Improving Functional Modularity in Protein-Protein Interactions Graphs using Hub-Induced Subgraphs

Duygu Ucar[1], Sitaram Asur[1], Umit Catalyurek[2], and Srinivasan Parthasarathy[12]

[1] Department of Computer Science and Engineering, The Ohio State University
[2] Department of Biomedical Informatics, The Ohio State University

Abstract. Dense subgraphs of Protein-Protein Interaction (PPI) graphs are believed to be potential functional modules and play an important role in inferring the functional behavior of proteins. PPI graphs are known to exhibit the scale-free property in which a few nodes (hubs) are highly connected. This scale-free topology of PPI graphs makes it hard to isolate dense subgraphs effectively. In this paper, we propose a novel refinement method based on neighborhoods and the biological importance of hub proteins. We show that this refinement improves the functional modularity of the PPI graph and leads to effective clustering into dense components. A detailed comparison of these dense components with the ones obtained from the original PPI graph reveal three major benefits of the refinement: i) Enhancement of existing functional groupings; ii) Isolation of new functional groupings; and iii) Soft clustering of multi-functional hub proteins to multiple functional groupings.

1 Introduction

Protein-Protein interaction (PPI) graphs have been obtained through accumulations of experimentally determined interactions between proteins. The presence of biologically relevant functional modules in PPI graphs has been theorized by many researchers [8, 14, 27]. Mining these graphs to isolate functional modules is a crucial task for the purposes of function prediction and identification in computational proteomics. However, extraction of these functional modules using traditional mining/clustering algorithms has proven to be difficult [21, 24].

The primary property of the PPI graph that is detrimental to traditional graph mining is its scale-free topology [22] with the degree distribution following the power law as $F(k) \sim k^{\alpha}$ where $\alpha < 0$. Most proteins in the graph participate in a small number of interactions while a few proteins, known as hubs, are involved in a large number of interactions. The topology typically consists of a giant central core containing a significant amount of proteins and their interactions. The rest of the proteins are either completely disconnected or part of small disconnected groups. The scale-free topology, (i.e. the tendency of the hubs to interact with a high fraction of proteins), makes isolation of modules hidden inside the central core all but impossible [21].

Another challenge in clustering PPI graphs is the need to assign proteins to different groups (soft clustering) based on their functions. Hub proteins typically have multiple functions and are likely to be essential for the organism. Recently, Karypis *et al* [1] presented several multi-level graph partitioning algorithms to address the difficulty of partitioning scale-free graphs. Although the proposed algorithms result in better groupings compared to traditional algorithms, they still do not perform soft clustering.

In order to address these issues, we suggest a key refinement of the PPI graph, motivated by the topological and biological importance of the hub proteins [15]. Our aim is to target the neighborhood of these potentially multi-faceted proteins and isolate, for each of their functions, corresponding densely connected regions. Our approach consists of two stages. In the first stage, we refine the PPI graph to improve functional modularity, using hub-induced subgraphs. We employ the Edge betweenness measure [19] to identify dense regions within the neighborhoods. In the second stage, we cluster the refined graph using traditional algorithms. Our end goal is to isolate components with high degree of overlap with known functional modules. An additional advantage of the refinement process is its ability to perform soft clustering of hub proteins.

Earlier approaches that focus on elimination of hubs from the scale-free graphs have found that this disconnects the graph and breaks down the modules as well [2, 12]. Recently, Costa [11] introduced a hub-centered community detection algorithm. We believe that this will not be effective in scale-free graphs since hubs have a large number of neighbors which cannot all be part of the same community. To the best of our knowledge, we are the first to suggest duplicating hubs to improve modular decomposition of scale-free graphs. Although, in this work, we focus on PPI graphs, our refinement technique is applicable to any scale-free graph.

Other groups have attempted to extract dense regions to isolate protein complexes from PPI graphs [5, 17] using concepts such as k-cores or cliques. Although dense regions of the PPI graph are highly associated with known functional modules, they are by themselves not entirely informative in terms of function prediction. Mining the entire PPI graph will definitely prove to be a superior source for novel discovery of protein functions. Hence, we aim to improve the modularity of a PPI graph, as a whole, which enables enhanced functional prediction/identification of every protein of the graph.

The proposed refinement technique is evaluated on the PPI graph of Saccharomyces Cerevisiae obtained from the DIP (Database of Interacting Proteins) database. In order to quantify the quality of our clustering, we employ both topology-based and domain based validation metrics. We find that the clusters we obtain after refinement match very well with known biological annotations. In addition, we obtain groupings after refinement that could not be obtained from the original graph. Our technique also allows soft clustering of multi-functional proteins. We find that each of these clusters include proteins sharing a certain function with the multi-functional protein.

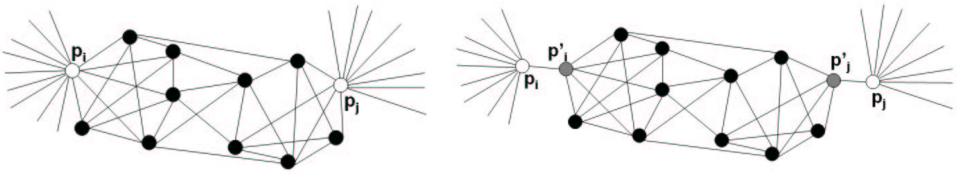
2 Graph Refinement

2.1 Evolutionary Implications

Recently, several groups [6, 10, 25] have suggested mathematical models to explain the evolutionary growth of Protein-Protein interactions graphs. They claim that preferential attachment is one of the main causes for the scale-free topology of interaction graphs. According to the duplication-divergence model proposed by Vazquez *et al* [25], there is a linear relation between a node's degree and the probability of a new node attaching to that node, known as preferential attachment. Since hubs have very high degrees, new proteins added to the graph are more likely to interact with hubs rather than other nodes. Hence, if a hub belongs to a functional module, most of the other proteins in that module will prefer to connect to the hub rather than a node with the same function but less degree. This suggests that proteins with the same function interact within themselves and also individually with at least one hub. For this reason, we believe that it is important to consider neighborhoods of hubs to isolate functional modules.

2.2 Hub-induced Subgraphs

As stated before, hubs typically tend to be essential proteins [15], having several important functions inside the cell. Hubs can therefore be linked to several functional modules. However, most of their interactions do not imply a functional similarity. In this work we are aiming to identify the neighbors of hubs that share functionalities with the hub protein. Hence, our goal is to identify all dense components that lie within the neighborhood of a hub. Once such components are identified, the neighboring hub is duplicated and all its interactions with the members of the group reassigned to the duplicate. In addition, all the duplicates will be linked to the original hub to preserve the original interactions of the proteins belonging to the isolated dense component. Note that we are not eliminating any interactions. We are merely re-assigning interactions between the proteins belonging to the dense components and the hub to the duplicate. If



(a) Example graph (b) Graph after hub duplications

Fig. 1. Illustration of Hub Duplication. Proteins p_i and p_j are duplicated and the duplicates, p'_i and p'_j are connected to dense components as well as the original proteins.

the proteins of a functional module are divided across neighborhoods of several

hubs, each of those hubs will be duplicated once and will be included in the functional module. This will isolate the functional module from the unrelated neighbors of the hubs and create a tightly knit group. An example can be seen in Figure 1.

We perform duplication of hubs into several new nodes for each dense component in the hub's neighborhood. In order to identify these dense components, we introduce the notion of a hub-induced subgraph.

Definition 1: Let $G = (V, E)$ be a graph. $G' = (V', E')$ is a *vertex-induced subgraph* of G if $V' \subseteq V$ and E' is all the edges of G between elements of V'.
Definition 2: A *hub-induced subgraph* of G is a graph $G'' = (V'', E'')$, where V'' corresponds to a hub's adjacency list.

Thus, for every hub of the graph, there exists a corresponding hub-induced subgraph obtained from the adjacency list of the hub. We isolate these hub-induced subgraphs to identify potential functional modules. For more details on our analysis of these hub-induced subgraphs, please refer our technical report [23].

2.3 Hub Duplication

To obtain information about the neighborhoods of hubs, we use the Edge betweenness measure which was first introduced by Newman *et al* [19]. This measure favors edges between communities and disfavors ones within communities. Newman *et al* introduced three different Edge betweenness measures; Shortest-path, Random-walk and Current-flow. We use the Shortest-path betweenness measure, which considers the number of shortest paths between all pair of nodes going along each edge.

Given a graph, $G(V, E)$ and 'known number of partitions'(k), the algorithm identifies k sub-groups such that the intra-group connections are dense and inter-group connections sparse, by repetitively removing edges with high Betweenness values. Our goal is to detect dense regions inside each hub-induced subgraph. We implement the algorithm without the k parameter and include the clustering coefficient of the subgraphs as the stopping criteria.

The clustering coefficient [26] is a measure that represents the interconnectivity of a vertex's neighbors. The clustering coefficient of a vertex v is defined as the proportion of edges between its direct neighbors to the number of edges that could possibly exist between them. The clustering coefficient for the whole graph is the mean over the coefficients of all vertices in it and lies between 0 and 1. Tightly knit groups are associated with high clustering coefficients.

Although the Shortest-path betweenness algorithm is computationally costly ($O(E^2V)$ running time), since the hub-induced subgraphs are small in size ($<$ 284 nodes), it is tractable for our purpose. The pseudo-code of our refinement algorithm with Shortest-path betweenness and clustering coefficient measures is given in Algorithm 1. Here, *most-between-edge(G_i)* returns the edge with the highest Shortest-path betweenness score in the G_i subgraph. T_{size} represents the minimum size(number of nodes) of the dense components we will consider and

T_{cc} represents the clustering coefficient threshold. When a component (of size $>= T_{size}$) is dense enough, algorithm calls $DuplicateHub$ function to duplicate the corresponding hub and re-assign its interactions with the members of the dense component to the duplicate. For each dense component identified from a hub-induced subgraph a duplication event takes place. [1]

Algorithm 1 Identify-Dense-Regions(G_i)

```
INPUT G_i = (V_i, E_i) : hub-induced subgraph of Hub_i
if size(G_i) < T_size then
  Return
else if CC(G_i) ≥ T_cc then
  DuplicateHub(Hub_i,G_i)
else
  e = most-between-edge(G_i)
  //remove e from G_i
  G_i ← G_i − e
  recalculate Edge betweenness values
  if G_i is partitioned into G_i^1 and G_i^2 then
    Identify-Dense-Regions(G_i^1)
    Identify-Dense-Regions(G_i^2)
  else
    Identify-Dense-Regions(G_i)
  end if
end if
```

3 Methods

3.1 Clustering Algorithms

Once the PPI graph is refined using hub-induced subgraphs, the resulting graph is clustered to separate out the functional modules. We used two graph clustering algorithms - a single-level Spectral algorithm and kMETIS [16], a multi-level partitioning algorithm. The Spectral-based algorithm uses Eigenvectors of the Laplacian matrix constructed from the graph to determine effective clusters of the graph. These clusters minimize the total weight of the edge cut. The kMETIS algorithm, obtains a k-way partition of the approximate graph and refines it to construct a k-way partitioning of the original graph. For more details about these algorithms please refer our technical report [23].

3.2 Validation Measures

Topological Measure To evaluate our clusters, we use a topology-based modularity metric proposed by Newman [19]. This metric considers a k X k symmetric

[1] Note that, Betweenness scores are recalculated whenever an edge is removed from the graph to capture the topology of the remaining graph.

matrix of clusters where each element a_{ij} represents the fraction of edges that link nodes between clusters i and j and each a_{ii} represents the fraction of edges linking vertices within cluster i. The modularity measure is given by

$$M = \sum_i (a_{ii} - (\sum_j a_{ij})^2) \tag{1}$$

Statistical Measure based on Domain Information To test if the clusters obtained correspond to known functional modules, we need to validate our dense components using known biological associations. We used the Gene Ontology Consortium Online Database [4] to look for biological relations between proteins assigned to the same cluster. The Gene Ontology (GO) is a controlled vocabulary designed to accumulate the result of all investigations in the area of genomics and biomedicine by providing a large database of known associations containing common terminology that can be used among researchers. GO provides three ontologies - cellular component(CC), molecular function(MF) and biological process(BP). Cellular component terms refer to the localization of proteins inside the cell. Molecular function terms refer to shared activities at the molecular level and biological process terms refer to entities at both the cellular and organism levels of granularity. We used all three annotations for validation and comparison in accordance with earlier works [24, 3].

Merely counting the proteins that share an annotation will be misleading since the underlying distribution of genes among different annotations is not uniform. Hence, we use p-values to calculate the statistical significance of a group of proteins that share a GO term. The p-values essentially represent the chance of observing that particular grouping, or better, given the background distribution. Assume we have a cluster of size n, out of which m proteins share a particular annotation. Also, there are N proteins in the database with M of them known to have that same annotation. Then using the Hypergeometric Distribution, the probability of observing m or more proteins that are annotated with the same GO term out of n proteins is:

$$p - value = \sum_{i=m}^{n} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}} \tag{2}$$

Smaller p-values imply that the grouping is not random and is more significant biologically than one with a higher p-value. A cut-off parameter (alpha level) is used to differentiate significant groups from the insignificant ones. If a group of proteins are associated with a p-value greater than the cut-off, they are considered insignificant. We used the recommended cut-off of 0.05 for all our validations.

As the p-value of a single cluster is statistically not representative, we define a Clustering score function in order to quantify the overall clusters. We defined this score as follows.

$$Clustering\ score = \frac{\sum_{i=1}^{n_S} min(p_i) + (n_I * cutoff)}{n_S + n_I} \tag{3}$$

where n_S and n_I denotes the number of significant and insignificant clusters, respectively. $cutoff$ stands for the alpha level(0.05) whereas $min(p_i)$ denotes the smallest p-value of the significant cluster i. Hence, each cluster is associated with one p-value for each of the three ontologies.

4 Experimental Results

In this section, we discuss our experimental results. **Topology-based Modularity:** First, we use the Modularity metric on the clusters obtained using both the kMETIS and Spectral algorithms. Figure 2-b shows the modularity compar-

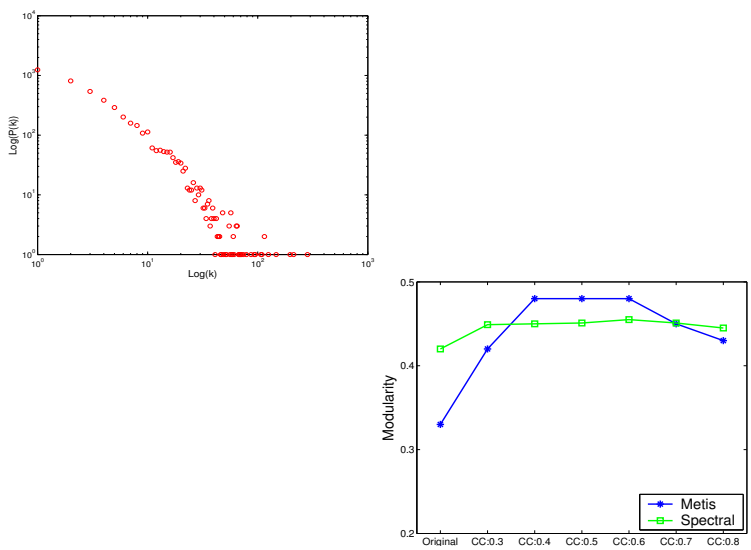
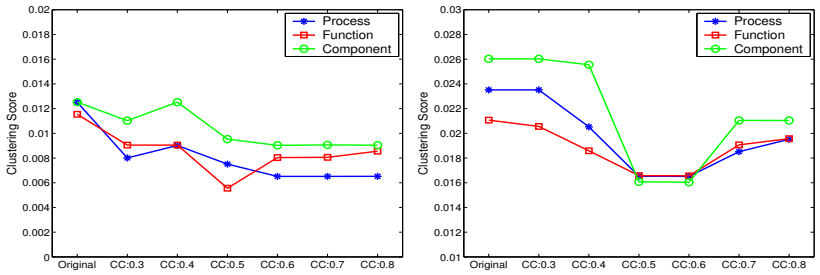


Fig. 2. a) Degree distribution of DIP dataset b) Modularity scores before($Original$) and after ($CC : 0.3$ to $CC : 0.8$) refinement

ison between the original graph and refined graphs for the two algorithms. We find that the refinement improves the modularity of the graph for both algorithms. From the curve, we find that the modularity scores peak at clustering coefficient values between 0.4 and 0.6 in both cases. Further, the modularity for clustering the original graph is much lower than for any of the refined graphs. kMETIS produces clusters with higher modularity(upto 45% better than the original) than Spectral(upto 8% better than the original) for the refined graphs.

Biological Modularity: Next, we test the effectiveness of our refinement technique by comparing with the clusters obtained from the original graph. The DIP dataset consists of 15147 interactions among 4741 proteins. In our work,

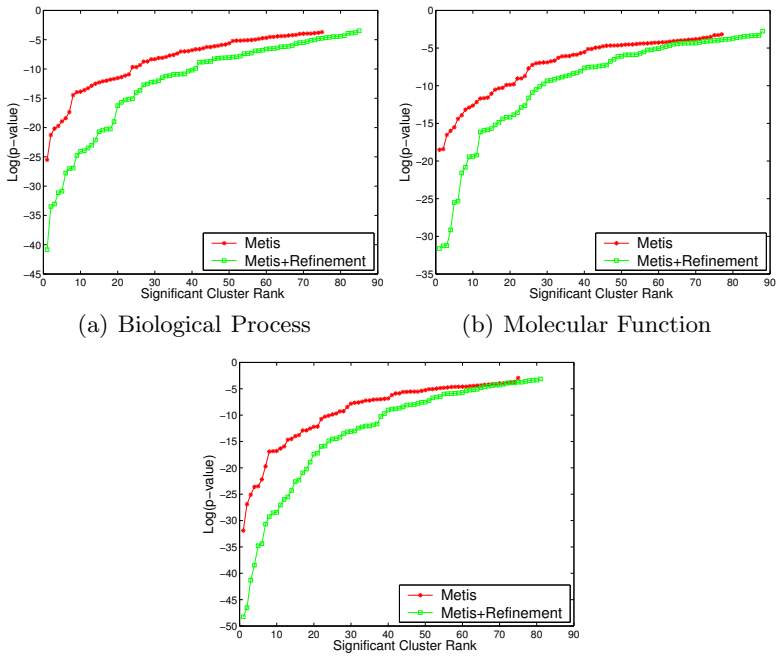
we analyzed the degree distribution of the graph (shown in Figure 2-a) and defined all nodes of degree greater than 25 (2% of all nodes) to be hubs. We ran the algorithm to find all dense components within every hub-induced subgraph using the clustering coefficient and size as stopping criteria. We chose 6 as the size threshold for dense components, since components with size smaller than 6 are likely to be insignificant. To choose a suitable threshold for the clustering coefficient, there are two things that we should consider. First, we want the resulting components to be dense enough to correspond to a functional module. We vary the clustering coefficient parameter (T_{cc}) between 0.3 and 0.8 and obtain refined graphs for each. We believe that, considering the incompleteness in PPI graphs and the need for obtaining dense components, a reasonable clustering coefficient value would be around 0.5-0.6. Components that have clustering coefficients within this range are likely to be dense enough to be considered as functional groups and would not be affected too much by the incomplete nature of the dataset. The refined graphs and the original graph are clustered by kMETIS and Spectral clustering algorithms separately. The results obtained are depicted in Figure 3. As can be seen from this figure, Clustering scores are reducing (improving) after refinement for both algorithms. Our above hypothesis is validated by the fact that, although an improvement is observed for every clustering coefficient threshold, this improvement is small for low and high values. Also, both algorithms have their smallest Clustering scores for the threshold values of 0.5, 0.6 and 0.7. If we consider the improvement in this clustering coefficient range, our refinement technique improves Clustering scores up to 52%, 48% and 28% for MF, BP and CC ontologies in the case of kMETIS and 30%, 21% and 38% for the same three ontologies for the Spectral algorithm. This confirms that kMETIS produces better clusters than the Spectral algorithm. Note that our Clustering score considers both significant and insignificant clusters. Next,



(a) Metis (b) Spectral

Fig. 3. Clustering scores before and after refinement algorithm is shown. Obtained by kMETIS and Spectral clustering algorithms. *Original* refers to the PPI dataset before the refinement. $CC : 0.3$, $CC : 0.4$,$CC : 0.5$,$CC : 0.6$, $CC : 0.7$, $CC : 0.8$ and $CC : 0.9$ refer to refined graphs with the respective clustering coefficient threshold. Process, Function and Component represent Biological Process, Molecular Function and Cellular Component ontologies respectively.

we evaluate the significance of our clustering results for all three ontologies. In Figure 4(a-b) we show the p-value distribution of significant clusters in both original and refined graphs for all three ontologies. For all ontologies, we find that the refined graph can be clustered into more biologically meaningful groups. For example, the best cluster we obtained on the original graph had a p-value of 8.2089e-25 for Biological Process, whereas the best cluster after refinement had a p-value of 5.4658e-41 for the same ontology. We obtained similar results for the other two ontologies. In addition, we are able to identify more significant clusters after the refinement for all three ontologies.



(a) Biological Process (b) Molecular Function

(c) Cellular Component

Fig. 4. P-value distribution of significant clusters before and after the refinement. The y axis represents the log(p-value) for each corresponding cluster.

5 Discussion and Conclusion

In this paper, we have proposed a refinement technique to improve modular decomposition of PPI graphs. We refined the PPI graph based on Shortest-path betweenness and clustering coefficient measures. From our experimental results, we found that duplicating the hubs of a scale-free PPI graph improves the modularity of the graph. Thus, we are able to obtain topologically and biologically

more significant clusters even using traditional clustering algorithms. A detailed examination of the obtained clusters revealed that the proposed method has three major benefits:

- Enhancement of available functional groupings: We obtain larger groups of proteins that are annotated with the same GO term from our refined graph than on the original graph.
- Isolation of new functional groupings: We find groupings of proteins that could not be obtained from the original graph.
- Soft-clustering: Our approach can identify multi-functional hub proteins and group them into modules corresponding to each of their functions.

We now provide some illustrations from our results for each of these cases. For more details, please refer our technical report [23]: KAP95 (karyopherin beta), an essential protein is known to take part in 'nucleocytoplasmic transport'. Specifically, it participates in a complex mediating nuclear import via a localization signal(NLS). It interacts with nucleoporins to guide transport across the nuclear pore complex [13]. When we cluster the original DIP dataset, this protein is correctly grouped with 8 proteins that are also annotated with 'nucleocytoplasmic transport' term with p-value 1.14e-08.

Using our refinement technique, KAP95 is duplicated once. The hub and its duplicate appear in two separate clusters when we use the kMETIS algorithm. In one, KAP95 is grouped with proteins 18 other proteins that share the same biological process ('nucleocytoplasmic transport') with p-value 1.07e-27. The major difference between this group and the one from the original graph are the inclusion of NUPs(Nucleoporins - 8 proteins) and KAPs(Karyopherins - 3 proteins). Transport through the nuclear pore complex is facilitated by transient interactions between the KAPs and the nuclear pore complex proteins (NUPs) [20]. Thus, locating NUPs and KAPs together is a noticeable benefit caused by our refinement. Clearly, our approach groups more proteins that belong to the same functional module together. This suggests that hub duplications make isolation of modules easier. These clusters are also valuable for predicting the functions of unknown proteins. In the above group, four proteins (YKL061W,YKR064W, YNL122C, YER004W) do not have a known function. Among these four, YKL061W is predicted by Brun et al [7] to take part in 'nucleus-cytoplasm transport' process which is in accordance with our findings. Since two different datasets and approaches are used to infer the same conclusion about protein YKL061W, the overlap is noteworthy. This also suggests that the other three proteins might have an unrevealed task in 'nucleocytoplasmic transport' biological process.

In addition to enhancing clusters, our method is able to assign hub proteins which were originally in insignificant clusters into significant clusters. To illustrate this, we consider the hub protein LSM8. The LSM(Sm-like) proteins interact with each other and with U6 snRNA complex and influence pre-mRNA splicing [18]. In the original dataset, this protein is assigned to a cluster which does not have any significant annotations. However, after the refinement, this protein is located into a cluster which has a biological process annotation with

p-value 1.2e-12. In addition to LSM8, ten other proteins in this group are associated with 'mRNA splicing'. LSM8 is located with the members of its complex (other SM-like proteins) as well as the components of U6 snRNP complex(PRP proteins). This example shows that our technique not only improves functional modules which can be identified from the original dataset, but also allows detection of functional modules which cannot be discovered from the original dataset.

Another advantage of our refinement technique is its ability to perform soft clustering on certain hub proteins. CKA1 is one of these multi-faceted proteins and is involved in several cellular events such as maintenance of cell morphology and polarity, and regulating the actin and tubulin cytoskeletons [9]. When the original dataset was clustered, CKA1 and seven other proteins, annotated with 'transcription, DNA-dependent' term are located in the same cluster(p-value 3.47e-05). On the other hand, our algorithm duplicates CKA1 twice. When we cluster, these 3 nodes are then assigned to different clusters resulting in three different groupings for protein CKA1. All three correspond to different functional modules of the CKA1 protein. One of these clusters is an enhancement of the 'transcription, DNA-dependent' functional module(very low p-value of 2.3e-19). The second cluster includes proteins which are annotated with the biological process term 'protein amino acid phosphorylation' with p-value 1.2e-05. CKA1 is itself annotated with the same term. The third cluster contains 21 proteins and CKA1, all of which are annotated for 'organelle organization and biogenesis' (with p-value 3.2e-12). Thus, we found that our technique, not only improved the obtainable clusters (by decreasing p-value from 3.47e-05 to 2.3e-19), but also grouped CKA1 with proteins that share its different functions. Altogether these examples indicate the effectiveness of our approach on isolation of functional modules from the PPI graphs.

Although, in this work, we have applied our refinement technique to PPI graphs, we strongly believe that it will be equally effective for all other scale-free graphs such as social networks.

References

1. A Abou-Rjeili and G Karypis. Multilevel algorithms for partitioning power-law graphs. *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2006.
2. R Albert, H Jeong, and A L Barabasi. Error and attack tolerance in complex networks. *Nature*, 406:378–382, 2000.
3. V Arnau, S Mars, and I Marin. Iterative cluster analysis of protein interaction data. *Bioinformatics*, 21:3:364–378, 2005.
4. M Ashburner and *et al.* Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet.*, 25(1):25–29, May 2000.
5. G D Bader and C WV Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinfomatics*, 4:2, 2003.
6. J Berg, M Lssig, and Andreas Wagner. Structure and evolution of protein interaction networks: a statistical model for link dynamics and gene duplications. *BMC Evolutionary Biology*, 4:51, 2004.

7. C Brun, F Chevenet, D Martin, J Wojcik, A Gunoche, and B Jacq. Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biology*, 5, 2003.
8. C Brun, C Herrmann, and A Guenoche. Clustering proteins from interaction networks for the prediction of cellular functions. *BMC Bioinformatics*, 5(95), July 2004.
9. D A Canton and D W Litchfield. The shape of things to come: An emerging role for protein kinase ck2 in the regulation of cell morphology and the cytoskeleton. *Cell Signalling*, 18:267–275, 2006.
10. F Chung, L Lu, T G Dewey, and D J Galas. Duplication models for biological networks, 2002.
11. L F Costa. Hub-based community finding. *http://www.citebase.org/cgi-bin/citations?id=oai:arXiv.org:cond-mat/0405022*, 2004.
12. P Crucitti, V Latora, M Marchiori, and A Rapisarda. Error and attack tolerance of complex networks. *Physica A*, 340:388–394, 2004.
13. D Gilchrist and M Rexach. Molecular basis for the rapid dissociation of nuclear localization signals from karyopherin alpha in the nucleoplasm. *J. Biol. Chem*, 278:51:51937–51949, 2003.
14. J Hua, D Koes, and Z Kou. Finding motifs in protein-protein interaction networks. *Project Final Report, CMU www.cs.cmu.edu/ dkoes/research/prot-prot.pdf*, 2003.
15. H Jeong, S P Mason, A L Barabasi, and Z N Oltvai. Lethality and centrality in protein networks. *Nature. 411:44.*, 411:41–42, 2001.
16. G Karypis and V Kumar. Unstructured graph partitioning and sparse matrix ordering system. technical report. *http://www-users.cs.umn.edu/ karypis/metis/metis/files/manual.pdf.*
17. X-L Li, S-H Tan, C-S Foo, and S-K Ng. Interaction graph mining for protein complexes using local clique merging. *Genome Informatics*, 16(2):260–269, 2005.
18. A E Mayes, L Verdone, P Legrain, and J D Beggs. Characterization of sm-like proteins in yeast and their association with u6 snrna. *EMBO J.*, 18(15):4321–4331, 1999.
19. M E J Newman and M Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.
20. L F Pemberton and B M Paschal. Mechanisms of receptor-mediated nuclear import and nuclear export. *Traffic*, 6:187, 2005.
21. A L Barabsi S Yook, Z N Oltvai. Functional and topological characterization of protein interaction networks. *Proteomics*, 4:928–942, 2004.
22. A Thomas, R Cannings, N A M Monk, and C Cannings. On the structure of protein-protein interaction networks. *Biochemical Society Transactions*, 31:1491–1496, 2003.
23. D Ucar, S Asur, U Catalyurek, and S Parthasarathy. Improving functional modularity in protein-protein interactions graphs using hub-induced subgraphs. *http://www.cse.ohio-state.edu/research/techReport.shtml. Electronic report under 2006/TR41.pdf*, 2006.
24. D Ucar, S Parthasarathy, S Asur, and C Wang. Effective preprocessing strategies for functional clustering of a protein-protein interactions network. *IEEE, International Symposium on Bioinformatics and Bioengineering, BIBE*, 2005.
25. A Vazquez, A Flammini, A Maritan, and A Vespignani. Modeling of protein interaction networks. *Complexus*, 1:38, 2003.
26. D Watts and S Strogatz. Collective dynamics of small world networks. *Nature*, 393(6684):440–442, June 1998.

27. L F Wu, T R Hughes, A P Davierwala, M D Robinson, R Stoughton, and S J Altschuler. Large-scale prediction of saccharomyces cerevisiae gene function using overlapping transcriptional clusters. *Nature Genetics*, 31:255–265, June 2002.