BIOINFORMATICS

Vol. 19 no. 15 2003, pages 1875–1881
DOI: 10.1093/bioinformatics/btg352

Learning to predict protein–protein interactions from protein sequences

Shawn M. Gomez[1,*], William Stafford Noble[2] and Andrey Rzhetsky[3]

[1] Unité de Biochimie et Biologie Moléculaire des Insectes, Institut Pasteur, 75724 Paris Cedex 15, France, [2] Department of Genome Sciences, University of Washington, Seattle, USA and [3] Columbia Genome Center, Center for Computational Biology and Bioinformatics (C_2B_2), Department of Biomedical Informatics, Columbia University, New York, USA

Received on May 1, 2003; revised and accepted on July 15, 2003

ABSTRACT

In order to understand the molecular machinery of the cell, we need to know about the multitude of protein–protein interactions that allow the cell to function. High-throughput technologies provide some data about these interactions, but so far that data is fairly noisy. Therefore, computational techniques for predicting protein–protein interactions could be of significant value. One approach to predicting interactions *in silico* is to produce from first principles a detailed model of a candidate interaction. We take an alternative approach, employing a relatively simple model that learns dynamically from a large collection of data. In this work, we describe an *attraction–repulsion* model, in which the interaction between a pair of proteins is represented as the sum of attractive and repulsive forces associated with small, domain- or motif-sized features along the length of each protein. The model is discriminative, learning simultaneously from known interactions and from pairs of proteins that are known (or suspected) not to interact. The model is efficient to compute and scales well to very large collections of data. In a cross-validated comparison using known yeast interactions, the attraction–repulsion method performs better than several competing techniques.

Contact: sgomez@pasteur.fr

INTRODUCTION

Stable homeostasis of a living cell is a direct result of the coordinated sequence of a large number of molecular interaction events. Among the numerous molecules participating in such interactions, proteins are probably the most important players. In particular, proteins transmit regulatory signals throughout the cell, catalyze a tremendous number of chemical reactions, and are critical for the stability of numerous cellular structures. While progress is being made in the identification of network components, significant challenges remain in their accurate characterization. For example, the total number of possible protein interactions within one cell is astronomically large, a potentially limiting factor for experimental analyses. In addition, some experimental methods suffer from high rates of both false positive and false negative predictions (Legrain *et al*., 2001; Edwards *et al*., 2002). In particular, recent work indicates that interactions found by such screens are far from complete, with thousands to tens of thousands of interactions as yet unknown within yeast (von Mering *et al*., 2002; Tong *et al*., 2002; Mrowka *et al*., 2001). As a result, complementary *in silico* methods capable of accurately predicting interactions would be of considerable value.

A number of approaches for predicting either physical interactions or functional relationships between proteins have been developed. These approaches consider such information as the conservation of gene order across genomes (Dandekar *et al*., 1998; Overbeek *et al*., 1999), the conservation of specific sets of proteins across species (Pellegrini *et al*., 1999), and the fusion of two separate proteins in one species into a single protein in another (Marcotte *et al*., 1999; Enright *et al*., 1999). Evolutionary information contained within phylogenetic trees likewise provides predictive power (Pazos and Valencia, 2001). Furthermore, with the recent development of high-throughput methods such as the yeast two-hybrid system, new techniques have been developed that exploit this experimental protein interaction data directly in the prediction of interactions among proteins (Bock and Gough, 2001; Wojcik and Schachter, 2001; Gomez *et al*., 2001; Sprinzak and Margalit, 2001; Deng *et al*., 2002). These techniques attempt to discover combinations of protein features in training data, such as protein domains and stretches of positive and negative charges, that have predictive value when applied to novel proteins.

In this study, we describe the extension of one of these methods (Gomez *et al*., 2001) and, using a comprehensive set of protein–protein interactions from *Saccharomyces cerevisiae*, compare its performance with two other algorithms that have also been used for interaction prediction, the support vector machine (SVM) and a method described by Sprinzak and Margalit (2001). The method described here is probabilistic,

[*] To whom correspondence should be addressed.

and in its new form, allows the incorporation of both ‘positive’ as well as ‘negative’ information (interactions that are not believed to exist within the cell). Results of this evaluation show that this model outperforms those used for comparison, providing the potential for more reliable inference of protein interactions.

ALGORITHM

The model described here assumes that evolutionarily conserved features within each protein of an interacting pair are responsible for the interaction. Interacting pairs found in one region of a network will often have homologs in another region of the same network. Our assumption of evolutionary redundancy implies that features that have predictive value for a portion of a large network will also have predictive value in the unknown part of the same network. Note also that in the present context we are primarily concerned with the prediction of physical or ‘binding’ interactions; however, functional relationships are also amenable to analysis within this framework.

We represent a molecular network as either a directed or undirected graph $G = \langle V, E\rangle$, with vertices V and edges E. Here, vertices correspond to proteins, and edges correspond to physical interactions between them. We assume that any pair of proteins within this network may potentially interact, and we assign probabilities to all such interactions. The model describing the stochastic generation of a complete network consists of two probabilistic components, *global* and *local*, which are assumed to be independent. The first component models the topology of a given network, while the second deals with individual interactions between pairs of proteins. Together, these two components can be used to predict the probability of any particular arrangement of edges among a set of proteins. The probability of a specific network Ω is thus $P(\text{data}|\Omega) = P(\text{global})P(\text{local})$.

For completeness we now describe the full model; however, the work described here focuses on predicting individual protein–protein interactions, and thus implements only the aspects of $P(\text{local})$ that carry out this function. We make this simplification because the methods used for comparison make predictions for individual edges and not complete networks. Also, note that $P(\text{global})$ can be used with any method that provides a network topology as input.

Given a set of vertices, the global model $P(\text{global})$ assigns probabilities to all possible edges among them. For an undirected graph, $P(\text{global})$ is a multinomial distribution with parameters M (equal to $|V|$) and π_i, where $i = 0, 1, \ldots, M$. For non-zero i the values π_i are generated according to a Zeta distribution [or a discrete power-law distribution Johnson and Kotz (1969)]

$$\pi_i = ci^{-\gamma}, \tag{1}$$

where parameters c and γ are assumed to be known from previous studies. For $i = 0$ we use $\pi_0 = 1 - \sum_{k=1}^{M} \pi_k$. Note that Equation (1) has been previously shown to describe the connectivity of metabolic and protein interaction networks (Jeong *et al.*, 2000, 2001; Gomez *et al.*, 2001). Thus $P(\text{global})$ assigns higher probabilities to those networks that ‘look’, in terms of their connectivity distributions, more biologically realistic.

In a complementary manner, $P(\text{local})$ assigns a probability to any set of edges that connect a group of vertices. Specifically,

$$P(\text{local}) = \prod_{(v_i,v_j)\in E} \hat{p}(v_i, v_j) \prod_{(v_i,v_j)\notin E} [1 - \hat{p}(v_i, v_j)], \tag{2}$$

where $\hat{p}(v_i, v_j)$ is the estimated individual edge probability between vertices v_i and v_j. Thus, when assigning a probability to a particular arrangement of edges, both existing and missing edges are taken into account. We now describe how these individual edge probabilities are calculated.

In this model, proteins are treated as sets or ‘bags’ of domains, and we assume that at least two features, one from each protein, are required for an interaction to exist between a protein pair. Because proteins are assumed to consist of multiple domains, all domains are considered in the context of protein–protein interactions and hence fall into two categories: domains that are informative with respect to predicting protein–protein interactions and domains that are not. Informative domains may indicate either the presence of an edge (e.g. domains that can physically interact), or the absence of an edge (e.g. a pair of domains that never occurs in interacting proteins). Non-informative domains are those that are distributed randomly and uniformly with regard to the ability of a protein to participate in interactions. In estimating domain–domain attraction probabilities, we want an estimator that gives an expected value of 0.5 for non-informative domain pairs, while giving estimates greater than 0.5 for edge-present domain pairs, and estimates below 0.5 for edge-absent domain pairs.

Let us consider a pair of uninformative protein domains, ϕ and ψ, that are distributed with uniform probability over the whole protein universe with densities ρ_ϕ and ρ_ψ, respectively. Imagine that we are estimating the probability of observing an interaction between a pair of proteins, one of which has domain ϕ and the other domain ψ, from a protein–protein interaction network with $|V|$ vertices (proteins) and $|E|$ undirected edges (real protein–protein interactions, each interaction counted just once). The expected number $N^{+}_{\phi\psi}$ of interacting pairs of proteins that have domains ϕ and ψ in separate proteins is

$$N^{+}_{\phi\psi} = |E| \cdot 2 \cdot \rho_\phi \cdot \rho_\psi. \tag{3}$$

Similarly, the expected number of non-interacting pairs of proteins containing domains ψ and ϕ in the same network is given by

$$N^{-}_{\phi\psi} = \left(\frac{|V|(|V| - 1)}{2} + |V| - |E|\right) \cdot 2 \cdot \rho_\phi \cdot \rho_\psi. \tag{4}$$

We are trying to find an estimator of the form

$$\hat{p}(\phi, \psi) = \frac{n^+_{\phi\psi}}{n^+_{\phi\psi} + \gamma n^-_{\phi\psi}}, \tag{5}$$

where $n^+_{\phi\psi}$ and $n^-_{\phi\psi}$ are, respectively, the number of times domain pair (ϕ, ψ) is seen in interacting and non-interacting proteins. (The observed random variables $n^+_{\phi\psi}$ and $n^-_{\phi\psi}$ have the expected values $N^+_{\phi\psi}$ and $N^-_{\phi\psi}$, respectively.) Parameter γ is a weighting coefficient that is selected so that the expectation of $\hat{p}(d_i, d_j)$ is equal to 0.5. In other words, we need to solve with respect to γ the following equation:

$$\frac{1}{2} = \frac{N^+_{\phi\psi}}{N^+_{\phi\psi} + \gamma N^-_{\phi\psi}}. \tag{6}$$

Doing so gives us the optimum value

$$\gamma = \frac{|E|}{|V|(|V| - 1)/2 + |V| - |E|}. \tag{7}$$

Note that, in the case of a *directed* network, γ is replaced by

$$\gamma = \frac{|E|}{|V|^2 - |E|}. \tag{8}$$

Further, we need to ensure that in the absence of observations ($n^+_{\phi\psi} = n^-_{\phi\psi} = 0$), we still have a non-zero probability of interaction between domains ϕ and ψ. This restriction can be accomplished by introducing a pseudocount, Ψ, in the following way:

$$\hat{p}(\phi, \psi) = \frac{n^+_{\phi\psi} + \Psi/2}{n^+_{\phi\psi} + \gamma n^-_{\phi\psi} + \Psi}, \tag{9}$$

where the value of Ψ is set to 0.01. In this work, we also compare the results of this model with our earlier one, which calculated the domain–domain interaction probability as

$$\hat{p}(\phi, \psi) = \frac{1}{2}\left(1 + \frac{n^+_{\phi\psi}}{n_\phi n_\psi + \Psi}\right), \tag{10}$$

where n_ϕ and n_ψ are the number of vertices containing domains ϕ and ψ, respectively, see Gomez *et al.* (2001); Gomez and Rzhetsky (2002). We refer to the previous model as the *attraction model*, and the model presented here as *attraction–repulsion model* due to the inclusion of negative interactions $n^-_{\phi\psi}$.

For the attraction–repulsion model, we combine multiple domain–domain interaction probabilities into a single edge probability by taking only the single most informative domain–domain probability

$$\hat{p}(v_i, v_j) = \underset{\hat{p}(\phi,\psi)}{\operatorname{argmax}} |(\hat{p}(\phi, \psi) - 0.5)| \tag{11}$$

for all $\phi \in v_i$, $\psi \in v_j$.

Similarly, for the attraction model we use our original method of averaging over all domain–domain interactions (Gomez *et al.*, 2001):

$$\hat{p}(v_i, v_j) = \sum_{\phi \in v_i} \sum_{\psi \in v_j} \frac{\hat{p}(\phi, \psi)}{|v_i||v_j|}, \tag{12}$$

where $|v_i|$ is the number of distinct protein domains observed in protein v_i.

METHODS

Data

In our experiments we use yeast protein–protein interaction data collected in an independent study (von Mering *et al.*, 2002). These data comprise interactions identified via six different methods: high-throughput yeast two-hybrid, correlated mRNA expression, genetic interaction (synthetic lethality), tandem affinity purification, high-throughput mass-spectrometric protein complex identification and computational methods. All interactions were classified into one of three confidence categories, high-, medium- and low-confidence, based on the number of different methods that identify an interaction as well as the number of times the interaction is observed. For the purposes of these experiments, high- and medium-confidence interactions are labelled ‘positive’, and low-confidence interactions are considered ‘unknown’. All other interactions are labelled ‘negative’. Clearly, this latter set of proteins will contain many errors—pairs of proteins that interact but are not known to do so. Consequently, the resulting performance measurements will tend to overestimate the false positive rate. However, without relying upon simulated data, this overestimation is currently unavoidable. Furthermore, for the purposes of comparing prediction algorithms, the resulting inaccuracy will be approximately uniform with respect to each computational method that we consider.

Feature extraction

In order to represent a pair of proteins, we consider two different types of features. The first is computed using a collection of hidden Markov models (HMMs) of protein domains from the Pfam 7.2 database (Sonnhammer *et al.*, 1997; Bateman *et al.*, 2002). These models represent evolutionarily conserved structures and are assumed to be related to protein function. Using HMMER 2.0 (Eddy, 1998), we compute the E-value of the best match of each Pfam domain to each yeast protein. Each such E-value serves as one feature in this representation.

The Pfam E-values, as well as the interaction data, are used to reduce the size of the data set. Following (Sprinzak and Margalit, 2001), we eliminate from consideration all proteins that do not match at least one Pfam model with an E-value less than 0.01. From within this set, we select all proteins that interact with at least one other protein. A total of

1714 proteins, containing 1015 different domain types, satisfy these criteria. The resulting set of 1.46 million protein pairs contains 7735 positive interactions and 19315 unknown interactions. The unknown interactions are not considered further.

This collection of 1714 proteins is also characterized using a second type of feature. Rather than identifying protein domains, these *4-tuple* features attempt to identify short amino acid subsequences that occur in interacting proteins. To compute these features, the sequence alphabet is first reduced from 20 amino acids to six categories of biochemical similarity [{IVLM}, {FYW}, {HKR}, {DE}, {QNTP}, and {ACGS} (Taylor and Jones, 1993)]. After this reduction, there are $6^4 = 1296$ possible substrings of length 4. For a given protein sequence, the 4-tuple feature representation is simply a binary vector of length 1296, in which each bit indicates whether the corresponding length-4 string occurs in the protein.

Support vector machine

The SVM is a binary classification algorithm (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000). As such, it is well suited to the task of discriminating between interacting and non-interacting protein pairs. The algorithm has found wide application in many fields (Cristianini and Shawe-Taylor, 2000), including bioinformatics applications such as recognition of translation start sites, protein remote homology detection, microarray gene expression analysis, functional classification of promoter regions, and peptide identification from mass spectrometry data. SVMs have previously been used in the prediction of protein–protein interactions (Bock and Gough, 2001). The experiments reported here employ the Gist 2.0 software (P. Pavlidis, I. Wapinski and W.S. Noble, submitted for publication) with the default parameter settings, including a linear kernel function and a 2-norm soft margin. Each pair of proteins is represented via the concatenation of the corresponding Pfam vectors. For fairness of comparison, we do not attempt to optimize the free parameters of the SVM, although experiments performed on a subset of the data indicate that varying the most important parameter (the weight associated with the soft margin) does not significantly improve the results (data not shown).

High-scoring sequence signatures

We also compare our method with that of Sprinzak and Margalit (2001). This method is similar to our original model (Gomez *et al*., 2001) in that it is trained with experimental interaction data, attempting to find domain or sequence signature pairs that are highly correlated with protein interactions. The model itself is straightforward, involving the creation of a contingency table detailing the number of times domain ϕ is seen in combination with domain ψ within an interaction. An interaction between a pair of proteins is predicted to take place if any single domain pair between a pair of proteins has a score S above a predefined threshold

$$S = \log_2\left(\frac{f_{\phi\psi}}{f_\phi f_\psi}\right), \tag{13}$$

where $f_{\phi\psi} = n^+_{\phi\psi}/|E|$, and $f_\phi = n_\phi/|V|$. When the observed frequency is zero, a dummy score is assigned (in this case -10.0) that is smaller than the minimum value found in the training data.

Note that for non-informative domains ϕ and ψ the expected value of f_ϕ is equal to ρ_ϕ, while the expected value of $f_{\phi\psi}$ is equal to $2 \cdot \rho_\phi \cdot \rho_\psi$. Therefore, the expected value of score S for non-informative domains is equal to one.

Cross-validation and scoring

The performance of each prediction algorithm is measured using three-fold cross-validation. In this paradigm, the data are split into three equal-sized parts. The learning algorithm is trained on two parts and tested on the remaining part. This train-test procedure is repeated three times, and the resulting collection of predictions is merged. In the experiments reported here, the entire cross-validation procedure is repeated five times in order to estimate variance.

In previously published cross-validation experiments on protein–protein interaction prediction, including our own, the cross-validation was performed on interactions. Here, by contrast, we perform cross-validation on individual proteins. Thus, each method is trained on all pairs of interactions from within a given training set of proteins, and each method is tested on interactions within a disjoint test set of proteins. In this approach, if the training set contains one interaction between protein A and B, then the test set will not contain an interaction between protein A and some other protein C. Performing cross-validation, as we do, on proteins rather than on interactions, ensures that the algorithm learns about the way proteins interact in general, rather than about the interaction characteristics of individual proteins.

The quality of a set of predictions is measured using the receiver operating characteristic (ROC) score. The output of each prediction algorithm is a ranking of protein pairs, in which the pair of proteins at the top of the list is estimated to be the most likely to interact. This ranked list can be converted into a set of binary predictions by applying a decision threshold by comparing the decision threshold T with the score s associated with each pair of proteins. If $s \geq T$, then the proteins are predicted to interact; otherwise, they are not. Rather than measuring the accuracy of the predictions resulting from a single decision threshold, the ROC score integrates over all possible decision thresholds, thereby evaluating the quality of the entire ranking. The ROC curve plots, for varying decision thresholds, the true positive rate as a function of false positive rate (Hanley and McNeil, 1982; Gribskov and Robinson, 1996), and the ROC score is the area under this curve. For a perfectly random classifier, these two rates will be approximately equal, yielding a diagonal curve and a score

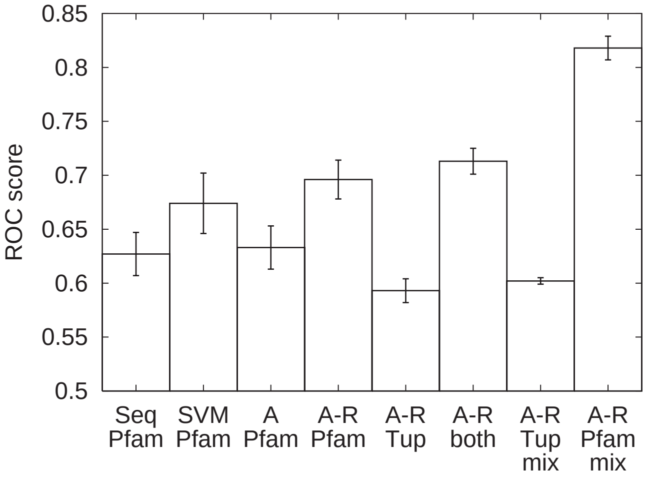

Fig. 1. ROC score summary. ROC scores are shown for eight different prediction methods. Under each bar, the first line of the label indicates the algorithm employed: 'Seq' for the sequence-signature method, 'SVM' for the support vector machine, 'A' for the attraction model, and 'A–R' for the attraction–repulsion model. The second line indicates the type of features used (Pfam domains, 4-tuples, or both). The methods marked 'mix' are tested on interactions for which one protein is in the training set and one is in the test set (see text). Error bars indicate standard deviation over five cross-validated tests.

of 0.5. For a perfect classifier, all of the positives will appear before all of the negatives, yielding a score of 1.0.

RESULTS

Our primary experimental results are summarized in Figure 1. These results demonstrate the utility of the attraction–repulsion model. Of the two types of features, Pfam domain profiles provide more information than 4-tuples. Overall, the prediction performance of the attraction–repulsion method is significantly better than that of the attraction-only model and the sequence-signature method. The attraction–repulsion model performs approximately as well as, but far more efficiently than, the SVM. In this section, we discuss these results in more detail.

First, note that all of the methods tested here perform significantly better than chance, indicating that some learning is occurring in every case. A random classifier receives an ROC score of 0.5, and the standard deviations on all of the observed ROC scores are clearly far better than random.

With respect to selecting a feature set, Figure 2 shows that the Pfam domain features are more informative than the 4-tuple features. The figure shows histograms of domain–domain interaction probabilities, computed from both types of features using Equation (9). Tuples have a mean probability of 0.46. This result implies that the majority of tuples carry very little information, i.e. nearly all are near the completely uninformative value of 0.5. The slight negative shift of the distribution suggests that when tuples are informative, it is with regard to the absence of an edge. By contrast, the shape of the corresponding distribution for Pfam domains suggests that a much greater proportion of pairs are informative. The distribution mean of 0.34 again suggests that when Pfam domain pairs are informative it is with regards to the absence of an edge. Note, however, that there are a number of pairs in the 0.9–1.0 probability range, indicating the existence of a significant number of pairs that are highly correlated with the existence of an edge between a pair of proteins.

The relative value of the Pfam and tuple features is further illustrated in Figure 1. The histogram bars labeled 'A–R Tup' and 'A–R Pfam' correspond to the attraction–repulsion model computed with tuple and Pfam vectors, respectively. A t-test indicates that these measurements are significantly different (with 95% confidence).

Focusing only on the Pfam features, we compare the recognition performance of the attraction–repulsion model [specifically, the rankings of probabilities generated from Equation (9)] with rankings produced by sequence signatures, the SVM, and the attraction-only model [Equation (10)]. These results are shown in Figure 1, with bars labelled 'Seq Pfam', 'SVM Pfam', 'A Pfam', and 'A–R Pfam'. The best ROC score of 0.696 ± 0.018 originates from the attraction–repulsion model. The SVM approach performs similarly, with a score of 0.674 ± 0.028. Overlap in the standard deviations of these two methods suggests that they perform nearly identically, and a t-test indicates that we cannot reject (with 95% confidence) the possibility that their means are the same. On the other hand, both methods significantly outperform the sequence-signature method, which has a score of 0.627 ± 0.020, as well as the attraction-only model with a score of 0.633 ± 0.021.

Of the two best-performing methods, the attraction–repulsion model is clearly more efficient. Indeed, the SVM as formulated here is barely efficient enough to be useful. Training the SVM on the entire collection of 1.2 million labeled interactions was not feasible, especially because each interaction is represented via a vector of length 7470 (twice the total number of available Pfam domains). Luckily, preliminary experiments (data not shown) indicated that sampling randomly from the negatively labeled examples (which predominate the training set) is sufficient to yield good performance: the ROC score stops improving after approximately 1% of the negatives are used. Still, training the SVM even on this subset requires $\sim$4 h. Thereafter, simply making a prediction for a single vector requires computing a scalar product with respect to most of the training set, which amounts to $12\,000 * 7470 = 89$ million multiplies per prediction. These calculations could be sped up by using feature selection techniques to reduce the input vector size and by more strongly encouraging sparseness of the SVM solution. But the SVM training is still fundamentally an $O(n^2)$ algorithm, and will

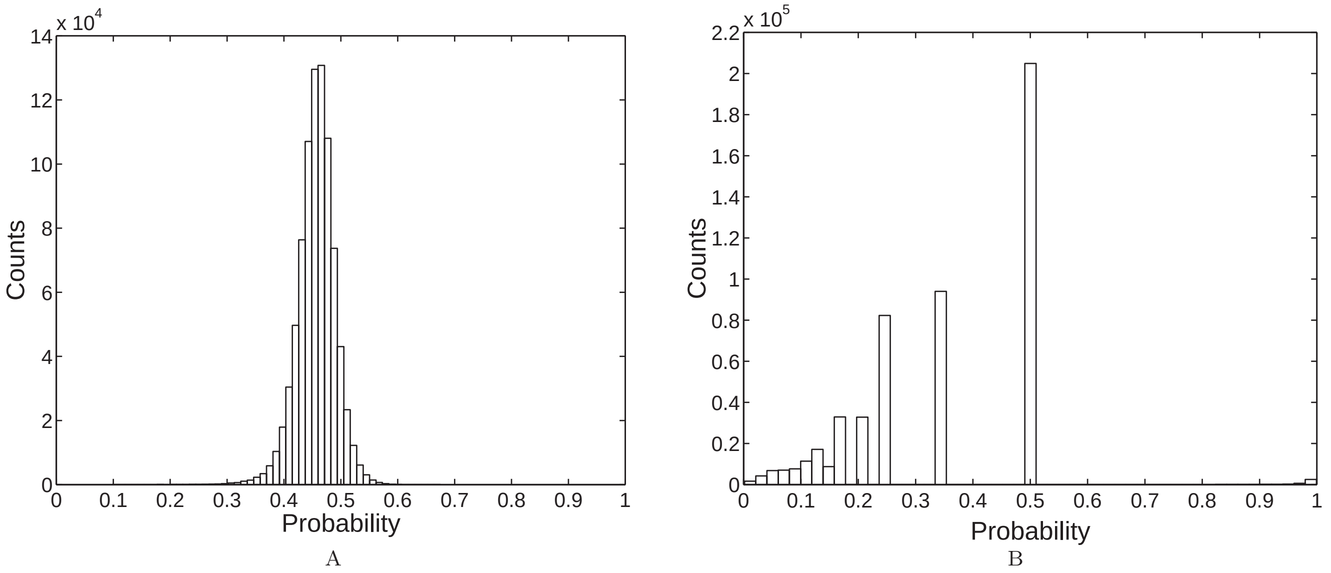

Fig. 2. Example distributions of (**A**) tuple–tuple and (**B**) Pfam domain–domain interaction probabilities. Each plot is a histogram of the probabilities calculated using Equation (9) across all features in one training set.

essentially always be slower than the much simpler, linear time attraction–repulsion model.

The cross-validated experiments described so far measure the ability of each algorithm to predict an interaction between two proteins for which no previous interactions are known. In practice, of course, a real predictor would also be tested on some proteins for which we have partial information about its interactions. In order to determine the effect of this partial information, we test the attraction–repulsion model on interactions between one training set protein and one test set protein. These results are shown in Figure 1 (bars labeled 'mix'). When learning from tuples, the effect of the partial information is minimal; however, for Pfam domains, the predictions of interactions involving a previously observed protein improve significantly, with mean ROC scores increasing from 0.696 ± 0.018 to 0.818 ± 0.011. This effect is understandable because tuples are extremely frequent, and in a given training set all tuples are observed at least once. Therefore, prediction of an interaction with a previously observed protein does not provide much additional information. On the other hand, the strong effect for Pfam domains can be attributed to the relative infrequency with which any given Pfam domain is observed. Thus, in situations where the interactions of at least some of the proteins have been observed before, we expect much higher prediction accuracies.

Finally, we briefly investigate the possibility of combining the two different sets of features, Pfam domains and 4-tuples, and learning from them simultaneously. The resulting learner, however, does not perform significantly better than a learner trained only on the Pfam domains. In order to make a prediction from combined features, we compare the probability assigned to each edge for each feature type. The final edge probability is calculated as before from each type of feature, and the maximally informative (i.e. farthest from 0.5) probability is assigned to the prediction. Thus, the final list of probabilities is a mixture of entries from both Pfam–Pfam and tuple–tuple interactions. Pfam–Pfam and tuple–tuple interactions contribute essentially equally to the final prediction with approximately 51% of predictions being generated from Pfam domains. Combining predictions in this manner generates an ROC score of 0.713 ± 0.012, shown in Figure 1 as 'A–R both'. A t-test at the 95% confidence level fails to distinguish this value from using Pfam features only. Thus, it is not clear whether this simple means of combining the Pfam and tuple–tuple interactions is of predictive value.

DISCUSSION

This work describes a probabilistic method for inferring the existence of protein–protein interactions. Within the context of our experiments, the attraction–repulsion model emerges as a strong candidate, benefiting greatly from the inclusion of negative information into predictions. In comparison, the attraction-only and sequence-signature models provide less accurate predictions, and the SVM provides comparable performance at increased computational expense. In general, the learning approach to prediction of protein–protein interactions provides the ability to better understand the functional role of newly discovered proteins by discovering potential links with proteins of known function or in previously studied pathways.

With the increasing amounts of molecular data, continued development of this and related methods will provide a useful set of tools for the understanding of molecular function.

Our results point to several paths for future research. For example, the SVM approach described here is simplistic and could be improved via a better feature selection algorithm, a knowledge-based kernel function, or by using a more light-weight learning algorithm, such as an ensemble of perceptrons. Similarly, we have experimented here only minimally with techniques for combining information from multiple sequence-based feature types. We expect to gain significant power by using more sophisticated data fusion methods across a wider variety of data types. Finally, the scope of this investigation can be expanded in several directions, including employing our global model and learning from interaction data from multiple species.

ACKNOWLEDGEMENTS

S.M.G. is supported by a grant from the Pasteur Foundation of New York and the Strategic Horizontal Programme on *Anopheles gambiae* at Institut Pasteur. W.S.N. is supported by a Sloan Research Fellowship and by National Science Foundation grants DBI-0078523 and ISI-0093302. A.R. is supported by grants EIA-0121687, DE-FG02-01ER25500, and GM61372 from National Science Foundation, Department of Energy, and National Institutes of Health, respectively.

REFERENCES

Bateman,A., Birney,E., Cerruti,L., Durbin,R., Etwiller,L., Eddy,S.R., Griffiths-Jones,S., Howe,K.L., Marshall,M. and Sonnhammer,E.L. (2002) The pfam protein families database. *Nucleic Acids Res.*, **30**, 276–280.

Bock,J.R. and Gough,D.A. (2001) Predicting protein–protein interactions from primary structure. *Bioinformatics*, **17**, 455–460.

Cristianini,N. and Shawe-Taylor,J. (2000) *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK.

Dandekar,T., Snel,B. and Bork,P. (1998) Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem. Sci.*, **23**, 324–328.

Deng,M., Mehta,S., Sun,F. and Chen,T. (2002) Inferring domain–domain interactions from protein–protein interactions. *Genome Res.*, **12**, 1540–1548.

Eddy,S. (1998) Profile hidden markov models. *Bioinformatics*, **14**, 755–763.

Edwards,A.M., Kus,B., Jansen,R., Greenbaum,D., Greenblat,J. and Gerstein,M. (2002) Bridging structural biology and genomics: assessing protein interaction data with known complexes. *Trends Genet.*, **18**, 529–536.

Enright,A.J., Iliopoulos,I., Kyrpides,N.C. and Ouzounis,C.A. (1999) Protein interaction maps for complete genomes based on gene fusion events. *Nature*, **402**, 86–90.

Gomez,S.M., Lo,S.H. and Rzhetsky,A. (2001) Probabilistic prediction of unknown metabolic and signal-transduction networks. *Genetics*, **159**, 1291–1298.

Gomez,S.M. and Rzhetsky,A. (2002) Towards the prediction of complete protein–protein interaction networks. *Proceedings of the Pacific Symposium on Biocomputing*, 413–424.

Gribskov,M. and Robinson,N.L. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput. Chem.*, **20**, 25–33.

Hanley,J.A. and McNeil,B.J. (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, **143**, 29–36.

Jeong,H., Mason,S.P., Barabasi,A.L. and Oltvai,Z.N. (2001) Lethality and centrality in protein networks. *Nature*, **411**, 41–42.

Jeong,H., Tombor,B., Albert,R., Oltvai,Z.N. and Barabasi,A.L. (2000) The large-scale organization of metabolic networks. *Nature*, **407**, 651–654.

Johnson,N. and Kotz,S. (1969) Discrete distributions. *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics*. Wiley, New York.

Legrain,P., Wojcik,J. and Gauthier,J.-M. (2001) Protein–protein interaction maps: a lead towards cellular functions. *Trends Genet.*, **17**, 346–352.

Marcotte,E., Pellegrini,M., Ng,H.-L., Rice,D., Yeates,T. and Eisenberg,D. (1999) Detecting protein function and protein–protein interactions from genome sequences. *Science*, **285**, 751–753.

Mrowka,R., Patzak,A. and Herzel,H. (2001) Is there a bias in proteome research? *Genome Res.*, **11**, 1971–1973.

Overbeek,R., Fonstein,M., D'Souza,M., Pusch,G.D. and Maltsev,N. (1999) The use of gene clusters to infer functional coupling. *PNAS*, **96**, 2896–2901.

Pazos,F. and Valencia,A. (2001) Similarity of phylogenetic trees as indicator of protein–protein interaction. *Protein Eng.*, **14**, 609–614.

Pellegrini,M., Marcotte,E.M., Thompson,M.J., Eisenberg,D. and Yeates,T.O. (1999) Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *PNAS*, **96**, 4285–4288.

Sonnhammer,E., Eddy,S. and Durbin,R. (1997) Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins*, **28**, 405–420.

Sprinzak,E. and Margalit,H. (2001) Correlated sequence-signatures as markers of protein–protein interaction. *J. Mol. Biol.*, **311**, 681–692.

Taylor,W.R. and Jones,D.T. (1993) Deriving an amino acid distance matrix. *J. Theor. Biol.*, **164**, 65–83.

Tong,A.H.Y., Drees,B., Nardelli,G., Bader,G.D., Brannetti,B., Castagnoli,L., Evangelista,M., Ferracuti,S., Nelson,B., Paoluzi,S. *et al*. (2002) A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science*, **295**, 321–324.

Vapnik,V.N. (1998) Adaptive and learning systems for signal processing, communications, and control. *Statistical Learning Theory*. Wiley, New York.

von Mering,C., Krause,R., Snel,B., Cornell,M., Oliver,S.G., Fields,S. and Bork,P. (2002) Comparative assessment of large-scale data sets of protein–protein interactions. *Nature*, **417**, 399–403.

Wojcik,J. and Schachter,V. (2001) Protein–protein interaction map inference using interacting domain profile pairs. *Bioinformatics*, **17**, 296S–305S.