Pacific Symposium on Biocomputing 7:413-424 (2002)

TOWARDS THE PREDICTION OF COMPLETE PROTEIN—PROTEIN INTERACTION NETWORKS

SHAWN M. GOMEZ[1] and ANDREY RZHETSKY[1,2]

Columbia Genome Center[1], and Department of Medical Informatics[2], Columbia University,
1150 St. Nicholas Avenue, Unit 109,
New York, NY 10032, USA

{smg42, ar345@columbia.edu}

We present a statistical method for the prediction of protein—protein interactions within an organism. This approach is based on the treatment of proteins as collections of conserved domains, where each domain is responsible for a specific interaction with another domain. By characterizing the frequency with which specific domain—domain interactions occur within known interactions, our model can assign a probability to an arbitrary interaction between any two proteins with defined domains. Domain interaction data is complemented with information on the topology of a network and is incorporated into the model by assigning greater probabilities to networks displaying more biologically realistic topologies. We use Markov chain Monte Carlo techniques for the prediction of posterior probabilities of interaction between a set of proteins; allowing its application to large data sets. In this work we attempt to predict interactions in a set of 40 human proteins, known to form a connected network, and discuss methods for future improvement.

1. Introduction

Increases in the number of sequenced genomes have led to rapid growth in the number of biological systems with characterized molecular components. Understanding of how these individual components are integrated together into a complete system, however, has lagged. Part of the difficulty in this undertaking originates in the fact that experimental data as to the existence of interactions between any two molecules are extremely sparse. While advances have also been made with, for example, high—throughput two—hybrid studies and complementary interaction databases, a comprehensive view of these molecular interaction networks is still lacking.

Networks consisting of proteins, DNA, RNA, and various small molecules, are formed due to one molecule's propensity to bind or otherwise influence another and hence alter system function. In this article, functional areas that provide this ability for one molecule to interact with another are referred to as domains. For example, subsequences of DNA where specific proteins bind are one class of domain (as are the amino acid subsequence responsible for binding activity within the protein). Interactions between proteins are of particular interest, as they are responsible for the

majority of "active" biological function. To date, protein—protein interactions are also the predominant type of interaction with significant quantities of supporting experimental data sets. As a result of these two factors, the work described here is focused on protein interaction networks, and more specifically, the *a priori* prediction of interactions between proteins as well as the prediction of whole networks.

Here we describe a statistical model capable of predicting protein—protein interactions which can be extended to other classes of molecules. While some previous methods have focused on using gene fusion events for the prediction of interactions [1, 2], our approach is more general in that any type of experimental evidence supporting an interaction can be used in prediction. Based upon experimentally verified interactions and estimates of network topology, this approach generates posterior probabilities conditioned on data for all possible interactions. The work described here is an extension of earlier work [3], which described the fundamentals of this model in some detail and presented small examples of its application. In this paper we describe the results of a more challenging application of the model and discuss methods for its improvement. A primary goal of this work is to provide a method for generating predictions that would be useful to the experimental biology community. In particular we feel that the prediction of molecular interactions, along with the ability to assign a probability to a given interaction, could be of significant benefit in the generation of new hypotheses and the prioritizing of appropriate (and perhaps more focused) experiments.

2. Model description

We start by representing a network as an oriented graph, $G = <V, E>$, where the vertices, V, of the graph are connected to each other through the edges, E. In this paper, edges represent a physical binding between corresponding proteins. Each vertex represents a protein, although extension of the model to handle other types of molecules (e.g. DNA) is rather straightforward. Each protein can be broken into smaller subunits consisting of one or more domains. We treat domains as evolutionarily conserved, elementary units of function. We assume that the domains are responsible for the generation of edges within the network; a simple example being a phosphorylation site and a kinase domain capable of phosphorylating that site. The upstream kinase domain is where the edge originates, and the edge terminates at the phosphorylation site. Domains themselves are found through the use of current tools and databases capable of assigning domains to proteins (e.g. Pfam)[4]. In addition to interaction data we use an additional parameter, characterizing the network topology, in the prediction of a network. While we describe the model in some detail here, greater description of certain aspects of the model can be found elsewhere [3].

2.1 Assigning probabilities to edges

Assigning a probability to a given network consists of two independent steps. The first step consists of assigning a probability p_{ij} to the existence of an edge connecting the proteins i and j or not connecting them (1 - p_{ij}). This process of assigning an edge can be thought of as the toss of biased coins (one coin per edge) for all possible edges, $|V|^2$ edges in all. The coin may be biased by prior information, assigning probabilities greater than 0.5 to vertices likely to be "attracted" to each other and form an edge. Probabilities of less than 0.5 can be assigned between proteins that "repel" each other and are thus unlikely to interact. Then for a protein network with a fixed number of vertices and a particular set of edges E between them, the probability of this network becomes

$$P(E) = \prod_{e_{ij} \in E} p_{ij} \prod_{e_{kl} \notin E} (1 - p_{kl}) .$$

How then do we define these individual edge probabilities p_{ij}?

We treat each protein as a collection of domains, and each of these domains has a tendency to attract or repel other domains between distinct proteins. Specifically, we define a probability of attraction $p(d_m, d_n)$ that exists for each upstream and downstream domain (as defined by moving with the "flow" or temporal sequence of the pathway), d_m and d_n, respectively. If the orientation is unknown, $p(d_m, d_n) = p(d_n, d_m)$, and the edge is undirected and both directed edges are present. Identical to edge probabilities between proteins, probabilities greater than 0.5 represent attraction while those less than 0.5, repulsion. For a pair of multidomain proteins i and j, where v_i and v_j are the set of unique protein domains for each, the probability of an edge forming between the two is

$$p_{ij} = \sum_{d_m \in v_i} \sum_{d_n \in v_j} \frac{p(d_m, d_n)}{|v_i||v_j|} .$$

Thus the probability of an edge forming between a pair of proteins is dependent on the relative attraction and repulsion of each protein's complement of domains, taken over all upstream—downstream pairwise combinations. This expression is a reasonable assumption as long as the number of edges incoming to or outgoing from a vertex is independent of the number of domains per protein; we have verified this assumption previously [3].

We determine the probability between a pair of domains, $p(d_m, d_n)$, by observing the frequency with which domain d_m appears upstream of domain d_n within experimental protein—protein interaction data. Specifically, we use

$$p(d_m, d_n) = \frac{1}{2}\left(1 + \frac{k_{mn}}{k_m k_n + \Psi}\right),$$

where Ψ is a positive real-valued pseudocount, k_{mn} is the number of edges in the training set that contain at least one domain d_m at the vertex of edge origin and at least one domain d_n at the vertex of edge destination, k_m is the number of distinct vertices that contain at least one domain d_m, and k_n is the number of distinct vertices that contain at least one domain d_n. This expression generates domain attraction probabilities greater than or equal to 0.5. As discussed later, probabilities of less than 0.5 are reserved for future modeling of repulsive interactions between domains, as observed, for example, in domain combination studies [5]. In this work, Ψ was assigned a value of 1. We assume that data supporting the existence of a particular interaction is usually backed by several experiments, while experiments showing the absence of an interaction are generally underrepresented by having either failed (and these failures not reported) or have not been performed. Thus this expression does not "penalize" for lack of an interaction, but assumes it to be the lack of supporting data. In the absence of any supporting data, all interactions between domains (and hence proteins) are equally likely.

In summary, we observe the frequency with which domain X lies immediately upstream or downstream of domain Y within experimental protein—protein interaction data. For an arbitrary pair of proteins, each with their own set of domains, we are then able to assign a probability to the likelihood of an edge forming between them. A complete network with a defined set of edges can similarly be assigned a probability; networks with many favorable edges will have a higher probability than a network with many unlikely edges.

2.2 Assigning probabilities to network topologies

The second part of our model deals with a global property of the network, that being its topology. The topology of a network is defined here as the distribution of edges going into and out of each vertex of the network. The number of edges going into a vertex is termed the *indegree*, and the number outgoing, the *outdegree*. In this model, we sort networks into a finite number of bins each representing a specific topology, where biologically realistic topologies have greater probabilities. Since multiple networks may be characterized by the same topology, each bin represents a collection of networks each with the same topological probability. For each network we compute the number of vertices that have outdegree zero, n_0^{out}, one, n_1^{out}, two, n_2^{out}, and so on to n_N^{out} . The vertices of a particular indegree are similarly computed. Networks with identical sets $\{n_x^{in}\}$ and $\{n_y^{out}\}$ are then grouped into a single bin. The probability of this bin is defined as

$$P(\{n_x^{in}\};\{\pi_x^{in}\},|V|) * P(\{n_y^{out}\};\{\pi_y^{out}\},|V|),$$

where

$$P\left(\{n_z\};\{\pi_z\},|V|\right)=\frac{|V|!}{n_0!\ldots n_N!}\prod_{z=0}^{N}\pi_z^{n_z}.$$

The probability distributions π_x^{in} and π_y^{out} give the probability of a network having x incoming and y outgoing edges respectively (described in more detail below).

It is easy to see that the probability of a network is simply the product of this distribution and $P(E)$ described in section *2.1*:

$$P(E)\times P(\{n_x^{in}\};\{\pi_x^{in}\},|V|)\times P(\{n_y^{out}\};\{\pi_y^{out}\},|V|).$$

Networks with favorable edge sets and favorable topologies will be more likely to be selected under our model.

The probability distributions π_x^{in} and π_y^{out} were estimated from yeast data taken from the Database of Interacting Proteins (http://dip.doe-mbi.ucla.edu/) [6, 7] and were found to follow a power—law distribution [3]; generally associated with scale—free behavior. This property was observed independently for the yeast protein network by Jeong and colleagues [8] and is also typical of a variety of other systems, both biological and man—made [5, 9, 10]. In this case, the probability of a vertex having k incoming or outgoing edges is

$$\pi_k = ck^{-\gamma},$$

with the values of c and γ different for each. For this work fits for each distribution gave c = 0.30 and γ = 1.97 for outgoing edges, and c = 0.56, γ = 2.80 for incoming. For π_0^{in} and π_0^{out} we used

$$\pi_0^{in}=1-\sum_{k=1}^{\infty}\pi_k^{in},\quad \pi_0^{out}=1-\sum_{k=1}^{\infty}\pi_k^{out}\ .$$

We used these distributions in the predictions described here, however, their use in other distributions is also possible. See the discussion section for further detail.

3. Methods

For training data, we used a combined dataset of protein—protein interaction data for both *Saccharomyces cereviseae* and *Homo sapiens*. We used the Pfam database (Pfam6.2; 2773 domains) and the HMMER package to determine the domains within each proteins (0.01 significance threshold). For the yeast data, we used a comprehensive list of interactions downloaded from Stanley Field's lab home page (http://depts.washington.edu/sfields/). This data included interactions from a number of sources [7, 11, 12]. We analyzed a total of 708 protein—protein interactions from yeast, all of which had at least 1 domain. For human data, we used a set of 778 inter-

actions downloaded from the Myriad Genetics Pronet Online web site (http://www.myriad-pronet.com/).

In this study, we attempted to predict interactions between a set of 40 human proteins known to form a fully connected network. Proteins were chosen from Pronet, with some of the proteins involved in the process of apoptosis. These interactions were not included as part of the original training set. Except for the requirement that all proteins of the network must be defined by at least one domain, this network was chosen at random. Proteins used in this analysis, and their indices in all figures, are given in Figure 1.

4. Results

4.1 Predictions of human protein—protein interactions

Edge probabilities based on domain—domain interaction data alone indicated that 97 edges had probabilities > 0.5 (see Figure 1). Note that we assumed that edges were not directed and thus the matrix shown here is symmetric. A total of 44 edges were in the original data set. Of these 44 edges, 8 are observed (18%) in the predicted 97 with probabilities > 0.5. Three out of eight interactions were involved in the heat shock pathway (read as (Y-axis, X-axis) on the figure); CHIP (12, 12) self—interaction, HSPA8—MRJ (24, 27), and HSPA8—PLCG1 (24, 30). The remaining 5 included FLN1—KSR1 (16, 25), PS2—CIB (32, 13), GDI2—RAB6 (20, 37), RAB6—GAPCenA (37, 18), and RAB6—RAB6KIFL (37, 38).

To see if any of the remaining 89 predicted edges represent known edges, we attempted a brief literature search. While often requiring significant expertise in a given pathway to adequately evaluate these results, we were still able to find obvious successes. The predictions of GDI1 (Guanine Nucleotide Dissociation Inhibitor, vertex 19) interacting with Rab11A, Rab3A, Rab5A, and Rab6 (vertices 34, 35, 36, 37 respectively) are in fact correct, and again not in the original data [13-15].

The prediction of CHIP interacting with TTC1 (tetratricopeptide repeat domain 1)(12, 40) is also understandable (though likely not a correct prediction, it may also be questionable in the original data) as the tetratricopeptide domain is a common protein—protein interaction motif, and a number of TPR containing proteins are known to interact with members of the heat shock protein family [16]. While purely speculative, the interaction of CIB (calcium and integrin binding protein with FLN1 (filamin) is interesting, as filamin has recently been shown to be a scaffold protein that interacts with calcium receptor and other cell signaling proteins [17]. While the prediction of only 8 known edges is disappointing, it is not unexpected due to limitations in the training data, and so it is quite possible that most of the predicted interactions are simply “noise.” The valid prediction of the GDI-Rab interactions, however,

were encouraging. Limitations of the data and methods for improving the model are presented in greater detail in the Discussion.

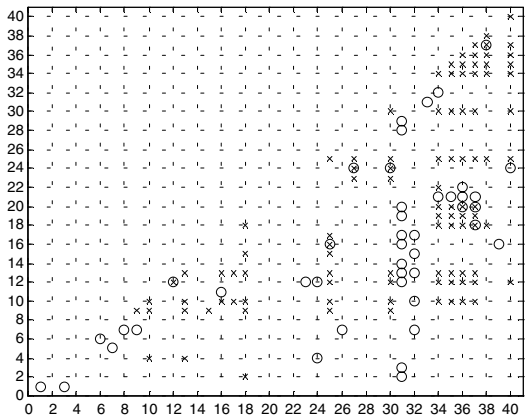

Figure 1. Known and predicted edges. Known edges are shown as open circles, while predicted edges are displayed as an "x." Proteins and their indices in all figures are: 1) ANT2, 2) APP (695), 3) B-CAT, 4) BAG3, 5) BAK, 6) Bax-beta, 7)Bcl-xL, 8) BCL2A1, 9)Bcl2-alpha, 10) Calsenilen, 11) CAV1, 12) CHIP, 13) CIB, 14) D-CAT, 15) DRAL, 16) FLN1, 17) FLNB, 18) GAPCenA, 19) GDI1, 20) GDI2, 21) GGTB, 22) GTPBP1, 23) HSPA4, 24) HSPA8, 25) KSR1, 26) MCL1, 27) MRJ, 28) PSAP, 29) PKP4, 30) PLCG1, 31) PS1 (467), 32) PS2 (448), 33) QM, 34) RAB11A, 35) RAB3A, 36) RAB5A, 37) RAB6, 38) RAB6KIFL, 39) TF, 40) TTC1. Values given in parentheses for proteins 2, 31, and 32 refer to alternative splice forms.

4.2 Markov chain Monte Carlo (MCMC) simulations

We used a MCMC simulation approach for computing the posterior probabilities of all edges within the network [18, 19]. This approach, particularly useful in generating posteriors from complicated distributions, allowed us to adequately sample from the astronomically large number of possible network configurations (for $|V|$ vertices there are $2^{|V||V|}$ possible networks). In our approach we used a uniform prior distribution over all networks, as we had no prior information that would cause us to prefer one network over another. Starting with an arbitrary network, and using a reversible—jump methodology [20], edges were both added and removed at each iteration of the algorithm. Addition and removal of edges moves the network from the current state X to a proposed state Y. Using a symmetric proposal distribution, the new state is accepted with probability

$$\alpha(x, y) = \min\left\{1, \frac{L(Y)}{L(X)}\right\},$$

where $L(.)$ is the likelihood of the network. If the proposed state is accepted, it becomes the current state. This method thus samples networks from the space of all

possible networks while keeping each edge occupied, or unoccupied over time, in proportion to its posterior probability.

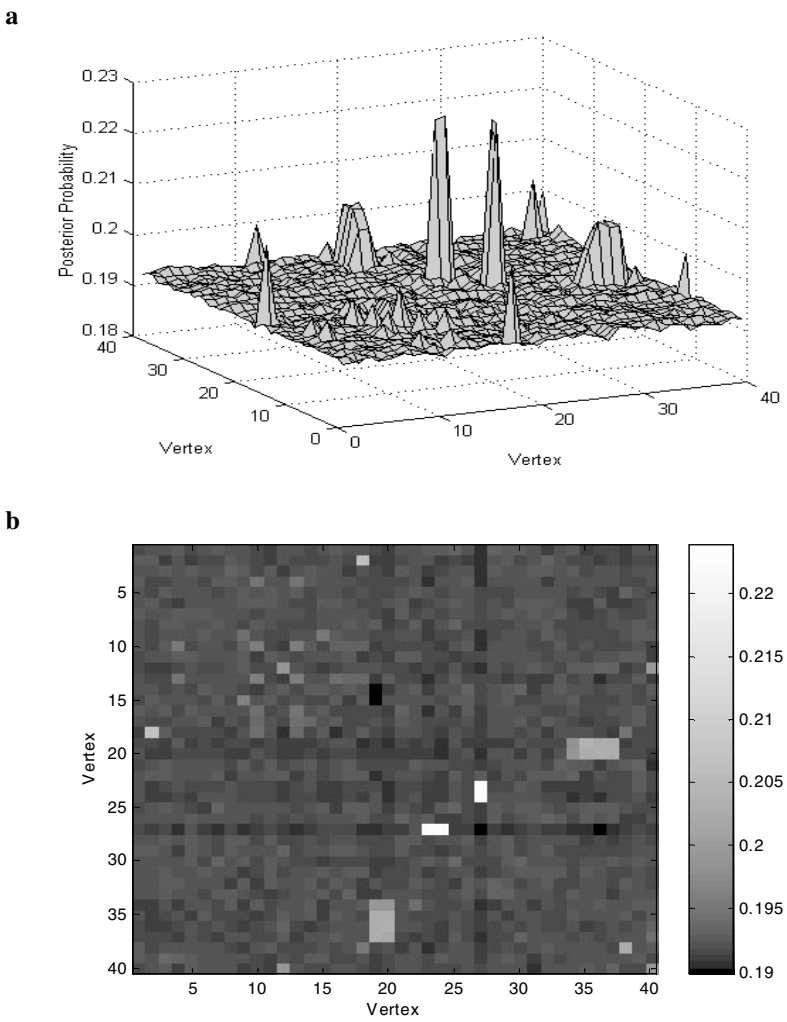

Figure 2. Network posterior probabilities. See text for further details.

The posterior distribution generated from approximately 10^7 samples is shown in Figure 2a and b. In 2a it can be seen that a few edges are readily apparent; rising well above the surrounding background. The two tallest peaks are of the HSPA8—MRJ interaction. Edges such as these show up rapidly in our simulations, while low—

probability edges can take considerably greater amounts of sampling to distinguish them from background. Figure 2b shows the posterior probabilities for each edge of the network. The lower probability (darker) "lines" running horizontally at vertices 20 and 27 and vertically along vertex 27 show the influence of the nonsymmetrical edge distributions. For example, since vertex 27 has a relatively high probability connection, the edge distribution tends to suppress the addition of new edges to the same vertex. Of course any vertex can have multiple incoming and outgoing edges, however due to the scale—free property of these networks, highly connected vertices are relatively rare.

5. Discussion

Obviously, we would have preferred to have better accuracy in our predictions, though the prediction of a few edges not in the data set and that were able to be confirmed through a literature search was very encouraging. However due to inadequacies in the training data and current limitations of the model, a large number of potential errors is unavoidable at this time. In our previous analysis, we used cross-validation to measure the effectiveness of the model. By starting with a large network (642 edges), and either adding or removing a single edge and determining whether the network probability increased or decreased as a result, we estimated 7% false negative and 10% false positive error rates [3]. The study described here, however, was significantly more challenging. We should also note that it is extremely difficult to evaluate the true accuracy of these results. Of the edges that could not be matched to known edges, it is quite possible that some of these are also correctly predicted. In fact, a primary goal of this effort is to generate just such predictions of currently unknown interactions.

Our approach, however, is primarily limited by the use of a rather limited set of previously defined domains. For example, of the 6202 proteins within yeast, nearly 40% were unable to be assigned any type of domain. Furthermore, of the 2238 edges used here, only 708 originated and terminated at proteins each with at least one domain. Similar limitations are seen within the human data. In addition, while yeast proteins tend to be characterized by a single domain, multidomain proteins are closer to the norm in humans, and thus the limited amount of training data is again a factor. Obviously, the lack of adequate domain coverage presents serious difficulties, as our model requires at least 1 instance of a particular domain—domain interaction in the training set to predict it in "real" data. To address this issue, we are in the process of developing a method that should be capable of providing 100% coverage.

As discussed in the model description, we currently use a multinomial distribution to characterize the distribution of edges going into and out of each vertex of the network, with the bin probabilities taken from fits to yeast data. While not optimal,

the use of yeast parameters seemed an acceptable first—pass attempt as, for example, edge distributions from metabolic networks (which also follow power—law behavior) have been shown to be very similar across species [21]. While we plan to acquire distributions for a number of species, it appears that the lack of reasonably large data sets could be a hindrance, with improper edge distributions perhaps masking interactions that would otherwise be apparent, particularly in interspecies predictions. In the interim, we plan to use parameters from a well—characterized system (e.g. yeast) in a distribution with identical mean but with greater variance. This requirement can be fulfilled with the incorporation of the negative multinomial distribution (instead of the multinomial distribution) into our simulations, defined as

$$P(n_1,n_2,\ldots n_k)=\frac{\Gamma\left(N+\sum_{i=1}^{k}n_i\right)}{\left(\prod_{i=1}^{k}n_i!\right)\Gamma(N)}Q^{-N}\prod_{i=1}^{k}\left(\frac{P_i}{Q}\right)^{n_i}\;;\quad\left(n_j\geq 0\right)\;\cdot$$

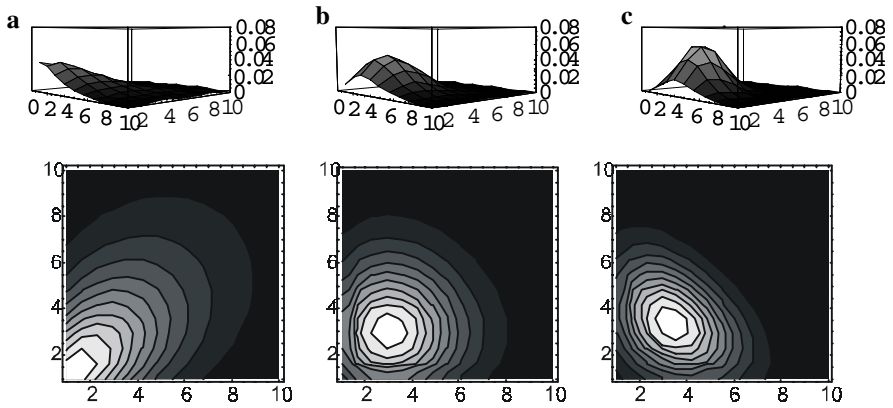

Figure 3. The negative multinomial distribution is an alternative to the multinomial. Parts (a) & (b) show the negative multinomial (suface plot above its corresponding contour plot) in comaprison to the multinomial distribution in (c). For the multinomial, $P_i = 0.25$ and N = 14. For the neg. multinomial, P_i was set equal to 0.25 times a constant, with *NP* held constant. For part (a) the const = 4, while for part (b), const = $1\text{x}10^{-6}$. See text for further details.

In Figure 3a and b, we show the negative multinomial with different parameters P_i, while Figure 3c shows a multinomial distribution. It can be seen that by increasing P_i we are able to increase the variance of the distribution while keeping the expected value identical to the multinomial distribution shown in part c. Note that

while we can match the expected value, we can only generate a variance that is greater than, but not equal to, the multinomial's. This is because a negative binomal distribution tends to a Poisson distribution as the variance decreases, and the Poisson distribution has typically larger variance than a multinomial distribution with the same mean.

From an implementation standpoint, this approach, while capable of handling large networks, benefits significantly from the use of appropriate computational resources. We ran a C programming language implementation of our method, which proved to be significantly more rapid than our previous implementation in Matlab. In addition, we have the benefit of a 5—node Beowulf cluster running Linux, with each node having 2, 1GHz CPUs. The availability of appropriate hardware and software was invaluable, as it can take a considerable amount of time to establish a stationary distribution (1-2 days in this case) and to generate the posterior (many days to generate a posterior with resolution of low—probability edges).

In the future, we plan on implementing "repulsive" interactions between domains. This can be achieved by assigning domain—domain interaction probabilities of < 0.5 to interactions that are never present. While requiring careful normalization and balancing with "attractive" probabilities, this feature should provide enhanced resolution of predicted interactions (bigger peaks and deeper valleys in the posterior plots). While having its own set of favorable and unfavorable properties, two—hybrid data should prove particularly valuable for this approach.

6. Conclusion

This work has attempted to describe a probabilistic approach to the prediction of protein—protein interactions *a priori*. Other approaches to predicting protein interactions are also being developed. Recently, work by Bock and Gough [22] described a Support Vector Machine approach for this prediction. This approach was based on primary structural data (the protein sequence) and utilized the DIP database for training data. A benefit of the approach described here is that we can assign probabilities to both edges and to complete networks (or subgraphs). Given a target protein(s), a ranking of most likely interaction candidates can be generated directly, providing some direct measure as to how confident one is as to the existence of a given interaction.

Our use of Markov chain Monte Carlo techniques provides a computationally feasible way to calculate the posterior probability of a network given data as:

$$P(network_i \mid data) = \frac{P(data \mid network_i)P(network_i)}{\sum_{all\ networks} P(data \mid network_j)P(network_j)} \quad .$$

While we have assumed a uniform prior distribution over all possible networks, the model does not require this. This framework allows new information (in the form of priors) to be added into the calculation as it becomes available.

In summary, while requiring further improvement, we feel that this approach holds significant potential. Its Bayesian basis allows the integration of disparate types of data into a single prediction. The discussed improvements should allow for more accurate predictions of both known and unknown interactions and will hopefully provide predictions of some value to the biological community.

References

[1] E. M. Marcotte, M. Pellegrini, et al., *Science* **285**, 751 (1999)
[2] A. J. Enright, I. Iliopoulos, et al., *Nature* **402**, 86 (1999)
[3] S. M. Gomez, S.-H. Lo, et al., *Genetics* (to appear, 2000)
[4] A. Bateman, E. Birney, et al., *Nucleic Acids Res* **28**, 263 (2000)
[5] G. Apic, J. Gough, et al., J. Mol. Biol. **310**, 311 (2001)
[6] I. Xenarios, E. Fernandez, et al., *Nucleic Acids Res* **29**, 239 (2001)
[7] I. Xenarios, D. W. Rice, et al., *Nucleic Acids Res* **28**, 289 (2000)
[8] H. Jeong, S. P. Mason, et al., *Nature* **411**, 41 (2001)
[9] A. L. Barabasi and R. Albert, *Science* **286**, 509 (1999)
[10] R. Albert, H. Jeong, et al., *Nature* **406**, 378 (2000)
[11] T. Ito, K. Tashiro, et al., *Proc Natl Acad Sci U S A* **97**, 1143 (2000)
[12] P. Uetz, L. Giot, et al., *Nature* **403**, 623 (2000)
[13] D. M. Hutt, L. F. Da-Silva, et al., *J Biol Chem* **275**, 18511 (2000)
[14] S.-K. Wu, P. Luan, et al., *J. Biol. Chem.* **273**, 26931 (1998)
[15] O. Ullrich, H. Stenmark, et al., *J. Biol. Chem.* **268**, 18143 (1993)
[16] C. A. Ballinger, P. Connell, et al., *Mol. Cell. Biol.* **19**, 4535 (1999)
[17] H. Awata, C. Huang, et al., *J. Biol. Chem.* **4**, 4 (2001)
[18] W. R. Gilks, S. Richardson, et al., "Markov chain Monte Carlo in practice." (Chapman & Hall/CRC, New York, 1996)
[19] W. K. Hastings, *Biometrika* **57**, 97 (1970)
[20] P. J. Green, *Biometrika* **82**, 711 (1995)
[21] H. Jeong, B. Tombor, et al., *Nature* **407**, 651 (2000)
[22] J. R. Bock and D. A. Gough, *Bioinformatics* **17**, 455 (2001)