

Decomposing PPI Networks for Complex Discovery

Guimei Liu Chern Han Yong
School of Computing
National University of Singapore
Singapore
liugm@comp.nus.edu.sg, cherny@nus.edu.sg

Hon Nian Chua
Data Mining Department
Institute for Infocomm Research
Singapore
hnchua@i2r.a-star.edu.sg

Limsoon Wong
School of Computing
National University of Singapore
Singapore
wongls@comp.nus.edu.sg

Abstract—Protein complexes are important for understanding principles of cellular organization and functions. With the availability of large amounts of high-throughput protein-protein interactions (PPI), many algorithms have been proposed to discover protein complexes from PPI networks. However, none of existing algorithms takes into consideration the fact that not all the interactions in a PPI network take place at the same time. As a result, predicted complexes often contain many spuriously included proteins, precluding them from matching true complexes.

We propose two methods to tackle this problem: (1) We utilize cellular component Gene Ontology (GO) terms to decompose PPI networks into several smaller networks such that the proteins in each decomposed network are annotated with the same cellular component GO term. (2) Hub proteins are more likely to fuse clusters that correspond to different complexes. To avoid this, we remove hub proteins from PPI networks, and then apply a complex discovery algorithm on the remaining PPI network. The removed hub proteins are added back to the generated clusters afterwards.

We tested the two methods on the yeast PPI network downloaded from BioGRID. Our results show that these methods can improve the performance of several complex discovery algorithms significantly. Further improvement in performance is achieved when we apply them in tandem.

Keywords—complex discovery; PPI network;

I. INTRODUCTION

High-throughput experimental techniques have produced large amounts of protein interactions, which makes it possible to discover protein complexes from protein-protein interaction (PPI) networks. A PPI network can be modeled as an undirected graph, where vertices represent proteins and edges represent interactions between proteins. Protein complexes are groups of proteins that interact with one another, so they are usually dense subgraphs in PPI networks. Many algorithms have been developed to discover complexes from PPI networks [1; 2; 3; 4; 5; 6; 7].

As a model organism, *Saccharomyces cerevisiae* (baker's yeast) has been extensively studied, and its PPI network is now relatively complete. However, the performance of existing complex discovery algorithms on the yeast PPI network is not very satisfactory. One reason behind this is that each protein do not necessarily participate in all its known interactions simultaneously. Existing complex

discovery algorithms do not take this into consideration. As a result, the clusters generated often contain extra proteins that preclude them from matching true complexes. An ideal solution would be to decompose the PPI network into several smaller networks such that interactions within each smaller network are contextually coherent. In reality, it is very difficult to know which subset of interactions take place together. Here we choose to use cellular component GO terms to decompose PPI networks because a protein complex can be formed only if its proteins are localized within the same compartment of the cell. We use only localization GO terms that are relatively general for decomposition.

The existence of hub proteins is another factor that makes it difficult for complex discovery algorithms to decide the boundary of clusters. Hub proteins are proteins that have a lot of neighbors in the PPI network, and these neighbors often belong to multiple complexes [8]. As a result, hub proteins often fuse clusters that correspond to different complexes. To avoid this, we remove hub proteins from PPI networks prior to clustering, and then add them back to the generated clusters after clustering.

We tested the above methods on the yeast PPI network downloaded from BioGRID [9]. The results show that these methods can improve the performance of existing complex discovery algorithms significantly.

The rest of the paper is organized as follows. Section II describes the two methods for decomposing PPI networks. Section III reports and discusses experiment results. Section IV concludes the paper.

II. DECOMPOSING PPI NETWORKS

A. The GO term decomposition method

A protein complex can only be formed if its proteins are localized within the same compartment of the cell. Hence we use cellular component GO terms to decompose a given PPI network into several smaller PPI networks such that all proteins in each smaller network are annotated with the same localization GO term. We use only localization GO terms that are relatively general for decomposition. There are several reasons for this. First, it is relatively easy to obtain the rough localization of proteins, compared with obtaining the precise and specific localization of proteins.

Secondly, very specific GO terms are annotated to very few proteins. Using them to decompose PPI networks produces many small fragments, and lots of information may be lost due to the decomposition. Finally, some very specific cellular component GO terms correspond to complexes, and they are just as hard to decide as complexes.

We use a threshold N_{GO} to select GO terms for decomposition, where N_{GO} should be large. The selected GO terms are annotated to at least N_{GO} proteins, and none of their descendant terms is annotated to at least N_{GO} proteins. If a GO term is selected, then none of its ancestor terms or descendant terms will be selected.

Given a selected GO term, we first remove all the proteins that are not annotated to the term from the given PPI network, and then apply a complex discovery algorithm on the resultant network. This process is repeated for every selected GO term. The final set of clusters is the union of the clusters discovered from every filtered network. Duplicated clusters are removed.

B. The hub removal method

Hub proteins are those proteins that have many neighbors in the PPI network. We call a protein a *hub protein* if it has at least N_{hub} neighbors, where N_{hub} is an integer. A hub protein often connects proteins that belong to different complexes, which makes it hard to decide the boundary of the complexes and the membership of the hub proteins.

To alleviate the impact of the hub proteins, we first remove hub proteins from a given PPI network, and then use an existing complex discovery algorithm to find clusters from the remaining network. Hub proteins are then added back to the generated clusters. We add a hub protein u back to a cluster C based on the connectivity between u and C , which is defined as follows:

$$Connectivity(u, C) = \frac{\sum_{v \in C} w(u, v)}{|C|}$$

where $w(u, v)$ is the weight of edge (u, v) , and it is calculated from the original PPI network using iterative AdjustCD [7] before removing hubs. If there is no edge between u and v , then $w(u, v)=0$. A hub protein u is added to a cluster C only if $Connectivity(u, C) \geq hub_add_thres$.

C. Combining the two methods

We combine the two methods together by first removing hub proteins from the given PPI network, and then decomposing the resultant PPI network using selected GO terms. The whole process is described below:

- 1) Let \mathcal{C} be the set of clusters generated. Initially \mathcal{C} is empty.
- 2) Remove hub proteins that have at least N_{hub} neighbors from the given PPI network G . Let G' be the resultant network.
- 3) Let g_1, \dots, g_m be the localization GO terms that are selected using threshold N_{GO} . For each g_i , do the following:
 - Remove proteins that are not annotated with g_i from G' . Let G'_i be the resultant network.
 - Apply a complex discovery algorithm on G'_i to find the set of clusters \mathcal{C}_i .
 - $\mathcal{C} = \mathcal{C} \cup \mathcal{C}_i$;
- 4) Remove duplicated clusters from \mathcal{C} .
- 5) Add hub proteins back to clusters in \mathcal{C} .

III. RESULTS

In this section, we first describe the datasets and the evaluation method used in our experiments, and then study the impact of the two decomposition methods on the performance of several complex discovery algorithms.

A. Experiment settings

PPI data. We used the yeast PPI dataset downloaded from BioGRID [9] (version 3.0.64) in our experiments. We kept only physical interactions. Self-interactions are removed. The dataset contains 5765 proteins and 52096 binary interactions.

Evaluation methods. Let S be a cluster and C be a reference complex. We define the matching score between S and C as the Jaccard index between S and C .

$$match_score(S, C) = \frac{|S \cap C|}{|S \cup C|}$$

Given a threshold $match_thres$, if $match_score(S, C) \geq match_thres$, then we say S and C match each other.

Given a set of reference complexes $\mathcal{C} = \{C_1, C_2, \dots, C_n\}$ and a set of predicted complexes $\mathcal{P} = \{S_1, S_2, \dots, S_m\}$, recall and precision are defined as follows:

$$Recall = \frac{|\{C_i | C_i \in \mathcal{C} \wedge \exists S_j \in \mathcal{P}, S_j \text{ matches } C_i\}|}{|\mathcal{C}|}$$

$$Precision = \frac{|\{S_j | S_j \in \mathcal{P} \wedge \exists C_i \in \mathcal{C}, C_i \text{ matches } S_j\}|}{|\mathcal{P}|}$$

There is often an inverse relationship between precision and recall. We use the harmonic mean of recall and precision, called F1-measure, to assess the overall performance.

Two sets of reference complexes are used in our experiments. One set of complexes are hand-curated complexes from MIPS [10], and the other set is generated by Aloy et al. [11]. We combine these two sets of complexes, and keep only those complexes with size no less than 4. Duplicated complexes are removed. Table I shows the number of complexes, number of proteins, the maximal, average and median size, and the average and median density of the complexes in the combined reference complex set.

Complex discovery algorithms. We used four complex discovery algorithms in our study. MCL [1] and RNSC

#cplx	#proteins	size			density	
		max	avg	median	avg	median
206	1318	95	13.60	8	0.631	0.700

Table I
STATISTICS OF REFERENCE COMPLEXES. ONLY COMPLEXES OF SIZE ≥ 4 ARE CONSIDERED.

algorithms	parameter settings
MCL	-I 1.8
RNSC	-e10 -D50 -d10 -t20 -T3
IPCA	-T0.4
CMC	<i>overlap_thres=0.5, merge_thres=0.4</i>

Table II
PARAMETER SETTINGS OF COMPLEX DISCOVERY ALGORITHMS.

[3] generate a partition of the PPI network, and they do not allow overlap among clusters. IPCA [6] and CMC [7] allow overlap among clusters. Unless stated explicitly, the parameters of the four algorithms are set as in Table II. Parameters not shown are set to their default values.

B. The GO term decomposition method

The first experiment studies the impact of the GO term decomposition method on the performance of the four algorithms. We use annotations in Gene Ontology [12] (dated 4 June, 2010) to select GO terms for decomposition.

Figure 1 shows the F1-measure of the four complex discovery algorithms when different N_{GO} thresholds are used for selecting localization GO terms. Overall, the performance of all the four algorithms improves. The precision of all the different N_{GO} values is improved considerably under all the different N_{GO} values. The recall is improved as well when $N_{GO} \geq 300$. When $N_{GO}=30$ or 100, recall of the four algorithms decreases. This is mainly because the GO terms selected are too specific in these two cases and too much information is lost. Hence we should use GO terms that are relatively general to decompose PPI networks to avoid breaking the whole network into tiny fragments. We have also tested other parameter settings of the four complex discovery algorithms besides that shown in Table II. The improvements achieved are all very similar to Figure 1.

We also compared the above improvement with that of using random protein groups for decomposition. Random protein groups are generated by replacing proteins of the selected GO terms with randomly picked proteins. We generated 100 sets of random protein groups and use their mean F1-measure as the result. Figure 2 shows that using random protein groups to decompose the PPI network decrease the performance of the four algorithms greatly, where the random protein groups were generated from GO terms selected at a threshold of 500.

C. The hub removal method

The second experiment studies the impact of the hub removal method on the performance of the four algorithms. Figure 3 shows the F1-measure of the four com-

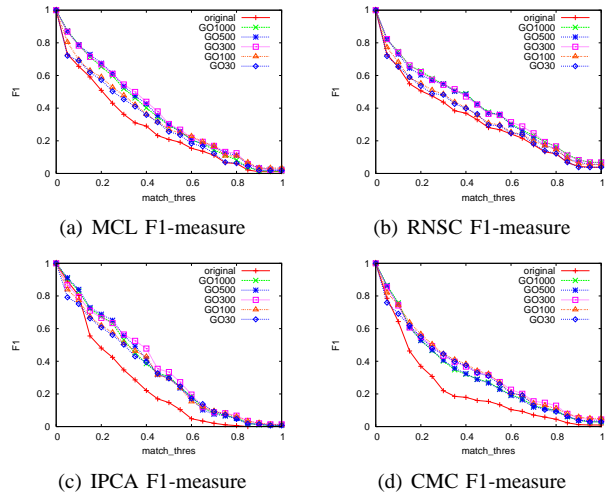


Figure 1. F1-measure of the four complex discovery algorithms when different sets of GO terms are used for decomposition. The X-axis is *match_thres*. “GO n ” means that the GO terms are selected using a threshold of n . For example, “GO1000” means that the GO terms are selected using a threshold of 1000. “original” means that complex discovery is performed on the original network without decomposition.

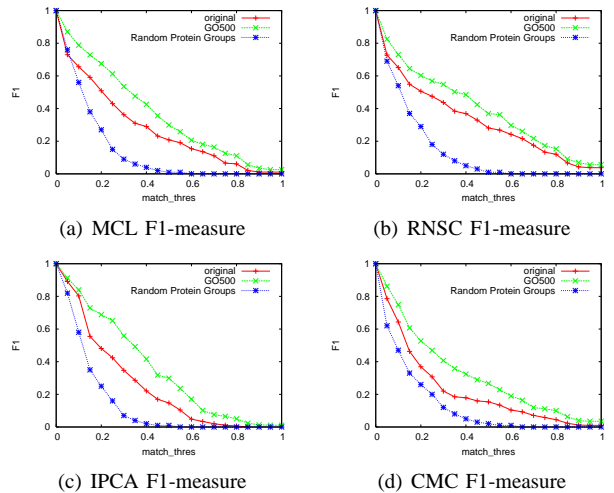


Figure 2. F1-measure of the four complex discovery algorithms when random protein groups are used for decomposition.

plex discovery algorithms when different N_{hub} thresholds are used for removing hub proteins. We use parameter *hub_add_thres* to control when to add a hub back to a cluster. In our experiments, we found that the proper range for *hub_add_thres* is [0.2, 0.9]. In the rest of the experiments, we set *hub_add_thres* to 0.3. The hub removal strategy is not helpful for RNSC and MCL, but is very helpful for IPCA and CMC. The main improvement of IPCA and CMC is on precision. The recall of the four algorithms decreases greatly when $N_{hub} \leq 30$, which indicates that too many hub proteins are removed when N_{hub} is too small.

D. Combining the two methods

The last experiment is to examine the combined impact of the two decomposition methods. Figure 4 shows the results.

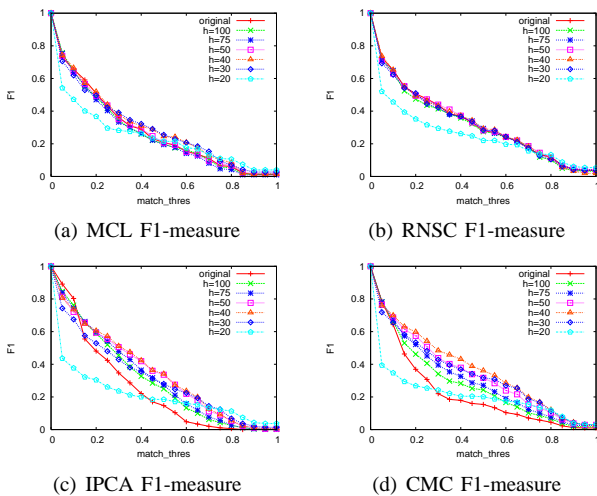


Figure 3. F1-measure of the four complex discovery algorithms when different N_{hub} values are used for removing hubs. The X-axis is *match_thres*. “h=n” means that a value of n is used to define hub proteins. For example, “h=100” means proteins with at least 100 neighbors are regarded as hubs. “original” means that complex discovery is performed on the original network without hub removal.

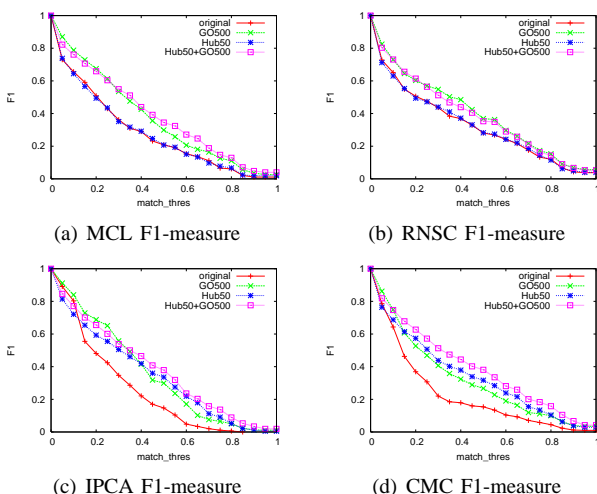


Figure 4. F1-measure of the four complex discovery algorithms when the two methods are performed in tandem. The X-axis is *match_thres*. “original” means the original network with neither hub removal nor GO term decomposition. “GO500” means that the network is decomposed using GO terms selected at a threshold of 500. “Hub50” means that hub proteins with at least 50 neighbors are removed. “Hub50+GO500” means that first hub proteins with at least 50 neighbors are removed, and the network is then decomposed using GO terms selected at a threshold of 500.

RNSC and MCL do not benefit much from the hub removal method, so for these two algorithms, combining the two decomposition methods yields little improvement compared with using GO decomposition alone. The performance of IPCA and CMC improve when both methods are used.

IV. DISCUSSION

In this paper, we proposed two methods to decompose PPI networks for complex discovery. We used four complex discovery algorithms to experimentally study the effective-

ness of the two methods. The results show that the two decomposition methods help improve the performance of the four algorithms significantly. For the GO term decomposition method, we recommend using localization GO terms that are relative general because their annotations are easier to obtain and they also preserve more information than GO terms that are very specific.

ACKNOWLEDGEMENT

This work was supported in part by a Singapore National Research Foundation grant NRF-G-CRP-2007-04-082(d) (Wong, Liu) and by a National University of Singapore NGS scholarship (Yong).

REFERENCES

- [1] S. van Dongen, “Graph clustering by flow simulation.” Ph.D. dissertation, University of Utrecht, 2000.
- [2] N. Przulj and D. Wigle, “Functional topology in a network of protein interactions.” *Bioinformatics*, vol. 20, no. 3, pp. 340–348, 2003.
- [3] A. D. King, N. Przulj, and I. Jurisica, “Protein complex prediction via cost-based clustering.” *Bioinformatics*, vol. 20, no. 17, pp. 3013–3020, 2004.
- [4] 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, no. 207, 2006.
- [5] B. Adamcsek, G. Palla, I. Farkas, I. Derenyi, and T. Vicsek, “Cfinder: locating cliques and overlapping modules in biological networks.” *Bioinformatics*, vol. 22, no. 8, pp. 1021–1023, 2006.
- [6] M. Li, J. Chen, J. Wang, B. Hu, and G. Chen, “Modifying the dpluss algorithm for identifying protein complexes based on new topological structures.” *BMC Bioinformatics*, vol. 9, no. 398, 2008.
- [7] G. Liu, L. Wong, and H. N. Chua, “Complex discovery from weighted ppi networks,” *Bioinformatics*, vol. 25, no. 15, pp. 1891–1897, 2009.
- [8] J.-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L. V. Zhang, D. Dupuy, A. J. M. Walhout, M. E. Cusick, F. P. Roth, and M. Vidal, “Evidence for dynamically organized modularity in the yeast protein-protein interaction network.” *Nature*, vol. 430, pp. 88–93, 2004.
- [9] C. Stark, B.-J. B. T. Reguly, L. Boucher1, A. Breitkreutz1, and M. Tyers, “Biogrid: a general repository for interaction datasets,” *Nucleic Acids Research*, vol. 34, no. Database Issue, pp. 535–539, 2006.
- [10] H. W. Mewes, C. Amid, R. Arnold, D. Frishman, U. Guldener, G. Mannhaupt, M. Munsterkotter, P. Pagel, N. Strack, V. Stumpflen, and J. Warfsmann, “Mips: analysis and annotation of proteins from whole genomes,” *Nucleic Acids Research*, vol. 32, no. Database issue, pp. 41–44, 2004.
- [11] P. Aloy, B. Bottcher, H. Ceulemans, C. Leutwein, C. Mellwig, S. Fischer, A.-C. Gavin, P. Bork, G. Superti-Furga, L. Serrano, and R. B. Russell, “Structure-based assembly of protein complexes in yeast,” *Science*, vol. 303, no. 5666, pp. 2026–2029, 2004.
- [12] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, and et al., “Gene ontology: tool for the unification of biology,” *Nature Genetics*, vol. 25, pp. 25–29, 2000.