# A Two-Step Approach for Clustering Proteins based on Protein Interaction Profile [*]

Pengjun Pei and Aidong Zhang
Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260
{ppei, azhang}@cse.buffalo.edu

## Abstract

*High-throughput methods for detecting protein-protein interactions (PPI) have given researchers an initial global picture of protein interactions on a genomic scale. The huge data sets generated by such experiments pose new challenges in data analysis. Though clustering methods have been successfully applied in many areas in bioinformatics, many clustering algorithms cannot be readily applied on protein interaction data sets. One main problem is that the similarity between two proteins cannot be easily defined. This paper proposes a probabilistic model to define the similarity based on conditional probabilities. We then propose a two-step method for estimating the similarity between two proteins based on protein interaction profile. In the first step, the model is trained with proteins with known annotation. Based on this model, similarities are calculated in the second step. Experiments show that our method improves performance.*

## 1 Introduction

Proteins seldom act alone; rather, they must interact with other biomolecular units to execute their function. An examination of these interactions is essential to discovering the biological context of protein functions and the molecular mechanisms of underlying biological processes.

Recently, new approaches have been developed for a genome-wide detection of protein interactions. Studies using *yeast two-hybrid system (Y2H)* [14, 25, 11, 16] and *mass spectrometric analysis (MS)* [10, 13, 24] have generated large amounts of interaction data. A protein interaction network (PIN) [4] can be constructed from existing protein-protein interaction data by connecting each pair of vertices (proteins) involved in an interaction. With the ever-growing size of the interaction network, computational methods to break them into smaller and more manageable modules are in great demand. These modules are expected to reflect biological processes and pathways. We can also get some hints on the function of uncharacterized proteins by looking at other known proteins in the same module. Clustering proteins based on the protein interaction network provides a natural solution to this problem.

This paper will identify and investigate the problem of how to define the similarity between two proteins for clustering algorithms that use similarity matrix as input. Firstly, we propose a novel definition of similarity measurement based on conditional probabilities. Then we propose a model for predicting the conditional probability. In our two-step approach of estimating the probability, we use annotated proteins available to train our model, followed by probability prediction based on the model. Our method is very thin-supervised because the properties of the annotated proteins are captured in the few parameters in our model and we do not force any constraint on whether two proteins should stay in the same cluster. Our experiments show that our new similarity measurement outperforms current available measurement. Finally, we conclude the paper and propose some future work.

## 2 Related Work

Clustering algorithms have been widely applied in dealing with large data sets in bioinformatics, including gene expression data analysis [8], DNA and protein sequence data analysis [12]. These algorithms are shown to be capable of grouping similar objects and detecting the underlying patterns. Many successful clustering algorithms in bioinformatics domain such as agglomerative hierarchical clustering [8], CAST [2] and CLICK [22] require an input of a

[*] This research was partially supported by National Science Foundation Grants DBI-0234895, IIS-0308001 and National Institutes of Health Grant 1 P20 GM067650-01A1

similarity or distance matrix. However, a protein interaction data set is represented in a different format. We do not have a straightforward similarity definition for two proteins in the data set. Though some approaches like superparamagnetic clustering (SPC) and optimization Monte Carlo algorithms in [23] have been proposed to cluster proteins directly based on the network, we expect more choices of clustering algorithms, especially for those that were successfully used in related domains. To apply those clustering algorithms, the similarity definition for two proteins is essential for effectively analyzing the data.

In [20], the similarity between two proteins is defined by the significance of neighborhood sharing of two proteins. Based on this similarity measurement, a hierarchical clustering method is applied to find protein clusters. However, how this measurement value can be related to protein functional similarity is still unknown. Also, as this similarity value is non-zero only for those protein pairs that share at least one neighbor, many protein pairs that have direct connections (interactions) will also get a similarity value of zero.

Meanwhile, some function prediction methods have been applied to map proteins onto known functional categories given some annotated proteins in the network. In [15], a binomial model of local neighbor function labelling probability is combined with a Markov Random Field (MRF) propagation algorithm to assign function probabilities for proteins. In [6], another MRF-based method is proposed. It considers a given network and the protein function assignments as a whole and scores the assignment accordingly. In [3], authors analyze previous MRF-based methods and propose a novel machine-learning method. As these methods require large amounts of training data and are limited to known function categories, they can not replace explorative clustering methods especially for organisms that our knowledge is quite limited. Meanwhile, as their algorithms are designed to predict function given a sufficient large number of annotated proteins, their models are not fully suited for our objective.

In this paper, by incorporating very limited annotation data, we provide a theoretically sound framework to define the similarity between two proteins. We expect this work will provide more choices for researchers in exploring protein interaction data.

# 3 Method

In this section, we propose using conditional probability to define the similarity between two proteins based on their protein interaction profiles. We then propose a model to define the conditional probability. Based on the model, we introduce a two-step method for calculating the conditional probability. In the first step, the model is trained with known protein annotation. Then based on this model, the similarities are calculated in the second step.

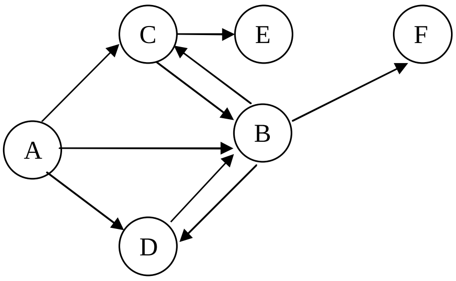


**Figure 1. Function Propagation from Source Protein $A$ to Other Proteins in the Network**

## 3.1 Novel Similarity Definition

As observed in [21], two interacting proteins show high homogeneity in annotation. In [27], it is observed that the farther away two proteins are in the network, the less homogenous they are. Therefore, we can regard the edges in the network as a means of message passing. Each protein tries to propagate its function to neighboring proteins. Meanwhile, each protein receives these function messages from its neighboring proteins to decide its own function. The final probability of a protein having a specific function is therefore a conditional probability defined on its neighbors' status of having this function annotation.

Figure 1 illustrates function propagations using one single protein $A$ as the source of information. $A$'s function is propagated towards its direct neighbors and then its indirect neighbors. In this process, the message will fade with the increase of distance (the length of path). E.g. its function is propagated to protein $B$ via paths $A \rightarrow B$, $A \rightarrow C \rightarrow B$, and $A \rightarrow D \rightarrow B$. Therefore, for the protein $B$, it receives messages from all these paths and shows a certain degree of function homogeneity with the source protein $A$. Protein $C$ also propagates its function to $E$. Protein $B$ propagates its function to proteins $C$, $D$, and $F$. Though the protein interaction network is undirected, the information flow from one vertex (source vertex) to another (sink vertex) can be conveniently represented by a directed graph. Throughout the paper, we use *protein* and *vertex* interchangeably and use $A$ to represent the *source vertex* and $B$ to present the *sink vertex*. $|P|$ is used to denote the total number of vertices (proteins) in the network.

For a certain functional label in consideration, we denote the probability of $A$ having this function as $P(A)$. As the result of the function propagation, $B$ shows certain degree of similarity with $A$'s function. $B$'s probability of having this function by propagation using $A$ as the information source can then be represented as a conditional probability $P(B|A)$. This conditional probability gives the capability of $A$'s function being transferred to $B$ via the network. Larger $P(B|A)$ value predicates closer function homogeneity and therefore higher similarity.

This measurement, however, is not symmetric, i.e., generally, $P(A|B) \neq P(B|A)$. Therefore, we define the similarity between proteins $A$ and $B$ as the product of two conditional probabilities:

$$Similarity_{AB} = P(A|B) * P(B|A). \qquad (1)$$

This measurement reflects the cohesiveness of two proteins' functions.

### 3.2 Model for Predicting Conditional Probability

For the function annotation of the sink protein $B$, its probability of having a certain function is determined by all the messages it receives from its neighbors. We call the message that favors this function annotation a *positive message*. For a protein that has a functional annotation with probability higher than a random protein in the network, it can propagate a positive message to its neighbors. When we consider the sink protein, it will also receive messages from other neighboring proteins. Therefore, the strength of homogeneity will depend on both the sum of positive messages towards the vertex, denoted as $PM$, and the degree of the vertex, denoted as $D$. Then the probability of a vertex having a specific function can be expressed as a function of these two values. Similar to [3], we can use a potential function $U(x; PM, D)$ to express this probability:

$$P(x|PM, D) = \frac{e^{-U(x;PM,D)}}{Z(PM, D)}, \qquad (2)$$

where $x$ is a binary value $x \in \{0, 1\}$ with 1 indicating the protein has the function under consideration. The normalization factor $Z(PM, D)$ is the summarization over all configurations :

$$Z(PM, D) = \sum_{y=0,1} e^{-U(y;PM,D)}.$$

We also adopt a linear combination of variables:

$$U(x; PM, D; \alpha) = (\alpha 0 + \alpha 1 * PM + \alpha 2 * D) * x.$$

We choose this model instead of the binomial-neighborhood model in [15] because the latter assumes that the neighbors of a vertex behave independently (the probabilities of taking a function are independent). As the clustering algorithm is designed to find dense part of the protein interaction network, our similarity measurement must handle such situations. In this case, the independence assumption is problematic [3].

Though our model is similar to the model in [3], our major interest lies in defining the similarity between two proteins and therefore (a) we always treat only one single protein as annotated protein, and (b) we consider proteins beyond direct neighbors of the source protein.

### 3.3 Iterative Function Propagation

For each protein $B$ that is connected (either directly connected or indirectly connected via some intermediary proteins) with the protein $A$, we can have a layer associated with it which is the shortest path length between the two proteins, denoted as $Dist(A, B)$. We use $N^{(k)}(A)$ to denote the set of proteins whose shortest path length to $A$ is $k$:

$$N^{(k)}(A) = \{B|Dist(A, B) = k\}.$$

We abbreviate $N^{(1)}(A)$ as $N(A)$. We call a protein $B \in N^{(k)}(A)$ a $k$-step neighbor of $A$.

Given a source protein $A$, we iteratively calculate the conditional probability of all other proteins's having the function annotation: we start from protein $A$'s direct neighbors and calculate the conditional probability for them. Then using the conditional probability of these direct neighbors of $A$, we calculate the conditional probability of the direct neighbors of these direct neighbors, i.e., $A$'s 2-step neighbors. This iteration continues until we get a conditional probability for each protein that is connected with $A$.

Having introduced the order for conditional probability estimation, we define the positive message in Equation (2) as follows:

We start from proteins belonging to $N^{(1)}(A)$, i.e, the direct neighbors of $A$. As each protein $B$ of this layer is directly connected with the source protein equally, we omit this direct connection message $A \rightarrow B$ and consider only the messages for its same-layer neighbors. Therefore, we can use the number of shared neighbors between $A$ and $B$ as the value of positive messages for protein $B$.

For a general case of a protein $B$ belonging to $N^{(k)}(A)$ with $k > 1$, we only regard those messages from its neighbors that belongs to $N^{(k-1)}(A)$ as positive. Each protein in layers less than $k - 1$ will have to propagate its information to proteins in $N^{(k-1)}(A)$ to affect the function annotation. Therefore, this information has already been captured in $A$'s $(k - 1)$-step neighbors. The messages from the same layer proteins, as we will show in the experiment part, are generally weak for $k > 1$ and therefore, omitted. The positive messages can be expressed as the sum of the product of two conditional probabilities:

$$PM_{B \leftarrow A} = \sum_{C \in N(B) \cap N^{(k-1)}(A)} P(B|C) * P(C|A), \qquad (3)$$

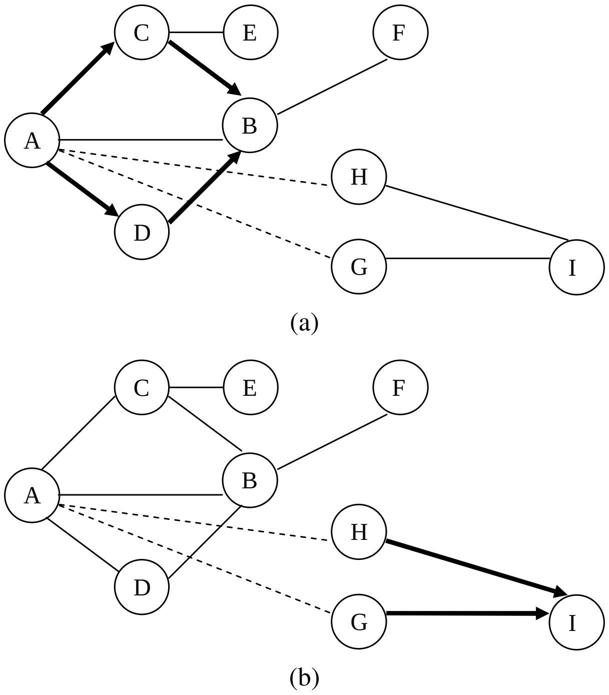


**Figure 2. Iterative Estimation of Conditional Probabilities**

where we use $PM_{B \leftarrow A}$ to explicitly represent that the positive message is from source $A$ to sink $B$ via the network.

The product of two conditional probabilities $P(B|C) * P(C|A)$ measures the probability of $A$'s function being successfully propagated to $B$ via the path $A \rightarrow ... \rightarrow C \rightarrow B$. Summing up all these probabilities over proteins that are both $B$'s direct neighbors and $A$'s $(k-1)$-step neighbors gives the strength of the message propagation from $A$ to $B$ via the network. As from the previous $k-1$ steps, we have already estimated the conditional probabilities $P(Y|X)$ for each $X$ and $Y \in \bigcup_{i=1,..k-1} N^{(i)}(X)$, we have the conditional probabilities $P(B|C)$ and $P(C|A)$ available.

Figure 2 gives an example of our function propagation process. We use vertex $A$ as the source and start with estimating the conditional probability of its direct neighbors: $P(B|A)$, $P(C|A)$, and $P(D|A)$. Figure 2(a) shows our function propagation messages from $A$ to $B$ in layer one. The messages towards vertex $B$ are marked in dark lines, which are the messages from vertices $C$ and $D$. Figure 2(b) shows our function propagation from $k$-step neighbors $H$ and $G$ to a $(k+1)$-step neighbor $I$.

After we calculate the value of positive messages, we can use Equation (2) to get the representation of the probability.

### 3.4 Two-Step Method for Similarity Estimation

From our definition above, we get both the representation of the conditional probability for each two vertices in the graph and the order of estimating the probability. However, this probability is not a numerical value yet. Instead, it is represented as a function of the parameters $\alpha$ in our model. In this section, we use a two-step method to estimate parameters and calculate the conditional probabilities.

In most organisms, there are some annotation data for proteins. Therefore, we have some training samples with known $x_i$, $PM_i$, and $D_i$ values.

In the ***first step (model training step)***, we use these training samples to estimate the parameters ($\alpha$) that maximize the joint probability:

$$P = \prod_i P(x_i | PM_i, D_i).$$

We use the simplex method (the Nelder-Mead algorithm) [19] to estimate these parameters. To get a comparatively accurate estimation, we estimate these parameters separately for each layer.

In the ***second step (conditional probability estimation step)***, the numerical values of the conditional probabilities are calculated using Equation (2) and the parameters ($\alpha$) estimated in the previous step.

### 3.5 Pruning Algorithm to Reduce Space and Time Complexity

Our method above estimates the conditional probability for each pair of vertices in the network and therefore, a total number of $|P| * (|P| - 1)$ probabilities must be calculated. However, this is unnecessary because for most pairs, the function propagation is very weak and the conditional probability value is very close to that of a random pair. Therefore, we make an improvement of the above method by adopting two pruning heuristics:

a. Positive message-based pruning: In this type of pruning, we try to identify non-promising pairs before representing its probability using Equation (2). After we calculate the positive message towards a vertex, we test whether the positive message is too low. Generally, there are a large number of such pairs. The probabilities of these pairs are expected to be comparatively low while the variance inside is very small. Therefore we simply combine them into a separate set $LowSet$. Then we estimate the probability of the data set $LowSet$ using annotated protein pairs in the data set. We use this probability estimate for all such pairs without going through the normal two-step training and estimation process.

b. Probability-based pruning: After we calculate each conditional probability, we prune the further propagation of non-promising vertices. Initially, we can estimate the conditional probability of two random vertices and we denote this value as $randPrb$. With the information degradation in message passing, when one vertex' conditional probability is only slightly higher than $randPrb$, it will send very weak positive messages to the farther neighbors. Therefore, when we meet a vertex $B$ that has the conditional probability $P(B|A) < randPrb + \epsilon$, we stop the propagation from this vertex $B$ as if it does not have any connection towards one more step vertices.

The whole algorithm is presented in Figure 3.

```
Algorithm ConditionalProbabilityEstimation()
Input: PIN, protein annotation data
Output: conditional probability set $CPS$ and $randPrb$
1. Estimate $randPrb$ using annotated proteins
2. for all vertex $A$ do
3.     $N^{(0)}(A) \leftarrow \{A\}$
4. for $k \leftarrow 1$ to $|P| - 1$ do
5.     $LowSet \leftarrow \phi$
6.     for all vertex $A$ do
7.         Get $N^{(k)}(A)$ from $N^{(k-1)}(A)$
8.         for all vertex $B \in N^{(k)}(A)$ do
9.             Calculate $D(B)$ and $PM_{B \leftarrow A}$
10.            if $(PM_{B \leftarrow A} < PM_{threshold})$ then
11.                $LowSet \leftarrow LowSet \cup \{(A, B)\}$
12.            else
13.                Represent $P(B|A)$ using Equation (2)
14.    Estimate parameters $\alpha$ using annotated proteins
15.    Estimate average probability $P_{low}$ for $LowSet$
16.    $stopPropagation \leftarrow true$
17.    for all vertex $A$ do
18.        for all vertex $B \in N^{(k)}(A)$ do
19.            if $(A, B) \in LowSet$ then
20.                $P(B|A) \leftarrow P_{low}$
21.            else
22.                Calculate $P(B|A)$ using Equation (2)
23.            if $P(B|A) < randPrb + \epsilon$ then
24.                $N^{(k)}(A) \leftarrow N^{(k)}(A) - \{B\}$
25.            if $P(B|A) > randPrb$ then
26.                $CPS \leftarrow CPS \cup \{< A, B, P(B|A) >\}$
27.        if $N^{(k)}(A) \neq \phi$ then
28.            $stopPropagation \leftarrow false$
29.    if $stopPropagation = true$ then
30.        break
31. return $CPS$ and $randPrb$
```

**Figure 3. Algorithm for Conditional Probability Estimation.**

We use the protein interaction network and protein annotation data as input. The output is a conditional probability set $CPS$ and the $randPrb$ probability value. For each significant conditional probability $P(B|A)$, we put $< A, B, P(B|A) >$ into the set $CPS$. For the rest of conditional probabilities, we approximate their values by $randPrb$. The pseudo code from Step 5 to Step 14 is the model training phase, while from Step 15 to Step 28 is the probability estimation phase, and Steps 10 and 23 are our positive message-based and probability-based pruning techniques, respectively. Though we iterate $k$ from 1 to $|P| - 1$ in Step 4, the iteration will most likely end early at Step 30 when none of the vertices has farther neighbors to propagate messages.

## 4 Experiments and Results

In this section, we compile several data sets and construct our protein interaction network. We analyze the network and find some properties on annotation transferring. Then we show that our estimated conditional probabilities correspond well with the real probabilities. Also, using a hierarchical clustering algorithm, we compare our method with previous measurement based on neighbor-sharing.

### 4.1 Data Sets

We compiled four data sets of yeast protein interactions:

**Table 1. Data Sets of Protein-Protein Interactions.**

| Data Set | Interactions | Proteins |
|---|---|---|
| Ito | 4392 | 3275 |
| DIPS | 3008 | 1586 |
| Uetz | 1458 | 1352 |
| MIPS4 | 788 | 469 |
| Combined | 9049 | 4325 |

Table 1 includes the four data sets we used for our experiments: Ito data set is the “full” data set by Ito et al. [14], DIPS data set is the set of yeast interactions in DIP [26] database that are generated from small-scale experiments, Uetz data set includes published interactions in [25] and unpublished interactions on their website[1], and MIPS4 data set includes four data sets [24, 18, 7, 9] deposited in MIPS [17]. Combining these together, we construct a protein interaction network with 4325 proteins and 9049 interactions.

For protein annotation, we use “subcellular localization” and “cellular role” in YPD [5] as protein localization and protein function annotation, respectively.

## 4.2 Statistics of the Data

Firstly, we investigate the relationship between the conditional probability and the shortest path length. In [27], two proteins with short distance are shown to be likely to share function and this function homogeneity degrades very fast with the increase of shortest path length. Similar result is observed in our experiments. Also, we calculate the conditional probability of random protein pairs and list result in Table 2.

**Table 2. Conditional Probability vs Distance.**

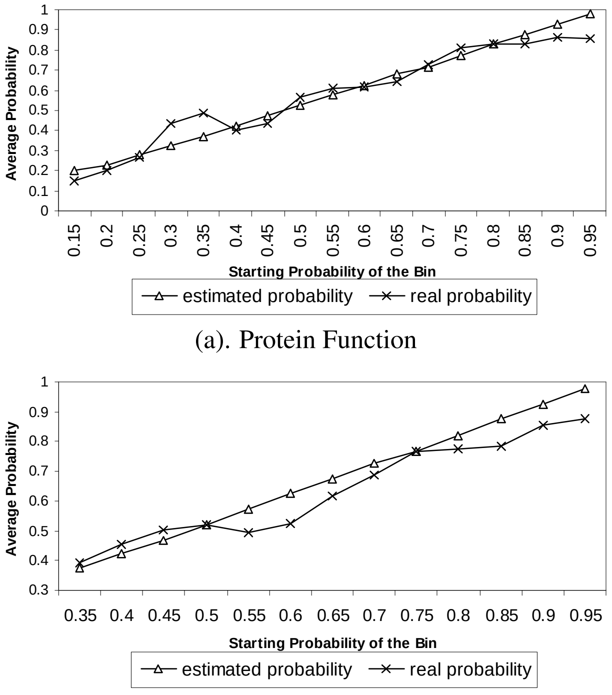
| distance | function | localization |
|---|---|---|
| 1 | 0.466739 | 0.601983 |
| 2 | 0.168622 | 0.356103 |
| 3 | 0.116772 | 0.312443 |
| 4 | 0.0974867 | 0.276011 |
| Random | 0.0982235 | 0.257639 |

Table 2 shows the fading of annotation transferring with the increase of distance. Direct connections strongly affect one protein's annotation, while indirect neighbors show a much weaker influence. Proteins more than three steps away are generally negligible. Therefore, paths with length greater than three are very weak in affecting a protein's annotation. This justifies our calculation of positive messages for a $k$-step vertex using only $(k-1)$-step neighbors except for 1-step vertices, in which case we use the same layer propagation. Also, this property of the data guarantees that our pruning method will dramatically reduce the computational complexity. In our experiments, very rarely do we need to calculate conditional probabilities four steps away.

## 4.3 Effectiveness of Conditional Probability Predicted

To validate the probabilities estimated by our model, we bin the protein pairs based on their predicted conditional probabilities. We bin them into bins with a span of 0.05 in probability. We calculate the *real probability* of the bin and the *estimated probability* of it. The real probability is the average probability of the bin estimated using known protein annotation in the bin without going through our predicting process. The estimated probability is the average predicted probability of the pairs using our method. The result is shown in Figure 4.

Figure 4 shows that our predicted probabilities are very close to real probabilities. Therefore, we conclude that our model can effectively predict the conditional probabilities.



(a). Protein Function

(b). Protein Localization

**Figure 4. Effectiveness of Conditional Probability Estimation.**

## 4.4 Comparing with the Unsupervised Method

We compare our similarity definition with the similarity defined in [20]. The previous method computes the P-values for all protein pairs: It defines the distance between two proteins $A$ and $B$ as the P-value of observing the number of shared neighbors under the hypothesis that neighborhoods are independent. The P-value, denoted as $PV_{AB}$, is expressed as:

$$PV_{AB} = \sum_{i=|N(A)\cap N(B)|}^{\min(|N(A)|,|N(B)|)} \frac{\begin{pmatrix} |N(A)| \\ i \end{pmatrix} \times \begin{pmatrix} |P|-|N(A)| \\ |N(B)|-i \end{pmatrix}}{\begin{pmatrix} |P| \\ |N(B)| \end{pmatrix}}.$$

When merging two subclusters, geometric means of two individual P-values are used for the new group's P-value. Therefore, if we define the similarity $Similarity_{AB}$ between proteins $A$ and $B$ as:

$$Similarity_{AB} = -\log(PV_{AB}),$$

then arithmetic means of two individual similarities can be used to define new similarity values when merging clusters. Therefore, the transformed algorithm based on standard *UPGMA (Unweighted Pair Group Method with Arithmatic Mean)* is equivalent to the original algorithm.

As our main focus here is to find a good similarity measurement for clustering methods that require a similarity matrix, we apply the same hierarchical clustering method, i.e., *UPGMA* and compare the result. After creating the dendrogram, we cut at various levels of the tree to get multiple clusters. We treat each cluster as a set of predictions for the homogeneity of two proteins' annotation: each protein shares its function (localization) with each another protein in the cluster. Therefore, we make $M*(M-1)/2$ predictions for a cluster of size $M$. Then we calculate the number of predictions and the precision of the predictions. Using different cutoffs, we can compare the precision of different methods at various numbers of predictions in a *Number-Of-Predictions* vs *Precision* plot.

To make a fair comparison, we use randomly selected 20% annotated proteins as our training set and treat the rest 80% annotated proteins as 'unknown' in our clustering process. Then we exclude those proteins pairs that are both among the annotated 20% training proteins in the resulting predicted protein pairs. Therefore, the previously known annotation relationships in the clustering process are *not* counted in our testing process. Making too few predictions will severely limit the power of the prediction while making too many predictions will bring a large number of false positives. Here, we choose the number of predictions at every 1000 interval up to 40000 and present the result in Figure 5.

Figure 5 shows that our method almost always outperforms the method using P-value as similarity measurement and therefore, the effectiveness of our method.

As we used only 20% annotated proteins for training and we did not force any two proteins into a certain cluster, we conclude that our method is "light" supervised.

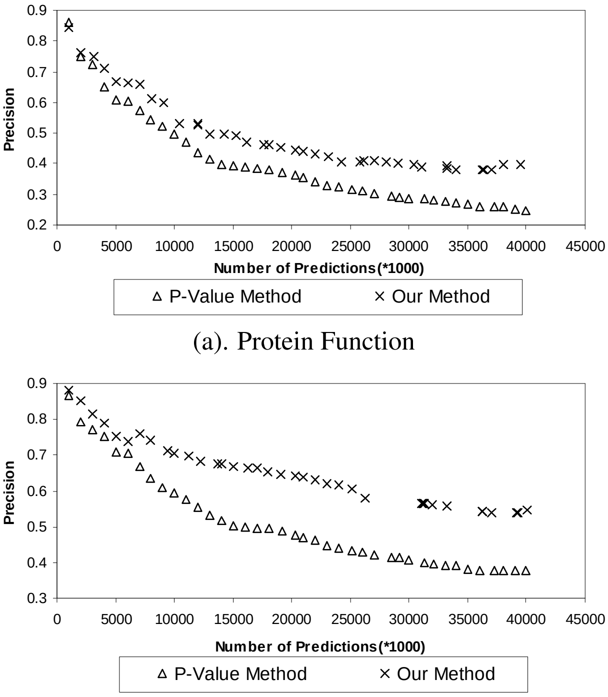
We list here the estimated $\alpha$ values of the first layer for protein function:

$$\alpha = (1.1678, -1.2529, 0.00456244).$$

We observe that $\alpha 1 < 0$ and $\alpha 2 > 0$, i.e., the conditional probability is positively related to the positive messages and negatively related to its degree. Also, we observe that positive messages have much more effect than the degree on the final probability, i.e., $|\alpha 1| > |\alpha 2|$. As for the first layer (direct neighbors), the positive message from $A$ to $B$ is the number of shared neighbors between $A$ and $B$, these observations agree well with the P-value-based method in [20]. Comparatively, our method is shown to be more accurate in representing the relationship between different factors.

## 5 Discussion and Future Work

We have presented a model for systematically defining the similarity between two proteins based on protein interaction profile. Then we used a two-step approach to build the model and calculate the similarity. To speed up calculation, we exploit the property of the protein interaction network and propose two pruning heuristics. Experiments show the advantage of our method comparing with previous work.



(a). Protein Function

(b). Protein Localization

**Figure 5. Comparison of The Clustering Result Using Different Similarity Measurements.**

Besides hierarchical clustering algorithms, many other clustering algorithms have been proposed to effectively mine large scale biological data sets. The measurement proposed here can also be used in these clustering algorithms. We plan to investigate the performance of various clustering algorithms.

Although our model for conditional probability definition is shown to be effective, it is still a simplified model for the complex biological system. In the future, we will explore different kinds of models.

## References

[1] http://depts.washington.edu/sfields/yplm/data/new2h.html.

[2] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6:281–297, 1999.

[3] C. Best, R. Zimmer, and J. Apostolakis. Probabilistic methods for predicting protein functions in protein-protein inter-

action networks. In *German Conference on Bioinformatics*, 2004.

[4] P. Bork, L. J. Jensen, C. von Mering, A. K. Ramani, I. Lee, and E. M. Marcotte. Protein interaction networks from yeast to human. *Curr Opin Struct Biol*, 14:292–299, 2004.

[5] M. C. Costanzo, J. D. Hogan, M. E. Cusick, B. P. Davis, A. M. Fancher, P. E. Hodges, P. Kondu, C. Lengieza, J. E. Lew-Smith, C. Lingner, K. J. Roberg-Perez, M. Tillberg, J. E. Brooks, and J. I. Garrels. The yeast proteome database (ypd) and caenorhabditis elegans proteome database (wormpd): comprehensive resources for the organization and comparison of model organism protein information. *Nucleic Acids Res*, 28:73–76, 2004.

[6] M. Deng, S. Mehta, F. Sun, and T. Chen. Inferring domain-domain interactions from protein-protein interactions. *Genome Res.*, 12:1540–1548, 2002.

[7] B. L. Drees, B. Sundin, E. Brazeau, J. P. Caviston, G. C. Chen, W. Guo, K. G. Kozminski, M. W. Lau, J. J. Moskow, A. Tong, L. R. Schenkman, r. McKenzie, A., P. Brennwald, M. Longtine, E. Bi, C. Chan, P. Novick, C. Boone, J. R. Pringle, T. N. Davis, S. Fields, and D. G. Drubin. A protein interaction map for cell polarity development. *J Cell Biol*, 154:549–571, 2001.

[8] M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A.*, 95:14863–14868, 1998.

[9] M. Fromont-Racine, A. E. Mayes, A. Brunet-Simon, J. C. Rain, , A. Colley, I. Dix, L. Decourty, N. Joly, F. Ricard, J. D. Beggs, and P. Legrain. Genome-wide protein interaction screens reveal functional networks involving sm-like proteins. *Yeast*, 17:95–110, 2000.

[10] A. C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, A. M. Michon, C. M. Cruciat, M. Remor, C. Hofert, M. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M. A. Heurtier, R. R. Copley, A. Edelmann, E. Querfurth, V. Rybin, G. Drewes, M. Raida, T. Bouwmeester, P. Bork, B. Seraphin, B. Kuster, G. Neubauer, and G. Superti-Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.

[11] L. Giot, J. S. Bader, C. Brouwer, A. Chaudhuri, B. Kuang, Y. Li, Y. L. Hao, C. E. Ooi, B. Godwin, E. Vitols, G. Vijayadamodar, P. Pochart, H. Machineni, M. Welsh, Y. Kong, B. Zerhusen, R. Malcolm, Z. Varrone, A. Collis, M. Minto, S. Burgess, L. McDaniel, E. Stimpson, F. Spriggs, J. Williams, K. Neurath, N. Ioime, M. Agee, E. Voss, K. Furtak, R. Renzulli, N. Aanensen, S. Carrolla, E. Bickelhaupt, Y. Lazovatsky, A. DaSilva, J. Zhong, C. A. Stanyon, J. Finley, R. L., K. P. White, M. Braverman, T. Jarvie, S. Gold, M. Leach, J. Knight, R. A. Shimkets, M. P. McKenna, J. Chant, and J. M. Rothberg. A protein interaction map of drosophila melanogaster. *Science*, 302:1727–1736, 2003.

[12] V. Guralnik and G. Karypis. A scalable algorithm for clustering sequential data. In *ICDM*, pages 179–186, 2001.

[13] Y. Ho, A. Gruhler, A. Heilbut, G. D. Bader, L. Moore, S. L. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier, L. Yang, C. Wolting, I. Donaldson, S. Schandorff, J. Shewnarane, M. Vo, J. Taggart, M. Goudreault, B. Muskat, C. Alfarano, D. Dewar, Z. Lin, K. Michalickova, A. R. Willems, H. Sassi, P. A. Nielsen, K. J. Rasmussen, J. R. Andersen, L. E. Johansen, L. H. Hansen, H. Jespersen, A. Podtelejnikov, E. Nielsen, J. Crawford, V. Poulsen, B. D. Sorensen, J. Matthiesen, R. C. Hendrickson, F. Gleeson, T. Pawson, M. F. Moran, D. Durocher, M. Mann, C. W. Hogue, D. Figeys, and M. Tyers. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415:180–183, 2002.

[14] T. Ito, K. Tashiro, S. Muta, R. Ozawa, T. Chiba, M. Nishizawa, K. Yamamoto, S. Kuhara, and Y. Sakaki. Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc. Natl. Acad. Sci. USA*, 93(3):1143–1147, 2000.

[15] S. Letovsky and S. Kasif. Predicting protein function from protein/protein interaction data:a probabilistic approach. *Bioinformatics*, 19:i197–204, 2003.

[16] S. Li, C. Armstrong, N. Bertin, H. Ge, S. Milstein, M. Boxem, P. Vidalain, J. Han, A. Chesneau, T. Hao, D. Goldberg, N. Li, M. Martinez, J. Rual, P. Lamesch, L. Xu, M. Tewari, S. Wong, L. Zhang, G. Berriz, L. Jacotot, P. Vaglio, J. Reboul, T. Hirozane-Kishikawa, Q. Li, H. Gabel, A. Elewa, B. Baumgartner, D. Rose, H. Yu, S. Bosak, R. Sequerra, A. Fraser, S. Mango, W. Saxton, S. Strome, S. Van Den Heuvel, F. Piano, J. Vandenhaute, C. Sardet, M. Gerstein, L. Doucette-Stamm, K. Gunsalus, J. Harper, M. Cusick, F. Roth, D. Hill, and M. Vidal. A map of the interactome network of the metazoan c. elegans. *Science*, 303:540–543, 2004.

[17] H. W. Mewes, D. Frishman, U. Guldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Munsterkotter, S. Rudd, and B. Weil. Mips: a database for genomes and protein sequences. *Nucleic Acids Res*, 30:31–34, 2002.

[18] J. R. Newman, E. Wolf, and P. S. Kim. A computationally directed screen identifying interacting coiled coils from saccharomyces cerevisiae. *Proc Natl Acad Sci U S A*, 97:13203–13208, 2000.

[19] W. H. Press, S. A. Teukosky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipe in C: The Art of Scientific Computing*. Cambridge University Press, New York, 1992.

[20] M. P. Samanta and S. Liang. Predicting protein functions from redundancies in large-scale protein interaction networks. *Proc Natl Acad Sci U S A*, 100:12579–12583, 2003.

[21] B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeas. *Nat Biotechnol*, 18:1257–1261, 2000.

[22] R. Sharan and R. Shamir. Click: a clustering algorithm with applications to gene expression analysis. In *Proc Int Conf Intell Syst Mol Biol.*, pages 307–316, 2000.

[23] V. Spirin and L. A. Mirny. Protein complexes and functional modules in molecular networks. *Proc Natl Acad Sci U S A.*, 100:12123–12128, 2003.

[24] A. H. Tong, B. Drees, G. Nardelli, G. D. Bader, B. Brannetti, L. Castagnoli, M. Evangelista, S. Ferracuti, B. Nelson, S. Paoluzi, M. Quondam, A. Zucconi, C. W. Hogue,

S. Fields, C. Boone, and G. Cesareni. A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science*, 295:321–324, 2002.

[25] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg. A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, 403:623–627, 2000.

[26] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S. M. Kim, and D. Eisenberg. Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res*, 30:303–305, 2002.

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