FORMATICS

Vol. 21 Suppl. 1 2005, pages i38–i46
doi:10.1093/bioinformatics/bti1016

Kernel methods for predicting protein–protein interactions

Asa Ben-Hur[1,*] and William Stafford Noble[1,2]

[1]Department of Genome Sciences and [2]Department of Computer Science and Engineering, University of Washington, Seattle, WA, USA

Received on January 15, 2005; accepted on March 27, 2005

espite advances in high-throughput methods for rotein–protein interactions, the interaction net- well-studied model organisms are sketchy at ing the continued need for computational meth- irect experimentalists in the search for novel

present a kernel method for predicting protein– ctions using a combination of data sources, ein sequences, Gene Ontology annotations, s of the network, and homologous interactions es. Whereas protein kernels proposed in the lit- e a similarity between single proteins, prediction requires a kernel between pairs of proteins. pairwise kernel that converts a kernel between s into a kernel between pairs of proteins, and he kernel's effectiveness in conjunction with a r machine classifier. Furthermore, we obtain ormance by combining several sequence-based on k-mer frequency, motif and domain content augmenting the pairwise sequence kernel with are based on other sources of data.

r method to predict physical interactions in yeast m the BIND database. At a false positive rate of fier retrieves close to 80% of a set of trusted We thus demonstrate the ability of our method rate predictions despite the sizeable fraction of that are known to exist in interaction databases. The classification experiments were performed vailable at http://pyml.sourceforge.net. Data are ttp://noble.gs.washington.edu/proj/sppi

These methods include the yeast two-hybrid screen and methods based on mass spectrometry (see von Mering *et al.*, 2002 and references therein). The data obtained by these methods are partial: each experimental assay can identify only a subset of the interactions, and it has been estimated that for the organism with the most complete interaction network, namely yeast, only about half of the complete 'interactome' has been discovered (von Mering *et al.*, 2002). In view of the very small overlap between interactions discovered by various high-throughput studies, some of them using the same method, the actual number of interactions is likely to be much higher. Computational methods are therefore required for discovering interactions that are not accessible to high-throughput methods. These computational predictions can then be verified by more labor-intensive methods.

A number of methods have been proposed for predicting protein–protein interactions from sequence. Sprinzak and Margalit (2001) have noted that many pairs of structural domains are over-represented in interacting proteins and that this information can be used to predict interactions. Several authors have proposed Bayesian network models that use the domain or motif content of a sequence to predict interactions (Deng *et al.*, 2002; Gomez *et al.*, 2003; Wang *et al.*, 2005). The pairwise sequence kernel was independently proposed in a recent paper (Martin *et al.*, 2005) with a sequence representation by 3mers. Other sequence-based methods use coevolution of interacting proteins by comparing phylogenetic trees (Ramani and Marcotte, 2003), correlated mutations (Pazos and Valencia, 2002) or gene fusion which works at the genome level (Marcotte *et al.*, 1999). An alternative approach is to combine multiple sources of genomic

ods, and in particular support vector machines
ölkopf and Smola, 2002), have proven use-
difficult classification problems in bioinform-
2004). The learning task we are addressing
ationship between pairs of protein sequences:
pairs of sequences are interacting or not. The
nce kernels (A kernel is a measure of similarity
he additional condition of being a dot product
e space; see Schölkopf and Smola, 2002 for
bed in the literature measure similarity between
s. We propose a method for converting a ker-
single proteins into a pairwise kernel, and we
ature space produced by that kernel.
ethod uses motif, domain and *k*mer composi-
pairwise kernel, and achieves better perform-
ple methods based on BLAST or PSI-BLAST.
ause it is difficult to predict interactions from
e, we incorporate additional sources of data.
kernels based on similarity of GO annotations,
core to interacting homologs in other species
al-clustering coefficient (Goldberg and Roth,
asures the tendency of neighbors of interact-
interact as well. Adding these additional data
cantly improves our method's performance rel-
hod trained using only the pairwise sequence
kernel methods for combining data from het-
urces of data allows us to use high-dimensional
whereas other studies on predicting protein–
tions (Zhang *et al.*, 2004; Lin *et al.*, 2004) use
onal representations which are appropriate for
ssifier.

LS FOR PROTEIN–PROTEIN CTIONS

er kernel methods derive much of their power
ility to incorporate prior knowledge via the
n. Furthermore, the kernel approach offers the
y apply kernels to diverse types of data, includ-
th vectors (e.g. microarray expression data),
n strings (DNA and protein sequences), graphs
this work, we employ a diverse collection of
bed in this section.

proteins X_1 and X_2 compared with proteins X_1' and X_2'. We call a kernel that operates on individual genes or proteins a 'genomic kernel', and a kernel that compares pairs of genes or proteins a 'pairwise kernel'. Pairwise kernels can be computed either indirectly, by way of an intermediate genomic kernel, or directly using features that characterize pairs of proteins.

The most straightforward way to construct a pairwise kernel is to express the similarity between pairs of proteins in terms of similarities between individual proteins. In this approach, we consider two pairs to be similar to one another when each protein of one pair is similar to one protein of the other pair. For example, if protein X_1 is similar to protein X_1', and X_2 is similar to X_2', then we can say that the pairs (X_1, X_2) and (X_1', X_2') are similar. We can translate these intuitions into the following pairwise kernel:

$$K((X_1, X_2), (X_1', X_2')) = K'(X_1, X_1')K'(X_2, X_2') + K'(X_1, X_2')K'(X_2, X_1'),$$

where $K'(\cdot, \cdot)$ is any genomic kernel. This kernel takes into account the fact that X_1 can be similar to either X_1' or X_2'.

An alternative to the above approach is to represent a pair of sequences (X_1, X_2) explicitly in terms of the domain or motif pairs that appear in it. This representation is motivated by the observation that some domains are significantly over-represented in interacting proteins (Sprinzak and Margalit, 2001). A similar observation holds for sequence motifs as well. Given a pair of sequences X_1, X_2 represented by vectors $\mathbf{x}_1, \mathbf{x}_2$, with components $x_i^{(1)}, x_i^{(2)}$ we form the vector $\mathbf{x}_{12}$ with components $x_i^{(1)} x_j^{(2)} + x_i^{(2)} x_j^{(1)}$. We can now define the explicit pairwise kernel:

$$K((X_1, X_2), (X_1', X_2')) = K'(\mathbf{x}_{12}, \mathbf{x}_{12}'), \qquad (1)$$

where $\mathbf{x}_{12}$ is the pairwise representation of the pair (X_1, X_2), and $K'(\cdot, \cdot)$ is any kernel that operates on vector data. It is straightforward to check that for a linear kernel function, the pairwise and explicit pairwise kernels are identical. The explicit representation can be used in order to rank the relevance of motif pairs with respect to the classification task. This ranking is accomplished, e.g. by sorting the motif pairs according to the magnitude of the corresponding weight vector components.

ce models for our motif kernel are discrete ifs, providing a count of how many times a dis- e motif matches a sequence. To compute the e used discrete sequence motifs from the eMotif ill-Manning *et al.*, 1997). Yeast ORFs contain f 17 768 motifs out of a set of 42 718 motifs.

Pfam kernel uses a set of hidden Markov mod- o represent the domain structure of a protein, ted by comparing each protein sequence with the Pfam database (Sonnhammer *et al.*, 1997). rotein–HMM comparison yields an E-value version 10.0 contains 6190 domain HMMs; protein is represented by a vector of 6190 log s Pfam kernel has been used previously to pre- rotein interactions (Gomez *et al.*, 2003), though tion with the pairwise kernel described above. sequence kernels we use a a normalized linear $y)/\sqrt{K(x,x)K(y,y)}$; in the case of the Pfam performed an initial step of centering the kernel.

equence kernels

to using the pairwise kernel is the following:

$$2), (X_1', X_2')) = K'(X_1, X_2)K'(X_1', X_2'). \quad (2)$$

appropriate when similarity within the pair is d to the likelihood that a pair of proteins inter- his is a valid kernel even if K' is not a kernel, s formulation K' is simply a feature of the pair of sider GO annotations, for example: a pair of pro- ikely to interact if the two proteins share similar n addition to GO annotation we also consider es of the interaction network, and homologous other species. We summarize these properties scores $\mathbf{s}(X_1, X_2)$, such that the kernel for the data can be any kernel appropriate for vector

$$1, X_2)), (X_1', X_2')) = K'(\mathbf{s}(X_1, X_2), \mathbf{s}(X_1', X_2')), \quad (3)$$

se to use a Gaussian kernel for K'.

kernel Proteins that are not present in the same onent or that participate in different biological less likely to interact. We represent this prior

We consider two ways in which to define the dot product in this space. When the non-zero components are set equal to 1, then when each protein has a single annotation, and the annotatinos are on a tree, the dot product between two proteins is the height of the lowest common ancestor of the two nodes. An alternative approach assigns annotation a a score of $-\log p(a)$, where $p(a)$ is the fraction of proteins that have annotation a. We then score the similarity of annotations a, a' as $\max_{a'' \in \text{ancestors}(a) \cap \text{ancestors}(a')} -\log p(a'')$. In a tree topology, this score is the similarity between the deepest common ancestor of a and a', because the node frequencies are decreasing along a path from the root to any node. The score is a dot product with respect to the infinity norm on the annotation vector space. This also holds when the proteins have more than one annotation and the similarity between their annotations is defined as the maximum similarity between any pair of annotations. When one of the proteins has an unknown GO annotation, the kernel value is set to 0.

2.3.2 Interactions in other species It has been shown that interactions in other species can be used to validate or infer interactions (Yu *et al.*, 2004): the existence of interacting homologs of a given pair of proteins implies that the original proteins are more likely to interact. We quantify this observation with the following homology score for a pair of proteins (X_1, X_2):

$$h(X_1, X_2) = \max_{i \in \mathcal{H}(X_1), j \in \mathcal{H}(X_2)} I(i,j) \times \min(l(X_1, X_i), l(X_2, X_j)),$$

where $\mathcal{H}(X)$ is the set of non-yeast proteins that are significant BLAST hits of X, $I(i,j)$ is an indicator variable for the interaction between proteins i and j, and $l(X_k, X_i)$ is the negative of the log E-value provided by BLAST when comparing protein k with protein i in the context of a given sequence database. We used interactions in human, mouse, nematode and fruit fly to score the interactions in yeast.

2.3.3 Mutual clustering coefficient Protein–protein interaction networks tend to be 'cliquish'; i.e. the neighbors of interacting proteins tend to interact. Goldberg and Roth (2003) quantified this cohesiveness using the mutual clustering coefficient (MCC). Given two proteins u, v, their MCC can be quantified, by the Jaccard coefficient $|N(v) \cup N(u)|/|N(v) \cap$

nels, while the feature space for $\sum_i K_p(K_i)$ of features that originate from the same gen- n practice, the results from these two different ere very close, and the mixing approach was of its lower memory requirement. A Gaussian kernel can be introduced at several stages: inear genomic kernel as: $\exp(-\gamma(K_p(P,P) - K_p(P',P'))$, where P, P' are two pairs of pro- not tried introducing a non-linear kernel at the nomic kernel; a Gaussian kernel at the level of ernel performed similar to the 'linear' pairwise the high dimensionality of the resulting feature ults reported in this paper are computed using se kernels.

orating interaction reliability in g

s of protein–protein interaction data have noted experimental assays produce varying levels of and have proposed methods for finding which e likely to be reliable (von Mering *et al.*, 2002; , 2003; Deane *et al.*, 2002) (see Section 3.1 for corporate this knowledge about the reliability of n interactions into the training procedure using -margin parameter C (Schölkopf and Smola, rameter puts a penalty on patterns that are mis- re close to the SVM decision boundary. Each ple receives a value of C that depends on its a training set with an equal number of positive xamples we use two values: C_{high} for interac- to be reliable and for negative examples; C_{low} amples that are not known to be reliable.

DS

tion data

he prediction of physical interactions in yeast tion data from the BIND database (Bader *et al.*, ncludes published interaction data from high- periments as well as curated entries derived d papers. The advantage of BIND is that explicit distinction between direct physical d comembership in a complex.

e and negative examples We use physical negative examples is likely to contain very few proteins that interact.

High-throughput protein–protein interaction data contain a large fraction of false positives, estimated to be up to 50% in some experiments (von Mering *et al.*, 2002). Therefore, we prepared a set of BIND interactions that are expected to have a low rate of false positives. We use these reliable interactions in two ways. We evaluate the performance of our method on the reliable interactions because they are more likely to reflect the true performance of the classifier. We also use reliability to set the value of the SVM soft-margin parameter as discussed in Section 2.5. 'Gold standard' interactions can be derived from several sources:

- Interactions corroborated by interacting yeast paralogs. Deane *et al.* (2002) find 2829 interactions from the DIP database that are supported by their paralogous verification method (PVM). The estimated false positive rate of this method is 1%.
- Interactions that are supported by interacting homologs in multiple species are likely to be correct (Yu *et al.*, 2004).
- Interactions that are discovered by different experimental assays were estimated to be correct 95% of the time (Sprinzak *et al.*, 2003).
- Highly reliable methods, e.g. interactions derived from crystallized complexes.

We do not use PVM-validated interactions because they contain several biases.

- The test set is biased toward interactions that can be easily discovered by sequence similarity.
- The list of PVM-validated interactions cannot be used as-is to set the SVM soft-margin parameter in training because this may incorporate information about interactions that are in the test set.

Also, we do not include interactions validated by interacting homologs in other species, since that information is included in the data as a feature. Therefore, for the purpose of assessing performance we use a list of 750 interactions that were validated by high-quality or multiple assays. For setting the SVM soft-margin parameter we augment the 750 interactions with PVM-validated interactions that are computed on the basis of the training data alone. Training is performed on all

ores for the various methods computed using 5-fold cross-validation

	Kernel	ROC score	ROC_{50} score
	—	0.74	0.18
	—	0.78	0.11
	$K_{\text{non-seq}}$	0.95	0.37
	$K_{\text{p}}(K_{\text{motif}})$	0.76	0.17
	$K_{\text{p}}(K_{\text{Pfam}})$	0.78	0.20
	$K_{\text{p}}(K_{\text{spec}})$	0.81	0.05
	$K_{\text{p}}(K_{\text{motif}} + K_{\text{Pfam}})$	0.82	0.22
spectrum	$K_{\text{p}}(K_{\text{motif}} + K_{\text{Pfam}} + K_{\text{spec}})$	0.86	0.17
	$K_{\text{feat}} + K_{\text{p}}(K_{\text{motif}} + K_{\text{Pfam}} + K_{\text{spec}})$	0.97	0.44
	$K_{\text{feat}} + K_{\text{p}}(K_{\text{motif}} + K_{\text{Pfam}} + K_{\text{spec}})$	**0.97**	**0.58**

e all BIND physical interactions. ROC scores are computed on reliable interactions that do not include PVM-validated interactions. The BLAST and PSIBLAST ctions according to Equation (4). The 'kernel' column of the table shows which kernel was used in conjunction with the SVM classifier. The notation $K_{\text{p}}(K_{\text{g}})$ wise kernel was derived from a genomic kernel K_{g}. The $K_{\text{non-seq}}$ is a Gaussian kernel over the non-sequence features; in each method it participates in, the width determined by cross-validation as part of the classifier's training. The all-reliable method uses information on reliability to set the SVM soft-margin parameter ion 2.5.

itive for significant matches and increases as the match increases. The score for a query (X_1, X_2)

$$\max_{i \in \mathcal{P}} I(i, j) \min(l(X_1, X_i), l(X_2, X_j)) , \qquad (4)$$

e set of all proteins in the training set. In these we use PSI-BLAST scores computed in the e Swiss-Prot database (version 40, containing ins).

s of merit

his paper we evaluate the quality of a predictive two different metrics. Both metrics—the area iver operating characteristic curve (ROC score), lized area under that curve up to the first 50 false C_{50} score)—aim to measure both sensitivity and integrating over a curve that plots true positive on of false positive rate. We include both metrics ccount for two different types of scenarios in in–protein interaction prediction method might

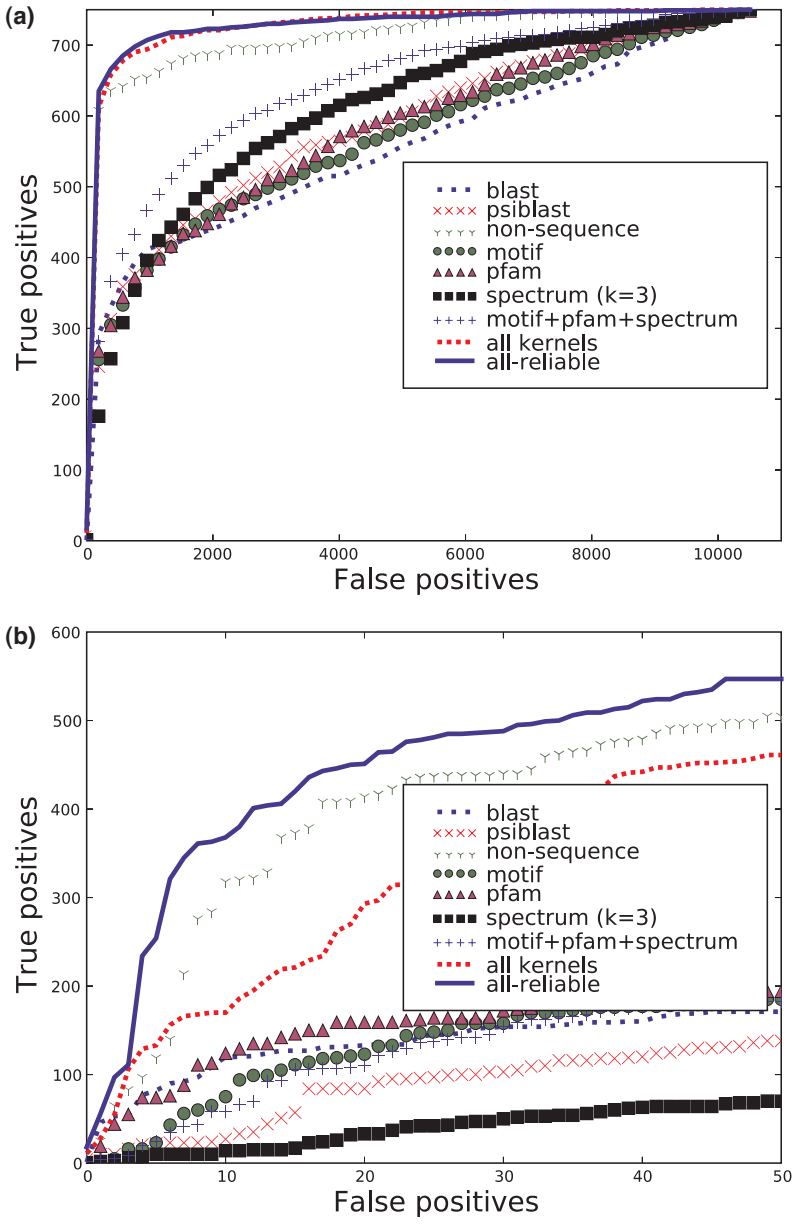
scenario, imagine that you have developed a ut method for detecting whether a given pair of acts. Rather than testing your method on ran- d pairs of proteins, you could use a predictive whether they participate in any predicted interactions. In this case, you do not care about the high-confidence interactions above; instead, you would like to be sure that the complete set of predictions is of high quality. In this case you are interested in the ROC score of the classifier.

4 RESULTS

We report, in this section, the results of experiments in predicting protein–protein interactions using an SVM classifier with various kernels, and compare these with a simple method based on BLAST or PSI-BLAST. All the experiments were performed using the PyML machine learning framework available at http://pyml.sourceforge.net. We begin this section with results obtained using the various kernels and kernel combinations, followed by a discussion of the choice of negative examples, and a section that shows the effects of choosing a non-redundant set of proteins.

4.1 Main results

We report results that are computed using 5-fold cross-validation on all BIND physical interactions. The SVM soft-margin parameter was not optimized—we used the default low value for this parameter to account for the noise in the data. The ROC/ROC_{50} curve is then computed for those reliable interactions that were not obtained using the PVM method



and ROC_{50} (**b**) curves for several methods. Best performance is obtained using a kernel that combines all the kernels presented ditional results are summarized in Table 1, along with a description of the methods.

he Pfam and motif kernels. The higher ROC

We now explore the effect of adding to the sequence ker-

tions provided another significant boost to the Its ROC and ROC_{50} scores were 0.98 and vely; at a false positive rate of 1% the classifier % of the trusted interactions. In this experiment to optimize the ratio between the two soft margin used $C_{low} = 0.01C_{high}$.

ontribution to the gain in performance comes process kernel feature. Its ROC score by itself the BIND interactions and 0.95 when limiting positive examples. The difference between the is probably due to the sizable fraction of false the BIND dataset. In the following subsection cenarios where the GO data are not useful. The r the MCC feature was 0.68 on all BIND inter- 53 when computed on the reliable interactions. erence for the MCC feature is a result of the fact requires a large number of interactions to be BLAST cutoff of $1e^{-10}$, 329 interactions from upported by interactions from other species, as negative examples. The ROC score for this fea- s low since it is sparse, i.e. is informative for a of interactions.

le of GO annotations

erstand the difference in the role of the sequence e non-sequence kernels, we compared the two task of distinguishing between physically inter- s pairs and those that are members of the same his case, the negative examples are chosen as hat are known to belong to the same complex own to physically interact. This set of negative kely to be more noisy than the non-interacting omplexes that are not accessible by yeast two- ly contain many physical interactions. But still, rwise method achieves an ROC score of 0.78, he value obtained with non-interacting negative his task, a classifier based on the non-sequence th an ROC score of 0.5. This is due to the fact that proteins, such as physically interacting proteins, imilar GO annotations and network properties, motif and Pfam rely on a signal that is often dir- o the interaction site itself (Wang *et al.*, 2005). vations can be made for other features used to

Significant attention has been paid to the problem of selecting gold standard interacting protein pairs for the purposes of training and validating predictive computational methods (Jansen *et al.*, 2003). However, less emphasis has been placed on the choice of non-interacting protein pairs. In this study, we selected negatives uniformly at random. We find that this strategy leads to consistent behavior and avoids bias.

The possibility for bias due to the method of constructing negative examples is evidenced by results reported in a related paper (Martin *et al.*, 2005). In this work, the authors report that a pairwise spectrum kernel provides highly accurate predictions of yeast interactions using a dataset studied in Jansen *et al.* (2003). The positive examples in this dataset satisfy our criteria of trusted interactions, and one might conclude that the use of highly reliable interactions is the reason for the success of the predictive method. However, we found that the method of choosing negative examples has a strong effect on the performance: the negative examples from Jansen *et al.* (2003) were chosen as pairs of proteins that are known to be localized in different cellular compartments. This makes these protein pairs much less likely to interact than randomly selected pairs, but the selection constraints impose a bias on the resulting distribution that makes the overall learning task easier [note that this is less likely to affect the results of non-sequence based methods, such as the one used by Jansen *et al.* (2003)]. To illustrate this effect, we created datasets with negative examples taken as pairs whose GO component similarity, as measured by our kernel, is below a given threshold. The performance of the resulting classifier varied as we varied this threshold (Table 2). This constrained selection method was tested with the spectrum and motif kernels using both the BIND interaction data and a set of trusted interactions similar to the one used by Martin *et al.* (2005) extracted from DIP and MIPS (Mewes *et al.*, 2000; Xenarios *et al.*, 2002). For the spectrum kernel, the ROC (ROC_{50}) scores varied from 0.87 (0.08) to 0.97 (0.46) on the DIP/MIPS data and from 0.77 (0.04) to 0.95 (0.36) on the BIND data, as the threshold was lowered from 0.5 to 0.04. Similarly, although slightly less pronounced, results were obtained for the motif pairwise kernel.

4.4 The dependence on interacting paralogs

The yeast genome contains a large number of duplicated genes. Since we are using a sequence-based method to pre-

pendence of the performance of the spectrum pairwise imilarity between localization annotations in negative

Threshold	ROC	ROC_{50}
0.50	0.77	0.04
0.10	0.89	0.15
0.07	0.91	0.21
0.05	0.92	0.25
0.04	0.95	0.36
0.5	0.87	0.08
0.1	0.94	0.22
0.07	0.95	0.32
0.05	0.96	0.34
0.04	0.97	0.46

tion that no two proteins in the set of negative examples have a less than a given threshold puts a constraint on the distribution of his constraint makes it easy for the classifier to distinguish between e examples, and the effect gets stronger as the threshold becomes ed the experiment on the BIND interaction dataset and on a dataset ns derived from DIP and MIPS interactions.

LAST (BLAST) method went down from 0.78 (0.62). This illustrates that the kernel combina- pendent on the presence of interacting paralogs r PSI-BLAST.

SION

we presented several kernels for prediction of n interactions and used them in combination for ormance. The concern regarding the pairwise gh dimensionality of its feature space, which is e number of features of the underlying kernel. d an alternative kernel which uses summation multiplication used in the expression for the el, similar to the work of Gomez *et al*. (2003). nce of the summation kernel is not as good as ding pairwise kernel, showing the advantage of features.

ng a classifier to predict protein–protein inter- is a balance between placing in the training d interactions as opposed to trying to maxim- r of positive examples by adding interactions

We also made no attempt to purge from our dataset examples that contain missing data (missing GO annotations). When trying to make predictions on unseen data, these data will contain missing data and so, the method is more likely to generalize if presented with examples containing missing data during training.

During the time of writing this paper we found that the pairwise approach was proposed by Martin *et al*. (2005). They used only the spectrum kernel, whereas here we considered several sequence kernels. We found that the spectrum kernel works better than the motif and Pfam kernels according to the ROC metric, but the spectrum kernel does not work as well as the motif and Pfam kernels according to the ROC_{50} metric. Apparently, the signal that the spectrum kernel generates is not as specific as that of the other kernels.

In addition, we have illustrated that pairwise sequence kernels can be successfully combined with non-sequence data. In this work, we have not attempted to learn the weights of the various kernels as done by Lanckriet *et al*. (2004). This is an avenue for future work, although solving the resulting semi-definite programming problem promises to be computationally expensive, owing to the large training sets involved. We also plan to consider additional sources of data such as gene expression and transcription factor binding data, which have also been shown to be informative in predicting protein–protein interactions (Zhang *et al*., 2004).

ACKNOWLEDGEMENTS

The authors thank Doug Brutlag, David Baker, Ora Schueler-Furman and Trisha Davis for the helpful discussions. This work is funded by NCRR NIH award P41 RR11823, by NHGRI NIH award R33 HG003070, and by NSF award BDI-0243257. W.S.N. is an Alfred P. Sloan Research Fellow.

REFERENCES

Bader,G.D. Donaldson,I., Wolting,C. Ouellette,B.F., Pawson,T. and Hogue,C.W. (2001) BIND—the biomolecular interaction network database. *Nucleic Acids Res*., **29**, 242–245.

Ben-hur,A. and Brutlag,D. (2003) Remote homology detection: a motif based approach. *Bioinformatics*, **19** (Suppl 1), i26–i33.

Deane,C. Salwinski,L., Xenarios,I. and Eisenberg,D. (2002) Two methods for assessment of the reliability of high throughput

I., Greenbaum,D., Kluger,Y., Krogan,N.J., Chung,S., nyder,M., Greenblatt,J.F. and Gerstein,M. (2003) networks approach for predicting protein–protein from genomic data. *Science*, **302**, 449–453.

G., Deng,M., Cristianini,N., Jordan,M.I. and 2004) Kernel-based data fusion and its application to ion prediction in yeast. In Altman,R.B., Dunker,A.K., ung,T.A. and Klein,T.E. (eds), *Proceedings of the osium on Biocomputing*, World Scientific, Singapore,

n,E. and Noble,W.S. (2002) The spectrum kernel: nel for SVM protein classification. In Altman,R.B., , Hunter,L., Lauderdale,K. and Klein,T.E. (eds), *Pro- he Pacific Symposium on Biocomputing*, New Jersey. tific, Singapore, pp. 564–575.

Jansen,R., Gerstein,M. and Zhao,H. (2004) Inform- ment on predicting protein–protein interactions. *ormatics*, **5**, 154.

Pellegrini,M., Ng,H.-L., Rice,D.W., Yeates,T.O. g,D. (1999) Detecting protein function and protein– ractions from genome sequences. *Science*, **285**,

,D. and Faulon J.-L. (2005) Predicting protein– actions using signature products. *Bioinformatics*, **21**,

rishman,D., Gruber,C., Geier,B., Haase,D., Kaps,A., Mannhaupt,G., Pfeiffer,F., Schüller,C., Stocker,S. (2000) MIPS: a database for genomes and protein *Nucleic Acids Res.*, **28**, 37–40.

g,C.G., Sethi,K.S., Wu,T.D. and Brutlag,D.L. (1997) and ranking discrete motifs. In *Proceedings of ternational Conference on Intelligent Systems for iology*, pp. 202–209.

004) Support vector machine applications in com- iology. In Schoelkopf,B., Tsuda,K. and Vert,J.-P. *el Methods in Computational Biology*. MIT Press, MA, pp. 71–92.

Pazos,F. and Valencia,A. (2002) *In silico* two-hybrid system for the selection of physically interacting protein pairs. *Proteins*, **47**, 219–227.

Ramani,A. and Marcotte,E. (2003) Exploiting the co-evolution of interacting proteins to discover interaction specificity. *J. Mol. Biol.*, **327**, 273–284.

Schölkopf,B. and Smola,A. (2002) *Learning with Kernels*. MIT Press, Cambridge, MA.

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.

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

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

Wang,H., Segal,E., Ben-Hur,A., Koller,D. and Brutlag,D.L. (2005) Identifying protein–protein interaction sites on a genome-wide scale. In Lawrence K. Saul, Yair Weiss and Léon Bottou (eds), *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, pp. 1465–1472.

Xenarios,I., Salwinski,L., Duan,X.Q.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.

Yu,H., Luscombe,N., Lu,H., Zhu,X., Xia,Y., Han,J., Bertin,N., Chung,S., Vidal,M. and Gerstein,M. (2004) Annotation transfer between genomes: protein–protein interlogs and protein–DNA regulogs. *Genome Res.*, **14**, 1107–1118.

Zhang,L., Wong,S., King,O. and Roth,F. (2004) Predicting co-complexed protein pairs using genomic and proteomic data integration. *BMC Bioinformatics*, **5**, 38–53.