Analysis of Biological Networks: Protein-protein Interaction Networks – Functional Annotation[*]

Lecturer: Roded Sharan Scribe: Shaul Karni And Aharon Sharim

Lecture 8, December 14, 2006

1 Introduction

Until the year of 2000, most scientific efforts in the fields of genomics and proteomics were directed towards achieving complete sequenced genomes. The recent availability of large-scale genomics and proteomics data has shifted the focus towards understanding the functionality of genes and the mechanisms of their activity. Known functional annotations such as the Gene Ontology [5] and genome-wide data such as whole genomes sequences, gene expression patterns, protein-protein interactions and others may help us to infer functions of the entire proteomes.

Gene ontology (GO) is a systematic scheme for function assignment. The functional annotations are hierarchial, organized as a directed acyclic graph from general to more specific terms (Figure 1). GO annotations are divided into three categories: cellular component, molecular function and biological process. The cellular component ontology describes location at the levels of subcellular structures and macromolecular complexes. The molecular functions of a gene product describes the actions of the gene product at the molecular level or the ”abilities” that it has. A biological process is a recognized series of events or molecular functions. A higher level of GO is given by the reduced GO-slim annotation.

One way for evaluating a protein function is examining its interactions with other proteins assuming that proteins involved in similar functions are more likely to interact. It was previously shown [9] that data of protein-protein interactions has a large predictive value for protein function (Figure 2). In the following sections we survey methods for inferring the function of proteins based on their physical interactions.

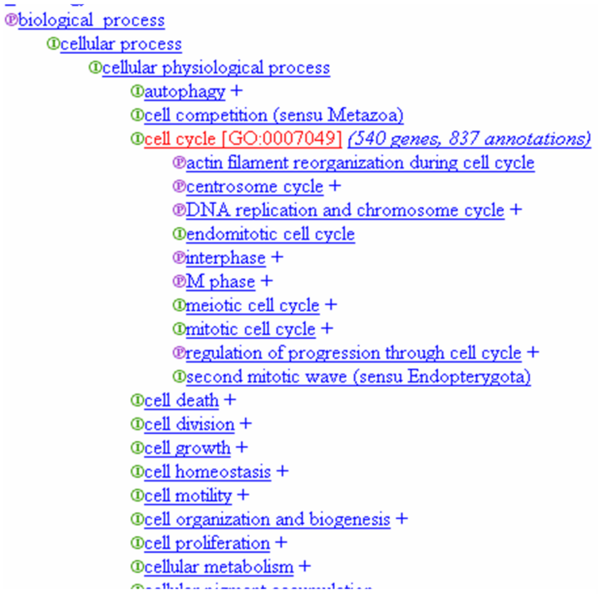
2 Local Methods

2.1 Neighborhood Counting

Neighborhood Counting is a method published by Schwikowski et al [11] which predicts the function of a protein based on the functions of its neighbors. The annotated functions of all the protein's neighbors are ordered in a list, from the most frequent to the least frequent. The first three or fewer functions are stated as predictors of the function of the protein. This method can be generalized to the $k - neighborhood$ of the predicted protein (i.e. the proteins of up to k edges from the source) rather than for its immediate neighbors. This method is simple but has several disadvantages:

1. No significance value is associated with a prediction.

[*] Based on a scribe by Inbar Cohen-Gihon and Mira Abraham.



ⓟbiological_process
ⓘcellular process
ⓘcellular physiological process
ⓘautophagy +
ⓘcell competition (sensu Metazoa)
ⓘcell cycle [GO:0007049] (540 genes, 837 annotations)
ⓟactin filament reorganization during cell cycle
ⓟcentrosome cycle +
ⓟDNA replication and chromosome cycle +
ⓘendomitotic cell cycle
ⓟinterphase +
ⓟM phase +
ⓘmeiotic cell cycle +
ⓘmitotic cell cycle +
ⓟregulation of progression through cell cycle +
ⓘsecond mitotic wave (sensu Endopterygota)
ⓘcell death +
ⓘcell division +
ⓘcell growth +
ⓘcell homeostasis +
ⓘcell motility +
ⓘcell organization and biogenesis +
ⓘcell proliferation +
ⓘcellular metabolism +
ⓘcellular pigment accumulation

Figure 1: Source: [14]. Example of a Gene Onthology categorization.

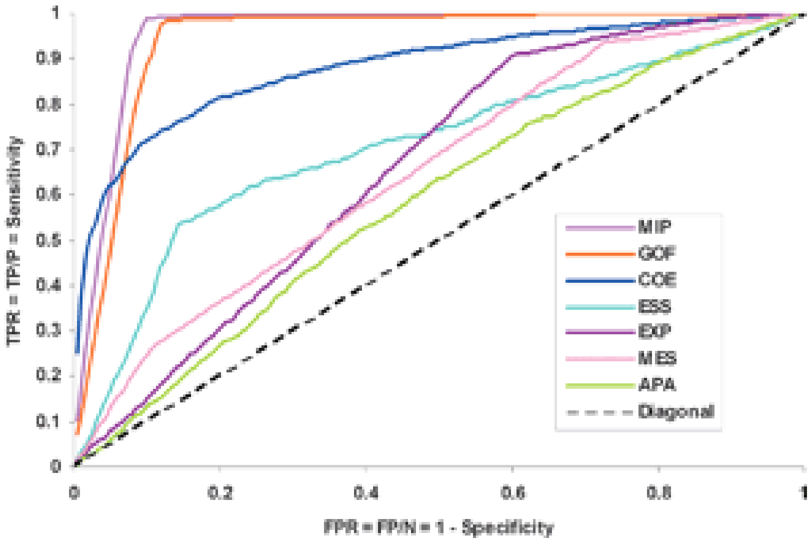


Figure 2: Source: [9]. Predictive power of individual features illustrated by ROC curves. ROC (receiver operating characteristic) curves graphically represent the performance of a classification method for different costs. It consists of a set of points, where for each point the vertical coordinate is a true positive rate (TPR) given by the ratio TP/[TP+FN]. The horizontal coordinate is a false positive rate (FPR) given by the ratio FP/[FP+TN). MIP and GOF are very strongly correlated, this shows a correlation between function and interaction.

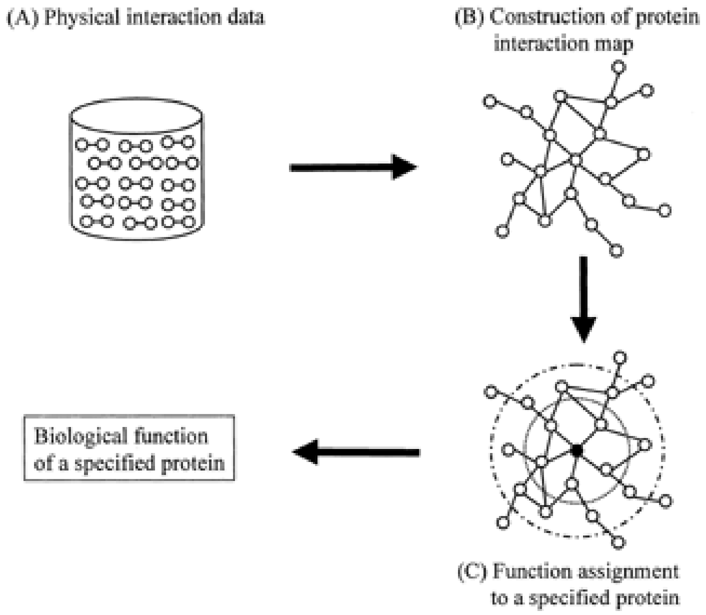

(A) Physical interaction data
(B) Construction of protein interaction map
Biological function of a specified protein
(C) Function assignment to a specified protein

Figure 3: Source: [6]. Overview of the local prediction methods. White circles represent proteins and the black circle represents a query protein for which function is predicted. (A) Physical interaction data deposited in the public databases. (B) Construction of the protein interaction map by integrating all physical interaction data. (C) Assignment of function to a query protein. This is done based on the functions of neighboring proteins on the map.

2. This method gives an equal weight for distant neighbors and immediate ones, while in practice immediate neighbors are more likely to share the same function with the predicted protein.
3. The method ignores the size of functional classes, tending to assign the more general functions to nodes.

2.2 Chi Square

This method attempts to eliminate the third problem (assigning the more general class) from the neighbor - counting method, by taking into account the expected appearance of each class (we expect to see the larges classes more). In this method, suggested by Hishigaki et al [6], the function of each protein in the protein-protein interaction graph (black circle in figure 3C) is predicted based on the functions of $n-neighbors$. For each functional category j, a χ^2-like score is calculated as follows. For each node i and it's neighbors $N(i)$, define $n_i(j)$ as the number of neighboring proteins that have the function j. Let's define the backround frequency, i.e. the frequency of j among all proteins as $f(j)$. Also let's define the expected number of proteins neighboring i with function j as $e_i(j) = |N(i)|f(j)$. We can now calculate $S_i(j)$ (the χ^2 score):

$$S_i(j) = \frac{(n_i(j) - e_i(j))^2}{e_i(j)} \tag{1}$$

The protein of interest is assigned the function with the highest score among the functions of the n-neighborhood. The method can also be extended to assigning more the one function by taking the k highest scoring functions.

Next , a *self consistency test* is performed. This test goal is to find the optimal n value , meaning the best maximal distance the algorithm should scan nodes for their function. The predicted functions of all proteins

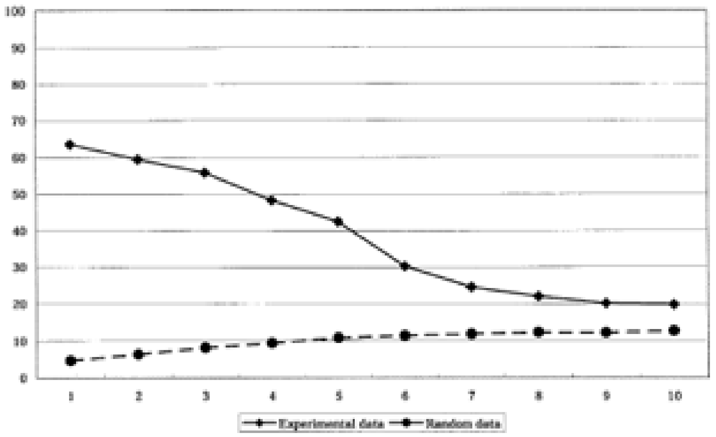


Figure 4: Source: [6]. Results of self-consistency test: prediction of cellular role. The horizontal axis represents the distance from a query protein (the n value) and the vertical axis represents the percentage prediction accuracy (data shown in filled diamonds). For reference, prediction results with randomly-assigned functions conserving the size distribution are also shown (in filled circles).

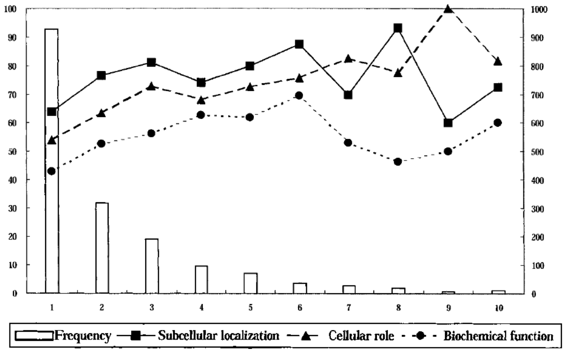


Figure 5: Source: [6]. Frequency of the number of binding partners and the dependency of prediction accuracy on that number. The horizontal axis represents the number of binding partners. The bar chart shows the distribution of the number of binding partners, with the right vertical axis showing their frequency. The line graphs show the prediction accuracy (%) with the left vertical axis.

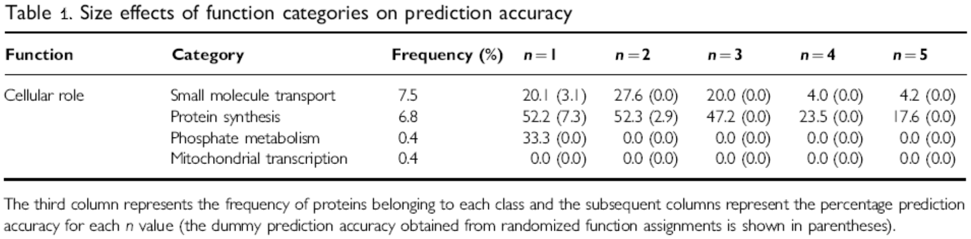


Table 1. Size effects of function categories on prediction accuracy

Function	Category	Frequency (%)	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
Cellular role	Small molecule transport	7.5	20.1 (3.1)	27.6 (0.0)	20.0 (0.0)	4.0 (0.0)	4.2 (0.0)
	Protein synthesis	6.8	52.2 (7.3)	52.3 (2.9)	47.2 (0.0)	23.5 (0.0)	17.6 (0.0)
	Phosphate metabolism	0.4	33.3 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
	Mitochondrial transcription	0.4	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)

The third column represents the frequency of proteins belonging to each class and the subsequent columns represent the percentage prediction accuracy for each n value (the dummy prediction accuracy obtained from randomized function assignments is shown in parentheses).

Figure 6: Source: [6]. Size effects of function categories on prediction accuracy.

in the map are compared with their annotated functions for each distance n. In Figure 4, the prediction accuracy of cellular role (one of the three functional categories used in the Yeast Proteome Database - YPD) with various n values is shown. The optimal n value is determined by a self-consistency test. Results of random predictions using the randomized function assignments are also shown for reference. Randomized function assignments results are the average of 100 trials in which for each protein, the assignment of function was changed to another, conserving the size distribution of functional categories. The maximum value was 63.6% with 1-neighboring proteins. Figure 5 shows both the distribution of the number of interaction partners (the degree of the protein) and the dependence of the prediction accuracy on this number for three kinds of functional categories. We see that the accuracy function is monotonically increasing only for degrees smaller than 7. This can be explained by the relatively small amount of proteins with a degree of at least 7 leading to non-robust results.

Figure 6 shows the two largest and the two smallest categories from the predictions of cellular role. It can be seen that this method is not effective for predicting small categories, and although the prediction accuracy for larger categories is better, and there are 20.1%, 27.6% and 20% prediction accuracy for n=1, n=2 and n=3, respectively, these values are not particularly high.

Disadvantages of the χ^2 method:

1. As in Neighbor-Counting, the method does not take under consideration the topology of the network. (i.e. a neighbor farther away has the same weight as a closer one).

2. Only one measure for annotation quality was used - the specificity (what part of the assignments we found were true). However, sensitivity should be taken under consideration as well (i.e. what part of the true assignments were found).

3. This method is not effective for predicting very small or very large categories: For small categories , the chance of a node to have a neighbor is small, and for very large categories, due to the very high random score, the random graph gets very high scores(close to 80%).

2.3 Application for Reliability Estimation

Trying to estimate the accuracy of function prediction, Deng et al [3] considered two methods: the Neighborhood-Counting method and the Chi-Square method for the prediction of protein function. Considering the functional annotation based on the cellular role of proteins, they examined 6,416 proteins, among which 3,894

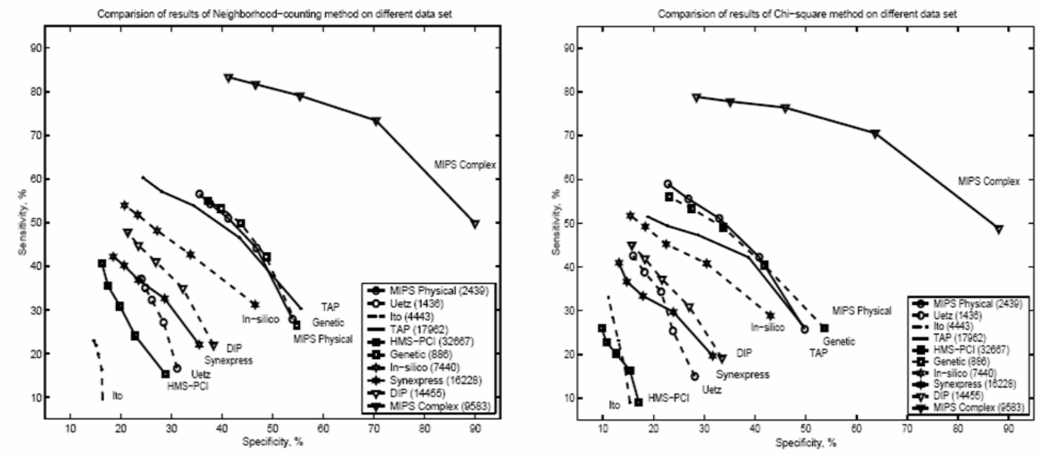


Figure 7: Source: [3]. Specificity and sensitivity of functional predictions for different protein-protein interaction data sets using the Neighborhood-Counting (left) and the Chi-Square (right) methods.

proteins have been assigned to one or more functions of 43 cellular roles, and the rest 2,522 proteins are unknown. They used a leave-one-out method to estimate the accuracy of predictions, and compared the results of different data sets.

Figures of Merit To test how successful a method is, we divide the results into 4 categories, *TP* - True Positives, *FP* - false Positives, *TN* - True Negatives and *FP* - False Positives. These can then be combined into two sets of measures:

1. *precision* and *recall precision* is the fraction of the true-positive predictions out of all the positive predictions (TP/(TP+FP)). *recall* is the fraction of the true-positive predictions out of all the true predictions (TP/(TP+FN)).

2. *specificity* and *sensitivity*: *sensitivity* is the same as recall (TP/(TP+FN)). *specificity* is the fraction of the true negatives out of all the negatives (TN/(TN+FP)).

One of the methods used to evaluate methods is the *leave one out method*. In this method, one known node is left out, and its function is calculated using all other nodes. The left out node's real function is then compared to the calculated function to check the correctness of the method.

In [3] the leave-one-out method was repeated for k known proteins $P_i, ..., P_k$, where n_i is the number of functions of protein P_i in YPD, m_i is the number of *predicted* functions for protein P_i and k_i is the overlap between known and predicted functions. The results can be seen in figure 7 which shows the relationship between specificity and sensitivity for the Neighborhood-Counting method and the Chi-Square method on different protein interaction data sets. One can see that the MIPS protein complex data have the best performance (high specificity and high sensitivity) both in the Neighborhood-Counting and the Chi-Square methods. The main advantage of this work is the use of a sensitivity score, where the previous methods described only the specificity of their predictions.

3 Global Methods

3.1 Graph Theoretic

3.1.1 Minimizing Inter-class Interactions

Similarly to local methods, global methods are based on the assumption that proteins involved in similar functions are more likely to interact. However, global methods use different strategies for the prediction of protein function, some of the works will be described hereinafter. Vazquez et al [13] aim at finding the assignment of functional classes to proteins which, minimizes the number of protein interactions across different functional categories. The scoring function is set to be the number of proteins with the same functional annotation. This score is associated with any given assignment of functions for the whole set of unclassified proteins. The contribution to the total score of a given functional assignment is computed from the number of classified and unclassified neighbor proteins with that function. To calculate the energy function we define f_i as function i assigned to protein i and $J_{i,j}$ is 1 if i and j interact and areboth unclassified, 0 otherwise. $h_i(f_i)$ is the number of classified partners of protein i with function f_i. Where $\delta(i,j)$ is the discrete δ function.
The above parameters are chosen to globally minimize the following energy function:

$$E = -\sum_{ij} J_{ij}\delta(f_i, f_j) - \sum_i h_i(f_i) \tag{2}$$

This energy function is an entropy function , containing the information about a node's neighbors and the interaction data between pairs of nodes. By minimizing the energy function , we minimize the functional difference between interacting nodes. This problem bears similarity to the *Min. Multiway k-Cut* [2], defined in the following way: Given a weighted graph and a set of k terminal vertices, find a set of minimal weight edges whose removal disconnects all terminals from one another. This is a polynomially solvable problem when $k = 2$ (max-flow-min-cut) and is NP-hard for $k = 3$. This problem has a trivial 2-approximation by the "isolation" algorithm: for a given terminal s_i, an isolating cut is any set of edges that cuts all paths between s_i and all the other terminals. Finding such cut ($\hat{E}_i$), whose weight is minimal is done by merging all the terminals other than s_i into a special vertex s_0, and then finding the minimum $s_i - s_0$ cut in the resulting graph by a standard max-flow-min-cut.

Let E^* be the optimal group of edges disconnecting all terminals and let V_i be the set of vertices left connected to i by E^*. Let E_i^* be the group of edges out of E^* having exactly one endpoint in V_i (The importance of E_i^* is that it isolates i). Hence, W($\hat{\text{E}}_i$) $\leq W(E_i^*)$. On the other hand, each edge in E^* is found exactly in two different subsets of E^* (i.e. in E_i^* and in E_j^*), thus the sum of weights of E_i^* is exactly twice the optimal weight:

$$\sum W(E_i^*) = 2W(E^*) \tag{3}$$

Thus:

$$W(\hat{E}) \leq \sum W(\hat{E}_i) \leq \sum W(E_i^*) = 2W(E^*) \tag{4}$$

Vazquez et al [13] used simulated annealing to minimize the energy function. Starting with an initial random configuration f_i, at each step, a random protein is selected and its state is changed from f_i to f_i', where f_i' is selected at random among the possible states of protein *i*. The energy difference (the change in energy) is $\Delta E = E' - E$ between the two configurations. If $\Delta E \leq 0$, the new configuration is accepted. If $\Delta E > 0$, they accept the new configuration with probability $r = \exp(-\Delta E/T)$.

This functional prediction method was applied to the analysis of the yeast *S. cerevisiae* protein-protein interaction network consisting of 1826 proteins and 2238 interactions. The functional classification was obtained from the MIPS database containing 424 functional categories.

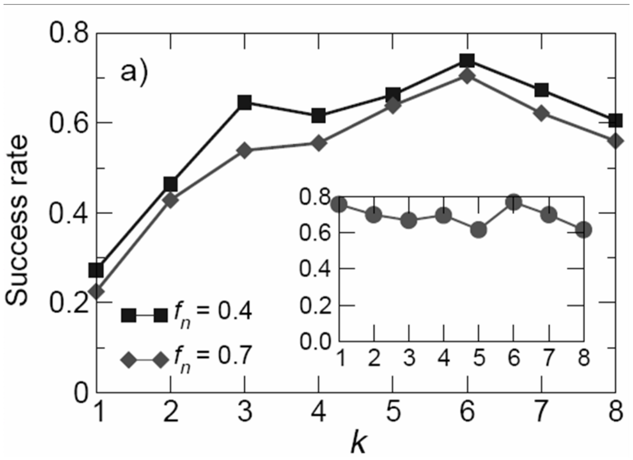


Figure 8: Source: [13]. Self-connectivity test. Each point represents the probability that the functional classification of proteins with *k* interacting partners. The success rate is reported here for the values of f_n=0.4 and f_n=0.7 in the upper and lower curve, respectively. In the inset the $f_n \to 0$ is reported, that is, when only a single protein is set unclassified.

Figure 8 shows the rate of successful predictions as a function of the degree of the proteins for different values of f_n, using the most detailed functional classification scheme available (424 functional classes). The success rate is estimated by hiding 40% or 70% (f_n=0.4 or 0.7) of the protein annotations and predicting their functions. The prediction quality of poorly connected nodes (degree 1 and 2) decreases to 30%. When only a single protein is hidden, very high rate of successful predictions (up to 80% success) is observed.

Figure 9 describes a comparison of the success rate of the global optimization (GO) method proposed here and the Neighborhood Counting. The success rate is shown as a function of the number of interacting partners k. N_k is the number of proteins having k interacting partners. The columns GO1 and GO2 are two different levels of functional classification that have been used. In GO1, the most stringent classifications, containing 424 functional categories were taken. In GO2, the less detailed classifications were taken, containing 20 functional categories. One can see that GO2 has a higher percentage of prediction accuracy. The comparison of the values in Figure 9 clearly indicates that the global optimization method is more effective, with a higher percentage of correct predictions.

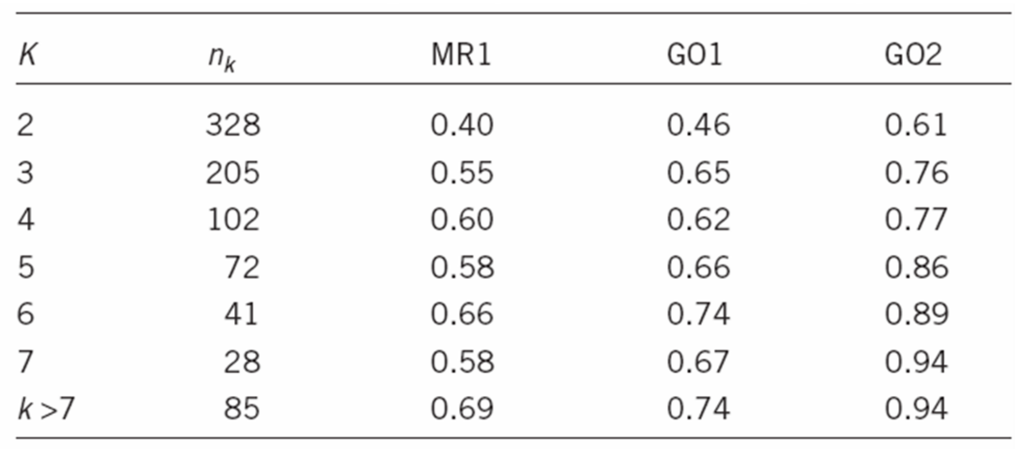
Disadvantages of the ”minimize inter-class interactions” method:

3.1.2 Maximize intra-class interactions

Similarly to the previous method, Karaoz et al. [8] sought to maximize the weight of consistent edges minus the weight of inconsistent ones. The energy function in this case is:

$$E = -\sum_{i,j} w_{ij} s_i s_j \tag{5}$$

Where $s_i = 1$ if i is assigned the function and $s_i = -1$ otherwise. The goal of this procedure is to assign a state of -1 or +1 to the nodes whose initial state is equal to 0. An important difference between this method and the previous one is that each function is considered separately, resulting with separate max-cut problems. The optimization is done by local search using applications of the following rule, which defines the dynamic behavior of the network, interactively to each node of the network until convergence (i.e. when



K	n_k	MR1	GO1	GO2
2	328	0.40	0.46	0.61
3	205	0.55	0.65	0.76
4	102	0.60	0.62	0.77
5	72	0.58	0.66	0.86
6	41	0.66	0.74	0.89
7	28	0.58	0.67	0.94
k >7	85	0.69	0.74	0.94

Figure 9: Source: [13]. Success rates for global optimization (GO) versus Neighborhood counting (MR). See text for explanations about each column.

further application of this rule does not change the state of any node).

$$s_i = sgn(\sum_{1\leq j\leq n_i} w_{ij}s_j - \Theta) \tag{6}$$

Here, n_i is the number of neighbors of protein i and Θ is an ”activation threshold”. The right side of this equation computes the weighted sum of the states of the neighbors of node i and compares this sum with Θ: if the sum is $> \Theta$, then the state of node i is set to +1, otherwise it is set to -1. This rule is a variant of the local guilt-by-association rule used in earlier studies. Iterative application of this rule achieves a more globally consistent functional annotation to all of the proteins in the network than a single application of this rule.

The evaluation of the quality of the predictions was done by the leave one out method (described above). *F-measure* is an harmonic mean of precision and recall, where higher *F-measures* correspond to a better quality. Their data set included protein-protein interactions that were supported by at least two experiments. They obtained 94% precision and 64% recall for 828 GO functions with *F-measures*> 0. However, there is no comparison to previous work, so it is hard to evaluate the quality of the results.

3.1.3 Functional-network flow

All the above local methods do not consider the global topology of the protein interaction graphs. Therefore, considering the two instances shown in Figure 10, all non-topological methods will produce the same result for protein a in both instances. These methods ignore information on the number of independent paths between protein a and the annotated proteins, in addition, these methods treat all distances equally. In the left figure, all paths to a from the known proteins are dependent on b, where in the right figure, all paths are independent. The method described in Nabieva et al [10] is a network flow where every protein with known function is a source of flow to the other proteins, and the edges are weighted according to their reliability. The functional flow algorithm generalizes the principle of guilt by association to groups of proteins that may or may not interact with each other physically. The algorithm is as follows:
Each protein with a known function is treated as a source of flow for that function. For each function, flow spread is simulated by an iterative algorithm using discrete time steps. The *capacity* of an edge is defined as its weight and will represent its reliability. The *reservoir* of a node is the amount of flow the node can

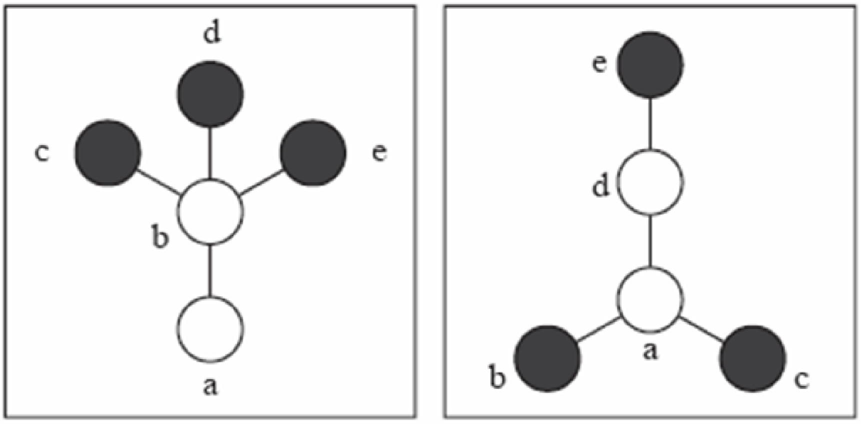


Figure 10: Source: [10]. Two protein interaction graphs that are treated identically by Neighborhood-Counting with radius 2 when annotating protein a. Dark colored nodes correspond to proteins that are known to take part in the same process.

pass to its neighbors. Source nodes have infinite reservoir; for all others it is initially zero. The algorithm is run for several iterations, typically six, enough to let the flow reach all the nodes in the network. In each iteration a node pushes flow residing in its reservoir such that flow is proportional to edge capacities of the node, capacity constraints are satisfied and flow only spreads from more filled to less filled reservoirs. A protein's *Functional score* the amount of flow that entered the it's reservoir throughout the algorithm.

The formal algorithm: $R(u)$ - reservoir of u. Infinite for sources; initially 0 otherwise.
$g(u, v)$ flow from u to v. Initially (in time 0), there is no flow on all edges.
At each subsequent time step, the reservoir of each protein is recomputed by considering the amount of flow that has entered the node and the amount that has left:

$$R_t^a(u) = R_{t-1}^a(u) + \sum_{v:(u,v)\in E} (g_t^a(v,u) - g_t^a(u,v)) \tag{7}$$

At each subsequent time step, the flow proceeding downhill and satisfying the capacity constraints:

$$g_t^a(v,u) = \begin{cases} 0 & \text{if } R_{t-1}^a(u) < R_{t-1}^a(v), \\ min(w_{u,v}, \frac{w_{u,v}}{\sum_{(u,y)\in E} w_{u,y}}) & \text{otherwise} \end{cases} \tag{8}$$

The algorithm is run for $d = 6$ iterations (enough to propagate flow from the source to all recipients). The final functional score for u is the sum of all flows which entered into the vertex during the whole algorithm run:

$$f_a(u) = \sum_{t=1}^{d} \sum_{v:(u,v)\in E} g_t^a(v,u) \tag{9}$$

Comparison to previous approaches: Since the functional flow method deals with weighted protein interaction graphs, in order to compare it to the other local methods described above, one has to generalize those to the weighted case. Here we describe for each relevant method, how this conversion can be achieved.

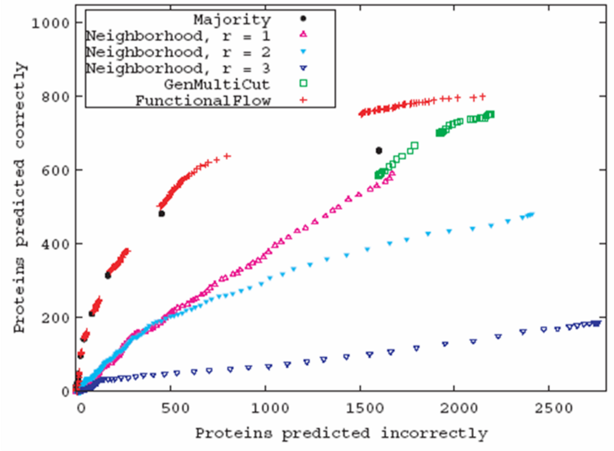


Figure 11: Source: [10]. ROC analysis of Neighborhood Counting (herein Majority), Chi-square (herein Neighborhood), GenMultiCut and FunctionalFlow on the yeast UNWEIGHTED physical interaction map

1. Neighborhood-counting is easily generalized to weighted data by taking a weighted sum (instead of simply summing up) the number of times each annotation occurs for each protein, as described in the original algorithm. For each protein, the score of a particular function is the corresponding sum.

2. The chi-square method does not extend naturally to the case of weighted interaction graphs.

3. The k-cut approach is solved optimally using integer linear programming (ILP), generalizing it to the weighted case, via the following formulation: Let $x_{u,a}$ be a node variable, indicating if u is assigned function a. If a protein u has known functional annotation, the variable $x_{u,a}$ is fixed as 1, and 0 otherwise. $x_{u,v,a}$ is an edge variable, indicating if both u, v are annotated with a. Minimizing the weighted number of neighboring proteins with different annotations is equivalent to maximizing the number with the same annotations, and the corresponding ILP is shown here:

   Maximize $\sum_{(u,v)\in E, a\in FUNC} x_{u,v,a} w_{u,v}$
   Subject to

   $$\begin{array}{ll} \sum_a x_{u,a} = 1 & \text{if} \quad annot(u) = \emptyset \\ x_{u,a} = 1 & \text{if} \quad a \in annot(u) \\ x_{u,a} = 0 & \text{if} \quad a \notin annot(u), annot(u) \neq \emptyset \\ x_{u,v,a} \leq x_{u,a} & \text{for } (u,v) \in E \; and \; a \in FUNC \\ x_{u,v,a} \leq x_{v,a} & \text{for } (u,v) \in E \; and \; a \in FUNC \\ x_{u,v,a}, x_{u,a} \in \{0,1\} & \text{for all u,v and a.} \end{array}$$

In this ILP solution, *annot(u)* is the set of known annotations for protein *u* , and FUNC= $\bigcup_u annot(u)$ is the set of all functional annotations. The first constraints specifies that exactly one functional annotation is made for any protein. The second and third constraints ensure that if protein *u* is annotated with function *a*, $x_{u,a}$ is set as a constant to 1, and if protein *u* is annotated but not with function *a*, $x_{u,a}$ is set as a constant to 0. The third and fourth constraints ensure that a particular function is picked for an edge only if it is also chosen for the corresponding proteins.

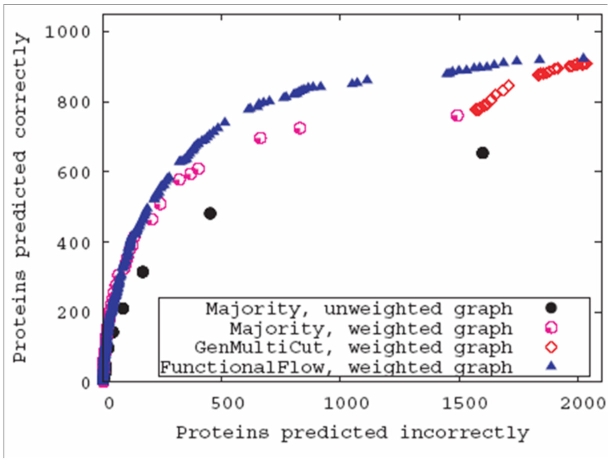


Figure 12: Source: [10]. Performance of Neighborhood-counting (here Majority), GenMultiCut and FunctionalFlow on the yeast physical interaction map where experimental reliabilities (weights) are incorporated. The performance of Neighborhood-counting (Majority) on the unweighted graph is also given as a reference.

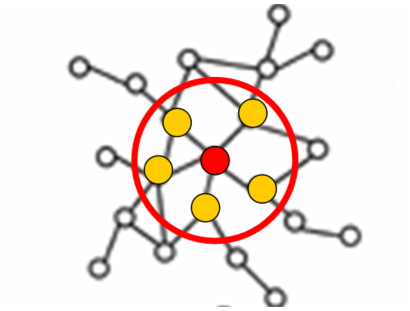


Figure 13: Illustration of the basic idea of probabilistic methods. The function of the red node is independent of all the nodes in the graph given the function of its direct neighbors (yellow)

Figure 11 shows that the Functional Flow algorithm identifies more true-positives (TPs) over the entire range of false positives (FPs) than either GenMultiCut or Chi-square, using radius 1,2 and 3. Functional Flow performs better than neighborhood-counting when proteins are not directly interacting with with at least three proteins of the same function.

Figure 12 shows a substantial improvement in predictions using all three methods when incorporating reliable estimations via edges weights.

3.2 Probabilistic Methods

A protein's function depends on its neighbors' function, the neighbors' functions depend on their neighbors and so on. The basic idea of the probabilistic methods is that a node's function is independent of non-direct neighbors functions given it's direct neighbors. This means that if we label a node in a particular function - it is conditionally independent of all other nodes given its neighbors (see illustration in Figure 13). Conditional independence of all other nodes given a node's neighbors is called *Markovian property*. In a simpler model, a *Markov chain* is a sequence $x_1, x_2, x_3, ...$ of random variables in which the conditional probability distribution of x_{n+1} on past states (values of previous variables, $x_1, ..., x_n$) is a function of x_n alone, i.e:

$$P(x_{n+1} = a_{n+1}|x_1 = a_1, x_2 = a_2, ..., x_n = a_n) = P(x_{n+1} = a_{n+1}|x_n = a_n) \tag{10}$$

In other words, the chain satisfies the Markovian property, in any state it ”remembers” only the previous state. Similarly we can define Markov Random Field (MRF) as an undirected graph in which the vertices represent the random variables and the edges reflect the relation between these random variables. The graph will also maintain the short memory property (Markovian property). The state of a given vertex x_t given its immediate neighbors states is independent of all the other vertices states, i.e:

$$P(x_t = a_t|x_s = a_s, s \neq t) = P(x_t = a_t|x_s = a_s, s\epsilon N(t)) \tag{11}$$

Now we will define a probability function and we will show that it is the probability function of an MRF. Let C be the group of all cliques in graph G. Let A_c represent an assignment to a clique $c\epsilon C$. Let us define $\psi_c(A_c)$, $c\epsilon C$, a positive potential function (which upon receiving A_c, a legal assignment to the clique vertices evaluates to a positive real number). Now we can define the probability of assigning all the random variables $x_1, ..., x_n$ and $A = (a_1, a_2, ..., a_n)$ is a legal assignment to $x_1, x_2, ..., x_n$

$$p(x_1 = a_1, x_2 = a_2, ..., x_n = a_n) = \frac{1}{Z} \prod_{c\epsilon C} \psi_c(A_c) \tag{12}$$

Z is the normalizing constant which evaluates to:

$$Z = \sum_{A\epsilon A^*} \prod_{c\epsilon C} \psi_c(A_c) \tag{13}$$

Where A^* is defined as the group of all legal assignments.

A probability distribution is called Gibbs if it can be expressed as a product over $|C|$ positive potential functions (each applied on a different clique) whose value depends only on the random variables within that clique (see Equation 12).

The relation between MRF and Gibbs distribution is described by the Hammersley-Clifford Theorem.

Hammersley-Clifford Theorem: If X_t has a positive joint probability distribution ($\forall A\epsilon A^* : p(A) > 0$) then they form a MRF on graph G iff G admits a Gibbs distribution.

Proof: A simple proof can be found in [1].

The computation of (12) is problematic because of the following reasons:

1. Finding all cliques is computationally difficult.

2. Evaluating Z requires summing over all possible assignments to x_1,..., x_n which in most cases is not feasible.

To tackle the first problem, one commonly assigns zero potential for all cliques of size greater than 2. In order to further increase the computational efficiency, potential functions are made homogeneous, i.e. the potential functions of cliques of the same size are equal and depend only on the assignment to the cliques's vertices. After homogenizing the potential functions the expression describing the probability of a legal assignment A is described by: $p(A) = \frac{1}{Z}e^{-H(A)}$

$$H(A) = \sum_{i=1}^{n} H_1(a_i) + \sum_{(i,j)\epsilon E} H_2(a_i, a_j) \tag{14}$$

Where H_1 and H_2 are the potential functions for all cliques of size 1 and 2, respectively. The second difficulty , the evaluation of Z, is eliminated through using conditional probability. According to this theory the conditional probabilities satisfies the Markovian property (11). Therefore we can write the conditional probability using the homogeneous potential function:

$$P(x_t = a_t|A_{s\neq t}) = \frac{e^{-H(a_t,A_{s\neq t})}}{\sum_{\alpha\epsilon A_t} e^{-H(a_t,A_{s\neq t})}} \tag{15}$$

Where $A_{s\neq t}$ is a legal assignment to all vertices other than x_t and A_t is all the possible legal assignments to vertex t. In a $2 - order$, homogeneous MRF:

$$P(x_t = a_t|A_{s\neq t}) = \frac{e^{-H_1(a_t)-\sum_{s\epsilon Nei(t)} H_2 A(a_t,a_s)}}{\sum_{\alpha\epsilon A_t} e^{-H_1(a_t)-\sum_{s\epsilon Nei(t)} H_2 A(\alpha,a_s)}} \tag{16}$$

The evaluation is done by using *Gibbs Sampling* which is discussed later.

3.2.1 An MRF for protein function prediction

The aim of this research, published by Deng et al [4], was given a protein *P*, function *F* ,protein-protein interaction data and the functional annotation of protein's interaction to predict the probability that P bares function F. This was done by applying Bayesian approaches and by employing the theory of Markov Random Fields on the protein interaction network. The usage of probabilistic model has two benefits which cannot be disregarded. First, in contrast to none-probabilistic methods the work described here is not limited in the number of functions which can be assigned to a single protein. Second, none-probabilistic methods are incapable of providing an estimation of the significance of their assignments.

MIPS database was used ($\sim$ 2400 interactions) in order to determine the interacting protein pairs. The functional annotation was retrieved from the Yeast Proteome Database, YPD [7], according to the following criteria: biochemical function($\sim$ 3400 annotated proteins), subcellular location ($\sim$ 3200 annotated proteins) and cellular role ($\sim$ 3900 annotated proteins).

The problem is formulated as follows: For a given function f_i let us assign 1 to vertices which are annotated and have that function, and assign 0 to vertices which are annotated and do not have that function.

Let $X = (x_1, ..., x_n, x_{n+1}, ..., x_{n+m})$ denute the functional annotation of all proteins where
$x_1 = \lambda_1, ..., x_n = \lambda_n$ are unannotated, $x_{n+1} = \mu_{n+1}, ..., x_{n+m} = \mu_{n+m}$ are annotated ($\forall i, x_i \epsilon \{0, 1\}$). Considering a function of interest we wish to infer the function of the unannotated proteins using the protein interaction network.

Defining Gibbs distribution for protein-protein interaction network is done as follows: Let π be the probability that a protein should bare f_i. If we consider only cliques of size one (ignoring the edges or any other information embedded in the whole graph) then the probability of an assignment X is proportional to:

$$p(X) = \prod_{i=1}^{N} \pi^{x_i}(1-\pi)^{1-x_i} = (\frac{\pi}{1-\pi})^{N_i}(1-\pi)^N \tag{17}$$

where N is the total number of vertices ($N = n + m$) and N_i is the number of vertices which were assigned with 1. Now we wish to handle cliques of size 2. For that purpose we need to define the following:

- N_{11} the number of edges in which both the edges vertices got assignment 1.
  $N_{11} = \sum_{(i,j)\epsilon E} x_i x_j$.
- N_{10} the number of edges in which the edges vertices have different assignments.
  $N_{10} = \sum_{(i,j)\epsilon E}(1 - x_i)x_j + (1 - x_j)x_i$.

- N_{00} the number of edges in which both the edges vertices got assignment 0. $N_{00} = \sum_{(i,j)\epsilon E}(1 - x_i)(1 - x_j)$.

The definition of $H_1(X)$ and $H_2(X)$ is now straightforward:

$$\begin{aligned} H_1(X) &= -\alpha N_1 \\ H_2(X) &= -\beta N_{10} - \gamma N_{11} - N_{00} \end{aligned} \tag{18}$$

where $\alpha = \log \frac{\pi}{1-\pi}$. α, β and γ define the model parameter set, θ, and require parameter estimation [1]. The Gibbs distribution can be written:

$$P(X|\theta) = \frac{1}{Z(\theta)} e^{-H_1(X) - H_2(X)} \tag{19}$$

where $\theta = (\alpha, \beta, \gamma)$. Notice that since Z, the normalization constant, is equal to the sum of potentials over all assignments it is now dependent on θ.

Gibbs sampling: In order to evaluate the unknown $\{x_i\}$ sampling was used. In this method multiple changes were applies to a set of starting values. After enough iterations it should converge to the requested distribution.

In Gibbs sampling , we start with a set of p known initial values

$$\psi^{(0)} = (\psi_1^{(0)}, \psi_2^{(0)}, .., \psi_p^{(0)}) \tag{20}$$

For each iteration $i + 1$. for each vertex ψ_j , we calculate it's value using conditional distribution:

$$\psi_j^{(i+1)} = P(\psi_j | \psi_1^{(i)}, ..., \psi_j - 1^{(i)}, \psi_j + 1^{(i)}, .., \psi_p^{(i)}) \tag{21}$$

I.e. : in each iteration we recalculate all the values of ψ. After enough iterations, there will not be any more changes (”burn in”), And the Gibbs sampling will return

$$\psi = (\psi_1, \psi_2, .., \psi_p) \tag{22}$$

Claim: Gibbs sampling satisfies detailed balance. Proof: It is enough to show that

$$p(x_1, .., x_n)P(x_i^* | x_1, .., x_{i-1}, x_{i+1}, .., x_n) = P(x_1, .., x_{i-1}, x_i^*, x_{i+1}, .., x_n)P(x_i | x_1, .., x_{i-1}, x_{i+1}, .., x_n) \tag{23}$$

By Bayes theorem

$$P(x_i^* | x_1, .., x_{i-1}, x_{i+1}, .., x_n) = P(x_1, .., x_{i-1}, x_i^*, x_{i+1}, .., x_n)/P(x_1, .., x_{i-1}, x_{i+1}, .., x_n) \tag{24}$$

$$P(x_i | x_1, .., x_{i-1}, x_{i+1}, .., x_n) = P(x_1, .., x_{i-1}, x_i, x_{i+1}, .., x_n)/P(x_1, .., x_{i-1}, x_{i+1}, .., x_n) \tag{25}$$

Thus,

[1] Even though the calculation of α given π is straightforward ($\alpha = \log \frac{\pi}{1-\pi}$) the careful reader will probably notice that we still need to evaluate α. We wish α to reflect some of the dependency of a vertex in its neighbors, however, when we estimated π we did not consider the networks topology.

$$P(x_1,..,x_{i-1},x_{i+1},..,x_n) = P(x_1,..,x_{i-1},x_i^*,x_{i+1},..,x_n)/P(x_i^*|x_1,..,x_{i-1},x_{i+1},..,x_n) \tag{26}$$

$$P(x_1,..,x_{i-1},x_{i+1},..,x_n) = P(x_1,..,x_{i-1},x_i,x_{i+1},..,x_n)/P(x_i|x_1,..,x_{i-1},x_{i+1},..,x_n) \tag{27}$$

Comparing 26 and 27 we get:

$$P(x_1,..,x_{i-1},x_i^*,x_{i+1},..,x_n)/P(x_i^*|x_1,..,x_{i-1},x_{i+1},..,x_n) = \tag{28}$$

$$P(x_1,..,x_{i-1},x_i,x_{i+1},..,x_n)/P(x_i|x_1,..,x_{i-1},x_{i+1},..,x_n) \tag{29}$$

Conclusion: Gibbs distribution will converge to the requested distribution.

As it is evident from Figure 14 by using a low threshold of the predicator achieves at least 45% sensitivity and specificity.

In order to compare the MRF method with other methods, the authors implemented Neighborhood Counting [11] and Chi-square [6]. In both methods the top $n = 1, ..5$ functions were assigned to each unannotated protein. For every n the sensitivity and specificity values were calculated in both methods. As can be seen in Figure 15 for any given specificity, the sensitivity of the Bayesian method is higher than those of Neighborhood Counting or Chi-square. It is worth mentioning, that even though the Neighborhood Counting method is much simpler than the Bayesian, the difference between the performances of the two is not so pronounced.

The relationship between quality of prediction and network's information was estimated by applying the leave-one-out method was applied to proteins which have at least one, two or six interacting partners. As it is evident in figure 16 as the amount of information increases (higher number of interactions) the quality of the prediction increases too.

3.3 Module Based Annotation

A trivial usage of modules in order to predict gene function will be 'guilt by association'. In other words, given a module that a significant number of its proteins are known to have a certain function we can predict that the other proteins of the module also have this function.

The SAMBA algorithm [12] is designed for identifying significant modules. It was applied to a data set compiled from various sources as was described in [12], whose annotation was determined according to the GO database [5]. The specificity of this method was evaluated by performing a five-way cross validation: repeatedly applying SAMBA to data sets in which one-fifth of the known gene annotations were hidden and tested the specificity of predicting the function of these genes. The obtained specificity is ranged from 40% to 100% for a variety of classes (for example: mating (GO:0007322) 65%, amino acid metabolism (GO:0006520) 40%, glucose metabolism (GO:0006006) 100%.) In many cases, the classification errors result from ambiguous annotation terms or too general categories and may represent missing information rather than misclassification.

4 Summary

In this lecture, we described diverse approaches for predicting protein functions based on its physical interactions. Local approaches included Neighborhood-Counting and Chi-Square, while global approaches that

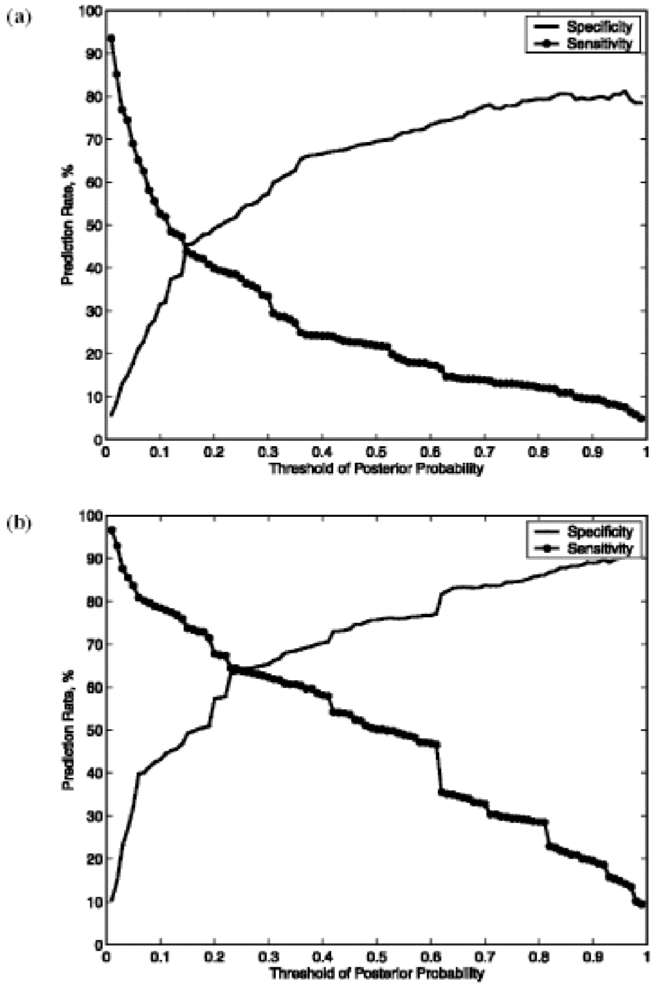


Figure 14: Source: [4]. Sensitivity and specificity values of the predicator as a function of the threshold. The upper and lower graphs represents predicator values for biochemical function and subcellular location respectively. Predicatory values for cellular role are similar to those of biochemical function (data not shown).

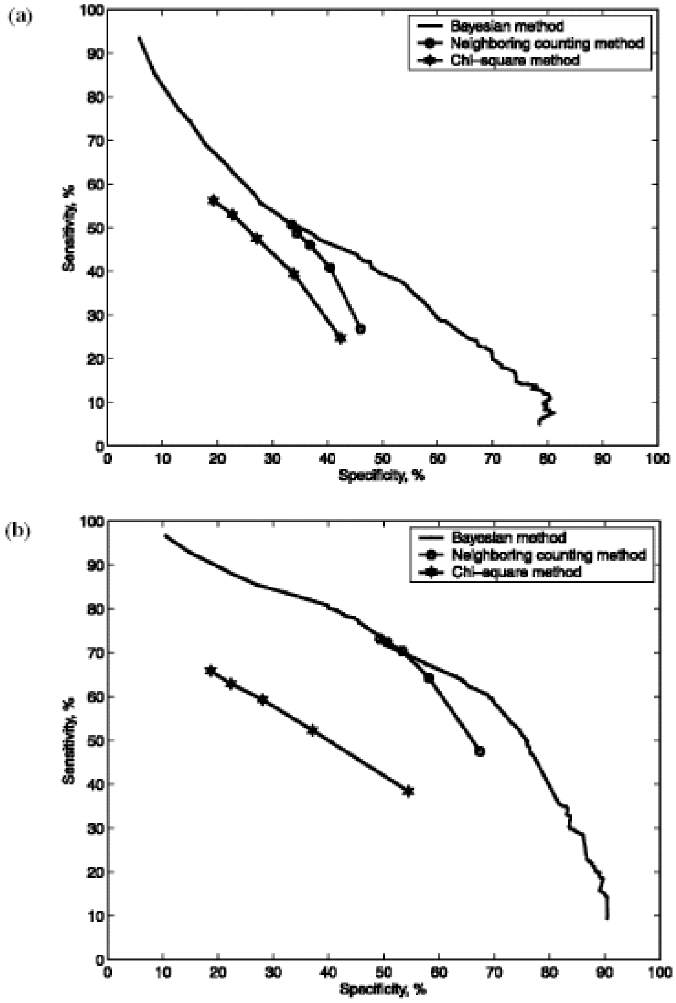


Figure 15: Source: [4]. The relationship between sensitivity and specificity in the prediction of all three models. upper graph and lower graph describe this relationship for biochemical function (fig a.) and subcellular location (fig b.) respectively. the relationships for cellular role are similar to those of biochemical function (data not shown).

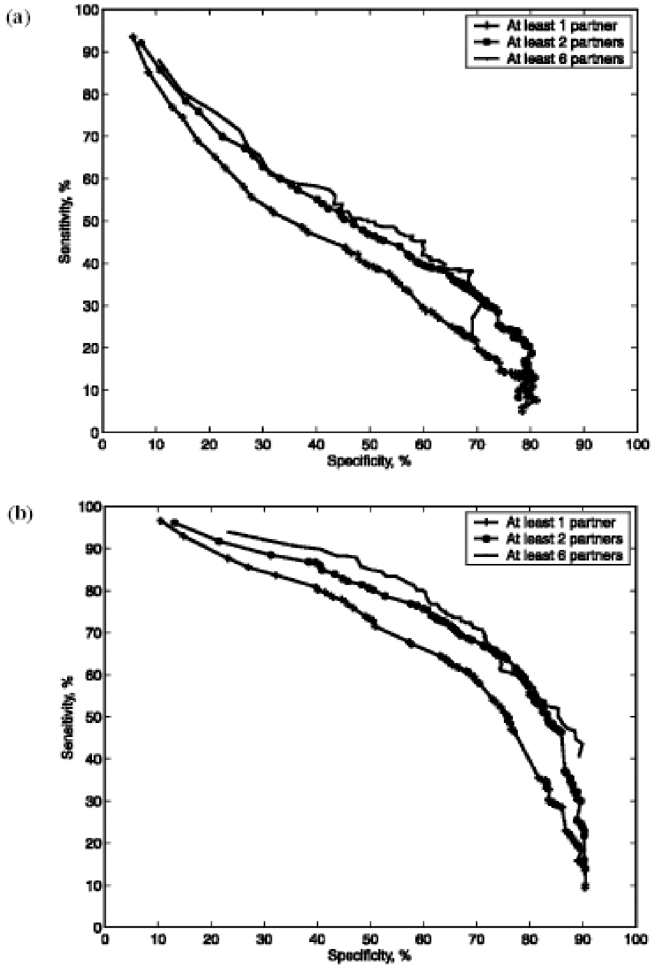


Figure 16: Source: [4]. The relationship between sensitivity and specificity in the prediction on proteins which have at least one, two or six interacting partners upper graph and lower graph describe this relationship for biochemical function (fig a.) and subcellular location (fig b.) respectively.

The relationships for cellular role are essentially similar to those of biochemical function and subcellular location(data not shown).

were mentioned here are based on either graph theoretic approaches or probabilistic models. The graph theoretic methods include minimizing inter-class interactions, maximizing intra-class interactions and a functional network flow algorithm. The probabilistic methods deal with Gibbs distribution and Markov Random Field and are able to accompany their function prediction with a level of confidence in their prediction. Finally, we mentioned in a nutshell a module-based approach. The survey shows that global approaches dominate the local ones, though they are not significantly better.

References

[1] J. Besag. Spatial interaction and statistical analysis of lattice systems. *J. Royal Statistical Society*, 36(2):192–236, 1974.

[2] E. Dahlhaus, DS. Johnson, CH. Papadimitriou, PD. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM J Comput*, 23:864–894, 1994.

[3] M. Deng, F. Sun, and T. Chen. Assessment of the reliability of protein-protein interactions and protein function prediction. *Pac Symp Biocomput*, pages 140–51, 2003.

[4] M. Deng, K. Zhang, S. Mehta, T. Chen, and F. Sun. Prediction of protein function using protein-protein interaction data. *J Comput Biol*, 10(6):947–60, 2003.

[5] MA. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, and RE. Foulger. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res.*, 1(32):D258–61, 2004.

[6] H. Hishigaki, K. Nakai, T. Ono, A. Tanigami, and T. Takagi. Assessment of prediction accuracy of protein function from protein-protein interaction data. *Yeast*, 18(6):523–31, 2001.

[7] PE. Hodges, AH. McKee, BP. Davis, WE. Payne, and JI. Garrels. The yeast proteome database (YPD): a model for the organization and presentation of genome-wide functional data. *Nucleic Acids Res*, 27(1):69–73, 1999.

[8] U. Karaoz, TM. Murali, S. Letovsky, Y. Zheng, C. Ding, CR. Cantor, and S. Kasif. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc Natl Acad Sci U S A*, 101(9):2888–93, 2004.

[9] LJ. Lu, Y. Xia, A. Paccanaro, H. Yu, and M. Gerstein. Assessing the limits of genomic data integration for predicting protein networks. *Genome Res.*, 15(7):945–53, 2005.

[10] E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, 21:302–310, 2005.

[11] B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. *Nat Biotechnol*, 18(12):1257–61, 2000.

[12] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc Natl Acad Sci U S A*, 101(9):2981–6, 2004.

[13] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Global protein function prediction from protein-protein interaction networks. *Nat Biotechnol*, 21(6):697–700, 2003.

[14] `http://www.informatics.jax.org/searches/GO_form.shtml/`.