KERNEL-BASED DATA FUSION AND ITS APPLICATION TO PROTEIN FUNCTION PREDICTION IN YEAST

GERT R. G. LANCKRIET
Division of Electrical Engineering, University of California, Berkeley

MINGHUA DENG
Department of Biological Sciences, University of Southern California

NELLO CRISTIANINI
Department of Statistics, University of California, Davis

MICHAEL I. JORDAN
Division of Computer Science, Department of Statistics, University of California, Berkeley

WILLIAM STAFFORD NOBLE
Department of Genome Sciences, University of Washington

Abstract

Kernel methods provide a principled framework in which to represent many types of data, including vectors, strings, trees and graphs. As such, these methods are useful for drawing inferences about biological phenomena. We describe a method for combining multiple kernel representations in an optimal fashion, by formulating the problem as a convex optimization problem that can be solved using semidefinite programming techniques. The method is applied to the problem of predicting yeast protein functional classifications using a support vector machine (SVM) trained on five types of data. For this problem, the new method performs better than a previously-described Markov random field method, and better than the SVM trained on any single type of data.

1 Introduction

Much research in computational biology involves drawing statistically sound inferences from collections of data. For example, the function of an unannotated protein sequence can be predicted based on an observed similarity between that protein sequence and the sequence of a protein of known function. Related methodologies involve inferring related functions of two proteins if they occur in fused form in some other organism, if they co-occur in multiple

Online supplement at `noble.gs.washington.edu/yeast`.

species, if their corresponding mRNAs share similar expression patterns, or if the proteins interact with one another.

It seems natural that, while all such data sets contain important pieces of information about each gene or protein, the comparison and fusion of these data should produce a much more sophisticated picture of the relations among proteins, and a more detailed representation of each protein. This fused representation can then be exploited by machine learning algorithms. Combining information from different sources contributes to forming a complete picture of the relations between the different components of a genome.

This paper presents a computational and statistical framework for integrating heterogeneous descriptions of the same set of genes, proteins or other entities. The approach relies on the use of kernel-based statistical learning methods that have already proven to be very useful tools in bioinformatics.[1] 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 semidefinite programming (SDP).[2] This SDP-based approach[3] 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 predicting the functions of yeast proteins. Following the experimental paradigm of Deng *et al.*,[4] we use a collection of five publicly available data sets to learn to recognize 13 broad functional categories of yeast proteins. We demonstrate that incorporating knowledge derived from amino acid sequences, protein complex data, gene expression data and known protein-protein interactions significantly improves classification performance relative to our method trained on any single type of data, and relative to a previously described method based on a Markov random field model.[4]

2 Related Work

Considerable work has been devoted to the problem of automatically integrating genomic datasets, leveraging the interactions and correlations between them to obtain more refined and higher-level information. Previous research in this field can be divided into three classes of methods.

The first class treats each data type independently. Inferences are made separately from each data type, and an inference is deemed correct if the various data agree. This type of analysis has been used to validate, for example, gene expression and protein-protein interaction data,[5,6,7] to validate protein-protein interactions predicted using five different methods,[8] and to infer protein function.[9] A slightly more complex approach combines multiple data sets using intersections and unions of the overlapping sets of predictions.[10]

The second formalism to represent heterogeneous data is to extract binary relations between genes from each data source, and represent them as graphs. As an example, sequence similarity, protein-protein interaction, gene co-expression or closeness in a metabolic pathway can be used to define binary relations between genes. Several groups have attempted to compare the resulting gene graphs using graph algorithms,[11,12] in particular to extract clusters of genes that share similarities with respect to different sorts of data.

The third class of techniques uses statistical methods to combine heterogeneous data. For example, Holmes and Bruno use a joint likelihood model to combine gene expression and upstream sequence data for finding significant gene clusters.[13] Similarly, Deng *et al.* use a maximum likelihood method to predict protein-protein interactions and protein function from three types of data.[14] Alternatively, protein localization can be predicted by converting each data source into a conditional probabilistic model and integrating via Bayesian calculus.[15] The general formalism of graphical models, which includes Bayesian networks and Markov random fields as special cases, provides a systematic methodology for building such integrated probabilistic models. As an instance of this methodology, Deng *et al.* developed a Markov random field model to predict yeast protein function.[4] They found that the use of different sources of information indeed improved prediction accuracy when compared to using only one type of data.

This paper describes a fourth type of data fusion technique, also statistical, but of a more nonparametric and discriminative flavor. The method, described in detail below, consists of representing each type of data independently as a matrix of kernel similarity values. These kernel matrices are then combined to make overall predictions. An early example of this approach, based on fixed sums of kernel matrices, showed that combinations of kernels can yield improved gene classification performance in yeast, relative to learning from a single kernel matrix.[16] The current work takes this methodology further—we use a *weighted* linear combination of kernels, and demonstrate how to estimate the kernel weights from the data. This yields not only predictions that reflect contributions from multiple data sources, but also yields an indication of the relative importance of these sources.

The graphical model formalism, as exemplified by the Markov random field model of Deng *et al.*, has several advantages in the biological setting. In particular, prior knowledge can be readily incorporated into such models, with standard Bayesian inference algorithms available to combine such knowledge with data. Moreover, the models are flexible, accommodating a variety of data types and providing a modular approach to combining multiple data sources. Classical discriminative statistical approaches, on the other hand, can provide superior performance in simple situations, by focusing explicitly on the boundary between classes, but tend to be significantly less flexible and less able to incorporate prior knowledge. As we discuss in this paper, however, recent developments in kernel methods have yielded a general class of discriminative methods that readily accommodate non-standard data types (such as strings, trees and graphs), allow prior knowledge to be brought to bear, and provide general machinery for combining multiple data sources.

3 Methods and Approach

Kernel Methods Kernel methods work by embedding data items (genes, proteins, etc.) into a vector space $\mathcal{F}$, called a *feature space*, and searching for linear relations in such a space. This embedding is defined implicitly, by specifying an inner product for the feature space via a positive semidefinite *kernel function*: $K(\mathbf{x}_1, \mathbf{x}_2) = \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2) \rangle$, where $\Phi(\mathbf{x}_1)$ and $\Phi(\mathbf{x}_2)$ are the embeddings of data items $\mathbf{x}_1$ and $\mathbf{x}_2$. Note that if all we require in order to find those linear relations 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, which is often much easier than computing the coordinates of the points explicitly. Evaluating the kernel on all pairs of data points yields a symmetric, positive semidefinite matrix K known as the *kernel matrix*, which can be regarded as a matrix of generalized similarity measures among the data points.

The kernel-based binary classification algorithm that we use in this paper, the *1-norm soft margin support vector machine*,[17,18] forms a linear discriminant boundary in feature space $\mathcal{F}$, $f(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x}) + b$, where $\mathbf{w} \in \mathcal{F}$ and $b \in \mathbb{R}$.

Given a labelled sample $S_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, $\mathbf{w}$ and b are optimized to maximize the distance ("margin") between the positive and negative class, allowing misclassifications (therefore "soft margin"):

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\Phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i \geq 0, \quad i = 1, \ldots, n \end{aligned} \tag{1}$$

where C is a regularization parameter, trading off error against margin. By considering the corresponding dual problem of (1), one can prove [18] that the weight vector can be expressed as $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i)$, where the support values α_i are solutions of the following dual *quadratic program* (QP):

$$\max_{\alpha} \quad 2\alpha^T\mathbf{e} - \alpha^T \text{diag}(\mathbf{y}) K \text{diag}(\mathbf{y})\alpha \;:\; C \geq \alpha \geq 0, \;\; \alpha^T\mathbf{y} = 0,$$

An unlabelled data item $\mathbf{x}_{new}$ can subsequently be classified by computing the linear function

$$f(\mathbf{x}_{new}) = \mathbf{w}^T\Phi(\mathbf{x}_{new}) + b = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}_{new}) + b.$$

If $f(\mathbf{x}_{new})$ is positive, then we classify $\mathbf{x}_{new}$ as belonging to class $+1$; otherwise, we classify $\mathbf{x}_{new}$ as belonging to class -1.

Kernel Methods for Data Fusion Given multiple related data sets (e.g., gene expression, protein sequence, and protein-protein interaction data), each kernel function 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, 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 semidefiniteness, and thus allow a simple but powerful algebra of kernels.[19] For example, given two kernels K_1 and K_2, inducing the embeddings $\Phi_1(\mathbf{x})$ and $\Phi_2(\mathbf{x})$, respectively, it is possible to define the kernel $K = K_1 + K_2$, inducing the embedding $\Phi(\mathbf{x}) = [\Phi_1(\mathbf{x}), \Phi_2(\mathbf{x})]$. Of even greater interest, we can consider parameterized combinations of kernels.

In this paper, given a set of kernels $\mathcal{K} = \{K_1, K_2, \ldots, K_m\}$, we will form the linear combination

$$K = \sum_{i=1}^{m} \mu_i K_i. \tag{2}$$

As we have discussed, fitting an SVM to a single data source involves solving a QP based on the kernel matrix and the labels. We have shown 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 (2) for problems involving multiple kernels.[3] In the case of the 1-norm soft margin SVM, we want to minimize the same cost function (1), now with respect to both the discriminant boundary and the μ_i. Again considering the Lagrangian dual problem, it turns out [3] that the problem of finding optimal μ_i and α_i reduces to a convex optimization problem known as a *semidefinite program (SDP)*:

$$\begin{array}{ll}\displaystyle\min_{\mu_i, t, \lambda, \nu, \delta} & t \\ \text{subject to} & \operatorname{trace}\left(\displaystyle\sum_{i=1}^{m} \mu_i K_i\right) = c, \\ & \displaystyle\sum_{i=1}^{m} \mu_i K_i \succeq 0, \\ & \begin{pmatrix} \operatorname{diag}(\mathbf{y})(\sum_{i=1}^{m} \mu_i K_i)\operatorname{diag}(\mathbf{y}) & \mathbf{e} + \nu - \delta + \lambda \mathbf{y} \\ (\mathbf{e} + \nu - \delta + \lambda \mathbf{y})^T & t - 2C\delta^T \mathbf{e} \end{pmatrix} \succeq 0, \\ & \nu, \delta \geq 0, \end{array} \tag{3}$$

where c is a constant. SDP can be viewed as a generalization of linear programming, where scalar linear inequality constraints are replaced by more general linear matrix inequalities (LMIs): $F(\mathbf{x}) \succeq 0$, meaning that the matrix F has to be in the cone of positive semidefinite matrices, as a function of the decision variables $\mathbf{x}$. Note that the first LMI constraint in (3), $K = \sum_{i=1}^{m} \mu_i K_i \succeq 0$, emerges very naturally because the optimal kernel matrix must indeed come from the cone of positive semidefinite matrices. Linear programs and semidefinite programs are both instances of convex optimization problems, and both can be solved via efficient interior-point algorithms.[2]

In this paper, the weights μ_i are constrained to be non-negative and the K_i are positive semidefinite and normalized ($[K_i]_{jj} = 1$) by construction; thus $K \succeq 0$ is automatically satisfied. In that case, one can prove [3] that the SDP (3) can be cast as a *quadratically constrained quadratic program (QCQP)*, which

Table 1: **Functional categories.** The table lists the 13 CYGD functional classifications used in these experiments. The class listed as "others" is a combination of four smaller classes: (1) cellular communication/signal transduction mechanism, (2) protein activity regulation, (3) protein with binding function or cofactor requirement (structural or catalytic) and (4) transposable elements, viral and plasmid proteins.

	Category	Size		Category	Size
1	metabolism	1048	8	cell rescue, defense & virulence	264
2	energy	242	9	interaction w/ cell. envt.	193
3	cell cycle & DNA processing	600	10	cell fate	411
4	transcription	753	11	control of cell. organization	192
5	protein synthesis	335	12	transport facilitation	306
6	protein fate	578	13	others	81
7	cellular transp. & transp. mech.	479			

improves the efficiency of the computation:

$$\begin{aligned} \max_{\alpha,t} \quad & 2\alpha^T \mathbf{e} - ct \\ \text{subject to} \quad & t \geq \frac{1}{n}\alpha^T \mathrm{diag}(\mathbf{y}) K_i \mathrm{diag}(\mathbf{y})\alpha, \quad i = 1, \ldots, m \\ & \alpha^T \mathbf{y} = 0, \\ & C \geq \alpha \geq 0. \end{aligned} \tag{4}$$

Thus, by solving a QCQP, 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.

4 Experimental Design

In order to test our kernel-based approach, we follow the experimental paradigm of Deng *et al.*[4] The task is predicting functional classifications associated with yeast proteins, and we use as a gold standard the functional catalogue provided by the MIPS Comprehensive Yeast Genome Database (CYGD—`mips.gsf.de/proj/yeast`). The top-level categories in the functional hierarchy produce 13 classes (see Table 1). These 13 classes contain 3588 proteins; the remaining yeast proteins have uncertain function and are therefore not used in evaluating the classifier. Because a given protein can belong to several functional classes, we cast the prediction problem as 13 binary classification tasks, one for each functional class.

The primary input to the classification algorithm is a collection of kernel matrices representing different types of data. In order to compare the SDP/SVM approach to the MRF method of Deng *et al.*, we perform two variants of the experiment: one in which the five kernels are restricted to contain precisely the same binary information as used by the MRF method, and a second experiment in which two of the kernels use richer representations and a sixth kernel is added.

For the first kernel, the domain structure of each protein is summarized using the mapping provided by SwissProt v7.5 (`us.expasy.org/sprot`) from protein sequences to Pfam domains (`pfam.wustl.edu`). Each protein is characterized by a 4950-bit vector, in which each bit represents the presence or absence of one Pfam domain. The kernel function K_{Pfam} is simply the inner product applied to these vectors. This bit vector representation was used by the MRF method. In the second experiment, the domain representation is enriched by adding additional domains (Pfam 9.0 contains 5724 domains) and by replacing the binary scoring with log E-values derived by comparing the HMMs with a given protein using the HMMER software toolkit (`hmmer.wustl.edu`).

Three kernels are derived from CYGD information regarding three different types of protein interactions: protein-protein interactions, genetic interactions, and co-participation in a protein complex, as determined by tandem affinity purification (TAP). All three data sets can be represented as graphs, with proteins as nodes and interactions as edges. Kondor and Lafferty [20] propose a general method for establishing similarities among 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 three interaction kernel matrices, K_{Gen}, K_{Phys} and K_{TAP}. Because direct physical interaction is not necessarily guaranteed when two proteins participate in a complex, a smaller diffusion constant is used to construct K_{TAP}, i.e., $\tau = 1$ instead of $\tau = 5$ for the others.

The fifth kernel is generated using 77 cell cycle gene expression measurements per gene.[21] Two genes with similar expression profiles are likely to have similar functions; accordingly, Deng *et al.* convert the expression matrix to a square binary matrix in which a 1 indicates that the corresponding pair of expression profiles exhibits a Pearson correlation greater than 0.8. We use this matrix to form a diffusion kernel K_{Exp}. In the second experiment, a Gaussian kernel is defined directly on the expression profiles: for expression profiles $\mathbf{x}$ and $\mathbf{z}$, the kernel is $K(\mathbf{x}, \mathbf{z}) = \exp(-||\mathbf{x} - \mathbf{z}||^2/2\sigma)$ with width $\sigma = 0.5$.

In the second experiment, we construct one additional kernel matrix by

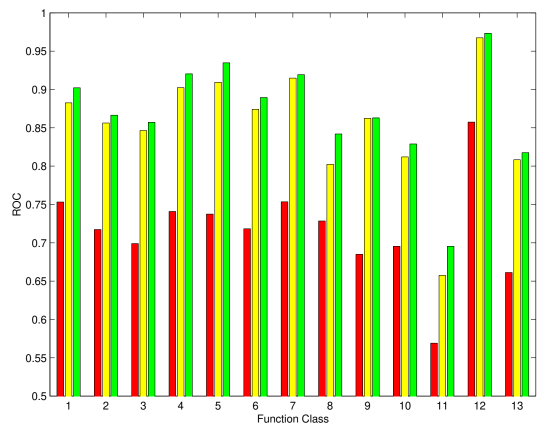

Figure 1: **Classification performance for the 13 functional classes.** The height of each bar is proportional to the ROC score. The standard deviation across the 15 experiments is usually 0.01 or smaller (see supplement), so most of the depicted differences are significant. Red bars correspond to the MRF method of Deng *et al.*; yellow bars correspond to the SDP/SVM method using five kernels computed on binary data, and green bars correspond to the SDP/SVM using the enriched Pfam kernel and replacing the expression kernel with the SW kernel.

applying the Smith-Waterman pairwise sequence comparison algorithm [22] to the yeast protein sequences. Each protein is represented as a vector of Smith-Waterman log E-values, computed with respect to all 6355 yeast genes. The kernel matrix K_{SW} is computed using an inner product applied to pairs of these vectors. This matrix is complementary to the Pfam domain matrix, capturing sequence similarities among yeast genes, rather than similarities with respect to the Pfam database.

Each algorithm's performance is measured by performing 5-fold cross-validation three times. For a given split, we evaluate each classifier by reporting the receiver operating characteristic (ROC) score on the test set. The ROC score is the area under a curve that plots true positive rate as a function of false positive rate for differing classification thresholds.[23] For each classification, we measure 15 ROC scores (three 5-fold splits), which allows us to estimate the variance of the score.

5 Results

The experimental results are summarized in Figure 1. The figure shows that, for each of the 13 classifications, the ROC score of the SDP/SVM method

Table 2: **Kernel weights and ROC scores for the transport facilitation class.** The table shows, for both experiments, the mean weight associated with each kernel, as well as the ROC score resulting from learning the classification using only that kernel. The final row lists the mean ROC score using all kernels.

Kernel	Binary data		Enriched kernels	
	Weight	ROC	Weight	ROC
K_{Pfam}	2.21	.9331	1.58	.9461
K_{Gen}	0.18	.6093	0.21	.6093
K_{Phys}	0.94	.6655	1.01	.6655
K_{TAP}	0.74	.6499	0.49	.6499
K_{Exp}	0.93	.5457	—	.7126
K_{SW}	—	—	1.72	.9180
all	—	.9674	—	.9733

is better than that of the MRF method. Overall, the mean ROC improves from 0.715 to 0.854. The improvement is consistent and statistically significant across all 13 classes. An additional improvement, though not as large, is gained by replacing the expression and Pfam kernels with their enriched versions (see supplement). The most improvement is offered by using the enriched Pfam kernel and replacing the expression kernel with the Smith-Waterman kernel. The resulting mean ROC is 0.870. Again, the improvement occurs in every class, although some class-specific differences are not statistically significant.

Table 2 provides detailed results for a single functional classification, the transport facilitation class. The weight assigned to each kernel indicates the importance that the SDP/SVM procedure assigns to that kernel. The Pfam and Smith-Waterman kernels yield the largest weights, as well as the largest individual ROC scores. Results for the other twelve classifications are similar (see supplement)

6 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 predicting the function of yeast proteins. The resulting SDP/SVM algorithm yields significant improvement relative to an SVM trained from any single data type and relative to a previously proposed graphical model approach for fusing heterogeneous genomic data.

Kernel-based statistical learning methods have a number of general virtues as tools for biological data analysis. First, the kernel framework accommodates non-vector data types such as strings, trees and graphs. Second, kernels provide significant opportunities for the incorporation of specific biological knowledge, as we have seen with the Pfam kernel, and unlabelled data, as in the diffusion and Smith-Waterman kernels. Third, the growing suite of kernel-based data analysis algorithms requires 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 semidefiniteness, and thus the powerful technique of semidefinite programming can be exploited to derive general procedures for combining data of heterogeneous format and origin.

Acknowledgements 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.

1. B. Schölkopf, K. Tsuda and J.-P. Vert. *Support vector machine applications in computational biology.* MIT Press, Cambridge, MA, 2004.
2. L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.
3. G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. In *Proc 19th Int Conf Machine Learning*, pp. 323–330, 2002.
4. M. Deng, T. Chen, and F. Sun. An integrated probabilistic model for functional prediction of proteins. *Proc 7th Int Conf Comp Mol Biol*, pp. 95–103, 2003.
5. H. Ge, Z. Liu, G. Church, and M. Vidal. Correlation between transcriptome and interactome mapping data from *Saccharomyces cerevisiae. Nature Genetics*, 29:482–486, 2001.
6. A. Grigoriev. A relationship between gene expression and protein interactions on the proteome scale: analysis of the bacteriophage T7 and the yeast *S*accharomyces cerevisiae. *Nucl Acids Res*, 29:3513–3519, 2001.
7. R. Mrowka, W. Lieberneister, and D. Holste. Does mapping reveal correlation between gene expression and protein-protein interaction? *Nature Genetics*, 33:15–16, 2003.
8. C. von Mering, R. Krause, B. Snel *et al.* Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417:399–403, 2002.
9. E. M. Marcotte, M. Pellegrini, M. J. Thompson, T. O. Yeates, and D. Eisenberg. A combined algorithm for genome-wide prediction of protein function. *Nature*, 402(6757):83–86, 1999.
10. R. Jansen, N. Lan, J. Qian, and M. Gerstein. Integration of genomic

datasets to predict protein complexes in yeast. *Journal of Structural and Functional Genomics*, 2:71–81, 2002.
11. A. Nakaya, S. Goto, and M. Kanehisa. Extraction of correlated gene clusters by multiple graph comparison. In *Genome Informatics 2001*, pp. 44–53, 2001.
12. A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18:S136–S144, 2002.
13. I. Holmes and W. J. Bruno. Finding regulatory elements using joint likelihoods for sequence and expression profile data. In *Proc Int Sys Mol Biol*, pp. 202–210, 2000.
14. M. Deng, F. Sun, and T. Chen. Assessment of the reliability of protein-protein interactions and protein function prediction. In *Proc Pac Symp Biocomputing*, pp. 140–151, 2003.
15. A. Drawid and M. Gerstein. A Bayesian system integrating expression data with sequence patterns for localizing proteins: comprehensive application to the yeast genome. *J. Mol. Biol.*, 301:1059–1075, 2000.
16. P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy. Gene functional classification from heterogeneous data. In *Proc 5th Int Conf Comp Mol Biol*, pp. 242–248, 2001.
17. B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pp. 144–152, 1992.
18. B. Schölkopf and A. Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.
19. C. Berg, C. J. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions.* Springer, New York, NY, 1984.
20. R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proc Int Conf Machine Learning*, pp. 315–322, 2002.
21. P. T. Spellman, G. Sherlock, M. Q. Zhang *et al.* Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol Biol Cell*, 9:3273–3297, 1998.
22. T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–197, 1981.
23. J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.