BIOINFORMATICS

Vol. 20 no. 0 2004, pages 1–9
doi:10.1093/bioinformatics/bth483

Predicting protein–protein interactions using signature products

Shawn Martin[1,*], Diana Roe[2] and Jean-Loup Faulon[3]

[1]Sandia National Laboratories, Computational Biology, 9212, P.O. Box 5800, MS 310, Albuquerque, NM, 87185, USA, [2]Biosystems Research and [3]Computational Biology, 9212, P.O. Box 969, MS 9951, Livermore, CA, 94551, USA

Received on March 8, 2004; revised on August 12, 2004; accepted on August 13, 2004
Advance Access publication …

ABSTRACT

Motivation: Proteome-wide prediction of protein–protein interaction is a difficult and important problem in biology. Although there have been recent advances in both experimental and computational methods for predicting protein–protein interactions, we are only beginning to see a confluence of these techniques. In this paper, we describe a very general, high- throughput method for predicting protein–protein interactions. Our method combines a sequence-based description of proteins with experimental information that can be gathered from any type of protein–protein interaction screen. The method uses a novel description of interacting proteins by extending the signature descriptor, which has demonstrated success in predicting peptide/protein binding interactions for individual proteins. This descriptor is extended to protein pairs by taking signature products. The signature product is implemented within a support vector machine classifier as a kernel function.

Results: We have applied our method to publicly available yeast, *Helicobacter pylori*, human and mouse datasets. We used the yeast and *H.pylori* datasets to verify the predictive ability of our method, achieving from 70 to 80% accuracy rates using 10-fold cross-validation. We used the human and mouse datasets to demonstrate that our method is capable of cross-species prediction. Finally, we reused the yeast dataset to explore the ability of our algorithm to predict domains.

Contact: smartin@sandia.gov.

INTRODUCTION

Protein–protein interactions are of interest in biology because they regulate a variety of cellular processes, including metabolic cycles, DNA transcription and replication, different signaling cascades and many additional processes. The importance of understanding these interactions has prompted the development of various experimental methods for measuring protein–protein interaction. These methods include two-hybrid systems (Fields and Song, 1989), mass spectrometry (Ho *et al*., 2002) and protein chips (Zhu *et al*., 2001). Of these methods, the two-hybrid method is mature enough to have been used to obtain full protein interaction networks for *Saccharomyces cerevisiae* (Ito *et al*., 2000; Uetz *et al*., 2000) and *Helicobacter pylori* (Rain *et al*., 2001).

At the same time, computational techniques have been developed to determine protein–protein interactions based on genomic sequence analysis. These techniques generally employ algorithms to search sequence information for patterns that might be expected to occur based on a priori biological knowledge. Different algorithms search for the conservation of gene neighborhoods and gene order (Dandekar *et al*., 1998), the occurrence of fused genes (Enright *et al*., 1999; Marcotte *et al*., 1999), the coevolution of interacting proteins (Pazos *et al*., 1997) and similarity of phylogentic trees (Goh *et al*., 2000; Pazos and Valencia, 2001). A good overview of these and other approaches can be found in Valencia and Pazos (2002).

An alternative approach is to use experimental data rather than a priori expectations to guide the discovery of inherent patterns in the sequence data. Both Sprinzak and Margalit (2001) and Bock and Gough (2001, 2003) use this approach. Finally, the approach in Jansen *et al*. (2003) attempts to combine both experimental and a priori knowledge when making predictions.

The method in Sprinzak and Margalit (2001) characterizes proteins using known InterPro structural domains, and then uses experimental binding data to identify structural domain pairings correlated with protein–protein binding. This approach uses the inherent assumption that all binding interactions occur within these well-defined domain–domain interactions. Bock and Gough (2001, 2003) take a more general approach by creating protein structural and physiochemical descriptors based on the primary sequence information, and then training a support vector machine (SVM) learning system to identify protein–protein binding interactions from these descriptors.

Although independent, our approach combines positives from both Bock and Gough (2001) and Sprinzak and Margalit

*To whom correspondence should be addressed.

(2001) while achieving similar or better performance. In particular, we introduce a novel and even more general descriptor called signature product, which is a product of subsequences and an expansion of the signature descriptor from chemical informatics. The fact that we use a product descriptor gives us the advantages of the method in Sprinzak and Margalit (2001) while the fact that we use primary sequence gives us the advantages of Bock and Gough (2001). Our simple and easy to calculate descriptor performs well, is symmetric, depends only on combinations of local sequence information and can handle large datasets. We have tested our algorithm using yeast, human, mouse, *H.pylori* and *Escherichia coli* datasets available over the Internet. We obtained the yeast data from Tong *et al*. (2002); Sprinzak and Margalit (2001), Jansen *et al*. (2003) and the Database of Interacting Proteins (DIP) (Xenarios *et al*., 2002); the human, mouse and *E.coli* data came from the DIP; and the *H.pylori* data came from Rain *et al*. (2001). Using these same datasets we benchmark our algorithm and provide comparisons with the works of Sprinzak and Margalit (2001); Bock and Gough (2001, 2003) and Jansen *et al*. (2003).

SYSTEMS AND METHODS

Support vector machines

SVMs are classifiers [there are also SVMs that perform regression (Smola and Schölkopf, 1998)] that are described thoroughly by their inventor (Vapnik, 1998). SVMs are very adaptable and have been applied successfully to a wide variety of problems. Recently, there has been interest in the application of SVMs to biological problems such as classification of gene expression data [e.g. Furey *et al*. (2000)], homology detection (Leslie *et al*., 2002) and prediction of protein–protein interaction (Bock and Gough, 2001), as well as many additional problems. For an introduction to SVMs, see Burges (1998) and Cristianini and Shawe-Taylor (2000).

To describe an SVM precisely, suppose our data are given as pairs $\{(x_i, y_i)\} \subset \mathbb{R}^n \times \{\pm 1\}$. In other words, suppose our data consist of two classes (1 and -1—in our case, binding and nonbinding protein pairs). Using this notation an SVM assumes the form $f(\mathbf{x}) = \sum_i \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b$, where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a decision function ($\mathbf{x}$ belongs to class 1 if $f(\mathbf{x})$ is greater than some threshold t, or to class -1 otherwise), $k\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a kernel function, otherwise known as a dot product in some vector space, and the constants b and α_i are obtained by solving a quadratic programming problem for details (see Burges, 1998). The threshold t is typically 0, although it may be varied to obtain classifiers that are more or less accurate on positive predictions.

SVMs have several advantages over other classifiers although we do not discuss them here. Instead, we refer to Vapnik (1998) and Bennett and Campbell (2000), among others. To implement the SVMs in this, we used the SVM$^{\text{light}}$ algorithm (Joachims, 1999) with a custom kernel based on signature (discussed in the next section).

Signature

One of the main computational challenges in using SVMs (or any method) for the prediction of protein–protein interactions is a suitable encoding of the protein information in some vector space. In our case, we have the problem of representing variable length amino acid sequences as vectors containing the information necessary to distinguish between binding and non-binding protein–protein pairs.

Our solution to this problem is to use the signature molecular descriptor (Visco *et al*., 2002; Faulon *et al*., 2003a,b; Churchwell *et al*., 2004).

We use a specific instance (height 1) of the more general model and formulate signature as a function s : {variable length amino acid sequences} $\to$ F defined by $s(A) = \sum_i \sigma_i \mathbf{z}_i$, where A is an amino acid sequence, $\mathbf{z}_i$ is a basis vector in the signature space $F \cong \mathbb{R}^N$ and σ_i is the number of occurrences of $\mathbf{z}_i$ in A. A signature consists of an amino acid and its neighbors, and the signature space consists of all possible signatures.

As an example, consider the six-letter amino acid sequence *LVMTTM*. All height 1 signatures are based on trimers and there are four trimers in this sequences: *LVM*, *VMT*, *MTT* and *TTM*. Each signature consists of a root (the middle letter) and its two neighbors, ordered alphabetically. Thus, the signatures corresponding to the four trimers are $V(LM)$, $M(TV)$, $T(MT)$ and $T(MT)$, so that $s(LVMTTM) = V(LM) + M(TV) + 2T(MT)$. Notice that MTT and TTM generate the same signature (due to symmetry) and therefore contribute two occurrences to the sum $s(A) = \sum \sigma_i \mathbf{z}_i$.

We should note that this is actually a very simple usage of signature, and that signature can be extended to handle longer subsequences (height $2, 3, \ldots$), as well as non-linear sequences. In fact, signature was originally developed to describe molecules in cheminformatics. Recently, however, signature has also been used successfully in applications to HIV protease-1 peptide prediction (Faulon *et al*., 2003a) and inverse design of LFA-1/ICAM-1 peptides (Churchwell *et al*., 2004). Signature has the following advantages over other descriptors (Visco *et al*., 2002; Faulon *et al*., 2003b): (1) signature has been shown to be competitive with other descriptors (e.g. Molconn-Z) in terms of deriving quantitative structure–activity relationships and predicting various properties; (2) signature is canonical in the sense that it can often be used to derive other descriptors (in the case of amino acid sequences we note that descriptors such as hydrophobicity [used in Bock and Gough (2001)] are encompassed using signature with a smaller alphabet; (3) signature encodes information about structure as well as sequence by keeping track of neighborhoods. Thus, signature is information rich, and, in particular, enables the solution of inverse problems.

Signature has been a useful descriptor in the past, and has the important practical advantage for us in that it provides a vector representation of an amino acid sequence. We exploit this fact to develop a signature-based SVM for use in the prediction of protein–protein interaction.

Signature kernel

As mentioned previously, signature can be used to obtain vector representations of variable length amino acid sequences. This is, of course, the minimum requirement for the application of an SVM to our problem. It is more computationally efficient, however, if we use the fact that SVMs do not actually require the storage/use of these very long sparse vectors. They only require dot products between the vectors. Since this dot product is the composition of the standard Euclidean dot product with the signature function, we call it the signature kernel. The signature kernel is given as $k : \{\text{variable length amino acid sequences}\}^2 \to \mathbb{R}$, where $k(A, B) = s(A) \cdot (B)$.

Here, we note that our kernel is very similar to the string kernels in Leslie *et al*. (2002). For us, the only real difference is that we use a symmetric formulation, while Leslie *et al*. (2002) do not, primarily due to problem domain. In fact, the string kernels in Leslie *et al*. (2002) could potentially improve the performance of our algorithm, as these kernels allow comparisons of sequences with mismatches. On the other hand, mismatches in subsequences of length 3 would be of nebulous value, and while we could have used longer subsequences, we did not find this to be necessary. In fact, we did some experiments (on the *H.pylori* data) where we tried longer subsequences: we found that these subsequences did not provide any improvement in performance and therefore discontinued their use. In the end, we focused more on the product signature formulation (discussed in the next section), and chose to use our simpler string kernel based on signature.

The signature kernel elegantly combines signature with SVMs and thus inherits all of the advantages of the two methods. Most importantly, the SVMs allow the use of a very large number of signatures (up to 100 k in the applications we consider later). This would be unfeasible with other methods [e.g. multilinear regression, as used previously in Visco *et al*. (2002); Faulon *et al*. (2003a,b); Churchwell *et al*. (2004)].

Product signature

The signature kernel SVM, as presented above, will only work with data points that consist of a single amino acid sequence. This is a problem since our data points are, in fact, pairs (protein–protein pairs) of amino acid sequences. In order to overcome this difficulty, we must define signature for pairs of amino acid sequences, and we must also provide a kernel that gives the inner product between two protein–protein pairs.

To define signature for protein–protein pairs, we use the notion of a tensor product between vectors (can be found in many standard texts on linear algebra). For our purposes, we define the tensor product between $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}} \in \mathbb{R}^n$ and $\mathbf{b} = (b_1, \ldots, b_n)^{\mathrm{T}} \in \mathbb{R}^m$ to be $\mathbf{a} \otimes \mathbf{b} = (a_1 b_1, a_1 b_2, \ldots, a_1 b_m, a_2 b_1, \ldots, a_n b_m)^{\mathrm{T}} \in \mathbb{R}^{nm}$, and we observe that the entries in $\mathbf{a} \otimes \mathbf{b}$ are the same as the entries contained in the outer product $\mathbf{a}\mathbf{b}^{\mathrm{T}}$. Using this definition, the signature for pairs, or the signature product, $s \otimes s : \{\text{amino acid sequences}\}^2 \to F \otimes F \cong \mathbb{R}^{N^2}$, is taken to be $s \otimes s(A, B) - s(A) \otimes s(B)$.

Using this definition of signature product, we can now apply an SVM to our problem by specifying a kernel $\tilde{k} : (F \otimes F) \times (F \otimes F) \to \mathbb{R}$ that gives a dot product in the signature product space. We, of course, use the standard Euclidean inner product so that the signature product kernel is defined by $\tilde{k}((A, B), (C, D)) = (s(A) \otimes s(B)) \cdot (s(C) \otimes (D))$.

We are now in a position to apply an SVM to the problem of predicting protein–protein interaction. Computationally, however, there is one final obstacle to overcome. Specifically, the use of the signature product effectively squares the complexity of the calculation. This is easily seen when we observe that $F \cong \mathbb{R}^N$ so that $F \otimes F \cong \mathbb{R}^{N^2}$. In reality, the complexity does not increase according to N but rather according to the lengths of the amino acid sequences involved. Nevertheless, the computational complexity is squared, and this causes a problem that must be addressed if we are to process large datasets.

Fortunately, there is a simple way to fix this problem. If we write (for clarity) $\mathbf{a} = s(A)$, $\mathbf{b} = s(B)$, $\mathbf{c} = s(C)$ and $\mathbf{d} = s(D)$, then we can see that

$$\begin{aligned}
\tilde{k}((A, B), (C, D)) &= (s(A) \otimes s(B)) \cdot (s(C) \otimes s(D)) \\
&= \text{trace}\,((\mathbf{a}\mathbf{b}^{\mathrm{T}})(\mathbf{c}\mathbf{d}^{\mathrm{T}})^{\mathrm{T}}) \\
&= \text{trace}\,(\mathbf{a}\mathbf{b}^{\mathrm{T}}\mathbf{d}\mathbf{c}^{\mathrm{T}}) \\
&= (\mathbf{b}^{\mathrm{T}}\mathbf{d})\text{trace}(\mathbf{a}\mathbf{c}^{\mathrm{T}}) \\
&= (\mathbf{b}^{\mathrm{T}}\mathbf{d})(\mathbf{a}^{\mathrm{T}}\mathbf{c}) \\
&= k(A, C)k(B, D), \qquad (1)
\end{aligned}$$

where $\text{trace}(X)$ is the sum of the diagonal elements of a square matrix X, and k is the signature kernel.

Equation (1) shows that in order to compute the signature product kernel of two protein–protein pairs, we need only to compute the signature kernel between combinations of the individual proteins. Thus, we have removed the squared computational complexity.

Symmetric signature product

We can obtain an additional improvement in the signature product by enforcing symmetry in the protein–protein order. In other words, we can make a protein pair (A, B) equivalent to protein pair (B, A). This symmetry is easily achieved by defining the symmetric signature product $\Gamma(A, B) = s(A) \otimes s(B) + s(V) \otimes s(A)$. The associated symmetric signature product

kernel is then $\gamma((A,B),(C.D)) = 2(k(A,C)k(B,D) + k(A,D)k(B,C))$.

Normalized signature product

Next, we can use a normalized dot product to compensate for potential differences in the length of the amino acid sequences involved in our calculations. In particular, a normalized version of the signature kernel can be implemented as $k(A,B)/\sqrt{(k(A,A)k(B,B))}$. This kernel extends directly to the signature product kernel and is only slightly more complicated in the case of the symmetric signature product.

Other adjustments

As mentioned previously, SVMs are very flexible and we found that we could occasionally achieve minor (2%) improvements in performance by adjusting kernels or using preprocessing techniques. In particular, we found that preprocessing by removal of signatures occurring only once in the dataset occasionally resulted in better performance. This improvement was used in the case of the yeast SH3 data.

We also found that using a Gaussian kernel in combination with the product signature kernel could result in better performance. In particular, we used the kernel $\exp[-\gamma(\tilde{k}(A,A) - 2\tilde{k}(A,B) + \tilde{k}(B,B))]$, where $\tilde{k}$ is the product signature as described previously, and γ was chosen to be 0.5. This kernel was used in the case of the full yeast proteome.

IMPLEMENTATION

We tested our algorithm on publicly available *S.cervisiae* (Tong *et al*., 2002; Sprinzak and Margalit, 2001; Jansen *et al*., 2003; Xenarios *et al*., 2002), *H.pylori* (Rain *et al*., 2001), human (Xenarios *et al*., 2002), mouse (Xenarios *et al*., 2002) and *E.coli* (Xenarios *et al*., 2002) datasets.

Yeast SH3 domains

We first tested our algorithm on the dataset in Tong *et al*. (2002). These data consist of 20 yeast SH3 domains and various associated ligands determined through phage display to bind (or not bind) to the 20 domains, resulting in a dataset of 709 domain–ligand pairs. These domain–ligand pairs were processed via the computation of different position-specific scoring matrices (PSSMs) to produce an interaction prediction. To compute the PSSMs, we followed the procedure described in Tong *et al*. (2002), resulting in one PSSM per domain.

We used the SH3 domain dataset to assess the usefulness of our technique from two perspectives. First, we wanted to compare predictions obtained using the product signature with predictions obtained without the product signature. We made these comparisons using ligand signatures only (domains were considered fixed) versus domain–ligand product signatures. Second, we wanted to compare our method with the PSSM method. The PSSM method considers only one domain at a time and so is not a product method.

Table 1. Comparison of methods

	Acc.	Prec.	Sens.
Yeast SH3			
Ligand-only	73.7	75.5	63.1
Product	80.7	81.4	75.2
PSSM	75.4	68.8	81.3
Full yeast			
Product	69.0	71.5	63.2
InterPro	70.8	86.5	49.2
Sprinzak	68.8	79.8	50.0
H.pylori			
Product	83.4	85.7	79.9
Bock and Gough	75.8	80.2	69.8
E.coli and *H.pylori*			
Product	56.1	17.4	76.7
Human			
Product	70.3	72.2	66.2
Mouse and human			
Product	74.3	68.6	77.4

Here, we compare the product signature method and its alternatives using yeast, *H.pylori*, human and mouse datasets. Ligand-only refers to using signature but without using the product method; PSSM is the method used in Tong *et al*. (2002); InterPro uses our method but with InterPro (Apweiler *et al*., 2001) entries instead of our sequence- based signatures; Sprinzak refers to the method in (Sprinzak and Margalit, 2001), which also used the InterPro entries; Jansen uses the method in Jansen *et al*. (2003); and Bock and Gough refers to the reported results in Bock and Gough (2003) on *H.pylori*.

The results of our analysis are shown in Table 1. In this table, we compare the product signature to the ligand-only signature, as well as to the PSSM method. We used 10-fold cross-validation and calculated accuracy, precision and sensitivity. To be precise, we first divided the dataset (at random) into 10 equally sized subsets. We used each subset in turn as a test set, while we trained our method on the union of the remaining 9 subsets. We evaluated the performance of our classifier by computing accuracy (TP + TN)/(TP + FP + TN + FN), precision TP/(TP + FP) and sensitivity TP/(TP + FN), where TP, TN, FP and FN are true and false positive and negative predictions. The values reported in Table 1 are the average values of the accuracy, precision and sensitivities averaged over the 10 test sets.

We note that the results of these comparisons may reveal more than simply which method performs better. In fact, we provided the comparison of the product method and the ligand-only method to investigate a particular hypothesis. Specifically, we know that because domains are ignored, the ligand-only method learns the tendency of a given ligand to bind. Therefore, the fact that the product method performs better than the ligand-only method suggests that the product method is learning something beyond just the tendency of a given ligand to bind. This was, in fact, one of the original motivations for developing the product method.

In addition to observations about specific classifiers, the accuracy, precision and sensitivity are useful for measuring the

behavior of a classifier in general. In particular, the accuracy gives the overall performance of a classifier, the precision gives the percentage of positive predictions that are actually positive and the sensitivity gives the percentage of actual positives that are predicted (note that TP/(FP + FN) = TP/P, where P is the total number of positives).

By looking at the precision and sensitivity statistics, we can determine if a classifier will identify positives correctly. If a classifier has a high precision and a low sensitivity, then it is likely to be correct when it makes a positive prediction, although it will make many false negative predictions. Conversely, a classifier with a low precision and a high sensitivity is likely to identify most true positives, even though many of its predictions will be false. In some sense, the first classifier is too conservative while the second is too optimistic.

Full yeast proteome

We next tested our algorithm by generalizing to the entire yeast proteome. In addition to testing our method on a larger, more heterogeneous dataset, the yeast proteome allowed us to compare our work with the work already done in Sprinzak and Margalit (2001). In particular, we first reproduced their work using the current InterPro database (as of December 2003), and the Swiss-Prot database (Boeckmann *et al*., 2003) for sequence information. Since the InterPro database has grown, we found more InterPro entries than previously reported in Sprinzak and Margalit (2001). From the 2908 protein pairs originally used, we obtained 2082 pairs describable using the InterPro entries (up from 1274), altogether using 1220 InterPro entries (up from 434). We checked our work by confirming the presence of many of the significant InterPro entry pairs reported in Sprinzak and Margalit (2001), as well as by obtaining a high sensitivity when using the protein pair subsets they describe.

Next, we tested both our method and the method in Sprinzak and Margalit (2001). As in the yeast SH3 dataset, we used 10-fold cross-validation to assess the methods. In contrast to the yeast SH3 dataset, noninteracting protein pairs were missing. As in Bock and Gough (2003), we had to assume that the pairs not specified explicitly as interacting were non-interacting. From these non-interacting pairs, we selected at random a set of negative examples (in fact, we selected a variety of random sets but always arrived at similar results). We started with 2082 interacting protein pairs and added 2082 noninteracting pairs, arriving at a dataset with 4164 protein pairs.

With this dataset, we tried our standard signature product method (using the Gaussian kernel as described in the Systems and methods section); our method using the InterPro entries instead of our sequence-based signatures; and the method in Sprinzak and Margalit (2001). These methods are shown in Table 1 as Product, InterPro and Sprinzak, respectively.

Yeast gold standard

Having achieved reasonably good results on our first two benchmarks, we decided to attempt a comparison with the work done in Jansen *et al*. (2003). Although the approach in Jansen *et al*. (2003) is not directly related to our method, it is another computational method for predicting protein–protein interactions.

In addition, the algorithm in Jansen *et al*. (2003) is benchmarked against a well-designed yeast gold-standard dataset. This dataset consists of 8250 positive interactions obtained from the MIPS database (Mewes *et al*., 2002), and 2 708 746 negative interactions. The negative interactions were generated from lists of proteins in separate subcellular compartments (Kumar *et al*., 2002). In our computations, we were forced to use a slightly smaller version of the gold-standard dataset. In particular, we obtained sequence information from Swiss-Prot for only 4596 of the ∼7500 proteins used in Jansen *et al*. (2003). This left us with 7714 positive interactions and 1 805 675 negative interactions. Our comparisons are based on this modified gold-standard dataset.

In Jansen *et al*. (2003), the gold-standard dataset was used with 7-fold cross-validation to benchmark a Bayesian network approach to protein–protein interaction prediction. This approach is capable of combining multiple sources of information (such as different experiments and/or *de novo* calculations) into the predictions. In particular, Jansen *et al*. (2003) combine experimental and *de novo* computations to produce proteome-wide predictions of probable yeast interactions. These predictions are denoted as PIT (Probabilistic Interactome Total) and are derived by combining the experimentally based predictions (PIE) and the computationally based predictions (PIP).

In Jansen *et al*. (2003), the PIE, PIP and PIT are compared using sensitivity and the ratio TP/FP (similar to precision). In particular, a confidence value L_{cut} is varied and the resulting sensitivities and TP/FP ratios are compared. L_{cut} essentially quantifies the certainity of algorithm about a given protein–protein interaction prediction being correct. By requiring a large value for L_{cut}, it is observed [Fig. 2C of Jansen *et al*. (2003)] that there is a decrease in sensitivity but an increase in the TP/FP ratio. This agrees with our previous remarks concerning sensitivity and precision. Namely, increasing L_{cut} makes the classifier more conservative and correspondingly more accurate on positive predictions.

Mainly to compare with Jansen *et al*. (2003), but also to verify that our method achieves greater accuracy when restricted to higher confidence positive predictions, we applied our method to the gold-standard dataset, again using 10-fold cross-validation. In the case of SVMs, the L_{cut} parameter has an analog in the threshold t described when we introduced the SVM (Systems and methods section). By increasing t, where the classifier $f(\mathbf{x})$ makes positive predictions when $f(\mathbf{x}) > t$, we can keep only high confidence predictions.

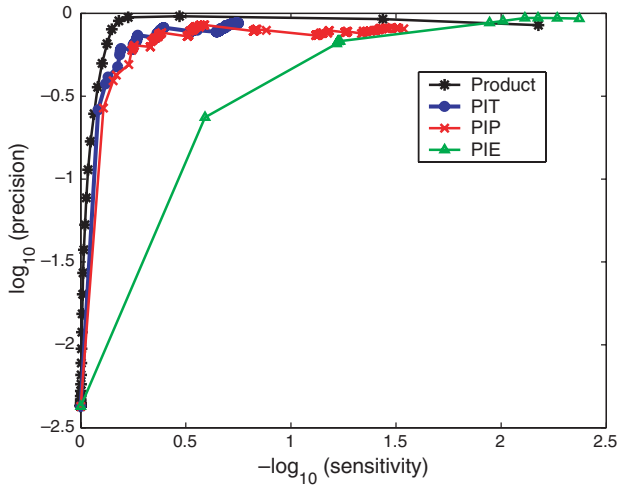

Fig. 1. Sensitivity versus precision. Here, we examine the trade-off between sensitivity and precision using our method and that Jansen *et al*. (2003). Both these methods can achieve good trade-offs, as indicated by the points in the upper left of the plot.

A comparison of our method with that of Jansen *et al*. (2003) can be seen in Figure 1. In the case of our method, the curve in Figure 1 was produced using 10-fold cross-validation on the gold-standard dataset while varying t in increments of 0.1. Since SVMs perform best when using balanced training sets, we did not use every negative training example during our cross-validation. To be precise, we did the following for each training/test set combination. In the training set, we selected at random a subset of the negative training examples equal in size to the set of positive training examples. We then trained our classifier using this subset of the training set and finally predicted every example (both positive and negative) in corresponding test set. The precision and sensitivity were again calculated by averaging the results over the 10 test sets.

In the case of Jansen *et al*. (2003), we avoided reimplementing their method by using the PIE, PIP and PIT predictions downloaded from their web page. Using these predictions, we recomputed the precision and sensitivity measures by varying L_{cut} parameter by increments of 20. We did not use Figure 2C in Jansen *et al*. (2003) directly because we had previously restricted the gold-standard dataset based on the use of proteins with sequence information, and because we have consistently used precision instead of TP/FP in our work.

Helicobacter pylori

We next tested our algorithm with another proteome-wide experiment using two-hybrid measurements of *H.pylori* produced in Rain *et al*. (2001). This dataset allowed us to examine an organism besides yeast and gave a comparison of our method with the method of Bock and Gough (2001, 2003). We again used 10-fold cross-validation to assess our method, and we also assumed that the pairs not specified explicitly as interacting where non-interacting. From these non-interacting pairs, we selected at random a set of negative examples so that altogether we used 1458 positives and 1458 negatives for a total of 2916 protein pairs. The results of the cross-validation study are shown in Table 1.

Human to mouse and *H.pylori* to *E.coli*

Finally, we tested the ability of our method to predict protein–protein interactions in one species using interactions from a different species. In particular, we used human protein–protein interactions archived in the DIP (Xenarios *et al*., 2002) to predict mouse interactions, also archived in the DIP database. There were 941 human and 239 mouse interactions in the database. Our results are shown in Table 1.

One caveat concerning this example is the fact that human and mouse protein–protein interactions are very closely related. In fact, we found that 50% of the proteins in the mouse dataset were also present in the human dataset, and that 40% of the interactions in the mouse dataset were present in the human dataset. Another caveat is that these datasets were not obtained by high-throughput proteome-wide methods and were small in comparison with the data in Rain *et al*. (2001).

Without such a close relation between species, our method may not work as well. In fact, when we tried to predict *E.coli* from *H.pylori* (*E.coli* is also available from the DIP database), we had very limited success, as can be seen by the poor results reported in Table 1.

DISCUSSION

Our method is most similar to the methods of Bock and Gough (2001) and Sprinzak and Margalit (2001). Both our method and that Bock and Gough (2001) use SVMs, sequence information and experimental data to predict protein–protein information. However, while (Bock and Gough, 2001) transform sequence information into physico- chemical information (charge, hydrophobicity and surface tension), the signature descriptor does not require us to perform such a transformation. Furthermore, Bock and Gough (2001) encode and compare protein pairs by concatenating normalized versions of the amino acid sequences of each protein, and hence use a global representation of a protein pair. Although local information can be encoded implicitly in Bock and Gough (2001) (by using a non-linear kernel, such as a polynomial kernel), our method uses explicit pair- oriented, local sequence information (product signature). In other words, we are looking for amino acid subsequence pairs, which occur together when two proteins interact.

Our use of subsequence pairs is similar to the correlated InterPro entry pairs described in Sprinzak and Margalit (2001). However, instead of using InterPro entries (Apweiler *et al*., 2001), we use an automatic method for generating signatures, which depends only on sequence information. Finally, our method has the advantage of using a principled method (SVMs) to obtain our final classifier.

Table 2. Top 10 predicted binding pairs

Swiss-Prot ID	Swiss-Prot ID	Prediction
P27895	P27895	2.36
P27895	P36022	1.95
P22579	Q00916	1.61
P00546	Q02821	1.56
P40064	Q00916	1.53
P50875	P19659	1.48
P19659	P09547	1.46
Q06142	Q02821	1.44
Q06245	P19524	1.41
Q00916	P08964	1.41

Top 10 yeast protein pairs predicted to bind using the signature product method.

To further establish the relationship between our technique and the InterPro domain method in Sprinzak and Margalit (2001), we explored the potential of our method for predicting domains. For this exercise, we returned to the full yeast dataset, where we selected the top 10 pairs predicted by our method to interact (Table 2). Using the 14 proteins present in these pairs, we constructed domain-sized amino acid subsequences by sliding a window across each of the protein sequences. Our window was of size 50, and we moved the window in increments of 10 amino acid residues. Using this method, we obtained 1681 subsequences, each 50 amino acids long.

From the model obtained using the full yeast dataset, we predicted which pairs of these subsequences would interact. By examining the positions of these interacting subsequences within the full protein sequences we could make domain predictions as shown in Figure 2.

In Figure 2, we examine the domain predictions for P09547 and P50875. In particular, Figure 2a shows that the region between 300 and 400 is more likely to bind with the other regions (windows) among the 14 proteins examined. We hypothesize that this is a domain. Figure 2b shows that this domain binds with itself and, therefore, that P09547 binds with itself. This is only a prediction, but when we examine a known interaction, P09547 with P50875, we see a similar result. In fact, we see in Figure 2c that P09547 binds with P50875 and that our previously hypothesized domain binds to regions 100–150, 200–250, 400–450 and 500–550. We again hypothesize that these are domains, this time in P50875.

To see that these predictions match known information, we looked up the domain information for P09547 and P50875 in the Swiss-Prot database. There were two domains mentioned for P09547, an Asn/Thr-rich region from 5 to 65, and a Gln-rich region from 337 to 385. Our domain correlates well with the Gln-rich region. For P50875, Swiss-Prot gives five domains: a Poly-Gln domain from 157 to 162, a Poly-Ser domain from 235 to 240, another Poly-Ser domain from 422 to 425, a Poly-Ala domain from 454 to 463, and a Poly-Asn domain from 552 to 559. Although not perfect, these domains also correlate with our predictions.

Finally, we observed that our domain prediction for P09547 matched with InterPro entry IPR001660 SAM domain also discovered in Sprinzak and Margalit (2001) in an InterPro entry pair with a count of 5 and a log-odds value of 2.55 (up to 6.47 in our calculations).

While predicting domains are not the focus of our method, we were encouraged by our preliminary results in this direction, and by how they matched with the InterPro predictions made in Sprinzak and Margalit (2001). We speculate that the ability of our method to identify domains is related to its ability to predict interactions.

Although we have benchmarked our algorithm fairly extensively and feel confident that if it performs well, we must issue some additional caveats. First of all, the algorithm requires some type of input, either from experiment or prediction. If these data are unreliable, as is often the case with protein–protein interaction data (von Mering *et al*., 2002; Sprinzak *et al*., 2003), then the results of our algorithm will also be unreliable.

Second, the fact that there are so few interacting protein pairs relative to the number of possible pairs can cause problems. In particular, there are bound to be many false positive predictions, even with a good classifier. Although we have shown that our method performs as well as the method in Jansen *et al*. (2003) in this regard, it nevertheless remains a practical problem (for any classifier) given the naturally unbalanced population of protein–protein interactions.

In total, however, we have proposed a very general method for predicting protein–protein interactions. Our method uses a unique product description of protein pairs called signature product. This description allows the clean integration of amino acid sequence information into a powerful SVM machine learning architecture. Our method performs as well or better than competing methods and combines advantages of the methods in Bock and Gough (2001) and Sprinzak and Margalit (2001). In addition, our method compares favorably to the method in Jansen *et al*. (2003), which uses information from many sources in making protein–protein interaction predictions.

Since our method is most similar to the methods of Bock and Gough (2001) and Sprinzak and Margalit (2001), we conclude with a brief discussion of the similarities and differences. First, our method uses only experimental and sequence information and can therefore be used to study organisms where little is known, like Bock and Gough (2001). Simultaneously, we use a local description of protein pairs, which may be more consistent with the actual biology of protein–protein interaction, like Sprinzak and Margalit (2001). On the other hand, our method eliminates disadvantages of both Bock and Gough (2001) and Sprinzak and Margalit (2001). In particular, our method does not require physico- chemical information, unlike in Bock and Gough (2001), and we do not need to have

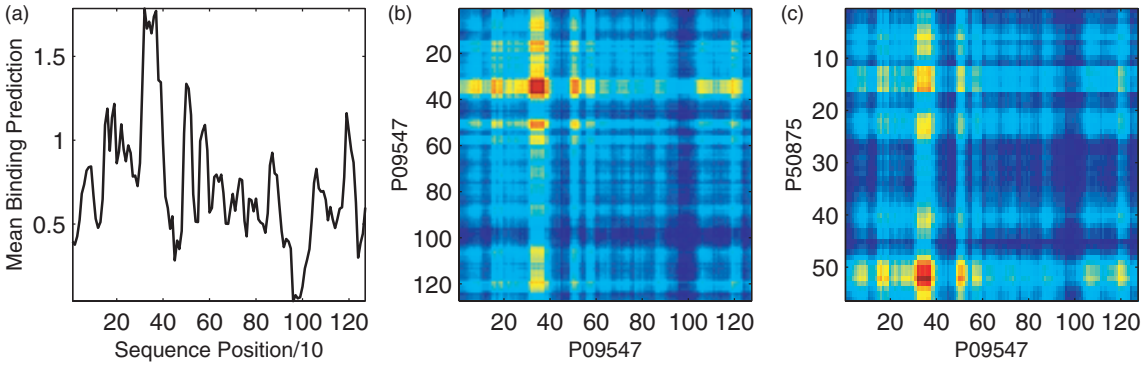

Fig. 2. Domain predictions. Plots showing the domain predictions for P09547. The x-axis of each plot gives the position of a 50-residue window moved 10 residues at a time across the full sequence of P09547. Plot (**a**) shows the mean binding activity of the windows along with x-axis with the other 1681 windows considered in the domain prediction example; plot (**b**) shows an intensity plot of the binding activities of all pairs of windows in P09547; and (**c**) shows an intensity plot of the binding activities of all pairs of windows in P09547 and P50875 (a known binder). In (b and c), red denotes activity and blue denotes inactivity.

prior knowledge of domains, unlike in Sprinzak and Margalit (2001).

ACKNOWLEDGEMENTS

Many thanks to Carla Churchwell for help with signature and associated computer codes. This work was funded by the US Department of Energy's Genomics: GTL program (www.doegenomestolife.org) under project, 'Carbon Sequestration in *Synechococcus sp.*: From Molecular Machines to Hierarchical Modeling' (www.genomes-to-life.org). Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

REFERENCES

Apweiler,R., Attwood,T.K., Bairoch,A., Bateman,A., Birney,E., Biswas,M., Bucher,P., Cerutti,L., Corpet,F., Croning,M.D. *et al.* (2001) The InterPro database, and integrated documentation resource for protein families, domains, and functional sites. *Nucleic Acids Res.*, **29**, 37–40.

Bennett,K.P. and Campbell,C. (2000) Support vector machines: hype or hallelujah. *ACM SIGKDD Explorations*, **2**, 1–13.

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

Bock,J. and Gough,D. (2003) Whole-proteome interaction mining. *Bioinformatics*, **19**, 125–135.

Boeckmann,B., Bairoch,A., Apweiler,R., Blatter,M.-C., Estreicher,A., Gasteiger,E., Martin,M.J., Michoud,K., O'Donovan,C., Phan,I., Pilbout,S. and Schneider,M. (2003) The Swiss-Prot protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.*, **31**, 365–370.

Pls provide the page range for ref. Burges 1998

Please update

Burges,C.J.C. (1998) A tutorial on support vector machines for pattern recognition. *Knowl. Discov. Data Mining*, **2**.

Churchwell,C.J., Rintoul,M.D., Martin,S., Visco,D., Kotu,A., Larson,R.S., Sillerud,L.O., Brown,D.C. and Faulon,J.L. (2004) The signature molecular descriptor. 3. Inverse quantitative structure–activity relationship of ICAM-1 inhibitory peptides. *J. Mol. Graph. Model.*

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

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

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**, 25–26.

Faulon,J.-L., Churchwell,C. and Visco,D.P.,Jr. (2003a) The signature molecular descriptor. 2. Enumerating molecules from their extended valence sequences. *J. Chem. Inf. Comput. Sci.*, **43**, 721–734.

Faulon,J.-L., Visco,D.P.,Jr. and Pophale,R.S. (2003b) The signature molecular descriptor. 1. Extended valence sequences vs. topological indices in QSAR and QSPR studies. *J. Chem. Inf. Comput. Sci.*, **43**, 707–720.

Fields,S. and Song,O.-K. (1989) A novel genetic system to detect protein–protein interactions. *Nature*, **340**, 245–246.

Furey,T., Cristianini,N., Duffy,N., Bednarski,D.W., Schummer,M. and Haussler,D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**, 906–914.

Goh,C.S., Bogan,A.A., Joachimiak,M., Walther,D. and Cohen,F.E. (2000) Co-evolution of proteins with their interaction partners. *J. Mol. Biol.*, **299**.

Please provide page range.

Ho,Y., Gruhler,A., Heilbut,A., Bader,G.D., Moore,L., Adams,S.L., Millar,A., Taylor,P., Bennet,K., Boutilier,K. *et al.* (2002) Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature*, **415**, 180–183.

Ito,T., Tashiro,K., Muta,S., Ozawa,R., Chiba,T., Nishizawa,M., Yamamoto,K., Kuhara,S. and Sakaki,Y. (2000) Toward a protein–protein interaction map of the budding yeast: a comprehensive system to examine two-hybrid interactions in all possible combinations between yeast proteins. *Proc. Natl Acad. Sci. USA*, **97**, 1143–1147.

Jansen,R., Haiyuan,Y., Greenbaum,D., Kluger,Y., Krogan,N.J., Chung,S., Emili,A., Snyder,M., Greenblatt,J.F. and Gerstein,M. (2003) A bayesian networks approach for predicting protein–protein interactions from genomic data. *Science*, **302**, 449–453.

Joachims,T. (1999) Making large-scale SVM learning practical. In Schölkopf,B., Burges,C.J.C. and Smola,A.J. (eds), *Advances in Kernel Methods–Support Vector Learning*. MIT Press, Cambridge, MA, pp. 169–184.

Kumar,A., Agarwal,S., Heyman,J.A., Matson,S., Heidtman,M., Piccirillo,S., Umansky,L., Drawid,A., Jansen,R., Liu, Y. *et al*. (2002) Subcellular localization of the yeast proteome. *Genes Dev.*, **16**, 707–719.

Please provide the page nos and other relevant details.

Leslie,C., Eskin,E., Weston,J. and Noble,W. (2002) Mismatch string kernels for SVM protein classification. In *Advances in Neural Information Processing Systems*. MIT Press.

Pls provide page nos.

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

Mewes,H.W., Frishman,D., Gruber,C., Geier,B., Haase,D., Kaps,A., Lemcke,K., Mannhaupt,G., Pfeiffer,F., Schuller,C., Stocker,S. and Weil,B. (2002) MIPS: a database for genomes and protein sequences. *Nucleic Acids Res.*, **30**, 31–34.

Please provide page nos.

Pazos,F., Helmer-Citterich,M., Ausiello,G. and Valencia,A. (1997) Correlated mutations contain information about protein–protein interaction. *J. Mol. Biol.*, **271**.

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

Rain,J.C., Selig,L., DeReuse,H., Battaglia,V., Reverdy,C., Simon,S., Lenzen,G., Petel,F., Wojcik,J., Schacter, V. *et al*. (2001) The protein–protein interaction map of *Helicobacter pylori*. *Nature*, **409**, 211–215.

Smola,A.J. and Schölkopf,B. (1998) A tutorial on support vector regression. *NeuroCOLT Technical Report NC-TR-98-030*, Royal Holloway College University of London, UK.

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

Sprinzak,E., Sattath,S. and Margalit,H. (2003) How reliable are experimental protein–protein interaction data? *J. Mol. Biol.*, **327**, 919–923.

Tong,A., 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.

Uetz,P., Giot,L., Cagney,G., Mansfield,T.A., Judson,R.S., Knight,J.R., Lockshon,D., Narayan,V., Srinivasan,M., Pochart,P. *et al*. (2000) A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. *Nature*, **403**.

Please provide page nos.

Valencia,A. and Pazos,F. (2002) Computational methods for the prediction of protein interactions. *Curr. Opin. Struct. Biol.*, **12**, 368–373.

Vapnik,V. (1998) *Statistical Learning Theory*. Wiley Interscience, New York.

Visco,D.P.,Jr, Pophale,R.S., Rintoul,M.D. and Faulon,J.L. (2002) Developing a methodology for an inverse quantitative structure–activity relationship using the signature molecular descriptor. *J. Mol. Graph. Model*, **20**, 429–438.

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.

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

Zhu,H., Bilgin,M., Bangham,R., Hall,D., Casamayor,A., Bertone,P., Lan,N., Jansen,R., Bidlingmaier,S., Houfek, T. *et al*. (2001) Global analysis of protein activities using proteome chips. *Science*, **293**, 2101–2105.