not only a dense structure in the original protein network but also have biological function.

C. Extending cluster

We introduce a new concept to measure how strongly a vertex v is connected to a subgraph K: the interaction probability \( E_{vk} \) of a vertex v to a subgraph K, where \( v \not\in K \), is defined by

\[
E_{vk} = \frac{e_{vk}}{w_k}
\]  

(2)

Where \( e_{vk} \) is the sum of the weights of edges between the vertex v and K, and \( w_k \) is the sum of weights of edges in K. We discuss the relationship between the parameter \( E_{vk} \) and \( IN_{vk} \) introduced in the algorithm IPCA[5]. According to [5],

\[
IN_{vk} = \frac{m_{vk}}{n_k}, \quad \text{where} \quad m_{vk} = \text{the number of edges between the vertex} \ v \ \text{and} \ K, \ \text{and} \ n_k = \text{the number of vertices in} \ K.
\]

By the expressions, our parameter \( E_{vk} \) is similar to the parameter \( IN_{vk} \). While our parameter considers with the biological weights, it have more biological meaning.

A cluster K is extended by adding vertices recursively from its neighbors according to the priority. The priority of a neighbor v of K is determined by the value \( E_{vk} \). This procedure is similar to the one proposed in IPCA [5], except that we do not use \( IN_{vk} \) to judge the extending.

III. RESULTS AND DISCUSSION

The protein interaction database is downloaded from the Gavin database [31] and BioGrid (version yeast HC-
The protein-complex dataset CYC2008 [16] which we used is a comprehensive catalogue of 408 manually curated heterometic protein complexes reliably backed by small-scale experiment reported. We apply the proposed algorithm OIIP to this two databases. And we used the various evaluation metrics introduced in previous paper [10] that have been used to evaluate their computational methods for complex detection. In the following subsections, we discuss the effect of the value $T_{in}$ on clustering, compare the predicted clusters with the known complexes, and evaluate the significance of the predicted clusters. We will also compare the algorithm OIIP to eight competing previous methods for their performance of identifying protein complexes. Since most proteins in the same complex have same or correlative function and involve in the same biological process, we employ biological annotation information, including GO cellular component annotation, GO Molecular Function annotation and GO Biological process annotation to assess the predicted protein-complexes.

### A. The effect of $T_{in}$ on clustering

To understand how the value of $T_{in}$ influences the outcome of the clustering, we generate 9 sets of clusters by using $T_{in} = 0.1, 0.2, ..., 0.9$ from the Gavin PPI dataset. The effect on the predicted clusters with different $T_{in}$ is given in Fig. 1.

Fig. 1(a) shows that the total number of the predicted clusters is increasing as $T_{in}$ increases. However, there is a abrupt decrease at $T_{in} = 0.5$. This is probably caused by the Hub structures in the protein interaction network. When $T_{in} = 0.5$, these Hub structures are decomposed into complexes that consist of only 2 proteins. Fig. 1(b) shows that the size of the biggest cluster is decreasing as $T_{in}$ increases. With the increasing of $T_{in}$, the probability of neighbors added to the cluster is decreasing. Thus, the size of the predicted clusters is also decreasing.

| Parameter $T_{in}$ | Precision | Recall | F-measure | sensitivity | PPV  | Accuracy |
|-------------------|-----------|--------|-----------|-------------|------|----------|
| $T_{in}=0.1$      | 0.4819    | 0.335784 | 0.395787 | 0.525       | 0.465604 | 0.494491 |
| $T_{in}=0.2$      | 0.498113  | 0.352941 | 0.413146 | 0.511979    | 0.48535 | 0.498487 |
| $T_{in}=0.3$      | 0.482112  | 0.365196 | 0.415588 | 0.501042    | 0.516012 | 0.508472 |
| $T_{in}=0.4$      | 0.559816  | 0.384804 | 0.456097 | 0.484896    | 0.560703 | 0.521424 |
| $T_{in}=0.5$      | 0.606195  | 0.335784 | 0.432177 | 0.420312    | 0.57216  | 0.490394 |
| $T_{in}=0.6$      | 0.582645  | 0.340686 | 0.429963 | 0.409375    | 0.621008 | 0.504207 |
| $T_{in}=0.7$      | 0.485768  | 0.343137 | 0.402181 | 0.386458    | 0.638547 | 0.496761 |
| $T_{in}=0.8$      | 0.458477  | 0.338235 | 0.389283 | 0.360417    | 0.653266 | 0.48523  |
| $T_{in}=0.9$      | 0.466227  | 0.345588 | 0.396944 | 0.340625    | 0.663043 | 0.475236 |

| Method        | Precision | Recall | F-measure | Accuracy | GO_Function | GO_Process | GO_Component |
|---------------|-----------|--------|-----------|----------|-------------|------------|--------------|
| CoreMethod    | 0.526596  | 0.279412 | 0.365101 | 0.491702 | 0.579787    | 0.430851   | 0.478723     |
| Mcode         | 0.733333  | 0.294118 | 0.419847 | 0.469031 | 0.511111    | 0.392593   | 0.474074     |
| MCL           | 0.540373  | 0.27451  | 0.364071 | 0.509139 | 0.503106    | 0.391304   | 0.456267     |
| RNSC          | 0.400651  | 0.348039 | 0.372497 | 0.489759 | 0.332248    | 0.247557   | 0.309446     |
| CMC           | 0.608     | 0.218   | 0.3211   | 0.474     | 0.692       | 0.55       | 0.633        |
| CFinder       | 0.663     | 0.191   | 0.297    | 0.419     | 0.602       | 0.439      | 0.551        |
| COACH         | 0.524     | 0.331   | 0.406    | 0.49      | 0.656       | 0.525      | 0.61         |
| IPCA          | 0.526032  | 0.338235 | 0.411731 | 0.48789  | 0.719928    | 0.587074   | 0.666068     |
| OIIP $T=0.4$  | 0.559816  | 0.384804 | 0.456097 | 0.521424 | 0.697853    | 0.553681   | 0.636503     |
| OIIP $T=0.5$  | 0.606195  | 0.335784 | 0.432177 | 0.490394 | 0.769912    | 0.639381   | 0.710177     |
As shown in Table 1, the Precision of the algorithm OIIP is about 0.6 when $T_{in}=0.5$, which implies that the clusters generated by OIIP are reliable. The F-measure is about 0.45 when $T_{in}=0.4$, which represents overall performance of an algorithm. The Accuracy takes into account of both the sensitivity and PPV, and is determined by the larger one. In this experiment, the Accuracy is mostly influenced by the PPV. The PPV of the clusters generated by OIIP increases with the increasing of $T_{in}$. Especially, an obvious increase appears when $T_{in} \geq 0.5$.

B. Comparison of OIIP and other methods

Since there have been protein complexes that were experimentally determined, a good protein complexes detecting algorithm should identify these known complexes as many as possible. Fig. 2 shows the numbers of known complexes (size>2) matched to the clusters generated by OIIP and by other eight previous known methods: CoreMethod[15], Mcode[9], MCL[13], RNSC[12], CMC[14], CFinder[11], COACH[10], IPCA[5]. Our method predicted more complexes than others on Gavin PPI dataset. As show in Table 2 OIIP also get the highest F-measure ($T_{in}=0.4$). Although Mcode has the highest precision, its recall is very low. The fact is that the predicted protein complexes are fewer than our method. Some predicted clusters which are not matched with criterion complexes are possible actual complexes which are undiscovered. So it is necessary to predict more clusters with high F-measure value.

And we count the number of clusters with p-value less than 0.01, a threshold which represents significant biological sense and compute the proportion of clusters which achieve low p-value. The proportion of clusters from various methods with low p-value is shown in Table 2. Table 2 also shows that the clusters predicted by our method have achieved highest biological significance than predicted clusters from others on all the three biological annotation datasets when T is set to 0.5. Compare to the IPCA, we have better performance in all evaluation measurements. So the ontology interaction to the PPI network is valuable to predict protein complexes.

As show in Fig. 3, RNSC has the highest recall and CMC has the highest precision, while our method OIIP also gets the highest F-measure on BioGrid PPI dataset.

C. Examples of predicted complexes

From the results of the experiment above we know that most of our predicted clusters have highly biological meanings. We give some examples of predicted clusters with detailed information which are matched with the benchmark complexes in Table 3. We also list the best matched benchmark complexes, NA and the p-value of three protein annotation.

Some of predicted clusters from our method are not matched with complex from criterion dataset. But we find that they have highly biologically significant and have high local density, so some of them may be real complexes which are still undiscovered. We also give some examples in Table 4 and Fig. 4. Their p-value of biological annotation shows that some of them may be the candidate protein complexes. The results are useful for biologists to find the new protein complexes.

IV. CONCLUSIONS AND FUTURE WORKS

It is believed that identification of protein complexes is useful to explain certain biological progress and to predict functions of proteins. In this paper, we developed an algorithm OIIP to identify protein complexes based on the new large weighted protein interaction networks. When we implemented the OIIP algorithm with weighted networks, the overall F-measure and accuracy of complexes is substantially improved. This result strongly appeals the necessity of integrating of functional information for the analysis of protein-protein interaction data.

The fact that biological properties are poor at the identification reveals that the higher-level structures (e.g., secondary and tertiary structure) of proteins cannot be accurately represented by the primary structure under the current coding techniques. The experimentally determined protein interaction network has not been used in the research, and a possible future research could combine the experimentally determined protein interactions with the GO estimated interactions to further improve the identification.

ACKNOWLEDGMENT

This work is supported by grant from the Natural Science Foundation of China (No. 60673039 and61070098), the
REFERENCES

[1] Uetz P, A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae, *Nature*, 2000, 403(1), pp.623-627.

[2] von Mering C, et al, Comparative assessment of large-scale data sets of protein-protein interactions, *Nature*, 2002, 417(1), pp. 399-403.

[3] Cho Y-R, Hwang W, Zhang A., Identification of overlapping functional modules in protein interaction networks: information flow-based approach, *Proceedings of 6th IEEE International Conference on Data Mining Workshops*, 2006, pp.147-152.

[4] Ideker T, Ozier O, Schwikowski B, Siegel AF, Discovering regulatory and signalling circuits in molecular interaction networks, *Bioinformatics*, 2002, 18(1), pp.S233-S240.

[5] Min Li, Jianer Chen, Jianxin Wang, Modifying the DPClus algorithm for identifying protein complexes based on new topological structures, *BMC Bioinformatics*, 2008, 9(398), pp.345-352.

[6] J. Ortiz, O. Stemmann, S. Rank, J. Lechner, A putative protein complex consisting of Ctf19, Mcm21, and Okp1 represents a missing link in the budding yeast kinetochore, *Genes Dev*, 1999, 13 (9), pp. 1140–1155.

[7] Resnik P, Using information content to evaluate semantic similarity in a taxonomy, *Proc. of 14th International Joint Conference on Artificial Intelligence*, 1995,1(1), pp. 448-453.

[8] Altaf-Ul-Amin M, Shinbo Y, Mihara K, Karokawa K, Kanaya S, Development and implementation of an algorithm for detection of protein complexes in large interaction networks, *BMC Bioinformatics*, 2006, 7(3), pp.134-163.

[9] Bader G, Hogue C. An Automated Method for Finding Molecular Complexes in Large Protein Interaction Networks. *BMC Bioinformatics* 2003, 4(2), pp. 455-470.

[10] Wu M, Li XL, Kwoh CK and Ng SK, A Core-Attachment based Method to Detect Protein Complexes in PPI Networks, *BMC Bioinformatics*, 2009, 10(169), pp.122-142.

[11] Adamcsek B, Palla G, Farkas I, Derenyi I, Vicsek T, CFinder: locating cliques and overlapping modules in biological networks, *Bioinformatics*, 2006, 22(8), pp. 1021-1043.

[12] King A, Przulj N, Jurisica I, Protein Complex Prediction via Cost-based Clustering, *Bioinformatics*, 2004, 20(1), pp. 3013-3020.

[13] Pereira-Leal JB, Enright AJ and Ouzounis CA, Detection of functional modules from protein interaction networks, *Protein Structure Function, and Bioinformatics* 2004, 54(1), pp. 49–57.

[14] Liu GM, Chua HN and Wong L, Complex discovery from weighted PPI networks, *Bioinformatics*, 2009, 25(15), pp. 1891–1897.

[15] Leung HC, Yiu SM, Xiang Q and Chin FY, Predicting Protein Complexes from PPI Data: A Core-Attachment Approach, *Journal of Computational Biology*, 2009, 16(2), pp. 133–144.

[16] Shuye Pu, Jessica Wong, Brian Turner, Up-to-data catalogues of yeast protein complexes. *Nucleic Acids Research*, 2009, 37(1), pp. 825-831.

Figure 4. Some predicted complexes which don’t match with benchmark complexes.
### TABLE III. Some predicted clusters which matched with benchmark complexes.

| ID | Predicted clusters | Benchmark complexes ID in CYC2008 | p-value | GO_Component | GO_Function | GO_Process |
|----|-------------------|-----------------------------------|---------|---------------|-------------|------------|
| 1  | YBR123C YOR110W YPL007C YDR362C YAL001C YGR047C | 402 1 | 0 | 3.16E-06 | 2.59E-05 |
| 2  | YLR208W YGL092W YDL116W YJR042W YKL057C YGL100W | 181 1 | 1 | 1.10E-05 | 2.50E-06 |
| 3  | YLR166C YBR102C YPR055W YIL068C YER008C YDR166C YGL233W YJL085W | 90 1 | 1 | 2.68E-06 | 2.06E-04 |
| 4  | YBR234C YLR370C YOR065C YDL029W YIL062C YNR035C YKL013C | 12 1 | 1 | 1.19E-03 | 4.95E-06 |
| 5  | YER157W YGR120C YPR105C YNL051W YML071C YGL223C YNL041C YGL005C | 112 1 | 1 | 1.05E-05 | 6.91E-02 |
| 6  | YHR081W YHR069C YOL021C YGR095C YGR195W YDR280W YGR158C YCR035C YDL111C YNL232W YOR001W YOL142W YOR076C | 178 1 | 0.923077 | 3.00E-03 | 0 |
| 7  | YGL048C YKL145W YHR027C YHL030W YLR421C YHR200W YDR427W YDL147W YFR052W YDL097C YPR108W YIL075C YFR004W | 2 1 | 0.911157 | 1.09E-05 | 2.68E-06 |
| 8  | YPL210C YDL092W YML010C YPR088C YPL243W YDL051W YKL122C | 248 1 | 0.857143 | 3.46E-06 | 1.19E-07 |
| 9  | YOR179C YDR195W YGR156W YER133W YAL043C YKL059C YPR107C YLR115W YDR301W YNL317W YKR002W YKL018W YLR277C | 341 1 | 0.816667 | 9.43E-04 | 0 |
| 10 | YMR223W YGL066W YBR081C YGR252W YDR448W YGL112C YDR145W YMR236W YDR167W YBR198C YOL148C YLR055C YPL254W YDR392W YCL010C YDR176W YHR099W YML007W | 227 1 | 0.802778 | 1.39E-04 | 0 |

### TABLE IV. Some predicted complexes which don’t match with benchmark complexes.

| ID | Main proteins | P-value | GO_Component | GO_Function | GO_Process |
|----|---------------|---------|---------------|-------------|------------|
| Complex a | YOR201C YNL284C YCR046C YBL038W YDR237W YDR462W YDR296W YDR322W YNL005C YMR024W | 9.89E-06 | 2.32E-06 | 8.94E-07 |
| Complex b | YIL070C YNL137C YHL004W YPL013C YBR251W YDR041W YDR036C | 5.39E-05 | 8.58E-06 | 4.95E-06 |
| Complex c | YNL112W YDR091C YMR309C YPR041W YBR079C YDR429C YOR096W YML063W | 0 | 2.03E-06 | 3.04E-04 |
| Complex d | YDR164C YOR204W YBL038W YML025C YGR220C YCR046C YNL005C YDR296W YDR237W YNL284C YDR322W YDR462W YLR439W | 9.36E-06 | 9.18E-06 | 7.45E-06 |