An Ensemble Framework for Clustering Protein-Protein Interaction Networks

Sitaram Asur, Srinivasan Parthasarathy[*]and Duygu Ucar
Department of Computer Science
The Ohio State University
Contact:srini@cse.ohio-state.edu

Abstract

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 these networks has been theorized by many researchers. However, the application of traditional clustering algorithms for extracting these modules has not been successful, largely due to the presence of noisy false positive interactions as well as specific topological challenges in the network. In this paper, we propose an ensemble clustering framework to address this problem. For base clustering, we introduce two topology-based distance metrics to counteract the effects of noise. We develop a PCA-based consensus clustering technique, designed to reduce the dimensionality of the consensus problem and yield informative clusters. We also develop a soft consensus clustering variant to assign multifaceted proteins to multiple functional groups. We conduct an empirical evaluation of different consensus techniques using topology-based, information theoretic and domain-specific validation metrics and show that our approaches can provide significant benefits over other state-of-the-art approaches. Our analysis of the consensus clusters obtained demonstrates that ensemble clustering can a) produce improved biologically significant functional groupings; and b) facilitate soft clustering by discovering multiple functional associations for proteins.

1. Introduction

Proteins are central components of cell machinery and life. In fact, as noted by Kahn [21], *it is the proteins dynamically generated by a cell that execute the genetic program.* Mering *et. al.* [36] note that, to fully understand cell machinery, simply listing, identifying and determining the functions of proteins in isolation is not enough – (clusters of) interactions need to be delineated as well, since proteins work with other proteins to regulate and support each other for specific functions. Recent advances in technology have enabled scientists to determine, identify and validate pair-wise protein interactions through a range of experimental and in-silico methods [13, 14, 26, 35]. Such data can be naturally represented in the form of interaction networks. The task of extracting relevant groupings or functional modules from such interaction networks, for the purposes of understanding the behavior of organisms, protein function prediction and drug design is challenging and an active area of research [7, 19, 38, 39, 34]. The challenges involved are manifold.

First, is the issue of data quality. Different experimental and in-silico methods can be used to compute interactions, each with its own strengths and weaknesses [13, 14, 26, 35]. Often, the overlap, in terms of common interactions across experimental settings, is not very high. An added complexity is that the data obtained from such methods is believed to be quite noisy - many interactions are conjectured to be false positives. Integrating data from such sources yields interaction networks that are inherently noisy [4]. To address this problem, various researchers have examined data preprocessing techniques to identify and eliminate potential false positives (and to identify potential false negatives) by examining the topological characteristics of such networks [34, 9, 28].

Second, even if the network is assumed to be noise free, partitioning the network using classical graph partitioning or clustering schemes is inherently difficult. A common characteristic of PPI networks is that, a few nodes (hubs) have very large degrees, while most other nodes have very few interactions. Applying traditional clustering approaches typically results in a clustering arrangement that is quite poor – containing one or a few giant core clusters and several tiny clusters (possibly singleton clusters). To address this problem, researchers have relied on various refinements that take into account domain expertise and topological information (e.g. targeting scale-free networks) to constrain the clustering process resulting in an improved clustering arrangement [29, 16].

Third, some proteins are believed to be multi-functional – effective strategies for soft clustering of these essential proteins are needed. This dictates the need to leverage or adapt *soft* cluster-

[*]This work is supported in part by the DOE Early Career Principal Investigator Award No. DE-FG02-04ER25611 and NSF CAREER Grant IIS-0347662.

ing approaches. To address this problem, recent research has examined strategies such as hub duplication [33] and partitioning the line-graph transform of the original PPI network. The former ensures the soft clustering of hub proteins, that are believed to be multi-functional [20], while the latter targets the clustering of edges in the original graph (nodes in the transformed graph) to dictate the eventual set of protein clusters [25].

In this work we examine an alternative approach, ensemble clustering, to resolve these three problems simultaneously. Ensemble clustering has been proposed in the literature as a useful approach to strengthen the quality of simple clustering algorithms [15, 17, 32, 31]. The goal is to combine multiple, diverse and independent clustering arrangements to obtain a single, comprehensive *consensus* clustering. Empirical evidence has suggested that intelligent combination of these clusters can lead to novel and meaningful cluster structures, even in the presence of noise [32]. We should also note that, one can weight individual clustering arrangements according to their strengths and weaknesses, potentially addressing the fusion problem[1].

However, naively applying ensemble clustering to the problem at hand will not work. There are certain key questions that need to be resolved. First, what are the base clustering methods to use for processing PPI networks? An appealing option here is to leverage domain and topological information to identify good candidate base clustering methods. Second, clustering ensembles typically do not scale very well – building a consensus is expensive and is affected by the dimensionality of the problem on hand. An attractive option here is to investigate the use of traditional dimensionality reduction options to improve the scalability of the consensus building step. Third, are there ways in which one can make the ensemble clustering more robust to noise effects? For example, by developing suitable pruning or weighting strategies. Fourth, the existing literature on ensemble clustering algorithms is limited to hard clustering problems – can one adapt such approaches for soft clustering? Faced with these challenges our contributions are:

- We have designed and evaluated the use of *two topology-driven distance metrics* for network clustering. We use three traditional graph partitioning algorithms with the two metrics to obtain six base clusterings that are *diverse and yet informative about the topological properties* of nodes in the network.
- We have designed and evaluated a consensus method that relies on *Principal Component Analysis (PCA)* to reduce the dimensionality of the consensus determination problem. The ensemble solution on the reduced dimensional space can then be efficiently computed using traditional consensus methods.
- We have also developed a topology-driven strategy for pruning *weak base clusters* that significantly improves the quality of the resulting ensemble cluster arrangement.
- We have designed an adaptation to the above approach that allows for *soft ensemble clustering of proteins* in interaction networks. This enables our method to model and account for multi-faceted proteins.
- We conduct a detailed empirical evaluation and comparison of our approaches with other state-of-the-art algorithms on the PPI network of budding yeast (*Saccharomyces Cerivisiae*). We use topological, information theoretic and domain-specific cluster validation metrics to evaluate and modulate the improvements gained from each component of the proposed ensemble clustering methodology.

Our experimental results show that our algorithms can provide significant improvement in cluster quality across the board (not just the top clusters), when compared to previously reported methods. We also show that ensemble clustering can effectively facilitate the discovery of multiple functional associations for proteins.

2. Related Work

Many clustering algorithms of various types have been applied to analyze PPI networks. Bader [5] proposed the three-stage Molecular Complex Detection (MCODE) algorithm to identify densely connected regions from a PPI graph. First, each vertex of the graph is associated with a weight based on the local neighborhood density of that vertex. Second, clusters are created around the top-weighted vertices (seed vertices) by iteratively adding high-scoring vertices to the cluster. Finally, clusters that are not dense enough are eliminated from the final set of partitions.

The MCL algorithm (Markov Clustering) [12], proposed by Dongen is a fast and scalable unsupervised clustering algorithm for graphs, based on the simulation of stochastic flow in graphs. The algorithm simulates random walks within a graph by alternation of two operators called expansion and inflation. Eventually, the iteration results in the separation of the graph into different segments (clusters). A recent study [6] compared four clustering algorithms, - Markov CLustering (MCL), Restricted Neighborhood Search Clustering (RNSC), Super Paramagnetic Clustering (SPC), and Molecular Complex Detection (MCODE) , on six protein-protein interaction networks to identify protein complexes. The clusters obtained from the algorithms were compared with known annotated complexes. Their conclusion was that Markov Clustering (MCL) algorithm far outperformed the other algorithms in the extraction of complexes from interaction networks.

The ensemble clustering problem has been studied previously in the machine learning community by many researchers, although it has been applied mainly to small classification datasets thus far. Fred *et al* [15] map clusterings produced by multiple runs of the k-means algorithm with different initializations into a co-association matrix. They then apply a hierarchical single-link algorithm to partition this matrix into the final consensus clusters. In a later work, Topchy *et al* [32] also present two approaches to prove the effectiveness of a cluster ensemble - using plurality voting and using a metric on the space of partitions.

[1] This aspect is not considered in this paper but we believe the approach is naturally amenable to fusing information from multiple experimental and in-silico interaction networks inculcating domain bias.

Gionis *et al* [17] provide a formal definition to the problem of cluster aggregation and discuss a few consensus algorithms with theoretical guarantees. The algorithms they propose use the distance matrix representation and are suitable mainly for small datasets. The Agglomerative algorithm proposed by Gionis *et al* merges clusters that have distances less than 1/2, which is a hard-coded threshold. If a point has distance greater than half with all other clusters, it is placed in a cluster by itself. The Balls algorithm tries to find ball-shaped clusters, grouping together proteins that are close to each other and far from other nodes. Both these algorithms have been evaluated only on small categorical datasets. They have not been evaluated on large graph datasets. We use these two algorithms for comparison with our techniques.

Strehl and Ghosh [31] define the cluster ensemble problem as an optimization problem and aim to maximize the normalized mutual information of the consensus clustering from the initial clusters obtained from ten base clustering algorithms. They use a hypergraph representation with an $n \times m$ matrix, where n is the number of points and m is the total number of clusters in all the clusterings. They introduce three different algorithms to obtain consensus clusterings, namely Cluster-based Similarity Partitioning (CSPA), HyperGraph Partitioning (HGPA), and Meta-Clustering (MCLA) algorithms. In CSPA, they construct a similarity matrix from the clusters obtained from the base clustering algorithms. This similarity matrix is treated as a weighted graph and partitioned using the Metis [22] algorithm to obtain the consensus clustering. In HGPA, the goal is to find a hyperedge separator that partitions the hypergraph into k unconnected components by cutting a minimal number of hyperedges. The HMetis algorithm is used for this purpose. In MCLA, the main idea is to group related hyperedges (base clusters) to obtain meta-clusters. A representative cluster is obtained for each meta-cluster. Finally, each data point is compared with the representative clusters and assigned to the meta-cluster it is most associated with. We use these three ensemble consensus techniques in our evaluation.

3. Algorithms

In this section, we describe our topological similarity metrics, base clustering algorithms and consensus methods in detail.

3.1 Similarity metrics

We introduce two different similarity metrics designed to capture diverse topological properties of PPI networks. Our goal is to weight edges of the PPI network to reflect the reliability of the corresponding interactions. Accordingly, edges with low values of weights will indicate potential false positive (noisy) interactions. Clustering algorithms can then use these weights to eliminate noisy edges and yield meaningful partitions. To assign suitable weights, we focus on two different topological features - Clustering Coefficient and Edge Betweenness.

3.1.1 *Clustering coefficient-based*

The first similarity metric is based on the Clustering coefficient, a popular metric from graph theory. The clustering coefficient [37] is a measure that represents the interconnectivity of a vertex's neighbors. The clustering coefficient of a vertex v with degree k_v can be defined as follows:

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

where n_v denotes the number of triangles that go through node v.

Essentially, if the edge between two nodes contributes significantly to the clustering coefficients of the nodes, then the nodes are considered similar and should be clustered together. To calculate the similarity of nodes v_i and v_j, we first calculate their clustering coefficients as CC_{v_i} and CC_{v_j}. We then remove the interaction(edge) between these nodes and re-calculate the clustering coefficient of each node as $CC^{'}_{v_i}$ and $CC^{'}_{v_j}$. The difference between these two values represent the importance of the edge for each node. Accordingly, the Clustering coefficient-based similarity of two nodes is then calculated as follows:

$$S_{cc}(v_i, v_j) = CC_{v_i} + CC_{v_j} - CC^{'}_{v_i} - CC^{'}_{v_j}$$

Note that if two nodes are not linked in the original network, their Clustering coefficient-based similarity score is zero. The similarity scores are normalized into the range [0-1] using min-max normalization.

3.1.2 *Betweenness-based*

The second metric is based on the Shortest-path Edge betweenness measure, which was first introduced by Newman *et al* [23]. It is a popular measure for clustering networks in sociology and ecology to obtain communities. This measure favors edges between communities and disfavors ones within communities. The Shortest-path betweenness measure computes, for each edge in the graph, the fraction of shortest paths that pass through it. To take advantage of the global information that is captured by the edge-betweenness measure [24], we use it as a similarity metric, as follows.

$$S_{eb}(v_i, v_j) = 1 - \frac{SP_{ij}}{SP_{max}}$$

where SP_{ij} is the number of shortest paths passing through edge ij and SP_{max} is the maximum number of shortest paths passing through an edge in the graph. Similar to the previous metric, this metric is defined only for connected pairs and rescaled into the range [0-1] using min-max normalization.

3.2 Base algorithms

We use three conventional graph clustering algorithms to obtain the base clusters.

3.2.1 *Repeated bisections (rbr):*

The Repeated bisections algorithm is a top-down clustering algorithm that computes the desired k-way clustering solution, by performing a sequence of $k - 1$ repeated bisections, where k is the required number of clusters. The input matrix is first clustered into two groups, after which one of the groups is selected and bisected further. This process continues until the desired number of clusters is found. During each step, a cluster is bisected so that the resulting 2-way clustering solution optimizes the I2 clustering criterion function, which is given as:

$$I2 = maximize \sum_{i=1}^{k} \sqrt{\sum_{v,u \in S_i} sim(v,u)} \qquad (1)$$

where k is the total number of clusters, S_i is the set of objects assigned to the i^{th} cluster, v and u represent two objects, and $sim(v,u)$ is the similarity between two objects.

3.2.2 *Direct k-way partitioning (direct):*

In this method, the desired k-way clustering solution is computed by simultaneously finding all k clusters. Initially, a set of k objects is selected from the data sets to act as the seeds of the k clusters. Then, for each object, its similarity to these k seeds is computed, and it is assigned to the cluster corresponding to its most similar seed. This initial clustering is then repeatedly refined to optimize the I2 clustering criterion function.

3.2.3 *Multilevel k-way Partitioning (Metis):*

Metis (kMetis) is a popular multilevel partitioning algorithm, developed by Karypis *et al* [22]. It works in three phases: coarsening, initial partitioning and refinement. In the coarsening phase, the original graph is transformed into a sequence of smaller graphs. An initial k-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.

3.3 Consensus Methods

Using the base algorithms with the two topological metrics we discussed in the first subsection, we obtain six sets of k clusters. Our goal is to combine these individual clusterings to obtain a meaningful consensus clustering. Given n individual clusterings $(c_1..c_n)$, each having k clusters, a consensus function F is a mapping from the set of clusterings to a single, aggregated clustering:

$$F : \{c_i | i \epsilon 1, .., n\} \rightarrow c_{consensus}$$

Ideally, the consensus clustering needs to be representative of the individual component clusterings.

3.3.1 *PCA-based Consensus*

The consensus technique we propose consists of three stages - Cluster Purification, Dimensionality Reduction and Consensus clustering.

Cluster Purification:

It has been well documented that different clustering algorithms typically yield diverse clusterings [27, 32]. This is due to the different criteria and similarity metrics employed for clustering. Hence, it is likely that some of the clusters obtained are less consistent with the topology of the original graph than others. We believe that such clusters contribute to noise and distort the consensus function. To find these clusters, we once again rely on a topological measure. We define a reliability measure for each cluster, that is based on the topology of the proteins in the cluster. The shortest path distance between two proteins i and j is the minimum number of interactions in the original graph that separate them. For each cluster, we compute the intra-cluster distance as the average shortest path distance between all pairs of proteins in that cluster.

$$ClusterDistance(cl_1) = \frac{\sum_{(i,j) \in V_{cl_1}} SP(i,j)}{|V_{cl_1}| * Diam_G} \qquad (2)$$

where V_{cl_1} represents the nodes in cluster cl_1 and $SP(i,j)$ represents the shortest path distance in terms of number of edges between nodes i and j. $Diam_G$ signifies the diameter of the original PPI graph and is used for normalization. The reliability of a cluster is inversely proportional to its intra-cluster distance.

$$Rel(cl_1) = \frac{1}{ClusterDistance(cl_1)} \qquad (3)$$

If the distance between nodes in a cluster is high, it indicates that the cluster is not very modular. Hence, we use a threshold value to prune away weak clusters. We choose a threshold value ensuring that each protein is represented in at least $(1/3)^{rd}$ of the reliable subset of clusters.

Dimensionality Reduction:

We then represent the remaining clusters in a binary format with an $n \times m$ matrix, where m is the total number of clusters obtained using all base algorithms. Each row represents a point while each column corresponds to a cluster. The value I(x,y) in the matrix represents the indicator function of point x wrt cluster cl_y.

$$I(x, cl_y) = \begin{cases} 1, & \text{if } x \in cl_y \\ 0, & \text{otherwise} \end{cases}$$

Even after pruning clusters, it is likely that the number of dimensions (remaining clusters) is too large for the direct application of clustering algorithms. For instance, in our case, we have six algorithm-metric combinations each producing k clusters after pruning. If the value of k is large, clustering the $6 \times k$-dimensional points would prove inefficient, since distance metric computations that are integral to clustering, do not scale well to high dimensions [1].

To obtain a more scalable and efficient representation for clustering, we use the technique of Principal Component Analysis (PCA). The idea is to reduce the number of dimensions of the matrix without compromising the information required for clustering. As we described above, each feature vector (row) in the matrix corresponds to the cluster membership pattern of a node. Since we are using hard clustering algorithms, a node can occur only in 6 clusters. For large values of k, the binary feature vectors will be very sparse. Also, since the occurrence of a node in a cluster is not independent of other clusters in a clustering, there is bound to be a lot of redundancy in the feature vectors. Several researchers [18, 11, 30] have suggested the application of dimensionality reduction techniques (such as PCA) as a pre-processing step to clustering sparse high-dimensional data. PCA uses the eigen decomposition of the correlation matrix to find orthogonal directions with total maximum variance of projections. In our case, it can use the correlations between the cluster membership patterns of nodes to eliminate redundancies reducing the matrix to a more compact representation, retaining only discriminatory information. Accordingly, we convert the $6 \times k$ clusters into a matrix and apply PCA to reduce the number of dimensions. Traditional clustering algorithms can then be applied on this reduced representation without performance concerns, to obtain consensus clustering arrangements.

Consensus Clustering:
To perform consensus clustering, we apply two different consensus clustering algorithms on the PCA representation - the *Recursive Bisection (PCA-rbr) algorithm*, which performed the best of the three base clustering algorithms, and the popular *Agglomerative Hierarchical (PCA-agglo) algorithm*. The agglomerative hierarchical clustering algorithm is a popular bottom-up clustering algorithm. In this method, the desired k-way clustering solution is computed using the agglomerative paradigm whose goal is to locally optimize (minimize or maximize) a particular clustering criterion function. The algorithm finds the clusters by initially assigning each object to its own cluster and then repeatedly merging pairs of clusters until either the desired number of clusters has been obtained or all of the objects have been merged into a single cluster leading to a complete agglomerative tree.

3.3.2 Weighted Consensus

An alternative approach to pruning, is to weight proteins based on the reliability of the clusters they belong to. The intuition here is that, if two proteins are present together in a cluster of poor reliability, the corresponding interaction between them can be deemed to be of low significance and given a low weight. The base clusters obtained can be used to construct a new graph, with an edge existing between proteins iff they have been clustered together at least once. The weights for these edges are proportional to the reliability of the clusters they belong to.

$$Weight(i,j) = \sum_{k=1}^{p} Rel(cl_k) \times Mem(i,j,cl_k) \qquad (4)$$

where $Rel(cl_k)$ is the Reliability score of cluster cl_k and $Mem(i,j,cl_k)$ is the cluster membership function.

$$Mem(i,j,cl_k) = \begin{cases} 1, & \text{iff } (i,j) \in cl_k \\ 0, & \text{otherwise} \end{cases}$$

The weighted graph is then clustered using the *Agglomerative Hierarchical (PCA-agglo) algorithm.*

3.3.3 Soft Consensus Clustering

As we mentioned earlier, several proteins are known to participate in several functions in the cell. By assigning all proteins to a single cluster each, we are inhibiting the number of functions that can be discovered. To overcome this issue, we construct a variant of the PCA-agglo consensus algorithm to perform soft clustering of proteins. The hard agglomerative algorithm places each protein into the most likely cluster to satisfy a clustering criterion. However, it is possible for a protein to belong to two clusters with varying degrees. The probability of a protein belonging to an alternate cluster can be expressed as a factor of its distance from the nodes in the cluster. If a protein has sufficiently strong interactions with the proteins that belong to a particular cluster, then it can be considered amenable to multiple membership. We use the average shortest path distance to quantify this measure.

$$P(i,cl_k) = 1 - \frac{\sum_{j \in V_{cl_k}} SP(i,j)}{|V_{cl_k}| * Diam_G} \qquad (5)$$

where $SP(i,j)$ denotes the length of the shortest path between i and j, $Diam(G)$ is the diameter of the PPI graph, and V_{cl_k} denotes the nodes in cluster cl_k. The algorithm computes the probability for each protein and each cluster. We use a global threshold to assign all nodes that have high propensity towards multiple membership into their respective alternate clusters. Note that, although we perform this operation for all nodes, the nodes with the highest probability for multiple membership are the hubs in the PPI graph, which have been hypothesized to be multi-functional in nature [20]. Owing to their high degrees, they are more likely to interact with proteins having different functions.

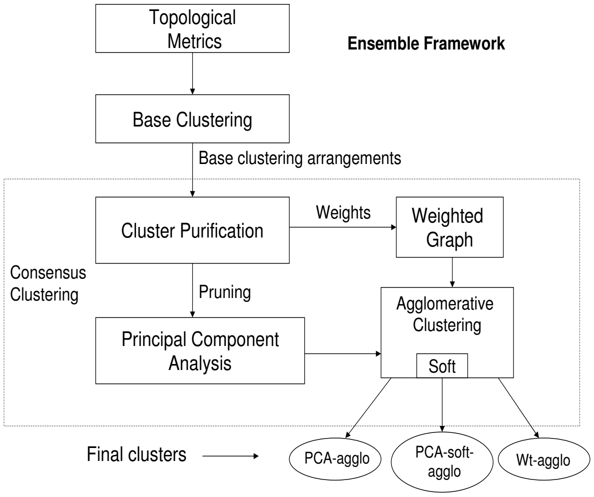

Figure 1. Overview of the Ensemble framework. Note that although we show only the agglomerative algorithm in the figure, the rbr algorithm can be used similarly

3.3.4 Putting It All Together

Figure 1 gives the overview of our ensemble framework. In the first step, the two topological metrics (Clustering Coefficient-based and Betweenness-based) are used with the three base clustering algorithms to reduce the noise in the PPI graph and produce

6 base clustering arrangements. In the consensus stage, the base clusters obtained are subjected to cluster purification to eliminate noisy clusters. We described two different techniques - pruning and weighting. The pruned clusters are fed into the PCA algorithm, which removes redundancies and noise and yields a compact representation. The result of the PCA step is a reduced matrix that contains only discriminatory information for proteins to be easily clustered. Alternately, the weights based on cluster reliability can be used to construct a new graph. For final consensus clustering, we use two algorithms as mentioned before - the Agglomerative algorithm and the RBR algorithm. Additionally, soft clustering can be performed to cluster certain proteins in multiple clusters.

4. Experiments

4.1 Dataset

The Protein-Protein Interactions (PPI) network of budding yeast (*Saccharomyces Cerevisiae*) has been studied earlier in several works [2, 34, 33, 38, 39]. This dataset is available from the Database of Interacting Proteins (DIP). It consists of 17194 interactions between 4928 proteins.

4.2 Validation Metrics:

Before presenting our experimental results, we would like to describe our validation metrics. We use both domain-specific and general metrics to evaluate the quality of the consensus clusters.

4.2.1 Topological Measure: Modularity

The first metric we use is a topology-based Modularity metric, originally proposed by Newman [23]. This metric uses a k X k symmetric matrix of clusters where each element d_{ij} represents the fraction of edges that link nodes between clusters i and j and each d_{ii} represents the fraction of edges linking nodes within cluster i. The modularity measure is given by

$$M = \sum_i (d_{ii} - (\sum_j d_{ij})^2)$$

4.2.2 Information Theoretic Measure: Normalized Mutual Information (NMI)

Another metric to evaluate the quality of clusters obtained is the amount of mutual information shared between clusterings. This metric was originally described by Strehl *et al* [31]. They define the optimal combined clustering as the one that shares the most information, in terms of mutual information, with the original clusterings. Assume r groupings denoted as $\Lambda = \{\lambda^q | q \epsilon \{1, .., r\}\}$. Suppose there are two clusterings λ^a and λ^b of sizes k^a and k^b respectively. Let n_h be the number of objects in cluster C_h according to λ^a, n_l the number of objects in cluster C_l according to λ^b and n_l^h is the number of objects in cluster C_h according to λ^a and in Cluster C_l according to λ^b. The [0-1] normalized mutual information ϕ^{NMI} can be calculated as follows:

$$\phi^{NMI}(\lambda^a, \lambda^b) = \frac{2}{n} * \sum_{k^a}^{l=1} \sum_{k^b}^{h=1} n_l^h * \log_{k^a * k^b} \frac{n_l^h * n}{n^h * n_l}$$

The average normalized mutual information (ANMI) [31] between a set of r labelings, Λ and a labeling named λ^i is defined as follows:

$$\phi^{NMI}(\Lambda, \lambda^i) = \frac{1}{r} * \sum_{r}^{q=1} \phi^{NMI}(\lambda^i, \lambda^q)$$

Here Λ is the set of base clusterings and λ^i is the consensus clustering.

4.2.3 Domain-based Measure: Clustering Score

For the PPI network, we need to test if the clusters obtained correspond to known functional modules. This can be done by validating the clusters using known biological associations from the Gene Ontology Consortium Online Database [3] [2]. The Gene Ontology (GO) database provides three vocabularies of known associations - *Cellular Component* which refers to the localization of proteins inside the cell, *Molecular Function* which refers to shared activities at the molecular level and *Biological Process* which refers to entities at both the cellular and organism levels of granularity. Earlier works have used these three ontologies to validate the biological significance of clusters [34, 2, 33]. We use all three annotations for validation and comparison. [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, p-values are used to calculate the statistical and biological significance of a group of proteins. The p-values essentially represent the chance of seeing that particular grouping, or better, given the background distribution. Assume a cluster of size n, with m proteins sharing a particular biological annotation. Also assume that 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:

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

Smaller p-values imply that the grouping is not random and is more significant biologically than one with a higher p-value. A $cutoff$ [4] parameter is used to differentiate significant groups from the insignificant ones. If a cluster is associated with a p-value greater than $cutoff$, it is considered insignificant. [5]

As the p-value of a single cluster is statistically not representative, we define a Clustering score function to quantify the overall

[2] http://db.yeastgenome.org/cgi-bin/GO/goTermFinder

[3] As of February 1, 2007, the GO database contains 6700 genes annotated with 1864 cellular component , 7527 molecular functions and 13155 biological processes.

[4] The GO ontology performs multiple hypothesis testing to adjust the $cutoff$ value.

[5] We used the recommended cut-off of 0.05 for all our validations.

clusters, as follows.

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

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

4.3 Experimental Results

4.3.1 Evaluation of Similarity Metrics

We first evaluate the two similarity metrics we have developed for base clustering. In particular, we wish to validate the benefits of using weighted metrics for eliminating noise. To do this, we apply the clustering algorithms on an unweighted graph, where all edges are treated the same ($= 1$). We then compare the clusters obtained using the domain-based Clustering score measure. To compare, we also implement a neighborhood metric based on the Czekanowski-Dice distance metric [8], which has been previously employed for clustering PPI graphs [10]. The neighborhood-based similarity metric is defined as:

$$S_n(v_i, v_j) = 1 - \frac{|Int(i) \Delta Int(j)|}{|Int(i) \cup Int(j)| + |Int(i) \cap Int(j)|} \quad (6)$$

Here, $Int(i)$ and $Int(j)$ denote the adjacency lists of proteins i and j, respectively, and Δ represents the symmetric difference between the sets. Note that using this metric, nodes that do not interact with each other may have non-zero similarity if they have common neighbors. The comparison, in terms of Clustering scores for the RBR algorithm [6], is given in Figure 2. The Betweenness and Clustering Coefficient-based metrics have high Clustering score values for all three ontologies. This indicates that the *Betweenness and Clustering Coefficient-based metrics can help reduce the effect of noise*, leading to meaningful clusters. The Neighborhood metric, on the other hand, performs worse than the unweighted scenario. The metric assigns non-zero scores to pairs of nodes that are not connected in the original graph, if they have common neighbors. The results suggest that this addition of new edges contributes to increased noise in the PPI graph.

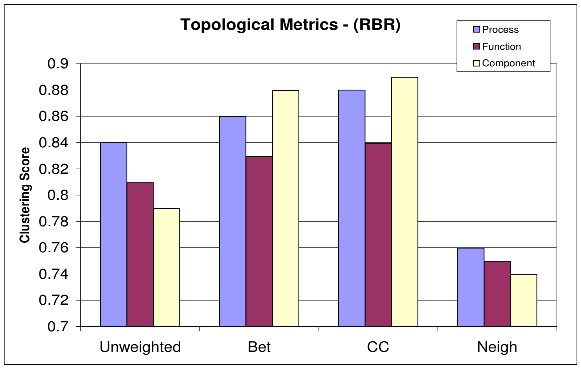

Figure 2. Domain-based Comparison of Base Similarity Metrics

4.3.2 Consensus Clustering

We use the three graph clustering algorithms with the two topology-based metrics to obtain six independent base clusterings each. Estimating the optimal number of clusters, k, is a serious issue in clustering. Earlier approaches [30] have suggested using the ratio between the inter-cluster and intra-cluster similarities to estimate the value. We used both similarity metrics with the Metis algorithm to estimate cluster quality for different values of k. We performed the same operation with the other two algorithms. Finally, one of the values optimal for all three algorithms was chosen as the value of k. Accordingly, the value of k for the PPI dataset was chosen to be 100. Once the base clusters are obtained, the cluster purification step is performed to prune away weak clusters. The remaining clusters are then represented in the form of a matrix, as described earlier, and PCA is applied to reduce the dimensions. We select the number of dimensions that capture 95% of the total variance. We then perform consensus clustering using three algorithms - the agglomerative hierarchical algorithm (PCA-agglo), the repeated bisections divisive algorithm (PCA-rbr) and the soft consensus (PCA-softagglo) algorithm. We also investigate the benefits of weighted (Wt-agglo) consensus clustering, for comparison.

To compare with our consensus technique, we use the three ensemble algorithms proposed by Strehl *et al* [31] - CSPA, HGPA and MCLA, and two ensemble algorithms - Balls (CE-balls) and Agglomerative (CE-agglo) proposed by Gionis *et al* [17]. The latter two algorithms do not accept the required number of clusters as a parameter. When we used the default settings for both, with a distance matrix based on shortest path distances, the CE-agglo algorithm produced 2121 clusters and the CE-balls algorithm yielded 2783 clusters for the 4928 proteins. Most of these clusters contained only singletons or pairs. Also, the CSPA algorithm ran out of memory for this dataset. It seems to be conducive only for small datasets.

Modularity and NMI: First, we compare the consensus algorithms in terms of their Modularity and Average Normalized Mutual Information scores. Figure 3 shows the comparative results in terms of both these metrics for 4 consensus methods. The CE-agglo and CE-balls algorithms, as we mentioned earlier, resulted in a large number of clusters, most of which contained only singletons and pairs. [7] Hence, the modularity and NMI scores were very low for these clusters and are not presented here.

It can be observed that the *PCA-agglo and PCA-rbr algorithms perform the best* with high scores in terms of both metrics.

Domain-based Evaluation: We proceed to evaluate the clusters obtained from the consensus algorithms using the domain-based metric. Figure 4 shows the comparison in terms of Clustering

[6] The trends for the other two clustering algorithms are similar and are omitted

[7] 1124 of the 2121 clusters produced by the CE-agglo algorithm contained singletons, whereas for the CE-balls algorithm, 1939 of the 2783 clusters contained singletons.

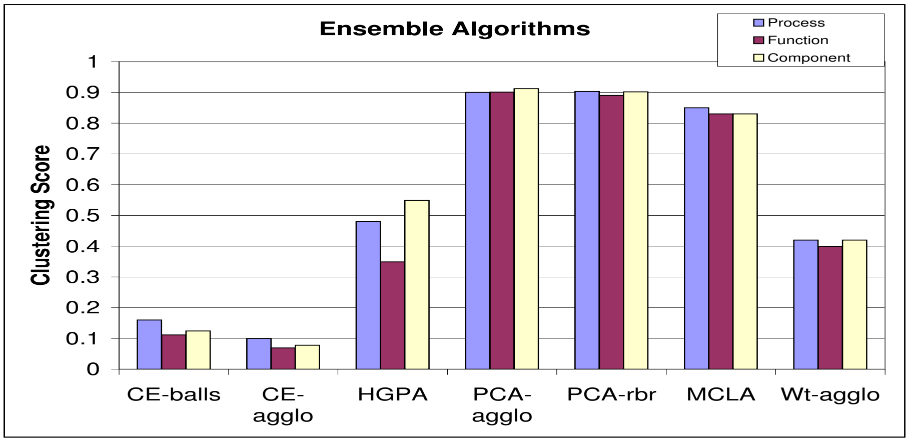

Figure 4. Domain-based Clustering scores for consensus algorithms. Comparisons with MCLA, HGPA, CE-Balls and CE-agglo.

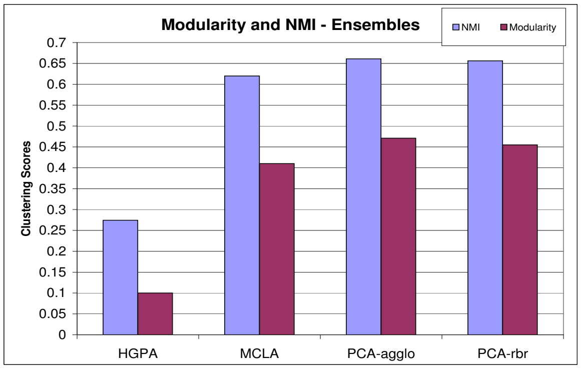

Figure 3. Modularity and NMI scores for consensus algorithms

Score for the *Biological Process*, *Molecular Function* and *Cellular Component* ontologies. Since the CE-agglo and CE-balls contain a large number of singletons, they have very few significant clusters. The PCA-based consensus methods once again do better than all the other algorithms. The *PCA-agglo and PCA-rbr algorithms* provide the *best clustering scores* overall. The CE-balls and CE-agglo, due to the large number of singletons and pairs, perform the worst, with very poor Clustering scores for all three ontologies. The Wt-agglo consensus method has poor results due to the fact that it produces 55 singleton clusters. However, we found that out of the other 45 clusters, most were significant. The fact that not all proteins were clustered by the weighted consensus method suggests that pruning with PCA is a better option.

Next, we further analyze the clusters obtained with the PCA-based consensus clustering. We consider the clusters obtained by the *PCA-rbr algorithm* and compare them against the *MCLA algorithm*, which was the best of the other consensus methods we compared against. Figure 5 shows the comparison between the two algorithms, in terms of p-value distribution of the clusters obtained, for the *Molecular Function* ontology [8]. The p-value distribution of the metis base algorithm is also provided for reference. The y-axis, in this case, corresponds to -log(pvalue), which means that higher values correspond to better biological significance. We find that both the consensus algorithms outperform the base clustering algorithm, as expected. The clusters obtained using the *PCA-rbr algorithm consistently outperform the MCLA clusters* in terms of biological significance. The MCLA algorithm results in 84 significant clusters for the *Molecular Function* ontology whereas the PCA-rbr algorithm provides 87. The best cluster we obtain with PCA-rbr for this ontology has a p-value score of *4.3e-58*. The best scoring cluster for the MCLA algorithm has a much worse p-value score of 2.73e-49. The best-scoring cluster for the PCA-rbr algorithm is composed of 64 proteins, among which 31 are annotated with the same *Molecular Function* term *GO:0004299 - proteasome endopeptidase activity*. In the whole genome, there are only 34 proteins (out of 6700 annotated proteins in the database) that are associated with this term. This result strongly emphasizes the quality of the clusters we obtained with the PCA-rbr algorithm. Such high-quality clusters are essential for predicting unknown functions of proteins. For instance, in the same cluster, there exist several proteins such as YPL066W, YCR001W, YBR204C and YLR040C that have not been previously annotated with a known *Molecular Function*. These results can be very effective in explaining and guiding wet-lab experiments for further analysis of the relation between these proteins and the specified GO term.

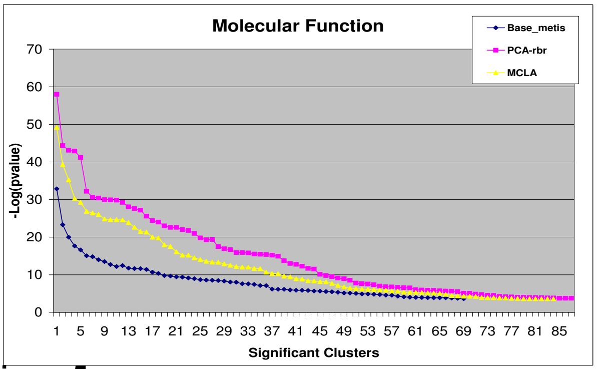

Figure 5. P-value distribution Comparison for Molecular Function ontology

In the case of MCLA, we obtain two clusters that are significantly annotated with the same GO term,*proteasome endopeptidase activity*. One of these clusters has 12 proteins (out of 40) and the other has 20 (out of 50) that are associated with this term. The p-value scores for these annotations are 9.8e-20 and 1.9e-36

[8] The plots for the other two ontologies follow similar trends and have been omitted due to lack of space.

respectively. On the other hand, as we previously stated, the PCA-rbr algorithm is able to assign *almost all these proteins (31 out of 34)* to a single cluster with a p-value score of e-58.

These results further demonstrate the effectiveness of the PCA-based clustering approach in finding biologically meaningful groups for the PPI dataset.

4.3.3 Comparison with MCODE and MCL

Next, we compare our consensus technique with two algorithms commonly utilized for extracting functional modules from PPI graphs - MCODE and MCL. A recent study [6] that compared these algorithms (among others) showed that the MCL algorithm, in particular, was very effective in identifying protein complexes from protein interaction networks. We wish to investigate the benefits of ensemble clustering when compared to these two algorithms.

We used the MCODE and MCL algorithm to extract clusters from the PPI graph. We used the default settings for MCODE (fluff option set to 0.1, mode score cut-off set to 0.2, degree cut-off set to 2), and obtained 59 clusters. One major drawback of this algorithm is that not all the proteins (vertices) in the network are clustered. The clusters we obtained consisted of only 794 proteins (out of 4928). From the domain-based metric, we found that among these 59 clusters, 46 clusters had significant *Cellular Component* annotations, 40 clusters had significant *Molecular Function* and 50 clusters had significant *Biological Process* annotations. On the other hand, the MCL algorithm generated 1246 clusters for the 4928 proteins. However, on examination, we found that most of these clusters were insignificant. Only 277 out of the 1246 were significant for *Biological Process*.

The p-value distributions for the 50 best clusters for PCA-agglo, MCODE and MCL for the *Biological Process* ontology are shown in Figure 6. Note that the graph illustrates improvements across the board and not merely among the best clusters. The MCODE algorithm produces only 50 significant clusters for this ontology. The biological significance of these clusters is very poor compared to the other two. The top 50 of these 277 clusters have consistently lower significance than the PCA-agglo clusters, as can be observed from the figure.

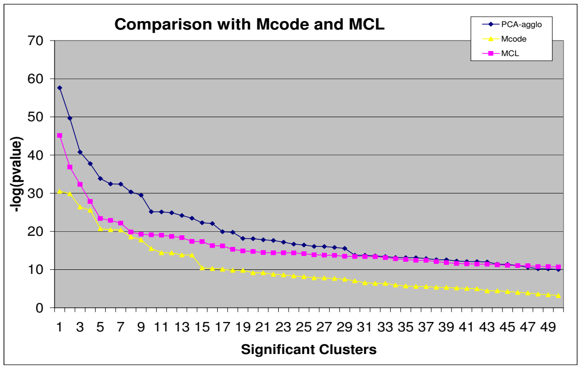

Figure 6. P-value distribution Comparison with MCODE and MCL for Biological Process ontology.

The PCA-agglo algorithm yielded a *large percentage of significant clusters* (90 out of the 100 clusters were significant for *Biological Process*) and with small p-values (high values of -log(pvalue)).

Algorithm	Modularity
PCA-agglo	**0.471**
PCA-rbr	0.46
MCLA	0.41
MCL	0.217
MCODE	0.372

Table 1. Modularity scores comparison

Moreover, PCA-agglo clustered all 4928 proteins whereas in the case of MCODE, a majority of the proteins (around 85%) were unclustered. In the case of MCL, the top 30 clusters are of much lower significance than the PCA-agglo clusters, although the two algorithms become comparable subsequently.

When we compared the modularity scores, we once again found the *PCA-based methods outperforming MCODE and MCL*. The modularity scores are given in Table 1 below. As we mentioned earlier, MCL produced a large number of clusters and most of the proteins in the clusters were sparsely connected. Since MCODE did not cluster all proteins, we only consider edges among the proteins clustered to compute the modularity. The results show that the ensemble methods produce denser clusters, with the PCA-agglo algorithm performing the best overall.

Qualitative Comparison with MCODE: We analyze the highest ranked cluster obtained by MCODE and the corresponding PCA-agglo cluster using the *Cellular Component* ontology to compare the effectiveness of these algorithms in terms of identifying protein complexes. The best scoring cluster in MCODE (with score 5.615) is composed of 26 proteins among which 15 belong to a known complex *proteasome regulatory particle* (GO:0005838). This grouping is associated with a small p-value of 8.5e-34. On the other hand, the PCA-agglo cluster that includes a majority of the same vertices has 21 proteins belonging to the *proteasome regulatory particle* complex. The significance of this result can be accentuated by the fact that out of the 6700 annotated proteins in the GO database, there exist only 23 proteins annotated with this complex. PCA-agglo groups 21 of them in one cluster (p-value 7.6e-49). The corresponding clusters produced by the two algorithms are plotted in Figure 7 (a) and (b). The white vertices represent proteins that are known to be part of this complex whereas the black ones do not have a known annotation in GO for that term. As can be seen from these two clusters, the cluster obtained by the PCA-agglo algorithm is denser compared to the MCODE cluster. In the MCODE cluster, there exist two separate dense regions, one composed of proteins in the *proteasome regulatory particle* complex and the other composed of proteins in the *snRNP U6* complex (GO:0005688). This example indicates that PCA-agglo can obtain dense and homogeneous clusters.

Qualitative Comparison with MCL: Next, we compare the clusters obtained by the MCL algorithm with the ones from PCA-rbr. The MCL algorithm partitioned our interaction network into 1246 clusters. Among these only 277 of them had significant *Biological Process* annotations , 216 of had significant *Molecular Function* and 226 of them had significant *Cellular Component* annotations. This meant that, around *900-1000 of the clusters were insignificant*. On the other hand, out of the 100 clusters produced

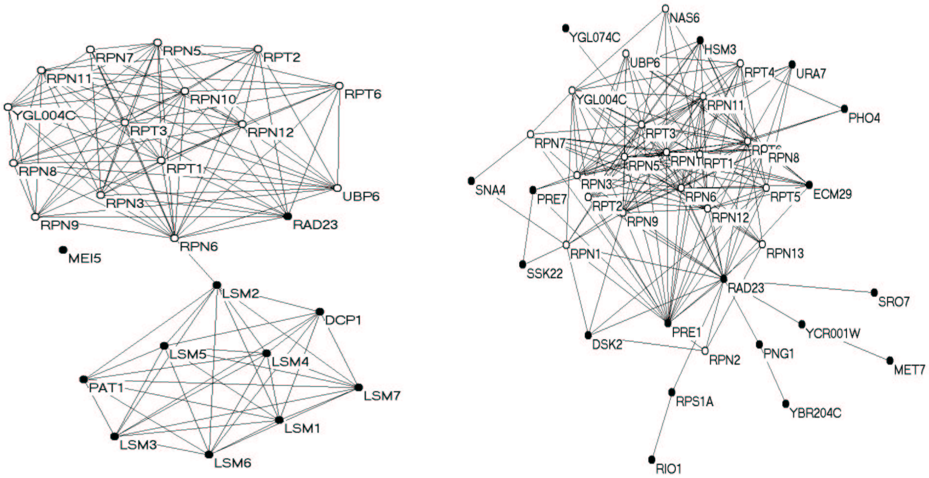

Figure 7. a)MCODE cluster b)PCA-agglo cluster

by PCA-rbr there exist 89 clusters with significant *Cellular Component* annotations, 87 clusters with significant *Molecular Function* annotations and 90 clusters with significant *Biological Process* annotations. Although MCL is able to produce more clusters, the precision (percentage of significant clusters) and the biological significance within the clusters is low.

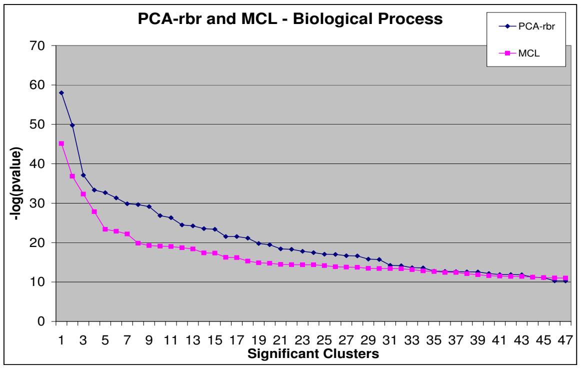

Figure 8. P-value distribution Comparison with PCA-rbr and MCL for Biological Process ontology.

For our analysis we considered the clusters with significant *Biological Process* annotations for the two algorithms. The corresponding distributions for the top 47 [9] clusters are shown in Figure 8. The MCL algorithm grouped 1940 proteins into 277 significant clusters with average cluster size of 7. Although PCA-rbr algorithm identifies only 90 clusters with significant annotations, 4145 proteins are grouped into these clusters (average cluster size is 46). To assess the biological homogeneity of these clusters, we label each of these clusters with the most significant GO annotations (p-values). Accordingly, the most significant annotation for MCL clusters for the *Biological Process* ontology has a p-value of 7.15e-46, whereas the most significant annotation for the PCA-rbr clusters is 2.38e-58. Furthermore, the average p-values for all significant clusters of MCL is 1.2e-04 whereas the average for PCA-rbr clusters are 1.4e-05. These results show that *MCL produces many small-sized clusters which are not as homogeneous as the clusters obtained by the PCA-rbr algorithm.*

To further analyze the effectiveness of these algorithms for protein complex identification purposes, we compared the most significant cluster obtained by MCL algorithm according to the *Cellular Component* ontology with its counterpart among the PCA-rbr clusters. The best cluster produced by the MCL algorithm (for this ontology) groups 31 proteins, among which 26 are known to be part of *organellar large ribosomal subunit* (GO:0000315). This arrangement is associated with a p-value of 5.7e-56. To find the corresponding PCA-rbr cluster, we identified the cluster that includes the most number of proteins from this cluster. As expected, the corresponding PCA-rbr cluster is also enriched with the proteins that are associated with *organellar large ribosomal subunit* . There exist 30 proteins (out of 40) in the corresponding PCA-rbr cluster which have known annotations with this complex (p-value is 1.3e-62). This cluster includes all 25 proteins that are correctly put together by the MCL algorithm as well as 5 other proteins (IMG1, MRP7, MRPL17, YDR115W, MRPL15) from the same complex that MCL fail to locate into this cluster. This illustrative example shows that the *PCA-rbr clusters are larger and more homogeneous* and may hence be better suited for the extraction of protein complexes.

4.3.4 Soft Clustering

As we mentioned earlier, many proteins in PPI networks are believed to exhibit multiple functionalities, interacting with different groups of proteins for different functions. To identify these multi-faceted proteins, we used the soft-clustering variant of the PCA-agglo algorithm, which allows proteins to belong to multiple clusters. The algorithm identifies proteins that have high propensity for multiple membership. We use a strict threshold of 0.2 and assign a protein to an alternate cluster only if its average shortest path distance to the cluster is below 0.2. When we obtain the soft clusters, we found that a majority of the proteins that had multiple membership were hub proteins (proteins with high degrees). This is consistent with our initial assumption, since hub proteins are likely to be well-connected and are believed to exhibit multiple functionalities.

To emphasize the benefits of performing soft clustering, we provide an illustrative example.

CKA1 is a multi-faceted hub protein, involved in multiple cellular events such as maintenance of cell morphology and polar-

[9] The remaining clusters have comparable pvalues

ity, and regulating the actin and tubulin cytoskeletons. When we analyze the base clusterings using the clustering scores, we find that the base clusterings associate this hub protein in different groups. Three of the base algorithms (direct-betweenness, rbr-clustering coefficient and rbr-betweenness) group CKA1 with all the other proteins (CKB1,CKB2,CKA2) in protein kinase CK2 complex. On the other hand, the direct-clustering coefficient base algorithm grouped CKA1 together with 33 other proteins that take part in RNA metabolism and the metis-betweenness base algorithm clusters it with proteins associated with cell organization and biogenesis (23 other proteins). These results indicate that most of the base clustering algorithms (except metis-clustering coefficient) are able to assign a multi-faceted protein to a cluster that includes proteins associated with one of its functions. A hard consensus clustering algorithm can only associate CKA1 with the most popular term. Accordingly, the pca-agglo consensus algorithm groups CKA1 with the protein kinase CK2 complex proteins in consensus with the majority of the base algorithms. This cluster, in which CKA1 has been placed by the PCA-agglo algorithm, has few proteins associated with the *cell organization and biogenesis* functionality. The soft clustering algorithm, on the other hand, places CKA1 into 3 clusters with significant enrichment scores. One of these clusters is consists of proteins associated with RNA metabolism with a significant p-value of 1.4e-23. The second cluster includes all protein kinase CK2 complex proteins (1.6e-09) whereas the third cluster is composed of cell organization and biogenesis proteins (4.8e-16). This example clearly shows that soft consensus clustering can lead to the discovery of multiple functionalities for proteins. The benefit of ensemble clustering is once again evident, since the *different base clustering algorithms uncover different functionalities*, which can be summed up adequately by the soft consensus clustering algorithm.

In our earlier work [33] we developed a soft clustering method based on hub-duplication for the PPI dataset. Now, we compare the performance of the PCA-based soft consensus method with the hub-duplication technique. The p-value distributions for the *Biological Process* ontology [10] is shown in Figure 9. It can be observed that the *PCA-soft-agglo method consistently yields clusters with higher biological significance* than the hub-duplication technique. It can be hypothesized that the good performance of the soft ensemble algorithm is due to the fact that it assimilates the results of different base clusterings, whereas typical soft clustering algorithms use a single clustering criterion.

5. Conclusion

In this paper, we have presented an ensemble framework for partitioning PPI networks. To obtain informative base clusters, we have developed two topological metrics that can counteract the effect of noisy (false positive) interactions in the PPI network. We have presented a detailed consensus technique involving Principal Component Analysis (PCA), designed to scale to large datasets and reduce the dimensionality of the consensus determination problem. Additionally, we have introduced topology-based pruning strategies to complement PCA in the task of eliminating redundant and noisy data. Finally, we have presented a soft consensus clustering algorithm, that is designed to discover multiple functional associations for proteins. Our thorough empirical evaluation and comparison of these consensus clustering algorithms with other state-of-the-art approaches using topological, information theoretic and domain specific validation metrics, demonstrate that the proposed PCA-based algorithms, apart from the scalability advantage, can lead to consensus clusters with high efficiency. Also, the PCA-based soft consensus clustering algorithm proves to be very effective in identifying multiple functionalities of proteins. The qualitative comparison of our clusters with those of popular algorithms such as MCODE and MCL reveals that ensemble algorithms can yield larger, denser clusters with improved biological significance. In the future, we would like to focus on extensions for the base algorithms. Also, we would like to extend the notion of ensembles to inculcate domain bias for fusing information from multiple experimental and in-silico PPI networks.

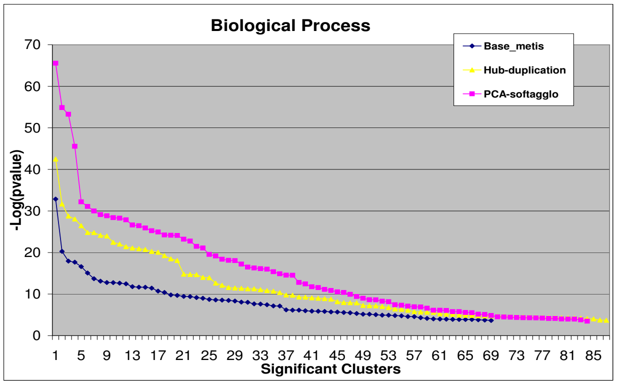

Figure 9. P-value distribution Comparison for Soft Clustering

6. References

[1] C. C. Aggarwal. Re-designing distance functions and distance-based applications for high dimensional data. *SIGMOD Record*, 30(1):13–18, 2001.

[2] V. Arnau, S. Mars, and I. Marin. Iterative cluster analysis of protein interaction data. *Bioinformatics*, 21:3:364–378, 2005.

[3] 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.

[4] G. Bader and C. W. Hogue. Analyzing yeast protein-protein interaction data obtained from different sources. *Nat Biotechnol.*, 20(10):991–997, 2002.

[5] G. Bader and C. W. V. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics.*, 4(2), 2003.

[6] S. Brohe and J. van Helden. Evaluation of clustering

[10] The plots for the molecular function and cellular component ontologies follow similar trends and have been omitted due to lack of space.

algorithms for protein-protein interaction networks. *BMC Bioinformatics.*, 7(488), 2006.

[7] C. Brun, C. Herrmann, and A. Guenoche. Clustering proteins from interaction networks for the prediction of cellular functions. *BMC Bioinformatics*, 5(95), July 2004.

[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] J. Chen, W. Hsu, M. L. Lee, and S. Ng. Increasing confidence of protein interactomes using network topological metrics. *Bioinformatics*, 22(16):1998–2004, 2006.

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

[11] C. Ding, X. He, H. Zha, and H. Simon. Adaptive dimension reduction for clustering high dimensional data. *Proc. ICDM 2002*, pages 107–114, 2002.

[12] S. V. Dongen. Graph clustering by flow simulation. *Centers for mathematics and computer science (CWI), University of Utrecht*, pages 49–57, 2000.

[13] S. Fields and O. Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340:245–246, 1989.

[14] S. Fields and R. Sternglanz. The two-hybrid system: an assay for protein-protein interactions. *Trends Genet.*, 10:286–292, 1994.

[15] A. Fred and A. Jain. Data clustering using evidence accumulation. *In Pmc. ICPR*, 2002.

[16] C. Friedel and R. Zimmer. Inferring topology from clustering coefficients in protein-protein interaction networks. *BMC Bioinformatics*, 7(519), 2006.

[17] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *21st International Conference on Data Engineering (ICDE'05)*, pages 341–352, 2005.

[18] D. C. Hoyle and M. Rattray. Pca learning for sparse high-dimensional data. *Europhysics Letters*, 62:117–123, 2003.

[19] J. Hua, D. Koes, and Z. Kou. Finding motifs in protein-protein interaction networks. *Project Final Report, CMU*, 2003.

[20] 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.

[21] P. Kahn. From genome to proteome. *Science*, 270, 1995.

[22] 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*.

[23] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[24] M. H. P Holme and H. Jeong. Subnetwork hierarchies of biochemical pathways. *Bioinformatics*, 19:532–538, 2003.

[25] J. Pereira-Leal, A. Enright, and C. Ouzounis. Detection of functional modules from protein interaction networks. *Proteins*, 54(1):49–57, 2004.

[26] E. M. Phizicky and S. Fields. Protein-protein interactions: methods for detection and analysis. *Microbiol.Rev*, 59:94–123, 1995.

[27] M. D. Richard and R. P. Lippmann. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Computation*, 3(4):461–483, 1991.

[28] R. Saito, H. Suzuki, and Y. Hayashizaki. Interaction generality, a measurement to assess the reliability of a proteinprotein interaction. *Nucleic Acids Research*, 30(5):1163–1168, 2002.

[29] R. Singh, J. Xu, and B. Berger. Struct2net: integrating structure into protein-protein interaction prediction. *Pac Symp Biocomput*, pages 403–414, 2006.

[30] A. Strehl and J. Ghosh. Relationship-based clustering and visualization for high-dimensional data mining. *INFORMS Journal on Computing*, 2002.

[31] A. Strehl and J. Gosh. Cluster ensembles - a knowledge reuse framework for combining partitionings. *AAAI*, pages 93–98, 2002.

[32] A. Topchy, M. Law, A. K. Jain, and A. Fred. Analysis of consensus partition in cluster ensemble. *IEEE International Conference on Data Mining, ICDM*, pages 225–232, 2004.

[33] D. Ucar, S. Asur, U. Catalyurek, and S. Parthasarathy. Improving functional modularity in protein-protein interactions graphs using hub-induced subgraphs. *PKDD*, 2006.

[34] D. Ucar, S. Parthasarathy, S. Asur, and C. Wang. Effective preprocessing strategies for functional clustering of a protein-protein interactions network. *BIBE*, 2005.

[35] J. Vasilescu, G. Xuecui, and J. Kast. Identification of protein-protein interactions using in vivo cross-linking and mass spectrometry. *Proteomics*, 4(12):3845–3854, 2004.

[36] D. von Mering, C. Krause, and *et al*. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 31:399–403, 2002.

[37] D. Watts and S. Strogatz. Collective dynamics of small world networks. *Nature*, 393(6684):440–442, June 1998.

[38] 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.

[39] S. Yook, Z. N. Oltvai, and A. L. Barabasi. Functional and topological characterization of protein interaction networks. *Proteomics*, 4:928–942, 2004.