An Ensemble Approach 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 PPI networks has been theorized by many researchers. However, PPI networks are known to contain noisy false positive interactions and possess the scale-free property, which makes the task of isolating these useful modules difficult. In this paper, we propose an ensemble clustering approach to address this problem. To perform initial clustering, we examine three topology-based distance metrics that are conducive for partitioning these networks. To perform consensus clustering, we develop a PCA-based hypergraph approach, designed to handle large interaction networks. We also develop a soft consensus clustering method to assign multifaceted hub 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 hub proteins.

1 Introduction

Proteins are central components of cell machinery and life. In fact, as noted by Kahn[13], it is the proteins dynamically generated by a cell that execute the genetic program. However it is insufficient to reason about their functionality in a stand-alone manner since proteins work with other proteins to regulate and support each other for specific functions [25]. Protein-Protein interaction (PPI) networks represent experimentally or in-silico determined interactions between proteins. The presence of biologically relevant functional modules in PPI networks has been theorized by many researchers [7, 12, 28]. The task of extracting these functional modules for the purposes of understanding the behavior of organisms, protein function prediction and drug design is an active research area in functional genomics. However, the application of traditional clustering algorithms for the extraction of these modules has yielded limited success [29, 23].

The challenges involved are manifold. First, the PPI networks exhibit classic scale-free properties [19], with a few nodes (hubs) having very large degrees, while most nodes have very few interactions. This is detrimental when adopting traditional partitioning or clustering algorithms to the task. The resulting clustering arrangement typically contains one or a few giant core clusters and several tiny clusters that are often not very useful. Second, the interactions data is known to be quite noisy - many detected interactions are conjectured to be false positives. Also, some proteins are believed to be multi-functional – effective strategies for soft clustering of these essential proteins are necessary. Finally, different experimental and in-silico methodologies have been used to detect these interactions with little overlap in terms of detected interactions. Fusion of information from multiple sources is a key problem.

Researchers to date have applied several techniques targeted at specific problems listed above [3, 6, 7, 12, 22, 23, 24, 28, 29]. In this work we present a uni-

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

fied solution, adapting the notion of ensemble clustering, to potentially attack these problems. Ensemble clustering has been proposed as a useful approach to strengthen the performance of simple clustering algorithms [9, 10, 20, 21, 18]. The goal is to combine multiple, diverse and independent clustering arrangements to obtain a single, comprehensive 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 [21]. Also, one can weight individual clustering arrangements according to their strengths and weaknesses, potentially addressing the fusion problem[1].

The critical challenges we address in this work include adapting ensemble clustering to work in a scale-free environment (none of the existing methods do), allowing these algorithms to scale to large PPI networks and finally, adapting them in a domain specific manner to enable the soft clustering of essential proteins.

Existing clustering algorithms cannot be applied directly to PPI networks, due to the large variance in degree between nodes (scale-free property) and the lack of a suitable metric to capture node similarity. In PPI networks, the only information available directly is the set of proteins and the interactions between them. Also, as we mentioned, several interactions are believed to be false positives. To address these problems, we develop and apply three different topological distance metrics based on neighborhood, clustering coefficient and shortest path betweennness of nodes in the network. These metrics, by design, provide low similarity scores for proteins having potentially-false interactions. We use three traditional graph partitioning algorithms with these metrics to obtain nine base clusterings that are diverse and yet informative about the topological properties of nodes in the network.

Existing approaches to ensemble clustering [9, 10, 20, 21, 18] have experimented with several graph-based, combinatorial and statistical consensus methods. However, most of these methods have been applied to small datasets. To represent clustering arrangements, some authors [9, 20] have used a co-association matrix representation. This representation scales quadratically with the size of the dataset and is hence, infeasible for large datasets. Strehl *et al* [18] propose a hypergraph representation which is inefficient when the number of clusters is large. To address the scalability problem, we rely on Principal Component Analysis (PCA) to reduce the dimensionality of the problem and yield an efficient representation for the clusterings, that can then be effectively clustered using traditional algorithms.

Another challenge, as we mentioned earlier, is the need to assign proteins to different groups (soft clustering) based on their functions. Hub proteins typically have multiple functions and are likely to be essential for the organism. In this regard, we implement a soft consensus clustering algorithm designed to discover multiple functional associations for hub 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 the consensus clusterings obtained. Our experimental results show that our algorithms can provide significant improvement in cluster quality, when compared to previously reported consensus methods. We also show that ensemble clustering can facilitate the discovery of multiple functional associations for hub proteins.

To summarize, the main points of this paper include

- The application of an Ensemble Clustering approach to Protein-Protein Interaction networks
- The use of three diverse topological distance metrics to obtain informative base clusterings for scale free PPI networks.
- A scalable PCA-based consensus method to obtain meaningful clusters efficiently.
- A soft ensemble clustering approach targeting essential (hub) nodes facilitating improved cluster quality for multi-functional proteins.

2 Related Work

Many clustering algorithms of various types have been applied to analyze scale-free networks. However, there is no single algorithm that can guarantee effective partitioning of natural groups from the core of a

[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

scale-free network. Karypis *et al* [1]; present multilevel graph partitioning algorithms to cluster scale-free networks. Wu *et al* [27] propose a geodesic path-based clustering approach to partition scale-free networks into natural divisions. They use these clusters to create meaningful approximations of the graph.

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* [9] 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. Topchy *et al* [20] reduce this problem into a maximum likelihood problem and propose using the EM algorithm to solve the corresponding problem. In a later work, Topchy *et al* [21] also present two approaches to prove the effectiveness of a cluster ensemble - using plurality voting and using a metric on the space of partitions.

Gionis *et al* [10] 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 scale-free graph datasets. We use these two algorithms for comparison with our techniques.

Strehl and Ghosh [18] 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 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

The general framework of our approach is provided in Algorithm 1. The call $EnsembleClustering(G, CA, k)$ returns k consensus clusters $C_1^{CA} \cup \ldots \cup C_k^{CA}$ for a given PPI network G=(V,E), using consensus algorithm CA. Initially, the base clustering algorithms are applied using the similarity metrics to obtain individual clusterings of k clusters each. This set of clusterings is represented appropriately and then the consensus clustering algorithm is applied to obtain the final set of consensus clusters. In the next few subsections, we describe our similarity metrics, base clustering algorithms and consensus methods in detail.

Algorithm 1 EnsembleClustering(G,CA,k)

```
Input: PPI network $G = (V, E)$ and $k$, the number of clusters required
Output: $C^{CA}$ = $C_1^{CA} \cup \ldots \cup C_k^{CA}$
for $i = 1$ to $|SimMetrics|$ do
  for $j = 1$ to $|BaseAlgorithms|$ do
    // Use each similarity metric with each base algorithm to obtain a clustering of $k$ clusters
    $C^{i*j}$ = $C_1^{i*j} \cup \ldots \cup C_k^{i*j}$
  end for
end for
// Convert the clusterings into representative matrix $M$
$M$ = represent($C^{1*1}$, $C^{1*2}$,...,C$^{|SimMetrics|*|BaseAlgorithms|}$)
// Cluster $M$ using CA
$C^{CA}$ = $C_1 \cup \ldots \cup C_k$
return($C^{CA}$)
```

3.1 Similarity metrics

We employ three different metrics designed to capture diverse topological properties of scale-free networks. We believe that together, they can provide enough information to partition scale-free graphs meaningfully, while reducing the effect of noise.

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 [26] 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 they 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* [15]. 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 [16], 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.

3.1.3 Neighborhood-based

The third metric we use is a Neighborhood-based similarity metric. We use the well-known Czekanowski-Dice distance metric [7] for this purpose. This metric uses the adjacency list of each node and favors nodes that have several common neighbors. Two nodes having no common neighbor will have zero similarity, while those interacting with exactly the same set of nodes will have the maximum value, 1. The similarity metric is defined as:

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

Here, $Int(i)$ and $Int(j)$ denote the adjacency list 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.

3.2 Base algorithms

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

3.2.1 Repeated bisections (rbr):

In this method, the desired k-way clustering solution is computed by performing a sequence of k - 1 repeated bisections. 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. Finally, the overall solution is globally optimized.

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 (kMetis):

kMetis is a popular multilevel partitioning algorithm, developed by Karypis *et al* [14]. 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 Algorithms

Using the base algorithms with the three similarity metrics we discussed in the first subsection, we obtain nine 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 Coassociation-based Consensus

We construct an $n \times n$ Coassociation matrix to represent the fraction of co-occurence of each pair of nodes in the 9 sets of base clusters. The coassociation value in the matrix for 2 points v_i and v_j is given by

$$C(v_i, v_j) = \frac{\|\text{Clusters containing both } v_i \text{ and } v_j\|}{\|\text{Total number of clusters}\|}$$

The matrix can thus be treated as a similarity matrix, with the points that occur together in all the clusterings having the maximum score value of 1. We obtain the Coassociation matrix for all nine clusterings and apply the Agglomerative algorithm with the UPGMA (average link) metric to obtain a consensus clustering. Apart from this *Coassociation-based Agglomerative Consensus (CBAC)* Algorithm, we also implement a variant with the single link metric (CB-slink), similar to the one used by Fred it et al [9].

3.3.2 Hypergraph-based Consensus

Strehl *et al* [18] used a hypergraph representation with an $n \times m$ binary 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 y.

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

As we described earlier, they proposed 3 algorithms - CSPA, HGPA and MCLA based on this representation.

3.3.3 PCA-based Consensus

The main disadvantage with the Coassociation matrix representation is that it is quadratic and therefore computationally expensive for large datasets. Although a hypergraph is a better representation, clustering algorithms cannot be directly applied to it when the number of clusters is large. For instance, in our case, we have nine algorithm-metric combinations each producing k clusters. If the value of k is large, clustering the $9 \times k$-dimensional points would prove inefficient since distance metric computations do not scale well to high dimensions [2].

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 hypergraph matrix without compromising the information required for clustering. As we described above, each feature vector (row) in the hypergraph matrix corresponds to the cluster membership pattern of a node. Since we are using hard clustering algorithms, a node can occur only in 9 clusters. For large values of k, the binary feature vectors will be very sparse. Also, since the occurence 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 [11, 8, 17] 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. Traditional clustering algorithms can then be applied on this reduced representation without performance concerns, to obtain consensus clustering arrangements.

Accordingly, we convert the $9 \times k$ clusters into a hypergraph matrix and apply PCA to reduce the number of dimensions. We then apply two different consensus clustering algorithms on the PCA representation - the *Recursive Bisection (PCA-rbr) algorithm* and the *Agglomerative Hierarchical (PCA-agglo) algorithm*.

3.3.4 Soft Consensus Clustering

As we mentioned earlier, hub proteins are known to participate in several functions in the cell. By assigning the hub proteins to a single cluster each, we are inhibiting the number of functions that can be discovered. To overcome this problem, we construct a variant of the PCA-agglo consensus algorithm to perform soft clustering for hub proteins. To identify hub proteins, we use the degree information of the nodes, similar to our earlier work [22]. By the theory of preferential attachment [24], there exists a strong positive correlation between the degree of a node and the probability of other nodes forming edges to it. Since hubs have very high degrees, new proteins added to the PPI network are more likely to interact with hubs than with other nodes. We analyze the degree distribution of the PPI network and use a degree threshold [22] to identify 60 hub proteins. We then perform clustering using the agglomerative algorithm with the additional constraint that these hub nodes can be placed 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 [3, 23, 22, 28, 29]. This dataset is available from the Database of Interacting Proteins (DIP). It consists of 15147 interactions between 4741 proteins. From the degree distribution shown in Figure 1, it can be observed that the PPI dataset is scale-free with a skewed distribution. The value of the scale parameter α is -1.7911.

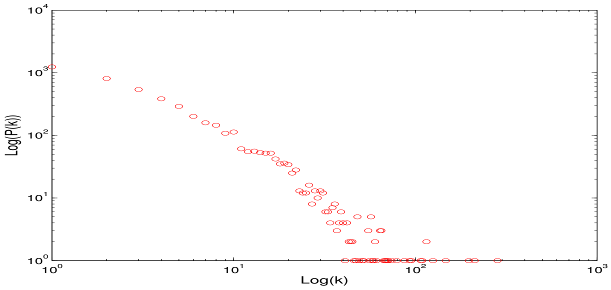


Figure 1. Degree distribution of the PPI network. k is the degree and $p(k)$ is the number of nodes with degree k.

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 [15]. 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* [18]. 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) [18] 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 [4] [2]. The Gene Ontology (GO) database provides three vocabularies of known associations - *cellular component (CC)* which refers to the localization of proteins inside the cell, *molecular function (MF)* which refers to shared activities at the molecular level and *biological process (BP)* 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 [23, 3, 22]. 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 significance of a group of proteins that share a GO term. 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 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 $cutoff$ 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. [4]

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

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

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 p-value for each of the three ontologies.

4.3 Experimental Results

We use the three graph clustering algorithms with the three topology-based metrics to obtain nine independent base clusterings each. Estimating the optimal number of clusters, k, is a serious issue in clustering. Earlier approaches [17] have suggested using the ratio between the inter-cluster and intra-cluster similarities to estimate the value. We used all three similarity metrics with the kMetis algorithm to estimate cluster quality for different values of k. Finally, one of the optimal values was chosen as the value of k. Accordingly, the value of k for the PPI dataset was chosen to be 100. The 9×100 clusterings obtained are then represented in the form of a hypergraph matrix 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 using three algorithms - the agglomerative hierarchical algorithm (PCA-agglo), the repeated bisections divisive algorithm (PCA-rbr) and the soft consensus (PCA-softagglo) algorithm. Apart from these three,

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

[3] As of May 2005, the GO database contains 7000 genes annotated with 1644 cellular component , 7502 molecular functions and 9706 biological processes.

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

we also implement the Coassociation based Agglomerative consensus (CBAC) algorithm, as discussed earlier. To compare with our techniques, we implement a Coassociation based single link agglomerative (CB-slink) algorithm, which was used by Fred *et al* [9, 20], the three algorithms proposed by Strehl *et al* [18] - CSPA, HGPA and MCLA and two algorithms - Balls and Agglomerative proposed by Gionis *et al* [10]. 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 the coassociation matrix, the agglomerative algorithm produced 1315 clusters and the Balls algorithm yielded 3494 clusters for the 4741 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.

4.3.1 Evaluation of Consensus Algorithms

Modularity and NMI: First, we compare the consensus algorithms in terms of the Modularity and Average Normalized Mutual Information scores. Figure 2 shows the comparative results in terms of both these metrics for 6 consensus methods. The Agglomerative and Balls algorithms, as we mentioned earlier, resulted in a large number of clusters, most of which contained only singletons and pairs. [5] Hence, the modularity and NMI scores were very low for these clusters and are not presented here.

It can be observed that the *CBAC, the PCA-agglo and PCA-rbr algorithms perform the best* with high scores in terms of both metrics, when compared to the other methods. The single link algorithm performs the worst overall. Although the MCLA algorithm performs better than HGPA and has a good NMI score, its performance in terms of modularity is worse than the PCA-based algorithms. Note that for the PCA-based methods, the number of dimensions is reduced from *900 to 50*. This makes their performance very impressive.

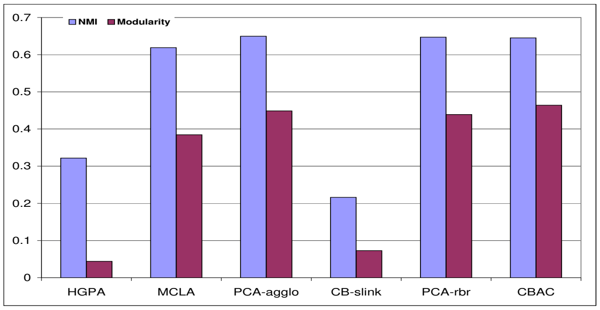


Figure 2. Modularity and NMI scores for consensus algorithms

Domain-based Evaluation: We proceed to evaluate the clusters obtained from the consensus algorithms using the domain-based metric. Figures 3a and 3b show the comparison in terms of the Clustering Score for the Biological Process, Molecular Function and Cellular Component ontologies. Note that in this case, *lesser values represent more meaningful clusters*. Also, a clustering score value of 0.05 represents the worst case, when all clusters obtained are insignificant. The *CBAC and PCA-based consensus methods* once again do better than all the other algorithms. The PCA-rbr algorithm provides the best clustering scores overall. The Balls and CB-slink algorithms have scores very close to 0.05. Although the CBAC algorithm performed well in both cases, clustering using a coassociation matrix is computationally quadratic. This makes it an inefficient and non-viable approach for large datasets. The PCA-based approach is more scalable and provides similar and even better quality clusters, in some cases. Its performance is admirable considering that we are reducing the number of dimensions by a factor of 18, *from 900 to 50*.

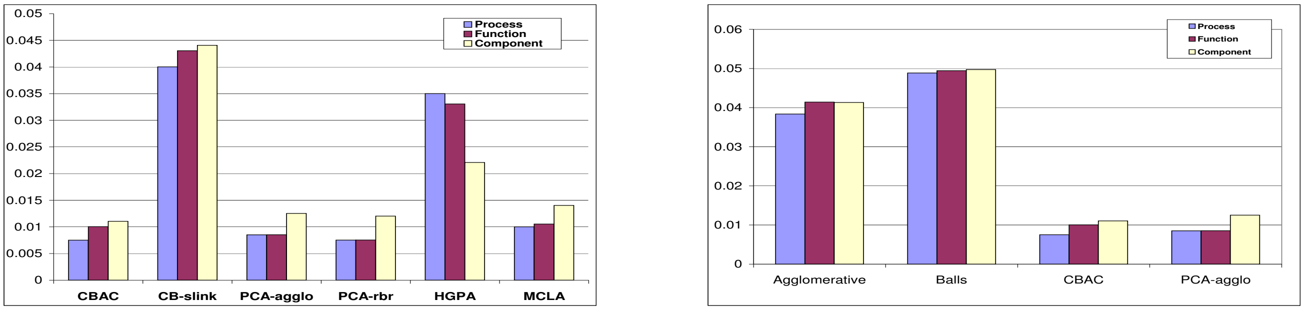
In the next experiment, we compare the number of significant clusters obtained, shown in Figure 4. Similar to the clustering score results, we find CBAC and PCA-agglo [6] having the largest percentages of significant clusters. As we mentioned earlier, the Agglomerative and Balls algorithms generate a large number of insignificant clusters.

Next, we further analyze the clusters obtained with the PCA-based consensus clustering. We consider the clusters obtained by the PCA-rbr algorithm. To emphasize the high quality of these clusters, we compare them against the MCLA algorithm. Figure 5 shows the comparison between the two algorithms, in terms of p-value distribution of the clusters obtained, for the biological process ontology [7]. The p-value distribu-

[5] 508 of the 1315 clusters produced by the Agglomerative algorithm contained singletons, and 337 contained pairs. For the Balls algorithm, 3204 of the 3494 clusters contained singletons.

[6] The PCA-rbr algorithm yielded similar number of significant clusters as the PCA-agglo method

[7] The plots for the other two ontologies follow similar trends



(a) (b)

Figure 3. Domain-based Clustering scores for consensus algorithms. Comparisons with a) MCLA and HGPA proposed by Strehl [18] and CB-slink proposed by Fred [9] b) Agglomerative and Balls algorithms proposed by Gionis *et al* [10]

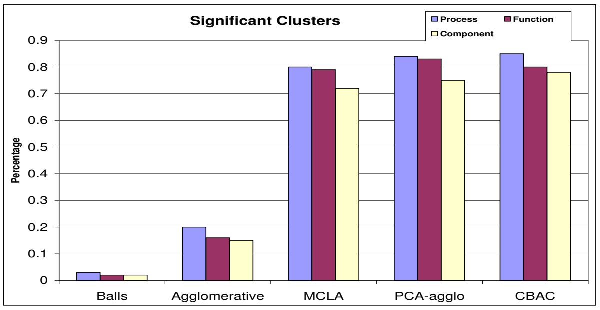


Figure 4. Percentage of Significant clusters obtained for consensus algorithms - PPI dataset

tion 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 78 significant clusters for the molecular function ontology whereas the PCA-rbr algorithm provides 83. The best cluster we obtain with PCA-rbr for this ontology has a p-value score of *5.8e-64*. The best scoring cluster for the MCLA algorithm has a much worse p-value score of 1.4e-40. 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 7000) 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.

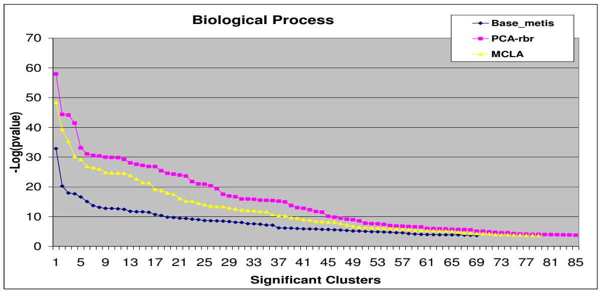


Figure 5. P-value distribution Comparison

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 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-64.

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

4.3.2 Soft Clustering

As we mentioned earlier, the hubs in PPI networks are believed to correspond to multi-functional proteins, which interact with different groups of proteins for different functions. To identify these different functions, we used the soft-clustering variant of the PCA-agglo

and have been omitted due to lack of space.

algorithm, which allows hub proteins to belong to multiple clusters. 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 polarity, 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. 6 out of the 9 base clusterings associate CKA1 with biological process term *transcription, DNA-dependent* with p-values ranging from e-05 to e-10. The metis base algorithm with the betweenness and neighborhood metrics associates CKA1 with biological process term *cell organization and biogenesis* with p-values of 1.81e-10 and 2.63e-12 respectively. A hard consensus clustering algorithm can only associate CKA1 with the most popular term. Accordingly, the pca-agglo consensus algorithm associates CKA1 in a large group associated with the biological process *transcription, DNA-dependent* with a much better p-value of *5.08e-13*. 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 *4 clusters* with significant biological process associations - *transcription, DNA-dependent* with p-value 1.20e-13, *cell organization and biogenesis* with p-value 6.36e-22, *cell ion homeostasis* with p-value 3.26e-19 and *regulation of transcription, DNA-dependent* with p-value 2.41e-10. Thus, we find that soft consensus clustering can lead to the discovery of multiple functionalities for hub 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 [22] 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 [8] is shown in Figure 6. It can be observed that the PCA-softagglo method consistently yields clusters with higher biological significance than the hub-duplication technique.

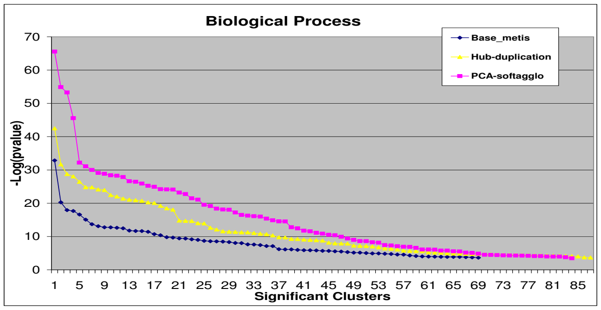


Figure 6. P-value distribution Comparison for Soft Clustering

We also conduct experiments comparing the performance of the base similarity metrics and clustering algorithms. For the metrics, the Betweenness metric performs the best and the Neighborhood metric, the worst. The Direct and rbr algorithms have similar modularity scores and outperform the Metis algorithm. For more details, please refer to our technical report [5].

4.3.3 Comparison of Similarity Metrics

In the next experiment, we compare the performance of the three similarity metrics that we employ to perform base clustering. We use the Modularity metric to make the comparison. For each metric, we take the average of the modularity score over all three base clustering algorithms for that metric. The results are presented in Figure 7. We find that the Betweenness metric provides the highest average modularity score and the Neighborhood metric performs the worst. As we mentioned earlier, the Betweenness and Clustering Coefficient metrics are defined only for nodes that have interactions between them while the Neighborhood metric can have non-zero values for two nodes even if they do not have an interaction between them (if they have common neighbors). The poor performance of the Neighborhood metric could be attributed to the fact that the metric tends to make the network more complex by creating artificial interactions between nodes that are not physically interacting.

4.3.4 Comparison of Base Clustering Algorithms

We now compare the performance of the three graph clustering algorithms used for base clustering. In this case, we take the average of the modularity scores over

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

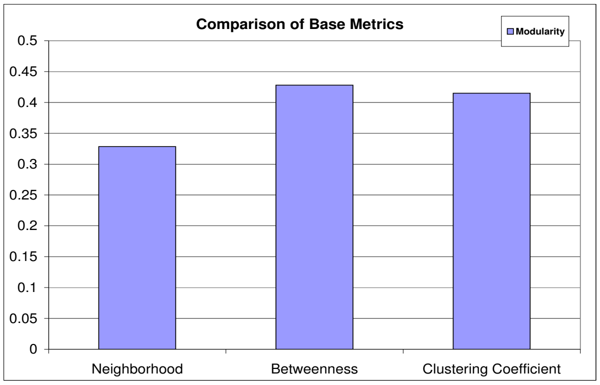


Figure 7. Modularity-based Comparison of Base Similarity Metrics

all three similarity metrics. Figure 8 shows the results for both datasets. The Direct and rbr algorithms have similar scores and outperform the Metis algorithm.

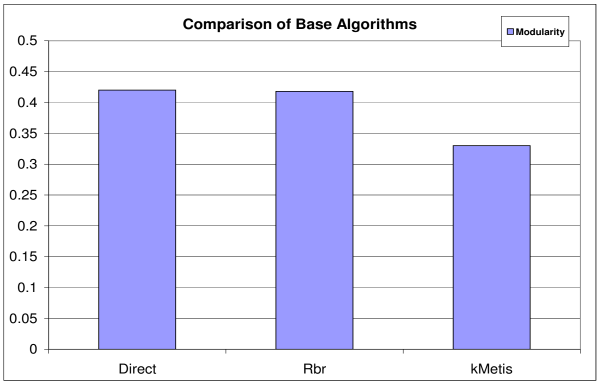


Figure 8. Modularity-based Comparison of Base Clustering Algorithms

5 Conclusion

In this paper, we have shown how an ensemble approach can be applied to effectively partition PPI networks. We have implemented three topological metrics using graph partitioning algorithms to reduce the effect of noise and obtain diverse base clusterings. We have presented several consensus algorithms including a PCA-based hypergraph approach, designed to scale to large datasets and a soft consensus clustering method, designed to discover multiple functional associations for hub proteins. The Coassociation-based Agglomerative Consensus (CBAC) Algorithm was found to perform well consistently, but the quadratic complexity of its representation is a serious disadvantage. The proposed PCA-based algorithms, apart from the scalability advantage, were found to lead to consensus clusters with high efficiency, in terms of all three metrics. We found that even with an 18x reduction in dimensionality, we were able to obtain clusters with high biological significance. Also, the PCA-based soft consensus clustering algorithm proved to be very effective in identifying multiple functionalities of hub proteins. In the future, we would like to extend the notion of ensembles to incorporate interactions obtained from multiple experimental and in-silico PPI networks.

References

[1] A. Abou-Rjeili and G. Karypis. Multilevel algorithms for partitioning power-law graphs. *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2006.

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

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

[4] M. Ashburner and *et al*. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nat Genet.*, 25(1):25–29, May 2000.

[5] S. Asur, S. Parthasarathy, and D. Ucar. An ensemble approach for clustering protein-protein interaction networks. *OSU Technical Report OSU-CISRC-9/06-TR73 - ftp://ftp.cse.ohio-state.edu/pub/tech-report/2006/TR73.pdf*, 2006.

[6] C. Brun, F. Chevenet, D. Martin, J. Wojcik, A. Gunoche, and B. Jacq. Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biology*, 5, 2003.

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

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

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

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

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

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

[14] 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*.

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

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

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

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

[19] A. Thomas, R. Cannings, N. A. M. Monk, and C. Cannings. On the structure of protein-protein interaction networks. *Biochemical Society Transactions*, 31:1491–1496, 2003.

[20] A. Topchy, A. K. Jain, and W. Punch. A mixture model for clustering ensembles. *in Proc. SIAM Conf. on Data Mining*, pages 379–390, 2004.

[21] 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.

[22] D. Ucar, S. Asur, U. Catalyurek, and S. Parthasarathy. Improving functional modularity in protein-protein interactions graphs using hub-induced subgraphs. *European Conference on Principles and Practice of Knowledge Discovery in Databases , PKDD*, 2006.

[23] D. Ucar, S. Parthasarathy, S. Asur, and C. Wang. Effective preprocessing strategies for functional clustering of a protein-protein interactions network. *IEEE International Symposium on Bioinformatics and Bioengineering, BIBE*, 2005.

[24] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Modeling of protein interaction networks. *Complexus*, 1:38, 2003.

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

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

[27] A. Y. Wu, M. Garland, and J. Han. Mining scale-free networks using geodesic clustering. *Conference on Knowledge Discovery in Data,Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 719 – 724, 2004.

[28] 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.

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