

Effective Pre-processing Strategies for Functional Clustering of a Protein-Protein Interactions Network

Duygu Ucar, Srinivasan Parthasarathy, Sitaram Asur and Chao Wang
Department of Computer Science and Engineering, The Ohio State University
Contact: {srini}@cse.ohio-state.edu

Abstract

In this article we present novel preprocessing techniques, based on topological measures of the network, to identify clusters of proteins from Protein-protein interaction (PPI) networks wherein each cluster corresponds to a group of functionally similar proteins. The two main problems with analyzing Protein-protein interaction networks are their scale-free property and the large number of false positive interactions that they contain. Our preprocessing techniques use a key transformation and separate weighting functions to effectively eliminate suspect edges, potential false positives, from the graph. A useful side-effect of this transformation is that the resulting graph is no longer scale free. We then examine the application of two well-known clustering techniques, namely Hierarchical and Multilevel Graph Partitioning on the reduced network. We define suitable statistical metrics to evaluate our clusters meaningfully. From our study, we discover that the application of clustering on the pre-processed network results in significantly improved, biologically relevant and balanced clusters when compared with clusters derived from the original network. We strongly believe that our strategies would prove invaluable to future studies on prediction of protein functionality from PPI networks.

1 Introduction

Protein-protein interaction (PPI) networks are believed to be important sources of information related to biological processes and complex metabolic functions of the cell. The presence of biologically relevant functional modules in PPI networks has been theorized by many researchers [6, 11, 18]. The task of extracting these functional modules for the purposes of functional prediction and identification is an active research area in functional genomics.

Clustering techniques are adequate to extract tightly connected modules from the interaction network. However, it has been observed that no single clustering algorithm can adequately reflect the underlying biological functions of proteins in their clusters [13]. Hence, various traditional

clustering and graph partitioning algorithms have been applied in this domain with mixed results. The cluster assignments have been found to vary significantly with the algorithm and its parameters. In this paper, we examine the application of two different clustering approaches - a hierarchical agglomerative clustering algorithm and a multi-way graph partitioning algorithm.

The primary property of the PPI network that is detrimental to traditional graph partitioning or clustering is its scale-free topology [16]. In scale-free networks, the node degree distribution follows a power-law as $P(k) \sim \frac{1}{k^\gamma}$. This produces a skew in the data-space, with a few highly connected proteins (hubs) linking the rest of the proteins to the system, which several conventional clustering algorithms cannot handle effectively. An additional problem is the unreliability of the interaction data. PPI data are typically obtained from high-throughput screen sources such as the Yeast two-Hybrid (Y2H) system [8]. Although this system produces a large number of protein interactions, the resultant interactions have been shown to include a large number of false positives [7]. These are caused mostly due to the testing of increasingly arbitrary protein-protein interactions. The interactions data, therefore, includes several physical interactions with no biological significance.

In this paper, we pre-process the data using *Line graph transformation* based on two topological metrics to transform the PPI network into a sparser network with reduced interactions. Our aim is to show that the transformed graph contains fewer false positives and leads to a more biologically relevant partitioning than the original graph. To validate our results, we use the Gene Ontology (GO) consortium database [5], which provides structured vocabularies (ontologies) to annotate genes in terms of their associations in biological processes, molecular functions and cellular components. The primary purpose of these annotations is to provide a common terminology to identify biologically relevant associations among genes. To summarize, our main contributions in this paper are :

- Novel pre-processing strategies to eliminate redundant false positive interactions from the original PPI dataset

- Application of two different clustering algorithms with suitable validation to obtain biologically meaningful results from the cleaned dataset.

2 Methodology

2.1 Dataset

The Database of Interacting Proteins (DIP) [19] is an online database, that accumulates experimentally determined protein-protein interactions from different sources. In this work, we focus on the budding Yeast (*Saccharomyces Cerevisiae*) proteome since it is a well-studied organism with large amounts of interactions data. As of May 2005, the database contains 4741 Yeast proteins having 15428 interactions. For the purpose of this study, the network can be visualized as a graph with nodes representing proteins and the edges between them denoting the corresponding interactions. Hence, we use the terms network and graph interchangeably in this paper.

2.2 Pre-processing

The majority of the interactions on the DIP database are obtained using the Yeast two-hybrid (Y2H) system. Fields and Song first described the features of the Y2H system in 1989 [8]. It has become one of the most commonly used technologies to detect protein-protein interactions. Its main advantages are its simplicity, low cost and high throughput. However, it is burdened by a tendency to produce a large number of false positives. A number of studies made to assess the quality of the data have demonstrated large number of erroneously identified interactions. Hence, the biological relevance of interacting proteins obtained from this system needs to be re-affirmed. In this paper, we provide a pre-processing technique that uses two different metrics to identify and eliminate interactions which are most likely to be false. We then proceed to partition the PPI graph using only the edges (interactions) which we believe to be reliable.

Although, it is impossible without experimental examination to determine if the interactions eliminated are indeed false, we believe that the presence of balanced, biologically significant clusters on the cleaned data serves as a preliminary validation of our technique. There has been work done by Saito *et al* [14], to eradicate false positive interactions using a metric called Interaction Generality. However, this metric focuses only on the degree of individual proteins without considering the topology of the network. We believe that the degree, by itself, is not sufficient. *It is important to consider connectivity and density of sub-networks to adequately deal with false positive interactions.* Our technique is therefore governed by the following intuition. If a node is strongly connected to its neighbors (i.e., lies inside

a dense subnetwork), it is obvious that the proposed interaction is supported by several other interactions. Hence, the edges (interactions) that are not part of dense subnetworks are more likely to be interactions that are falsely obtained. Edges that connect subnetworks are also potential false interactions. Hence we use topological metrics of the network, namely the Clustering Coefficient and Centrality (Betweenness and Closeness), to quantify the possibility of an interaction being false.

2.2.1 Clustering Coefficient

The Clustering Coefficient [17], is a metric commonly employed to identify well-connected sub-components in networks. It represents the interconnectivity of a node's neighbors. The Clustering Coefficient of a node v in a graph can be defined as follows:

$$CC(v) = \frac{2n_v}{k_v(k_v - 1)} \quad (1)$$

where n_v denotes the number of triangles that go through node v . The denominator gives the maximum number of triangles that can go through node v . It is implied that nodes having high Clustering Coefficient have neighbors that have higher probability to be neighbors.

2.2.2 Centrality

The Centrality of a node in a network is a measure of the structural importance of the node. There are three important aspects of Centrality: Degree, Closeness, and Betweenness. In this work we use Betweenness and Closeness as they are more informative than degree and more suitable for this problem.

Betweenness Centrality: Betweenness [10], is a measure of the centrality of a node and its influence over data flows in the network. For a node v , it is normally calculated as the fraction of the shortest geodesic paths between node pairs that pass through node v . More precisely, if $d_v(i, j)$ is the number of paths from i to j that pass through node v in a graph G having n nodes, then the Betweenness Centrality of node v can be calculated as

$$B(v) = \frac{\sum_{i,v,j \in G} d_v(i, j)}{(n-1)(n-2)} \quad (2)$$

Closeness Centrality : Closeness Centrality [9], is a measure of the closeness of a node, on average, to all the other nodes. Formally the closeness of a node v in a graph G is defined by the following expression:

$$C(v) = \frac{N-1}{\sum_{v,w \in G} d(v, w)} \quad (3)$$

where $d(v, w)$ denotes the pairwise geodesic distance between node v and w . N denotes the number of reachable

nodes from node v . Due to the scale free property, the nodes with the highest closeness scores in the PPI network are the hubs and hence they are viewed as core components of the network.

2.2.3 Line Graph Transformation

All the above metrics are defined for nodes of a graph. However we do not want to eliminate any nodes (proteins) of our PPI network since we wish to cluster them at the end. Our aim, therefore is to attack the edges (interactions) of a given network. In order to use metrics defined on nodes, we transform our data into a line graph representation [15]. In this representation, each node corresponds to an edge in the original graph and two nodes are connected if and only if they (the corresponding edges) have a common endpoint (i.e, protein) in the original graph.

Line graph transformation has several advantages for our purposes. First, it emphasizes the edges (interactions) rather than nodes. Since we are considering eliminating false positive interactions, this proves to be useful. Second, it retains information about the proteins involved. Hence, we are able to cluster all the proteins. Pereira *et al* [13], have used line graph transformation of a PPI network to deduce functional modules. Their work does not take into account the false positive interactions present in the PPI network. Another issue with clustering line graphs is the increased complexity. Since edges in the original graph form nodes in the line graph, the latter is significantly larger than the former. Conventional clustering algorithms do not scale well when confronted with graphs of this size.

Since a graph partition and its line graph representation have very different topologies in terms of compactness, we believe that finding dense subcomponents on the line graph will not reveal the actual dense regions of the original graph. Hence, we used the line graph representation only for pre-processing purposes. We transform the reduced line graph back to the original graph before clustering. Although various studies have been made on line graphs, no earlier work has focused on using the line graph transformation to perform pre-processing on the protein-protein interactions, to the best of our knowledge.

Some work has also been done on the Clustering Coefficient of line graphs. Nacher *et al* [12], showed that nodes with high Clustering Coefficient in the original graph will have high Clustering Coefficient after line graph transformation. Since proteins sharing a significant number of interaction partners are likely to participate in common cellular processes, nodes with high Clustering Coefficient in the line graph are more likely to participate in efficient partitions. As a result we decided to remove the nodes that have low Clustering Coefficient in line graph since they will correspond to edges that are not parts of any dense sub-

components in the original network and are therefore most likely to be false positives. FAS Research [4] studied the node betweenness of a line graph. They claimed that betweenness values calculated for the nodes of a line graph provide information about the contribution of each edge to the betweenness in the original network. In our work, we remove nodes with high Clustering Coefficient and low Centrality values. The remaining subgraph then contains only the nodes that belong to dense subcomponents.

2.3 Clustering-Partitioning Algorithms

As we mentioned earlier, we use two different clustering algorithms - an agglomerative hierarchical algorithm and a multi-level graph partitioning algorithm.

2.3.1 Agglomerative Hierarchical Clustering

Hierarchical clustering is a traditional clustering technique which is popular in this domain due to its robustness. Agglomerative clustering is a bottom-up hierarchical clustering paradigm which results in a nested set of clusters, where at each level, clusters are generated by merging clusters of a lower level, and at the bottom level each cluster is a single entity (e.g. a single protein in our case). At each level proteins that are most similar are merged to form higher level clusters. This hierarchical clustering process can be represented as a tree, or dendrogram, where each step in the clustering process is illustrated by a join of the tree. The algorithm requires a parameter specifying the number of clusters and a similarity criterion. In order to define similarity of proteins in the PPI network, we use the well-known Czekanowski-Dice distance metric [6]. This metric is ideal for this domain, since it increases the weight of shared interacting proteins, and two proteins having no common interactors will have the maximum distance value, while those interacting with exactly the same set of proteins will have zero value. Our similarity metric is defined as:

$$Sim(i, j) = 1 - \frac{|Int(i) \Delta Int(j)|}{|Int(i) \cup Int(j)| + |Int(i) \cap Int(j)|} \quad (4)$$

Here, $Int(i)$ and $Int(j)$ denote the set of interactors (including themselves) of proteins i and j , respectively, and Δ represents the symmetric difference between the sets. The value of this metric ranges from 0 to 1. To cluster the graph, we use a fast implementation of Agglomerative clustering from CLUTO [1], a clustering toolkit.

2.3.2 Multilevel k-way Graph Partitioning (kMetis)

Metis is a family of algorithms developed to partition graphs (and hypergraphs) [2]. The fundamental multilevel paradigm of Metis algorithms produces balanced and high quality partitions in a scalable manner. We have chosen

kMetis since it attempts to find balanced partitions and has been found to be very efficient. The multilevel partitioning algorithms have three major phases: coarsening, initial partitioning and refinement. In the coarsening phase, the original graph is transformed into a sequence of smaller graphs. An initial two-way partitioning of the coarsest graph that satisfies the balancing constraints while minimizing the cut value is obtained in the next phase. During the uncoarsening and refinement phase, the partitioning is projected back to the original graph by going through intermediate partitions. After projecting a partition, a partition refinement algorithm is employed to reduce the edge-cut while conserving the balance constraints.

We varied the number of clusters for both the above algorithms and picked the values that gave the best balance in terms of size of clusters. We discovered that for smaller values of k (number of clusters), the Hierarchical algorithm provided an imbalanced cluster arrangement. However at $k=500$ and 700 , the clusters obtained were balanced and suitable for analysis. In the case of Metis, since the algorithm is designed to obtain balanced clusters, a high value of k (500) results in extremely small sized clusters. Hence we picked the number of clusters as 120 for the experiments in the case of Metis.

2.4 Validation

To test the hypothesis that the final clusters obtained from the pre-processed data correspond to functional modules, we need to validate our clusters using biological information. We do this using gene annotations from the Gene Ontology Consortium Online database. The Gene Ontology (GO) is an important tool designed to support the work of researchers in the area of genomics and biomedicine by providing a common terminology to report the results obtained. GO consists of three terminologies comprising of biological process, molecular function and cellular component terms. The cellular components refer to entities included within a single cell. This provides anatomical and structure association information. The molecular function terms refer to shared activities at the molecular level. The biological process terms refer to entities at both the cellular and organism levels of granularity. Each of them provides valuable information in terms of biological significance of protein associations in the organism. As of May 2005, GO contains 7000 genes annotated in 1644 cellular component terms, 7502 molecular function terms and 9706 biological process terms.

For each cluster of proteins we obtain from our experiments, we query the GO-TermFinder tool [3], for the p-values in each of the three categories - biological process, molecular functions and cellular component. The tool examines a given group of proteins to find ontology terms to

which a large percentage of the given proteins are associated, compared to the actual percentage of the proteins in the database that are associated with these terms. The p-value thus represents the chance of obtaining a particular result or better, given a background distribution. Lower p-values represent extremely significant results. The p-value calculations are performed using the Hypergeometric Distribution, defined below.

If there are ' N ' proteins in the database with ' M ' of them assumed to share a particular property and our cluster contains ' n ' proteins out of which ' x ' have the same annotation. Then using the Hypergeometric Distribution, we get

$$pvalue = \frac{\binom{M}{x} \binom{N-M}{n-x}}{\binom{N}{n}} \quad (5)$$

where $\binom{n}{r}$ represents the number of combinations by which ' r ' entities can be selected from ' n ' entities. The tool also uses a cut-off (alpha-level) value to reduce the number of results obtained and displays only the significant terms. If a term has p-value smaller than the cut-off, it is considered to be significant and is displayed. We have used the recommended cutoff of 0.05 for all our validations. To assign a total p-value score for the clustering scheme, we define an average p-value notation.

$$avg_pvalue = \frac{\sum_{i=1}^n \min(pvalue_i)}{n} \quad (6)$$

where ' n ' denotes the number of partitions with significant p-values (smaller than the cut-off) and $\min(pvalue_i)$ denotes the smallest p-value of the partition ' i '. We calculate the average separately for each of the three ontologies. Apart from the p-value, we also calculate the variance of the p-values over the clusters and the number of clusters which have p-value higher than the threshold of 0.05 . The variance provides us information about the distribution of p-values while the latter quantifies the balance of the clusters obtained. Lower values for all these metrics signify high biological relevance.

3 Experiments

The first experiment that we perform is to highlight the effectiveness of our validation scheme. The subsequent experiments examine the effectiveness of our proposed pre-processing techniques.

3.1 Effectiveness of the Validation Metric

We use the p-value obtained from the GO annotations to validate our clusters. To test this method of validation, we use clusters obtained from the two clustering algorithms as well as clusters obtained by randomly partitioning the dataset. In order to make a fair comparison, we obtain random clusters having the same cluster-size distribution as the ones from the two algorithms.

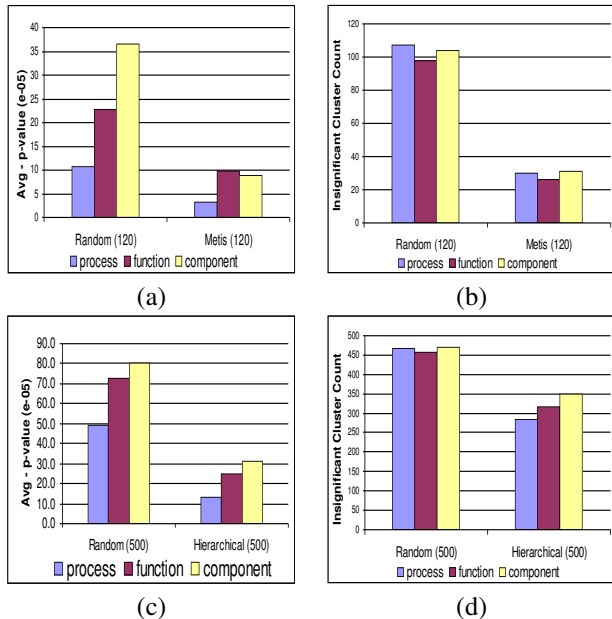


Figure 1. Clustering Algorithms vs. Random Clustering (a) Metis P-Value (b) Metis Insignificant Clusters (c) Hierarchical P-Value (d) Hierarchical Insignificant Clusters

Since this is not a test of our pre-processing method, we use the original PPI dataset. We expect the two clustering methods to yield better clusters than a random assignment of proteins to clusters. Hence a lower p-value on the clusters obtained from the two algorithms would indicate that our validation scheme is proficient in identifying better clusters. We use two specific criteria for evaluation - the average of the p-values of the clusters obtained and the number of insignificant clusters. Since we expect the clustering algorithms to work better than random assignments, smaller average p-values for the clusters from the two algorithms would suggest that our metric is capturing this difference. Figures 1a and 1b, present the comparison between Metis and random clustering. Figures 1c and 1d, present the comparison between Hierarchical and random clustering. From the results, we observe that the average p-values of Hierarchical and Metis clustering schemes are significantly smaller than the average p-values of randomized clustering. For the number of insignificant clusters, we see that both the clustering algorithms identify much more biologically relevant clusters than the random clustering technique. The results show that our validation metrics are appropriate, since they produce results that are consistent with our initial expectation (random clustering is expected to yield worse results).

3.2 Pre-processing with Clustering Coefficient

In this experiment, we examine the effect of using the Clustering Coefficient for pre-processing. We used the Clustering Coefficient on the line graph representation to pre-process the original dataset. We eliminated

30%,40%,50% and 60% of the edges this way. We then ran the two clustering algorithms on the original network and the pre-processed network separately. Figures 2 and 3 present the experimental results for two clustering algorithms respectively. To validate, we use the p-value, variance of p-values and the number of insignificant clusters obtained. The p-value score is the ratio between the p-values on clusters obtained on the preprocessed data and clusters obtained on the original data. The variance and insignificant scores are similarly defined. As before, smaller score values indicate high improvement achieved due to pre-processing. The variance of the p-values reflects the fluctuation of quality across the identified clusters. Smaller variance values indicate more stable clusters. From the data we see clearly the benefit due to the Clustering Coefficient pre-processing step. In most cases, both algorithms return better results on the pre-processed data, in terms of all the three metrics. Hierarchical works well even when the number of clusters is increased. This can be shown by the fact that most points are below the line where the scale value is 1. We compared the cluster size distribution of Metis with and without pre-processing. From Figures 4a-c we can observe that, the pre-processing reduces the variance in cluster size and also reduces the number of small-sized clusters.

3.3 Pre-processing with Centrality

We use a Centrality threshold to remove nodes from the graph. 15% of the nodes have Centrality scores over the threshold and are removed. The results of the two clustering algorithms are evaluated after pre-processing the data using Betweenness and Closeness Centrality respectively. Figures 5a-c depicts the results in terms of p-value, variance and insignificance scores for Betweenness Centrality. Similarly, results for Closeness Centrality are provided in Figures 6a-c. The score values are also significantly smaller than in the Clustering Coefficient experiments. This suggests that the Closeness Centrality might be a better metric to use. From the results in both the above cases, it is evident that the pre-processed data provides a significant improvement over the original dataset.

3.4 Comparison with Random Elimination

Here, we perform an experiment to highlight the fact that the improvement obtained from the pre-processing is due to its capability to eliminate possible false positive interactions, as opposed to merely reducing the number of the interactions. To do this, we eliminate interactions randomly from the original network. We then apply the Hierarchical clustering algorithm (with $k=500$) on the randomly eliminated data. We use the dataset pre-processed using Clustering Coefficients with 30% and 40% interactions eliminated. In order to achieve a fair measure of randomness, we take the average of the values of the clustering metrics

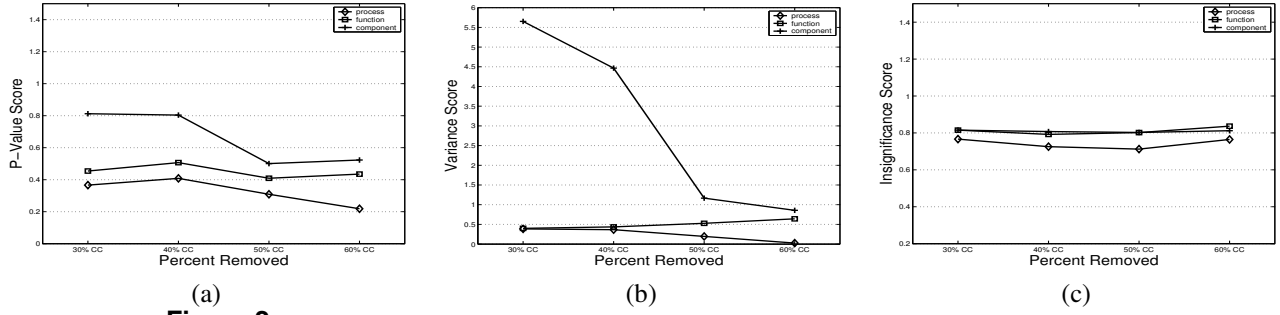


Figure 2. Hierarchical Algorithm(k=700) - Clustering Coefficient Pre-processing (a)P-Value (b)Variance (c) Insignificance

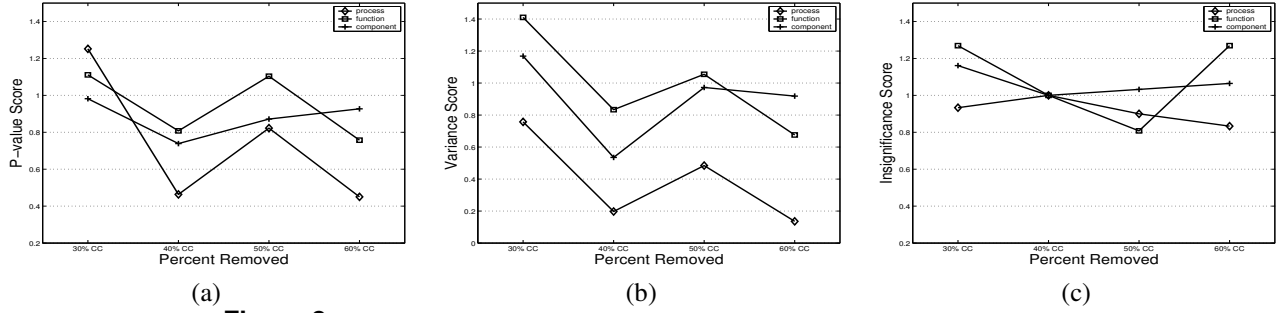


Figure 3. Metis Algorithm - Clustering Coefficient Pre-processing (a)P-Value (b)Variance (c) Insignificance

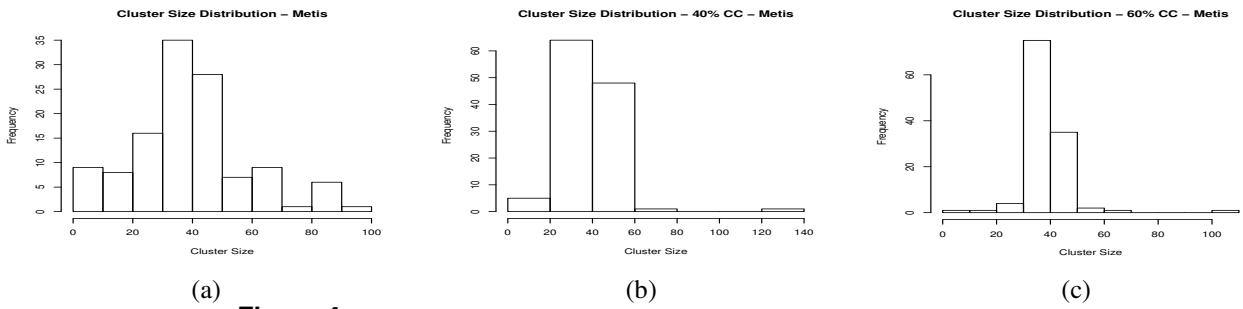


Figure 4. Metis Cluster Size Distribution (a)Original Data (b)40% Pre-processed (c)60% Pre-processed

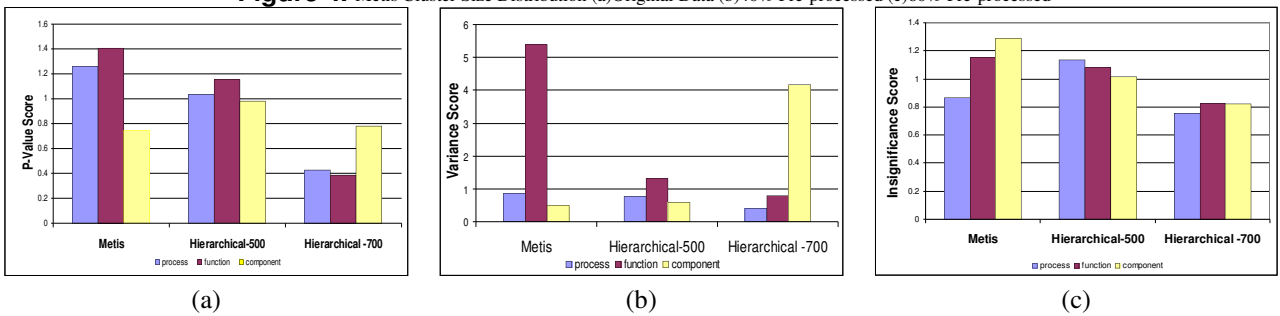


Figure 5. Betweenness Pre-processing: (a)P-Value (b)Variance (c) Insignificance

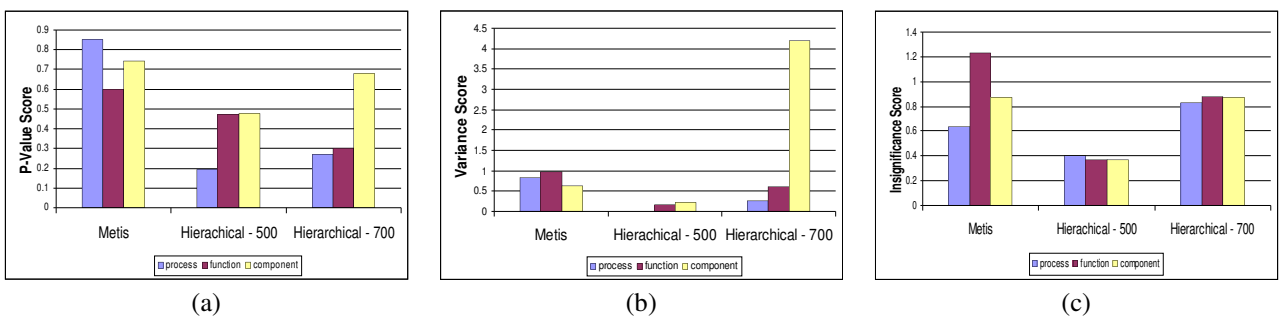


Figure 6. Closeness Centrality Pre-Processing (a)P-Value (b)Variance (c) Insignificance

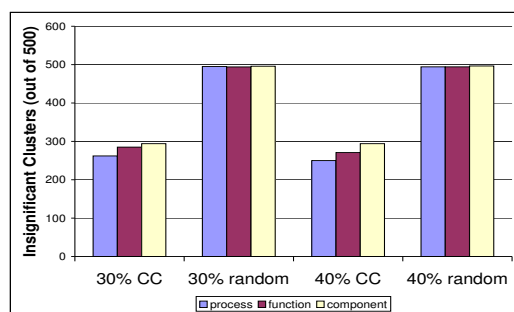


Figure 7. Clustering Coefficient Preprocessing vs. Random Elimination

over 5 runs for the random scheme. Figure 7 presents the experimental results comparing the performance of the Hierarchical algorithm on the pre-processing datasets and the randomly eliminated dataset. We choose the Hierarchical algorithm for this experiment. To evaluate the two strategies, we use the number of insignificant clusters since it provides an indication of how balanced the cluster assignments are. The benefit of our proposed methods over the random scheme can be clearly seen from the data. The Hierarchical clustering algorithm works much worse on the data with randomly eliminating interactions than on the pre-processed data. The data with randomly eliminated interactions results in a skewed clustering arrangement with a large cluster of size 4000 and a few small clusters.

4 Discussion

Our results clearly showed that we are able to quantify the biological meaning of a cluster in terms of three different ontologies defined by the GO Consortium. We believe that the proposed validation methodology will be helpful in order to interpret the results of a clustering algorithm applied on the PPI networks. Moreover, this validation method might also be effective in deciding which final groups to focus on when mining for novel biological findings. An example of a high scoring partition of the dataset, cleaned using Closeness Centrality has the smallest p-value of $8.93e-37$ for biological process. This partition is further analyzed to indicate how informative our partitioning can be. 19 proteins (UTP15, DIP2, IMP4, PWP2, UTP8, UTP4, NAN1, EMG1, RRP9, UTP10, UTP7, MPP10, UTP6, ENP1, NOP14, UTP9, NOP58, UTP13, IMP3) out of 41 in this cluster are annotated with term 'Processing of 20S Pre-rRNA' (GO:0030490), whereas there exist only 32 proteins associated with this term in the whole genome of 7000 proteins. Similarly, in the same group, 17 proteins (UTP10, UTP7, UTP15, DIP2, MPP10, IMP4, PWP2, UTP8, UTP6, UTP4, NAN1, NOP14, UTP9, NOP58, IMP3, UTP13,

| Go-Term | Cluster Freq | Genome Freq | p-value |
|---|--------------|-------------|-----------|
| P - mRNA splicing | 48 of 69 | 80 of 7000 | 2.13e-84 |
| P - mRNA splicing | 37 of 49 | 80 of 7000 | 6.02e-66 |
| P - proteolysis and peptidolysis | 27 of 31 | 113 of 7000 | 4.42e-46 |
| P - vacuolar acidification | 14 of 33 | 19 of 7000 | 1.22e-30 |
| P - processing of 20S pre-rRNA | 19 of 41 | 32 of 7000 | 8.93e-37 |
| F - pre-mRNA splicing factor activity | 29 of 69 | 45 of 7000 | 4.08e-50 |
| F - pre-mRNA splicing factor activity | 22 of 49 | 45 of 7000 | 5.58e-38 |
| F - proteasome endopeptidase activity | 26 of 31 | 34 of 7000 | 1.38e-61 |
| F - structural constituent of ribosome | 32 of 50 | 226 of 7000 | 2.29e-36 |
| F - snoRNA binding | 15 of 41 | 23 of 7000 | 8.45e-30 |
| F - RNA polymerase II transcription mediator activity | 17 of 42 | 20 of 7000 | 4.48e-37 |
| C - small nuclear ribonucleoprotein complex | 32 of 38 | 36 of 7000 | 7.08e-67 |
| C - small nuclear ribonucleoprotein complex | 38 of 49 | 65 of 7000 | 1.80e-73 |
| C - proteasome complex | 28 of 31 | 36 of 7000 | 9.48e-68 |
| C - organellar large ribosomal subunit | 29 of 50 | 47 of 7000 | 8.49e-55 |
| C - hydrogen-translocating V-type ATPase complex | 12 of 33 | 16 of 7000 | 2.23 e-26 |
| C - small nuclear ribonucleoprotein complex | 17 of 41 | 30 of 7000 | 2.71e-32 |

Table 1. The first column represents the ontology annotated with the specified cluster. PF and C stands for biological process, molecular function and cellular component respectively. GO-Term refers to the biological association for the proteins in each cluster. The Cluster Frequency represents the ratio of proteins annotated with the specified ontology term in the given cluster whereas the Genome Frequency column represents the ratio for the whole genome.

RRP9) are associated with 'Small Nucleolar Ribonucleoprotein Complex' (GO:0005732) whereas only 30 proteins are associated with this cellular component in the whole genome. Based on the very small p-value for the cluster in the annotated process ('Processing of 20S Pre-rRNA') we can hypothesize that all the proteins may be taking part in it. The proteins which are not annotated with this process might have an undetected functionality or interaction.

We are able to obtain clusters with extremely small p-values with hierarchical clustering. As an example, after Closeness Centrality reduction, we applied hierarchical clustering with the number of clusters as 500. One of the resulting clusters has a p-value score of $2.13e-84$ for biological process. 48 of the 69 proteins in this cluster are annotated with 'mRNA Splicing'. This process is only associated with 80 proteins in the whole genome. The same cluster has p-value scores $7.08e-67$, $4.08e-50$ for component and function respectively. Clearly, our preprocessing method, improves the biological value of the final clusters. We are able to obtain clusters that are enriched in proteins with the same function, process and component.

A cluster obtained by the Hierarchical algorithm was composed of the following proteins - PRE5, PRE1, UBP6, RPN8, RPT4, PRE9, ECM29, PRE7, RPT2, RPN9, RPT3, YGL004C, PRE6, RPT1, PRE4, PUP3, NAS6, RPN7, RPN10, RPN12, RPN11, RPN6, SCL1, PRE2, RPT5, RPN3, RPN4, RPN5, RPN13, PRE3, PHO4, RPT6. 28 of these proteins are annotated with the biological process 'Proteolysis and Peptidolysis'. This suggests that the remaining three proteins might have an unrevealed task in

the same process. Also, 27 of them are annotated with the molecular functionality 'Proteasome Endopeptidase Activity' whereas only 34 proteins in the whole genome have this molecular function annotation. This cluster is also valuable in terms of cellular component. It includes almost all the proteins in the proteasome complex. Our experimental results indicated that the pre-processing methods are more effective with Hierarchical algorithm. We provide details of some of the clusters we obtained in Table 1. The interactions between the proteins of the above two clusters (obtained from Hierarchical Clustering) are depicted in Figure 8.

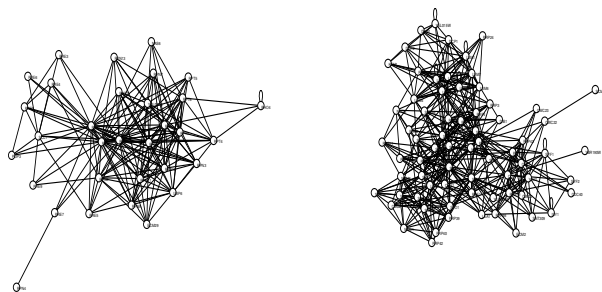


Figure 8.^a Example clusters

5 Conclusions and Future Work

In this work, we have proposed novel pre-processing strategies to eliminate redundant and potentially false interactions and facilitate the extraction of biologically relevant clusters from PPI datasets. We have demonstrated the effectiveness of this technique from our detailed experiments. Our results indicate clearly that our pre-processing strategies improve the quality of clusters obtained from standard clustering algorithms. Our comparative results for the two algorithms indicate that our strategies provide improvements regardless of the clustering algorithm applied.

In the future, we would like to extend this work to functional identification of proteins. Several proteins in genome databanks have not yet been characterized experimentally and their functionality is largely unknown. We believe that our pre-processing strategies will help in focusing on interactions that are highly reliable and can therefore lead to more information about unknown proteins. In this paper, we have considered only hard clustering. In the future we would like to test our pre-processing technique on soft clustering algorithms. We believe the improvements would be significantly better.

6 Acknowledgements

This work is supported in part by the following research grants : DOE Award No. DE-FG02-04ER25611; NSF CAREER Grant IIS-0347662;

References

- [1] <http://www-users.cs.umn.edu/karypis/cluto/index.html>.
- [2] <http://www-users.cs.umn.edu/karypis/metis/metis/files/manual.pdf>.
- [3] <http://db.yeastgenome.org/cgi-bin/GO/goTermFinder>.
- [4] <http://www.fas.at>.
- [5] 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.
- [6] C. Brun, C. Herrmann, and A. Guenoche. Clustering proteins from interaction networks for the prediction of cellular functions. *BMC Bioinformatics*, 5(95), July 2004.
- [7] C. Deane, L. Salwinski, I. Xenarios, and D. Eisenberg. Two methods for assessment of the reliability of high throughput observations. *Molecular and Cellular Proteomics*, 1(5):349–356, May 2002.
- [8] S. Fields and O. Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340(6230):245–246, July 1989.
- [9] L. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1:215–239, 1979.
- [10] L. Freeman. Centered graphs and the construction of ego networks. *Mathematical Social Sciences*, 3, 1982.
- [11] J. Hua, D. Koes, and Z. Kou. Finding motifs in protein-protein interaction networks. *Project Final Report, CMU*, 2003.
- [12] J. Nacher, T. Yamada, S. Goto, M. Kanehisa, and T. Akutsu. Two complementary representations of a scale-free network. *Physica A*, 349:349–363, 2005.
- [13] J. Pereira-Leal, A. Enright, and C. Ouzounis. Detection of functional modules from protein interaction networks. *Proteins*, 54(1):49–57, January 2004.
- [14] R. Saito and *et al.* Interaction generality, a measurement to assess the reliability of a protein-protein interaction. *Nucleic Acids Research*, 30(5):1163–1168, March 2002.
- [15] S. Skiena. Line graph 4.1.5 in implementing discrete mathematics: Combinatorics and graph theory with mathematica. *Reading, MA: Addison-Wesley*, pp. 128 and 135-139, 1990.
- [16] A. Thomas, R. Cannings, N. Monk, and C. Cannings. On the structure of protein-protein interaction networks. *Biochemical Society Transactions*, 31:1491–1496, 2003.
- [17] D. Watts and S. Strogatz. Collective dynamics of small world networks. *Nature*, 393(6684):440–442, June 1998.
- [18] L. Wu, T. Hughes, A. Davierwala, M. Robinson, R. Stoughton, and S. Altschuler. Large-scale prediction of *saccharomyces cerevisiae* gene function using overlapping transcriptional clusters. *Nature Genetics*, 31:255–265, June 2002.
- [19] I. Xenarios, D. Rice, L. Salwinski, M. Baron, E. Marcotte, and D. Eisenberg. Dip: the database of interacting proteins. *Nucl. Acids Res.*, 28(1):289–291, 2000.