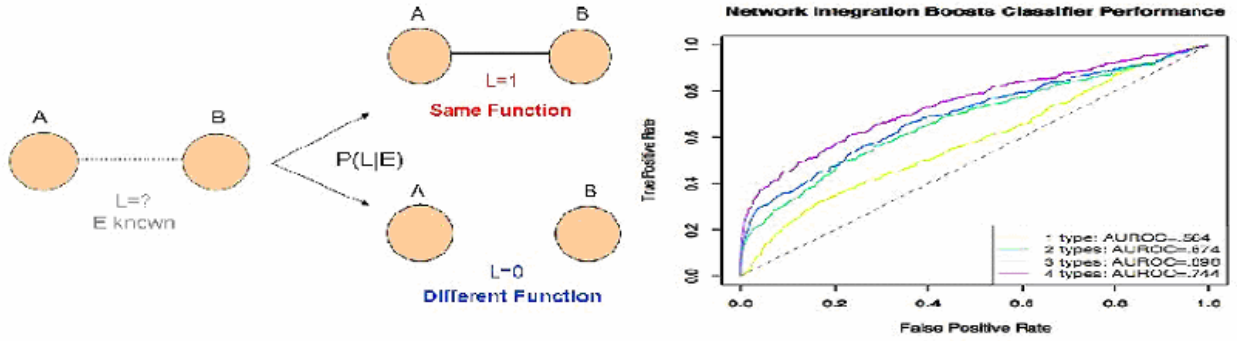
**Title:** ***Clustering and Segmentation of Probablistic Interaction Networks***
**Team Members:** ***Marcin Mejran, Dina Thomas***

## Introduction and Motivation

Protein-protein and protein-DNA interactions in the genome are modeled as interaction networks, and there are more than a dozen methods to detect these interactions. As a result at present there are a number of different interaction networks available for each sequenced organism. However even though most of these interaction predictors have been individually shown to predict experiment, the networks generated by different methods are often not superposable in any obvious way. This seeming paradox has simulated a burst of recent research in network integration. Integrating these different networks to arrive at a statistical summary of which proteins work together within a single organism can help detect linkages that would have been missed if only single predictor was used. It can also help strengthen the confidence of known linkages

By formulating the network integration problem as a binary classifier we can quantify the extent to which integration improves prediction accuracy over a single source. If two proteins have a shared functional category we say the link between them is labeled as L=1 and if they are in different functional categories the link is labeled L=0. The network integration problem is a binary classifier in high dimensional feature space as shown in Figure1(a)




1(a)binary classifier paradigm 1(b)ROC area for classifier

Figure 1(b) shows the area ROC curve for one data-set and shows that classifier performance increases monotonically as more data-sets are combined. Prof Serafim's group has developed an integration algorithm recently. The next step now is to extract information from these networks that have been integrated, potentially using this information to then create better integration and network comparison methods. There has been some success using regression methods on pairs of proteins, although other machine learning methods haven't been tested yet. The goal of this paper is to test clustering methods on these networks, and to extract information and summary statistics from them.

The data consist of protein interactions networks, which are represented as undirected graphs in which nodes represent proteins, edges represent interaction

probabilities. The nature of the edge weights biases clustering towards generating "correct" clusters. In our case, these desired clusters are those which group together proteins with similar function. This can allow us to find the function of unknown proteins based on what cluster they are in, and what other proteins are in that cluster. A potentially more important goal however is to find certain proteins which have important or otherwise interesting functions, from a biological point of view. These may include proteins that form the centers of a hub or which connect multiple together clusters. Such proteins can be thought of as vital to an organism's survival, and from a clustering point of view their removal is expected to greatly affect clustering results. We tested the reasonability of our clustering results using annotation information for each protein. The information used is a simple one line description which exists for roughly half the proteins. One of our main goals, besides finding clusters for our data, was to create a code base in R which can be used for future clustering of networks. We needed to take care so that our implementation is flexible as network characteristics differ from genome to genome. For example, some networks have more proteins with high probability linkages than others.

**Algorithms Implemented**

We implemented a number of clustering algorithms; hierarchical clustering, kmeans clustering and Markov clustering, and some extensions for these algorithms. For the actual clustering algorithms we mainly used pre-existing code from the R library "cluster" and the program MCL for Markov Clustering. The algorithms in the "cluster" library have the advantage of natively working on dissimilarity matrixes which is the form our data is in. The main goal as a result was to create a unified clustering mechanism in R so that all clustering could be performed at once.

The first algorithm we implemented was the traditional hierarchical clustering method, or rather a close cousin of it. We used the "agnes" and "diana" algorithms in the cluster library for R, which perform "divisive" and "agglomerative" clustering respectively. As the name implies the divisive algorithm, "diana", begins with one large cluster and recursively breaks it into smaller clusters based on the elements which are the furthest apart. Agglomerative on the other hand, use by "agnes", starts with many small clusters and then recursively combines them together into a tree. The second fundamental algorithms we applied are k-means and soft (fuzzy) k-means, or rather derivatives of them. As with hierarchical clustering we use method from the "cluster" library. "Pam" performs something very similar to k-means except that it uses medoids for cluster centers (which must be elements in the data set) and sums over dissimilarities instead Euclidian distances. "Clara" is an extension of "pam" designed to work with large data sets, which is the case for much of the data we use, by dividing the data into subsets of a fixed size. In soft k-means we do assign each data point to a cluster but instead we find the probability of each data point being in each cluster. "Fanny," like "pam," provides a soft k-means derivative for dissimilarity matrixes. We have also implemented the gap method for selecting the number of k-means clusters.

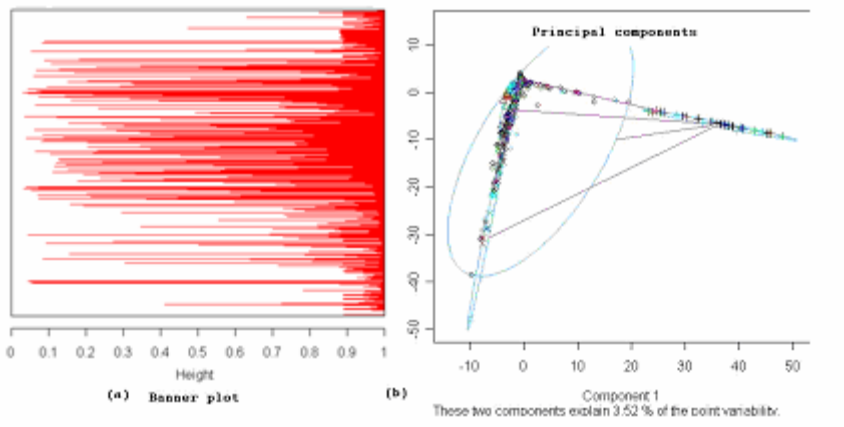
The last clustering algorithm we implemented was Markov Clustering (MCL) which works by simulating "flow" on a graph. This algorithm is based on the property that in a graph a random walk inside a dense cluster will visit many of the nodes before leaving the cluster. The basic idea is to simulate "flow" in a graph, promoting flow where the connections are strong and demoting it where they are weak, so that flow between clusters dies out but not within clusters. There are two phases in the algorithm: inflation and expansion. Expansion can be considered the flow going outwards into other areas, while inflation can be considered the strengthening/weakening of the flow within the structure. Mathematically the expansion is characterized by converting into a Markov graph and computing the powers of the associated stochastic Markov matrix. Inflation is performed by an entry-wise Hadamard-Schur product combined with diagonal scaling. We opted to use the MCL package for Markov Clustering instead of coding in R, as the package was optimized for large data sets (as some of ours are) which are known to cause problems (RAM usage) for un-optimized implementations. To characterize individual nodes in the network we implemented graph information algorithms like degree distribution, clustering coefficient and between-ness centrality. All these methods give an indication of which nodes can be potential hubs in the network. (See Glossary for definitions).

**Experiments**

We performed clustering on a number of data-sets. To extract only highly probable clusters we first threshold the network by removing edges with very low probability of interaction. Since dissimilarity matrixes need values for each entry we set the probabilities of interaction for these removed edges to 0 instead. The algorithms in general grouped biologically related proteins with good consistency, although there was a good amount of noise in the clustering. The threshold parameter is dependent on the data-set and was empirically determined. For the *Helicobacter pylori* data-set, the threshold was found to be 0.15 which gave good distribution of clusters i.e. not too many small clusters or a single large cluster.

**Observations**

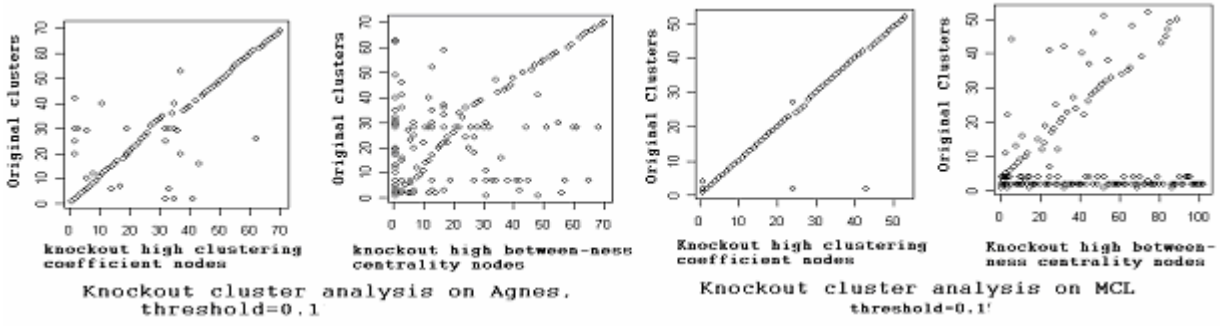
1. There are many weak linkages and very few strong ones, and the strong ones are disjoint. As a result if they are removed through the threshold, algorithms will create many clusters of single nodes.
2. There are a non-trivial number of nodes that are weakly connected to a large set of the remaining nodes, more than 70% in some cases. As a result a low threshold resulted in single large cluster, however as mentioned in point 1 a high threshold leads to many small clusters. These nodes have high between-ness centrality and low clustering coefficient, meaning that they are "hubs" between different sections of the graph but are not in any clique themselves. The effect of these nodes on clustering is seen in the plots below we obtained for a data-set. Plot (a) gives the banner plot which displays the hierarchy of clusters like a tree and it plots the distances at which members are merged. The concentration of bars at the right side indicates a lot of nodes are merged initially at the top of the hierarchy. Plot (b) gives the principal components as the ellipses. We can see concentration of points in one component.

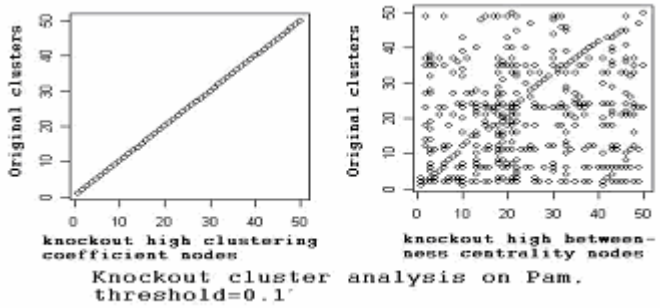



The existence of such proteins causes problems as they have a very large influence on clustering (see below). However are not very biologically significant for each cluster.

3. The stability of clusters for a node can be stated as its resistance to fragmentation when the node is deleted. A node that renders a cluster less stable is critical and gives the protein that is critical for the functioning of the biological network. These are the proteins that are essential for the survival of the organism. The detection of these proteins is critically dependent on the clustering method used as well as the threshold parameters.

To characterize critical nodes of the clusters, we knocked out certain nodes and performed clustering on the resulting network.
The scatter-plots for experiments on *Helicobacter pylori are* given below.




As seen from the plots above, nodes with high clustering coefficient do not affect the stability of the cluster in most cases. However nodes with high between-ness centrality, are found to be critical. It is also seen that MCL characterizes the critical nodes better

than Agnes or Pam, in some runs high clustering coefficient nodes also fragmented clusters.
4. MCL provides good clustering, and unlike the other methods can naturally work with removed edges. As a result it will place isolated proteins in their own clusters. The results are not always consistent, as the algorithm is not deterministic however in general it is quite sensitive to the removal of seemingly important proteins.

**Conclusions**

We were able to find many useful properties of such networks and the clustering of them. There are two sets of highly important nodes, those which are weakly connected to everything and those that are in well connected clusters. The former has a disproportionate influence on the clustering, meaning that many of the clusters may not be significant. At the same time, the removal of such nodes can lead to finding clusters which are separate and biologically significant. Markov clustering was found to create biologically significant clusters which and able to detect the removal of significant nodes. We were able to create a significant R codebase which can be used to run further analysis on these methods.

**Glossary**

1. *Between-ness centrality*: The between-ness centrality of a node *v* in a graph is the sum of the fraction of shortest paths between all pairs of nodes that pass through *v*.
2. *Clustering Coefficient*: Clustering coefficient of a node *v* having *n* neighbors is the ratio *N/(n*(n-1))* where *N* is the number of edges between the *n* neighbors.
3. *Pam (Partitioning around mediods)*: This finds k representative medoids from the data-set.
4. *Clara (Clustering Large Applications)*: This clustering method is used on large data-sets. The data is divided into sub-datasets of equal size and Pam is applied to each subset.
5. *Agnes* (*Agglomerative Nesting)*: This is a agglomerative hierarchical clustering algorithm.
6. *Diana (Divisive Analysis Clustering):* This is a divisive hierarchical clustering algorithm.

**References**

Markov Clustering: http://micans.org/mcl/
Cluster Package for R: http://cran.r-roject.org/src/contrib/Descriptions/cluster.html
Hierarchical cluster visualization: http://citeseer.ist.psu.edu/koren03twoway.html
K-mean gap statistic: http://citeseer.ist.psu.edu/tibshirani00estimating.html
Between-ness centrality: http://www.biomedcentral.com/1471-2105/6/213