A Framework for Genomic Data Fusion and its Application to Membrane Protein Prediction

Gert R. G. Lanckriet

gert@cs.berkeley.edu

Dept. of Electr. Eng. and Comp. Sc., University of California, Berkeley, CA 94720

Tijl De Bie

tijl.debie@esat.kuleuven.ac.be

Dept. of Electr. Eng., ESAT-SCD, K. U. Leuven, Belgium

Nello Cristianini

nello@wald.ucdavis.edu

Department of Statistics, University of California, Davis, CA 95616

Michael I. Jordan

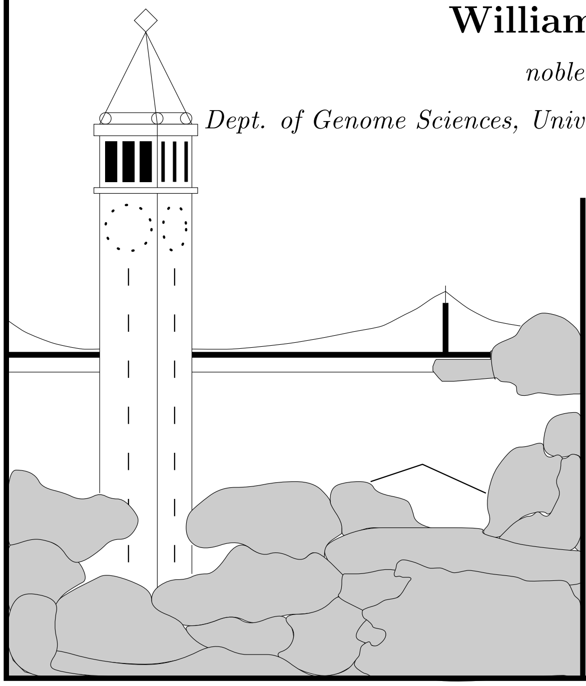
jordan@stat.berkeley.edu

Comp. Sc. Div. & Dept. of Stat., University of California, Berkeley, CA 94720

William Stafford Noble

noble@gs.washington.edu

Dept. of Genome Sciences, University of Washington, Seattle, WA 98195, USA



Report No. UCB/CSD-03-1273

September, 2003

Computer Science Division (EECS)
University of California
Berkeley, California 94720

A Framework for Genomic Data Fusion and its Application to Membrane Protein Prediction

Gert R. G. Lanckriet
gert@cs.berkeley.edu
Dept. of Electr. Eng. and Comp. Sc., University of California, Berkeley, CA 94720

Tijl De Bie
tijl.debie@esat.kuleuven.ac.be
Dept. of Electr. Eng., ESAT-SCD, K. U. Leuven, Belgium

Nello Cristianini
nello@wald.ucdavis.edu
Department of Statistics, University of California, Davis, CA 95616

Michael I. Jordan
jordan@stat.berkeley.edu
Comp. Sc. Div. & Dept. of Stat., University of California, Berkeley, CA 94720

William Stafford Noble
noble@gs.washington.edu
Dept. of Genome Sciences, University of Washington, Seattle, WA 98195, USA

September, 2003

Abstract

During the past decade, the new focus on genomics has highlighted a particular challenge: to integrate the different views of the genome that are provided by various types of experimental data. This paper describes a computational framework for integrating and drawing inferences from a collection of genome-wide measurements. Each data set is represented via a kernel function, which defines generalized similarity relationships between pairs of entities, such as genes or proteins. The kernel representation is both flexible and efficient, and can be applied to many different types of data. Furthermore, kernel functions derived from different types of data can be combined in a straightforward fashion—recent advances in the theory of kernel methods have provided efficient algorithms to perform such combinations in an optimal way. These methods formulate the problem of optimal kernel combination as a convex optimization problem that can be solved with semi-definite programming techniques. In this paper, we demonstrate the utility of these techniques by investigating the problem of predicting membrane proteins from heterogeneous data, including amino acid sequences, hydropathy profiles, gene expression data and known protein-protein interactions. A statistical learning algorithm trained from all of these data performs significantly better than the same algorithm trained on any single type of data and better than existing algorithms for membrane protein classification.

1 Introduction

The recent availability of multiple types of genome-wide data provides biologists with complementary views of a single genome and highlights the need for algorithms capable of unifying these views. In yeast, for example, for a given gene we typically know the protein it encodes, that protein's similarity to other proteins, its hydrophobicity profile, the mRNA expression levels associated with the given gene under hundreds of experimental conditions, the occurrences of known or inferred transcription factor binding sites in the upstream region of that gene, and the identities of many of the proteins that interact with the given gene's protein product. Each of these distinct data types provides one view of the molecular machinery of the cell. In the near future, research in bioinformatics will focus more and more heavily on methods of data fusion.

Different data sources are likely to contain different and thus partly independent information about the task at hand. Combining those complementary pieces of information can be expected to enhance the total information about the problem at hand. One problem with this approach, however, is that genomic data come in a wide variety of data formats: expression data are expressed as vectors or time series; protein sequence data as strings from a 20-symbol alphabet; gene sequences are strings from a different (4-symbol) alphabet; protein-protein interactions are best expressed as graphs, and so on.

This paper presents a computational and statistical framework for integrating heterogeneous descriptions of the same set of genes. The approach relies on the use of kernel-based statistical learning methods that have already proven to be very useful tools in bioinformatics (1, 2, 3, 4). These methods represent the data by means of a kernel function, which defines similarities between pairs of genes, proteins, etc. Such similarities can be quite complex relations, implicitly capturing aspects of the underlying biological machinery. One reason for the success of kernel methods is that the kernel function takes relationships that are implicit in the data and makes them explicit, so that it is easier to detect patterns. Each kernel function thus extracts a specific type of information from a given data set, thereby providing a partial description or view of the data. Our goal is to find a kernel that best represents all of the information available for a given statistical learning task. Given many partial descriptions of the data, we solve the mathematical problem of combining them using a convex optimization method known as semi-definite programming (SDP) (5, 6, 7). This SDP-based approach (8) yields a general methodology for combining many partial descriptions of data that is statistically sound, as well as computationally efficient and robust.

In order to demonstrate the feasibility of these methods, we address the problem of identifying membrane proteins. Integral plasma membrane proteins serve several important communicative functions between the inside and the outside of the cell (9). Classifying a protein as a membrane protein or not based on protein sequence is non-trivial and has been the subject of much previous research (10, 11, 12). This is a typical statistical learning problem in which a single type of feature derived from the protein sequence cannot provide the full story. We demonstrate that incorporating knowledge derived from the amino acid sequences, hydropathy profiles, gene expression data and known protein-protein interactions significantly improves classification performance relative to state-of-the-art methods (11) and relative to our method trained on any single type of data.

We begin by describing and motivating the problem of membrane protein classification. We then outline the main ideas of the kernel approach to pattern analysis and describe how different kernels defined on different data can be integrated using SDP to provide a unified description. In this section, we also define a number of kernels that are designed to capture different features of protein sequences, expression data, and protein-protein interactions, including a novel, knowledge-based kernel that is based upon the fast Fourier transform (FFT). This FFT kernel is designed to capture specific features of membrane protein sequences. Finally, we describe a series of computational experiments that demonstrate the validity and power of the kernel approach to data fusion.

2 Methods and Approach

2.1 Membrane Protein Classification

Membrane proteins are proteins that anchor in one of various membranes in the cell, including the plasma, ER, golgi, peroxisomal, vacuolar, cellular and mitochondrial inner and outer membranes. Many membrane proteins serve important communicative functions. Generally, each membrane protein passes through the membrane several times. The transmembrane regions of the amino acid sequence are typically hydrophobic, whereas the non-membrane portions are hydrophilic. This specific hydrophobicity profile of the protein allows it to anchor itself in the cell membrane.

Because the hydrophobicity profile of a membrane protein is critical to its function, this profile is better conserved in evolution than the specific amino acid sequence. Therefore, classical methods for determining whether a protein spans a membrane (12) depend upon a *hydropathy profile*, which plots the hydrophobicity of the amino acids along the protein (10, 13, 14). In this work, we build on these classical methods by developing a kernel function that is based on the low-frequency alternation of hydrophobic and hydrophilic regions in membrane proteins. However, we also demonstrate that the hydropathy profile provides only partial evidence for transmembrane regions. Additional information is gleaned from sequence homology and from protein-protein interactions.

Note that, in general, membrane protein prediction consists of predicting the locations of multiple transmembrane regions within a single protein. In this work, however, for the purposes of demonstrating the SDP method, we focus on the binary prediction task of differentiating between membrane and non-membrane proteins.

2.2 Kernel Methods

Kernel methods work by embedding data items (genes, proteins, etc.) into a vector space, called a *feature space* (15, 16, 17, 18, 19). A key characteristic of kernel methods is that the embedding in feature space is generally defined implicitly, by specifying an inner product for the feature space. Thus, for a pair of data items, x_1 and x_2, denoting their embeddings as $\phi(x_1)$ and $\phi(x_2)$, respectively, we specify the inner product of the embedded data, $\langle \phi(x_1), \phi(x_2) \rangle$, via a *kernel function* $K(x_1, x_2)$. Any symmetric, positive semi-definite function is a valid kernel function, corresponding to an inner product in some feature space. Note that if all we require are inner products, then we do not need to have an explicit representation of the mapping ϕ, nor do we even need to know the nature of the feature space. It suffices to be able to evaluate the kernel function.

Evaluating the kernel on all pairs of data points yields a symmetric, positive semi-definite matrix known as the *kernel matrix* or the *Gram matrix*. Intuitively, a kernel matrix can be regarded as a matrix of generalized similarity measures among the data points. The first stage of processing in a kernel method is to reduce the data by computing this matrix.

The reduction to a kernel matrix reflects the fact that kernel methods are generally based on linear statistical procedures in feature space. In particular, the classification algorithm that we use in this paper—known as a *support vector machine* (SVM, 20)—forms a linear discriminant boundary in feature space. Consider a data set consisting of n pairs (x_i, y_i), where x_i is the ith

Table 1: **Kernel functions.** The table lists the seven kernels used to compare proteins, the data on which they are defined, and the method for computing similarities. The final kernel, K_{RND}, is included as a control. All kernels matrices, along with the data from which they were generated, are available at `noble.gs.washington.edu/sdp-svm`.

Kernel	Data	Similarity measure
K_{SW}	protein sequences	Smith-Waterman
K_B	protein sequences	BLAST
K_{HMM}	protein sequences	Pfam HMM
K_{FFT}	hydropathy profile	FFT
K_{LI}	protein interactions	linear kernel
K_D	protein interactions	diffusion kernel
K_E	gene expression	radial basis kernel
K_{RND}	random numbers	radial basis kernel

data item (e.g., a protein), and $y_i \in \{-1, 1\}$ is a label (e.g., membrane or non-membrane). Compute the $n \times n$ kernel matrix whose (i, j)th entry is $K(x_i, x_j)$. Given this matrix, and given the labels y_i, we can throw away the original data; the problem of fitting the SVM to data reduces to an optimization procedure that is based entirely on the kernel matrix and the labels.

Different kernel functions correspond to different embeddings of the data and thus can be viewed as capturing different notions of similarity. For example, in a space derived from amino acid sequences, two genes that are close to one another will have protein products with very similar amino acid sequences. This amino acid space would be quite different from a space derived from microarray gene expression measurements, in which closeness would indicate similarity of the expression profiles of the genes. In general, a single type of data can be mapped into many different feature spaces. The choice of feature space is made implicitly via the choice of kernel function.

For the task of membrane protein classification we experiment with seven kernel matrices derived from three different types of data: four from the primary protein sequence, two from protein-protein interaction data, and one from mRNA expression data. These are summarized in Table 1.

2.2.1 Protein sequence: Smith-Waterman, BLAST and Pfam HMM kernels.

A homolog of a membrane protein is likely also to be located in the membrane. Therefore, we define three kernel matrices based upon standard homology detection methods. The first two sequence-based kernel matrices (K_{SW} and K_B) are generated using the BLAST (21) and Smith-Waterman (SW) (22) pairwise sequence comparison algorithms, as described previously (23). Because matrices of BLAST or Smith-Waterman scores are not necessarily positive semi-definite, we represent each protein as a vector of scores against all other proteins. Defining the similarity between proteins as the inner product between the score vectors (the so-called empirical kernel map, 24) leads to a valid kernel matrix, one for the BLAST score and one for the SW score. Note that including in the comparison set proteins with unknown subcellular locations allows the kernel to exploit this unlabelled data. The third kernel matrix (K_{HMM}) is a generalization of the previous pairwise comparison-based matrices in which the pairwise comparison scores are replaced by expectation values derived from hidden Markov models in the Pfam database (25). These similarity measures are not specific to the membrane protein classification task.

2.2.2 Protein sequence: FFT kernel.

In contrast, the fourth sequence-based kernel matrix (K_{FFT}) directly incorporates information about hydrophobicity patterns, which are known to be useful in identifying membrane proteins. The kernel uses hydropathy profiles generated from the Kyte-Doolittle index (26). This kernel compares the frequency content of the hydropathy profiles of the two proteins. After pre-filtering the hydropathy profiles, their Fourier transforms (describing the frequency content) are computed using an FFT algorithm. The frequency contents of different profiles are compared by applying a Gaussian kernel function, $K(x_1, x_2) = \exp(-||x_1 - x_2||^2/2\sigma)$ with width $\sigma = 10$, to the corresponding vectors of FFT values. This kernel detects periodicities in the hydropathy profile, a feature that is relevant to the identification of membrane proteins and complementary to the previous, homology-based kernels.

2.2.3 Protein interactions: linear and diffusion kernels.

We expect information about protein-protein interactions to be informative in this context for two reasons. First, hydrophobic molecules or regions of molecules tend to interact with each other. Second, transmembrane proteins are often involved in signaling pathways, and therefore different membrane proteins are likely to interact with a similar class of molecules upstream and downstream in these pathways (e.g., hormones upstream or kinases downstream). The two protein interaction kernels are generated using medium- and high-confidence interactions from a database of known interactions (27). These interactions can be represented as an interaction matrix, in which rows and columns correspond to proteins, and binary entries indicate whether the two proteins interact.

The first interaction kernel matrix (K_{LI}) is comprised of linear interactions, i.e., inner products of rows and columns from the centered, binary interaction matrix. The more similar the interaction pattern (corresponding to a row or column from the interaction matrix) is for a pair of proteins, the larger the inner product will be.

An alternative way to represent the same interaction data is to consider the proteins as nodes in a large graph. In this graph, two proteins are linked when they interact and otherwise not. Kondor and Lafferty (28) propose a general method for establishing similarities between the nodes of a graph, based on a random walk on the graph. This method efficiently accounts for all possible paths connecting two nodes, and for the lengths of those paths. Nodes that are connected by shorter paths or by many paths are considered more similar. The resulting *diffusion kernel* generates the second interaction kernel matrix (K_D).

An appealing characteristic of the diffusion kernel is its ability, like the empirical kernel map, to exploit unlabelled data. In order to compute the diffusion kernel, a graph is constructed using all known protein-protein interactions, including interactions involving proteins whose subcellular locations are unknown. Therefore, the diffusion process includes interactions involving unlabelled proteins, even though the kernel matrix only contains entries for labelled proteins. This allows two labelled proteins to be considered close to one another if they both interact with an unlabelled protein.

2.2.4 Gene expression: radial basis kernel.

Finally, we also include a kernel constructed entirely from microarray gene expression measurements. A collection of 441 distinct experiments was downloaded from the Stanford Microarray Database (`genome-www.stanford.edu/microarray`). This data provides us with a 441-element expression vector characterizing each gene. A Gaussian kernel matrix (K_E) is computed from these vectors by applying a Gaussian kernel function with width $\sigma = 100$ to each pair of 441-element vectors, characterizing a pair of genes. Note that we do not expect that gene expression will be particularly useful for the membrane classification task. We do not need to make this decision *a priori*, however; as explained in the following section, our method is able to provide an *a posteriori* measure of how useful a data source is relative to the other sources of data. We thus include the expression kernel in our experiments to test this aspect of the method.

2.3 Kernel Methods for Data Fusion

Each of the kernel functions described above produces, for the yeast genome, a square matrix in which each entry encodes a particular notion of similarity of one yeast protein to another. Implicitly, each matrix also defines an embedding of the proteins in a feature space. Thus, the kernel representation casts heterogeneous data—variable-length amino acid strings, real-valued gene expression data, and a graph of protein-protein interactions—into the common format of kernel matrices.

The kernel formalism also allows these various matrices to be combined. Basic algebraic operations such as addition, multiplication and exponentiation preserve the key property of positive semi-definiteness, and thus allow a simple but powerful algebra of kernels (29). For example, given two kernel functions K_1 and K_2, inducing the embeddings $\phi_1(x)$ and $\phi_2(x)$, respectively, it is possible to define the kernel $K = K_1 + K_2$, inducing the embedding $\phi(x) = [\phi_1(x), \phi_2(x)]$. Of even greater interest, we can consider parameterized combinations of kernels. In particular, given a set of kernels $\mathcal{K} = \{K_1, K_2, \ldots, K_m\}$, we can form the linear combination

$$K = \sum_{i=1}^{m} \mu_i K_i, \tag{1}$$

where the weights are constrained to be non-negative to assure positive semi-definiteness: $\mu_i \geq 0; i = 1, \ldots, m$. We consider this kind of kernel combination in this paper.

As we have discussed, fitting a kernel-based statistical classifier (such as the SVM) to data involves solving an optimization problem based on the kernel matrix and the labels. In particular, the SVM finds a linear discriminant in feature space that has maximal distance ("margin") between the members of the positive and negative classes. The algorithm for finding this optimal linear discriminant involves solving an optimization problem known as a *quadratic program*, a particular form of convex optimization problem for which efficient solutions are known (6).

In (8), we show that it is possible to extend this optimization problem not only to find optimal linear discriminant boundaries but also to find optimal values of the coefficients μ_i in Equation 1. In particular, in the case of the SVM, the problem of finding optimal μ_i reduces to a convex optimization problem known as a *semi-definite program (SDP).* Semi-definite programming can be viewed as a generalization of linear programming. Whereas a linear program involves the optimization of a linear function over the nonnegative orthant, a semi-definite program is a more general problem that involves the optimization of a linear function over the cone of semi-definite matrices. Linear programs and semi-definite programs are both instances of convex optimization problems, and both can be solved via efficient interior-point algorithms (5, 6, 7).

Thus, by solving an SDP, we are able to find an adaptive combination of kernel matrices—and thus an adaptive combination of heterogeneous information sources—that solves our classification problem. The output of our procedure is a set of weights μ_i and a discriminant function based on these weights. We obtain a classification decision that merges information encoded in the various kernel matrices, and we obtain weights μ_i that reflect the relative importance of these information sources.

2.4 Experimental Design

In order to test our kernel-based approach in the setting of membrane protein classification, we use as a gold standard the annotations provided by the Munich Information Center for Protein Sequences Comprehensive Yeast Genome Database (CYGD) (30). The CYGD assigns subcellular locations to 2318 yeast proteins, of which 497 belong to various membrane protein classes. The remaining approximately 4000 yeast proteins have uncertain location and are therefore not used in these experiments.

The primary input to the classification algorithm is the collection of kernel matrices from Table 1. Using the SDP techniques described above, we find an optimal combination of the seven kernel matrices, and the resulting matrix is used to train an SVM classifier.

For comparison with the SDP/SVM learning algorithm, we consider several classical biological methods that are commonly used to determine whether a Kyte-Doolittle plot corresponds to a membrane protein, as well as a state-of-the-art technique using hidden Markov models (HMMs) to predict transmembrane helices in proteins (11, 12). The first method relies on the observation that the average hydrophobicity of membrane proteins tends to be higher than that of non-membrane proteins, because the transmembrane regions are more hydrophobic. We therefore define f_1 as the average hydrophobicity, normalized by the length of the protein. We will compare the classification performance of our statistical learning algorithm with this metric.

Clearly, however, f_1 is too simplistic. For example, protein regions that are not transmembrane only induce noise in f_1. Therefore, an alternative metric filters the hydrophobicity plot with a low-pass filter and then computes the number, the height and the width of those peaks above a certain threshold (12). The filter is intended to smooth out periodic effects. We implement two such filters, choosing values for the filter order and the threshold based on (12). In particular, we define f_2 as the area under the 7th-order low-pass filtered Kyte-Doolittle plot and above a threshold value 2, normalized by the length of the protein. Similarly, f_3 is the corresponding area using a 20th-order filter and a threshold of 1.6.

Finally, the Transmembrane HMM (TMHMM) web server (`www.cbs.dtu.dk/services/TMHMM`) is used to make predictions for each protein. In (11), transmembrane proteins are identified by TMHMM using three different metrics: the expected number of amino acids in transmembrane helices, the number of transmembrane helices predicted by the N-best algorithm, and the expected number of transmembrane helices. Only the first two of these metrics are provided in the TMHMM output. Accordingly, we produce two lists of proteins, ranked by the number of predicted transmembrane helices (T_{PH}) and by the expected number of residues in transmembrane helices (T_{ENR}).

Each algorithm's performance is measured by splitting the data into a training and test set in a ratio of 80/20. We report the receiver operating characteristic (ROC) score, which is the area under a curve that plots true positive rate as a function of false positive rate for differing classification thresholds (31, 32). The ROC score measures the overall quality of the ranking induced by the classifier, rather than the quality of a single point in that ranking. An ROC score of 0.5 corresponds to random guessing, and an ROC score of 1.0 implies that the algorithm succeeded in putting all

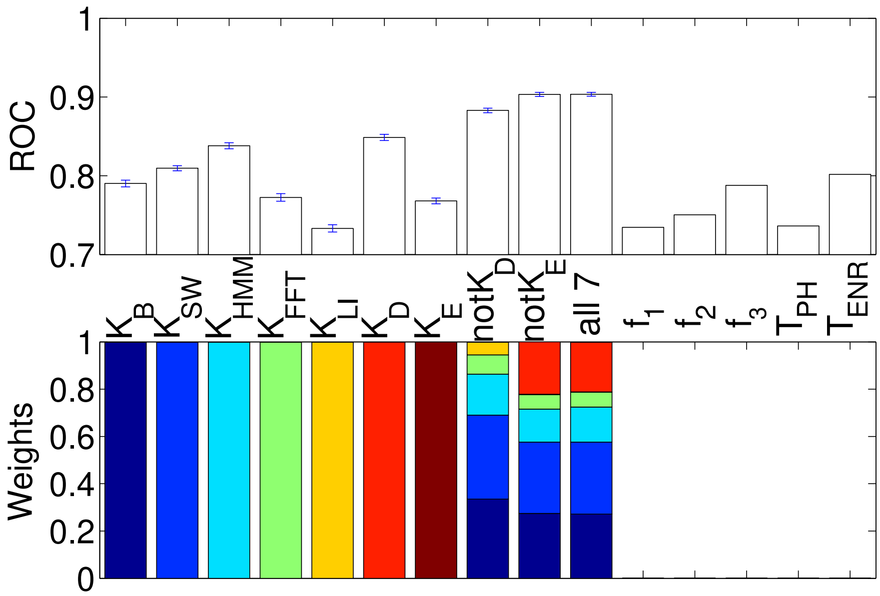


Figure 1: **Combining data sets yields better classification performance.** The height of each bar is proportional to the ROC score of the given membrane protein classification method. The bars labelled K correspond to SDP/SVM methods, the bars labelled f are hydropathy profile metrics, and the bars labelled T refer to the TMHMM methods as defined in the text. Error bars indicate standard error across 30 random train/test splits. The heights of the colored bars below each plot indicate the relative weight of the different kernel matrices in the optimal linear combination.

of the positive examples before all of the negatives. Each experiment is repeated 30 times with different random splits in order to estimate the variance of the performance values.

3 Results

We performed computational experiments which study the performance of the SDP/SVM approach as a function of the number of data sources, compare the performance of the method to classical biological methods and state-of-the-art techniques for membrane protein classification, and study the robustness of the method to the presence of noise.

The results from the first three experiments are summarized in Figure 1. The plot illustrates that SDP/SVM learns significantly better from the heterogeneous data than from any single data type. The mean ROC score using all seven kernel matrices (0.9035 ± 0.0025) is significantly higher than the best ROC score using only one matrix (0.8487 ± 0.0039 using the diffusion kernel). This improvement corresponds to a change in test set accuracy of 6.9%, from 81.3% to 88.2%.

As expected, the sequence-based kernels yield good individual performance. This is evident from the ROC scores. Furthermore, when all seven matrices are used at once, the SDP assigns relatively large weights to the sequence-based kernels. These weights are as follows: $\mu_B = 1.90$, $\mu_{SW} = 2.13$, $\mu_{HMM} = 1.04$, $\mu_{FFT} = 0.44$, $\mu_{LI} = 0.01$, $\mu_D = 1.47$ and $\mu_E = 0.01$. (Note that for ease of interpretation, we scale the weights such that their sum is equal to the number m of kernel matrices.) Thus, three of the four kernel matrices that receive weights larger than 1 are derived from the amino acid sequence. The Smith-Waterman kernel yields better results than the BLAST kernel, reflecting the fact that BLAST is a heuristic search procedure, whereas the Smith-Waterman algorithm guarantees finding the optimal local alignment of two sequences.

The results also show that the interaction-based diffusion kernel is more informative than the expression kernel. Not only has the diffusion kernel an individual ROC score which is significantly higher than the expression kernel, the SDP also assigns a weight of 1.47 to the diffusion kernel, whereas the expression kernel receives a weight of only 0.01. Accordingly, removing the diffusion kernel ("not K_D" in the plot) reduces the ROC score from 0.9035 to 0.8830, whereas removing the expression kernel ("not K_E") has almost no effect. Further description of the results obtained when various subsets of kernels are used is provided in Appendix A.

Figure 1 also compares the membrane protein classification performance of the SDP/SVM method with that of previously described techniques. The results confirm that using learning in this context dramatically improves the results relative to the simple hydropathy profile approach. Also, the SDP/SVM improves upon the performance of the TMHMM approach, even when the SVM algorithm uses only the sequence data K_{SW} or K_{HMM} (ROC of 0.8096 ± 0.0033 or 0.8382 ± 0.0038 versus 0.8018, respectively).

While the SDP/SVM algorithm is a discriminative method that attempts to find a decision boundary that separates positive and negative instances of membrane proteins, the TMHMM is a generative method that simply attempts to model the membrane proteins. As an illustration of the difference, it is known that the TMHMM tends to yield false positives for sequences containing signal peptides—hydrophobic sequences in the N-terminal regions of proteins (12). The SDP/SVM approach tends to avoid these false positives, because signal peptides appear among the negative instances in the training set. Indeed, as we show in Appendix B, signal peptides tend to be highly ranked by the TMHMM, and are more uniformly spread within the SDP/SVM rankings.

Finally, in order to test the robustness of our approach, a second experiment was performed in which a randomly generated kernel matrix K_{RND} was included among the kernel matrices used as input to our algorithm. This kernel matrix was generated by sampling 100-element vectors for each protein, where each component of each vector was sampled independently from a standard normal distribution, and then computing inner products of the 100-element vectors to form K_{RND}. A control classifier trained using only the random kernel yields an ROC score of 0.5, indicating that K_{RND} is indeed uninformative for the classification problem at hand. More importantly, when a classifier is trained using all seven real kernels plus K_{RND}, SDP assigns the random kernel a weight that is close to zero. Thus, the ROC score derived from seven matrices does not change when the random matrix is added, indicating that the method is robust to the presence of noisy, irrelevant data.

4 Discussion

We have described a general method for combining heterogeneous genome-wide data sets in the setting of kernel-based statistical learning algorithms, and we have demonstrated an application of this method to the problem of classifying yeast membrane proteins. The resulting SDP/SVM algorithm yields significant improvement in the state-of-the-art in membrane protein classification, with the performance of the algorithm improving consistently as additional genome-wide data sets are added to the kernel representation.

Kernel-based statistical learning methods have a number of general virtues as tools for biological data analysis. First, the kernel framework accommodates not only the vectorial and matrix data that are familiar in classical statistical analysis, but also more exotic data types such as strings, trees and graphs. The ability to handle such data is clearly essential in the biological domain. Second, kernels provide significant opportunities for the incorporation of more specific biological knowledge, as we have seen with the FFT kernel and the Pfam kernel. Third, the growing suite of kernel-based data analysis algorithms require only that data be reduced to a kernel matrix; this creates opportunities for standardization. Finally, as we have shown here, the reduction of heterogeneous data types to the common format of kernel matrices allows the development of general tools for combining multiple data types. Kernel matrices are required only to respect the constraint of positive semi-definiteness, and thus the powerful technique of semi-definite programming can be exploited to derive general procedures for combining data of heterogeneous format and origin.

We thus envision the development of general libraries of kernel matrices for biological data, such as those that we have provided at `noble.gs.washington.edu/sdp-svm`, that summarize the statistically-relevant features of primary data, encapsulate biological knowledge, and serve as inputs to a wide variety of subsequent data analyses. Indeed, given the appropriate kernel matrices, the methods that we have described here are applicable to problems such as the prediction of protein metabolic, regulatory and other functional classes, the prediction of protein subcellular locations, and the prediction of protein-protein interactions.

Finally, while we have focused on the binary classification problem in the current paper, there are many possible extensions of our work to other statistical learning problems. One notable example is the problem of *transduction*, in which the classifier is told *a priori* the identity of the points that are in the test set (but not their labels). This approach can deliver superior predictive performance (18), and would seem particularly appropriate in gene or protein classification problems, where the entities to be classified are often known *a priori*.

A Overview of all results

Table 2 lists results for various combinations of kernel matrices. This table includes the results depicted in Figure 1.

B Proteins with Signal Peptides

Figures 2, 3 and 4 and Table 3 illustrate the superior behavior of the SDP/SVM method with respect to proteins that contain signal peptides, as compared to TMHMM.

Signal peptides are identified by the SignalP web server (`www.cbs.dtu.dk/services/SignalP-2.0`). The server provides two types of predictions, based upon a neural network and an HMM. Here, the neural network score (NN) is the sum of the four values output by SignalP. Similarly, the HMM score is the sum of the signal peptide and signal anchor probabilities.

The figures show two complementary effects. First, many non-membrane proteins (points under the zero line) are ranked highly by T_{ENR}, while they are spread more uniformly over the ranking by T_{PH} and the SVM approach. This observation is confirmed by measuring the "distance to uniformity" for the three approaches (Table 3). This effect illustrates the sensitivity of T_{ENR} to signal peptides in non-membrane proteins, yielding false positives. Second, although both SVM and T_{PH} tend to rank the non-membrane proteins with signal peptides about equally uniformly

Table 2: **Performance of the SDP/SVM method using various combinations of kernels.** Each row in the table corresponds to one experiment, classifying the 497 known yeast membrane proteins versus the 1876 known non-membrane proteins in yeast. The data is split into train and test sets in a ratio of 80/20, and the classifier is a 1-norm soft margin SVM with C=1. The first seven columns indicate the average weight assigned via SDP to each of the seven kernel matrices. A hyphen indicates that the corresponding kernel is not considered in the combination. The rightmost columns list two performance metrics, test set accuracy (TSA) and ROC score, along with standard deviations computed across 30 randomly generated 80/20 splits.

K_B	K_{SW}	K_{HMM}	K_{HF}	K_{LI}	K_D	K_E	K_{RND}	TSA	ROC
1.00	-	-	-	-	-	-	-	$84.20 \pm 0.27\%$	$.7902 \pm .0043$
-	1.00	-	-	-	-	-	-	$84.94 \pm 0.28\%$	$.8096 \pm .0033$
-	-	1.00	-	-	-	-	-	$85.52 \pm 0.23\%$	$.8382 \pm .0038$
-	-	-	1.00	-	-	-	-	$83.31 \pm 0.27\%$	$.7725 \pm .0048$
-	-	-	-	1.00	-	-	-	$81.21 \pm 0.29\%$	$.7332 \pm .0046$
-	-	-	-	-	1.00	-	-	$81.30 \pm 0.27\%$	$.8487 \pm .0039$
-	-	-	-	-	-	1.00	-	$78.97 \pm 0.31\%$	$.7681 \pm .0037$
-	-	-	-	-	-	-	1.00	$78.39 \pm 0.31\%$	$.5409 \pm .0053$
1.13	0.87	-	-	-	-	-	-	$86.70 \pm 0.21\%$	$.8419 \pm .0029$
-	-	-	-	0.10	1.90	-	-	$81.24 \pm 0.29\%$	$.8535 \pm .0038$
1.35	-	-	-	-	0.65	-	-	$86.81 \pm 0.25\%$	$.8771 \pm .0034$
-	1.19	-	-	-	0.81	-	-	$87.34 \pm 0.21\%$	$.8822 \pm .0030$
1.84	-	-	-	0.16	-	-	-	$85.58 \pm 0.25\%$	$.8227 \pm .0042$
-	1.74	-	-	0.26	-	-	-	$86.11 \pm 0.23\%$	$.8462 \pm .0030$
1.07	1.17	-	-	-	0.76	-	-	$87.72 \pm 0.19\%$	$.8968 \pm .0026$
1.84	-	-	-	0.06	1.09	-	-	$86.99 \pm 0.26\%$	$.8821 \pm .0033$
1.35	1.41	-	-	0.23	-	-	-	$87.12 \pm 0.20\%$	$.8704 \pm .0027$
-	1.91	-	-	0.04	1.05	-	-	$87.18 \pm 0.21\%$	$.8759 \pm .0031$
-	2.27	1.21	0.50	-	-	0.03	-	$86.36 \pm 0.20\%$	$.8437 \pm .0036$
-	1.94	0.93	-	-	1.12	0.01	-	$87.67 \pm 0.19\%$	$.8914 \pm .0027$
-	2.31	-	0.44	-	1.24	0.01	-	$87.26 \pm 0.18\%$	$.8748 \pm .0030$
-	-	1.54	0.98	-	1.44	0.04	-	$87.44 \pm 0.22\%$	$.8916 \pm .0029$
-	1.27	0.73	-	-	-	-	-	$86.58 \pm 0.23\%$	$.8465 \pm .0034$
-	-	-	0.71	-	1.29	-	-	$85.87 \pm 0.24\%$	$.8587 \pm .0033$
-	1.72	0.92	0.36	-	-	-	-	$86.39 \pm 0.20\%$	$.8434 \pm .0035$
-	1.42	0.70	-	-	0.88	-	-	$87.74 \pm 0.19\%$	$.8925 \pm .0027$
-	1.73	0.87	0.33	-	1.07	-	-	$87.74 \pm 0.19\%$	$.8920 \pm .0028$
-	2.16	1.09	0.41	-	1.33	0.01	-	$87.69 \pm 0.17\%$	$.8928 \pm .0028$
-	2.60	1.31	0.50	0.01	1.57	0.01	-	$87.68 \pm 0.17\%$	$.8926 \pm .0028$
2.18	-	1.40	0.85	0.03	1.54	0.01	-	$88.02 \pm 0.20\%$	$.9061 \pm .0027$
1.96	2.15	-	0.42	0.02	1.44	0.01	-	$88.12 \pm 0.18\%$	$.8974 \pm .0027$
1.74	2.05	0.96	-	0.01	1.25	0.01	-	$87.93 \pm 0.22\%$	$.9007 \pm .0033$
1.66	1.81	0.87	0.37	-	1.28	0.01	-	$88.28 \pm 0.21\%$	$.9036 \pm .0028$
2.01	2.13	1.04	0.49	0.32	-	0.01	-	$87.51 \pm 0.19\%$	$.8830 \pm .0029$
1.65	1.81	0.84	0.37	0.01	1.33	-	-	$88.35 \pm 0.21\%$	$.9033 \pm .0026$
1.90	2.13	1.04	0.44	0.01	1.47	0.01	-	$88.23 \pm 0.23\%$	$.9035 \pm .0025$
2.16	2.51	1.27	0.52	0.01	1.40	0.01	0.12	$87.68 \pm 0.21\%$	$.9001 \pm .0026$

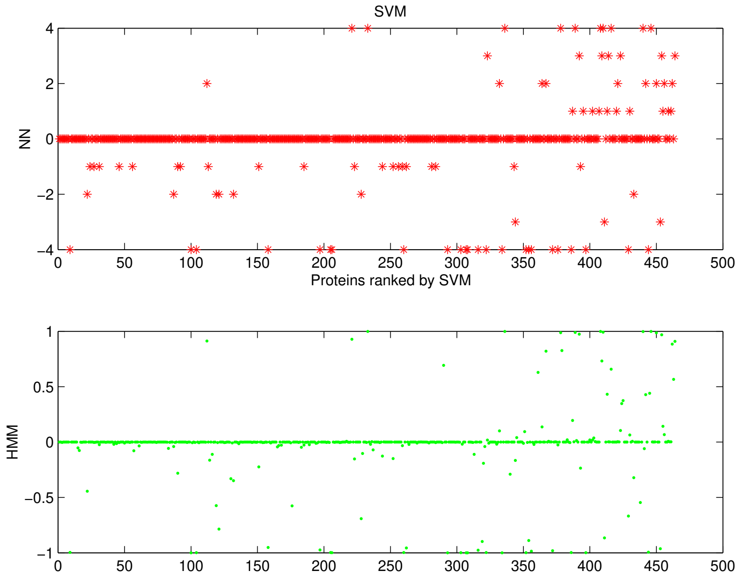


Figure 2: **Ranking of proteins by SVM, highlighting signal peptide properties.** The vertical axis plots the value of the NN and HMM scores multiplied by the true label of the protein (1 or -1). Hence, points below zero correspond to non-membrane proteins, while points above zero correspond to membrane proteins. The horizontal axis is the ranking of proteins induced by the SVM, with predicted membrane proteins on the left.

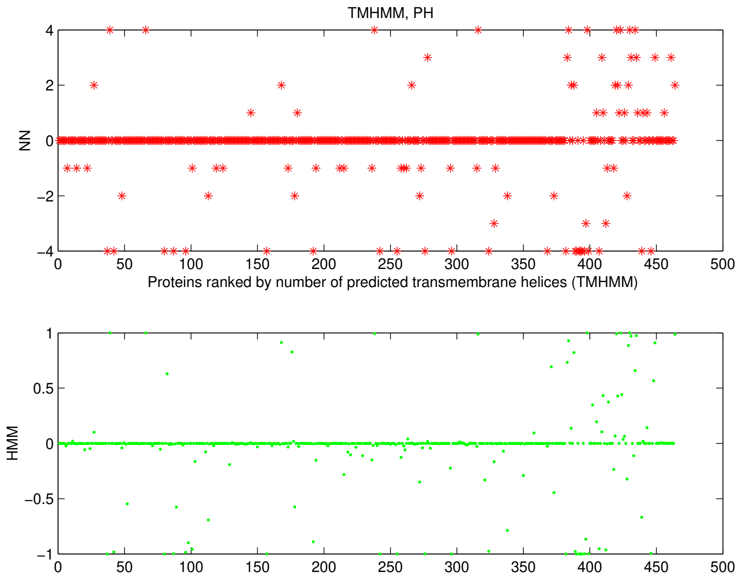


Figure 3: **Ranking of proteins by the number of TMHMM predicted transmembrane helices (T_{PH}), highlighting signal peptide properties.** This plot is similar to Figure 2.

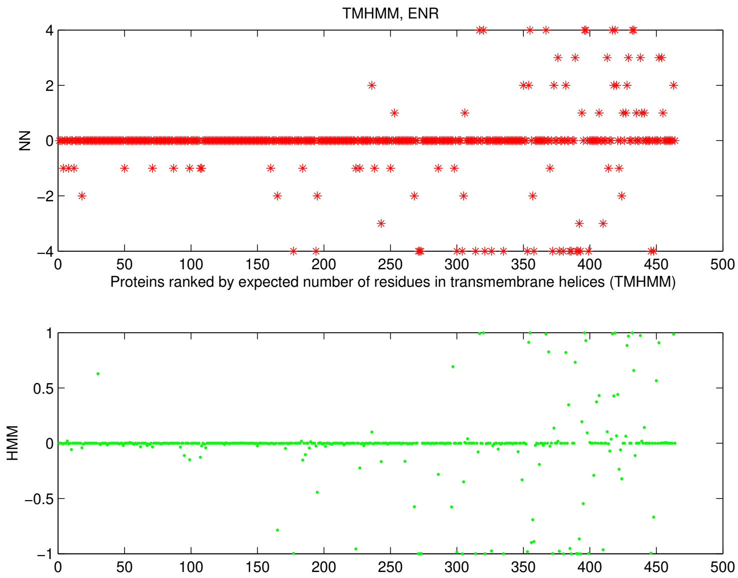


Figure 4: **Ranking of proteins by the TMHMM expected number of residues in transmembrane helices (T_{ENR}), highlighting signal peptide properties.** This plot is similar to Figure 2.

(when using HMM signal peptide predictions), T_{PH} ranks the true membrane proteins with signal peptides quite uniformly as well. This effect, which is also confirmed in Table 3, leads to a high false negative rate for T_{PH}.

Acknowledgments

WSN is supported by a Sloan Foundation Research Fellowship and by National Science Foundation grants DBI-0078523 and ISI-0093302. MIJ and GL acknowledge support from ONR MURI N00014-00-1-0637 and NSF grant IIS-9988642. TDB is a Research Assistant with the Fund for Scientific Research, Flanders (F.W.O.–Vlaanderen).

References

[1] Jaakkola, T., Diekhans, M,. & Haussler, D. (1999) in *ISMB99* (AAAI Press, Menlo Park, CA), pp. 149–158.

[2] Brown, M. P. S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C., Furey, T. S., Ares Jr., M., & Haussler, D. (2000) *PNAS* **97**, 262–267.

[3] Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D. (2000) *Bioinformatics* **16**, 906–914.

[4] Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lengauer, T., & Müller, K.-R. (2000) *Bioinformatics* **16**, 799–807.

Table 3: **"Distance to uniformity" of the ranking of membrane and non-membrane proteins with signal peptides, as provided by SVM and TMHMM.** Columns in the table correspond respectively to Figures 2, 3 and 4. The "distance to uniformity" for the ranking of non-membrane proteins (DU_{neg}) with signal peptides is obtained by plotting the cumulative absolute value of a given score (NN or HMM) of the below-the-zero-line points, and then computing the normalized 1-norm distance to the cumulative absolute value if the distribution was perfectly uniform, i.e., the line segment connecting the first and last point in the cumulative plot. The distance to uniformity for ranking of the membrane proteins (DU_{pos}) with signal peptides is obtained in a similar way, using the score of the above-the-zero-line points. Bold values indicate better behavior.

Signal Peptide	SVM		T_{ENR}		T_{PH}	
Prediction Method	DU_{neg}	DU_{pos}	DU_{neg}	DU_{pos}	DU_{neg}	DU_{pos}
NN	**0.15**	**0.66**	0.34	**0.69**	0.21	0.49
HMM	**0.17**	**0.64**	0.43	**0.65**	**0.16**	0.47

[5] Boyd, S., El Ghaoui, L., Feron, E., & Balakrishnan, V. (1994) *Linear Matrix Inequalities in System and Control Theory.* (SIAM, Philadelphia, PA).

[6] Nesterov, Y., & Nemirovsky, A. (1994) *Interior Point Polynomial Methods in Convex Programming: Theory and Applications.* (SIAM, Philadelphia, PA).

[7] Vandenberghe, L., & Boyd, S. (1996) *SIAM Review* **38**, 49–95.

[8] Lanckriet, G. R. G., Cristianini, N., Bartlett, P., El Ghaoui, L., & Jordan, M. I. (2002) in *ICML02*, eds. Sammut, C. & Hoffmann, A. (Morgan Kaufmann, San Francisco, CA).

[9] Alberts, B., Bray, D., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (1998) *Essential Cell Biology: An Introduction to the Molecular Biology of the Cell.* (Garland Science Publishing, New York).

[10] Engleman, D. M., Steitz, T. A., & Goldman, A. (1986) *Ann. Rev. Biophys. Biophys. Chem.* **15**, 321–353.

[11] Krogh, A., Larsson, B., von Heijne, G., & Sonnhammer, E. L. L. (2001) *Journal of Molecular Biology* **305**, 567–580.

[12] Chen, C. P., & Rost B. (2002) *Applied Bioinformatics* **1**, 21–35.

[13] Black, S. D., & Mould, D. R. (1991) *Anal. Biochem.* **193**, 72–82.

[14] Hopp, T. P., & Woods, K. R. (1981) *Proc. Natl. Acad. Sci. USA* **78**, 3824–3828.

[15] Cristianini, N., & Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines.* (Cambridge University Press, Cambridge, U.K.).

[16] Schölkopf, B., & Smola, A. (2002) *Learning with Kernels.* (MIT Press, Cambridge, MA).

[17] Wahba, G. (1990) *Spline Models for Observational Data.* (SIAM, Philadelphia, PA).

[18] Vapnik, V. N. (1998) *Statistical Learning Theory.* (Wiley-Interscience, New York, NY).

[19] Vapnik, V. N. (1999) *The Nature of Statistical Learning Theory.* (Springer, Heidelberg).

[20] Boser, B. E., Guyon, I., & Vapnik, V. (1992) in *COLT92* (ACM Press, Pittsburgh, PA), pp. 144–152.

[21] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990) *Journal of Molecular Biology* **215**, 403–410.

[22] Smith, T. F., & Waterman, M. S. (1981) *Journal of Molecular Biology* **147**, 195–197.

[23] Liao, L., & Noble, W. S. (2002) in *RECOMB02* (ACM Press, Washington, DC), pp. 225–232.

[24] Tsuda, K. (1999) in *ESANN99*, ed. Verleysen, M. (D-Facto Publications, Belgium), pp.183–188.

[25] Sonnhammer, E., Eddy, S., & Durbin, R. (1997) *Proteins* **28**, 405–420.

[26] Kyte, J., & Doolittle, R. F. (1982) *Journal of Molecular Biology* **157**, 105–132.

[27] von Mering, C., Krause, R., Snel, B., Cornell, M., Olivier, S. G., Fields, S., & Bork, P. (2002) *Nature* **417**, 399–403.

[28] Kondor, R. I., & Lafferty, J. (2002) in *ICML02*, eds. Sammut, C. & Hoffmann, A. (Morgan Kaufmann, San Francisco, CA).

[29] Berg, C., Christensen, J., & Ressel, P. (1984) *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions.* (Springer, New York).

[30] Mewes, H. W., Frishman, D., Gruber, C., Geier, B., Haase, D., Kaps, A., Lemcke K., Mannhaupt, G., Pfeiffer, F., Schüller, C., *et al.* (2000) *Nucleic Acids Research* **28**, 37–40.

[31] Hanley, J. A., & McNeil, B. J. (1982) *Radiology* **143**, 29–36.

[32] Gribskov, M., & Robinson, N. L. (1996) *Computers and Chemistry* **20**, 25–33.