Department of Computer Science
George Mason University
Technical Report Series

4400 University Drive MS#4A5
Fairfax, VA 22030-4444 USA
http://cs.gmu.edu/
703-993-1530

Multiple Kernel Learning for Fold Recognition

Technical Report GMU-CS-TR-2010-2

Huzefa Rangwala
rangwala@cs.gmu.edu

Abstract

Fold recognition is a key problem in computational biology that involves classifying protein sharing structural similarities into classes commonly known as "folds". Recently, researchers have developed several efficient kernel based discriminatory methods for fold classification using sequence information. These methods train one-versus-rest binary classifiers using well optimized kernels from different data sources and techniques.

Integrating this vast amount of data in the form of kernel matrices is an interesting and challenging problem. The semidefinite positive property of the various kernel matrices makes it attractive to cast the task of learning an optimal weighting of several kernel matrices as a semi-definite programming optimization problem. We experiment with a previously introduced quadratically constrained quadratic optimization problem for kernel integration using 1*-norm and* 2*-norm support vector machines. We integrate state-of-the-art profile-based direct kernels to learn an optimal kernel matrix* K^**. Our experimental results show a small significant improvement in terms of the classification accuracy across the different fold classes. Our analysis illustrates the strength of two dominating kernels across different fold classes, which suggests the redundant nature of the kernel matrices being combined.*

1 Introduction

In the past few decades, advances in sequencing technology has lead to an exponential increase in the volume of protein sequence data available. However, we are still lacking in the technical ability to characterize the experimental structures of these protein sequences. Remote homology detection and fold recognition play a central role in computational biology, where researchers are relying on computational techniques to classify proteins into functional and structural groups based solely on their amino acid sequences.

Several kernel based methods have been designed and improved for performing remote homology detection and fold recognition [12, 19, 17, 18, 10, 11, 27, 13, 24]. In particular, we [24] used evolutionary information and introduced two classes of well performing direct kernels (window based and local alignment based kernels).

The challenge was to combine the information obtained from several different data descriptors or vast number of carefully designed kernels for this problem. Some of the approaches have resulted in use of voting or jury based methods, generating a consensus from models learned using different kernel matrices[25]. Other approaches build formal graph models and Bayesian inferences for this integration. We approach the problem of integrating kernel matrices as a convex combination of several positive semidefinite matrices as done in a previous study [14] that integrated genomic data from different experimental sources.

In this project we combine two different kernels: (i) the window based, and (ii) local alignment based kernel for performing sequence classification. This method combines the different positive semidefinite kernel matrices with a semi-definite programming technique [15, 23]. These class of problems fall under the general framework of "kernel learning" or "multiple kernel learning". Semidefinite programming is one of the approaches for integrating the different kernel matrices. The multiple kernel learning problem has also been casted as a second order cone programming problem [4], semi-infinite linear program [30, 26] and sparsity exploiting semi-definite programming based approaches [32].

We trained the 1-norm and 2-norm SVM framework for weighting the previously developed profile-based kernel matrices [24] for the fold detection problem. Our results showed that two of the kernel matrices had the highest weights for classification, and the integration using the relaxed semi-definite program yielded a small improvement in performance. This work provides a brief

understanding of how and when multiple kernel learning methods should be used.

2 Problem Formulation and Methods

2.1 Fold Recognition Problem

The remote homology detection problem is defined as the identification of protein pairs sharing the same evolutionary ancestry, but having less than 30% sequence identity. Fold recognition is defined as the identification of protein pairs having similar structural topology and shape but no guarantee on the sequence identity. The two problems can be solved by classification of proteins into a particular class of proteins that are remote homologs or folds.

In this work we simulated fold detection by formulating as a fold classification problem within the context of SCOP's [22] hierarchical classification scheme. In this setting, protein domains within the same superfamily were considered as positive test examples, and protein domains within the same fold but outside the superfamily were considered as positive training examples. Since the positive test and training instances were members of different superfamilies within the same fold, the sequences in the different superfamilies do not have any apparent sequence similarity [22].

2.2 Support Vector Machines

Given a set of training samples $\{(x_i, y_i), \ldots, (x_n, y_n)\}$ where $y_i \in \{+1/-1\}$, the 1-norm soft margin support vector machine [5] (SVM) forms a linear discriminant boundary in a probably higher dimensional feature space $\mathcal{F}$, given by $f(x) = w^T\phi(x) + b$, where $w \in \mathcal{F}$, $b \in \Re$ and $\phi(x)$ represents the input to feature space transformation. The aim is to maximize the distance between the positive and negative classes, allowing for a few misclassification errors, so as to have a better generalization. This results in the following optimization problem:

$$\begin{array}{ll} \min\limits_{w,b,\xi} & w^T w + C\sum\limits_{i=1}^{n}\xi_i \\ \texttt{s.t.} & y_i(w^T\phi(x_i)+b) \geq 1-\xi_i, \\ & \xi_i \geq 0, \qquad \forall i = 1 \ldots n \end{array} \tag{2.1}$$

where C is the regularization parameter, ξ_i are the slack variables. Taking the dual of the problem in (2.1), the result is the following well known quadratic optimization problem [28]:

$$\begin{array}{ll} \max\limits_{\alpha} & 2\alpha^T e - \alpha^T\mathcal{D}(y)K\mathcal{D}(y)\alpha \\ \texttt{s.t.} & C \geq \alpha \geq 0 \\ & \alpha^T y = 0, \end{array} \tag{2.2}$$

where the Lagrangian multipliers α are the solutions of the optimization problem and the weight vector can be expressed as $w = \sum_{i=1}^{n}\alpha_i y_i \phi(x_i)$. $\mathcal{D}(y)$ denotes the diagonal matrix with entries given by $y = (y_1, \ldots, y_n)$. K denotes the kernel matrix or function and it defines the notion of "similarity" between pairs of training instances in their embedded feature space i.e $K(x_i, x_j) = \langle\phi(x_i), \phi(x_j)\rangle$ where $\langle, \rangle$ denotes the inner product. The resulting matrix K is known as the kernel matrix and satisfies Mercer kernel properties including semidefinite positiveness ($K \succeq 0$). Given this kernel matrix K, the learning problem in SVM is a quadratic optimization procedure dependent on the kernel matrix.

2.3 Profile-based Kernel Functions

The inputs to our classification algorithm are the various proteins and their profiles. A protein sequence X of length n is represented by a sequence of characters $X = \langle a_1, a_2, \ldots, a_n\rangle$ such that each character corresponds to one of the 20 standard amino acids. The profile of a protein X is derived by computing a multiple sequence alignment of X with a set of sequences $\{Y_1, \ldots, Y_m\}$ that have a statistically significant sequence similarity with X (i.e., they are sequence homologs).

We obtain the profiles using PSI-BLAST [2] as it combines both steps, is very fast, and has been shown to produce reasonably good results. However, the profile-based kernels can be used with other methods of constructing sequence profiles as well. For every sequence position, the profile captures the evolutionary information derived from the set of homologous sequences.

Many different schemes have been developed for determining the similarity between pairs of profiles that combine information from the original sequence, position-specific scoring matrix, or position-specific target and/or effective frequencies [21, 34, 20]. We use a profile-profile scoring scheme [24] that is derived from PICASSO [9, 21], found to have superior performance in building individual remote homology detection and fold recognition models.

2.3.1 Window-based Kernels In our previous study [24] we have developed a class of profile-based kernel functions that determines the similarity between a pair of sequences by combining the ungapped alignment scores of certain fixed length subsequences (referred to as $w$mers). Given a sequence X of length n and a user-supplied parameter w, the $w$mer at position i of X ($w < i \leq n - w$) is defined to be the $(2w+1)$-length subsequence of X centered at position i. That is, the $w$mer contains x_i, the w amino acids before, and the w amino acids after x_i. We will denote this subsequence as $w\text{mer}_X(i)$. Note that $w$mers are nothing more than the fixed-length windows used extensively in secondary structure prediction and in capturing local sequence information around a particular sequence position.

We developed three types of window-based kernel functions [24]: (i) The AF-PSSM kernel computes the similarity between a pair of sequences X and Y by adding-up the alignment scores of all possible $w$mers between X and Y that have a positive ungapped alignment score. (ii) The BF-PSSM kernel uses a scheme that computes the similarity between a pair of sequences based on a subset of the $w$mers used in the AF-PSSM kernel. Specifically, the BF-PSSM kernel selects $w$mers such that each position of X and each position of Y is present in at most one $w\mathrm{mer}$-pair and the sum of the $w\mathrm{mer}$ scores of the selected pairs are maximized. (iii) The BV-PSSM kernel is derived from the BF-PSSM kernel but operates with variable width $w$mers and selects for each position of X and each position of Y the variable length $w\mathrm{mer}$ with a maximum score.

In this paper we focus on combining the three window-based kernels with the local alignment kernels (discussed below) for improving the accuracy of the fold recognition problem.

2.3.2 Local Alignment Kernels The second class of profile-based kernels [24] that we use computes the similarity between a pair of sequences X and Y by finding an optimal alignment between them that optimizes a particular scoring function. We use the Smith-Waterman alignments [29] to derive our profile-based local alignment kernel, referred to as SW-PSSM.

Given two sequences X and Y of lengths n and m, respectively, the SW-PSSM kernel computes their similarity as the score of the optimal local alignment in which the similarity between two sequence positions is determined using the profile-to-profile scoring scheme of PICASSO [9], and a position independent affine gap model.

Any function can be used as a kernel as long as for any number n and any possible set of distinct sequences $\{X_1, \ldots, X_n\}$, the $n \times n$ Gram matrix defined by $K_{i,j} = \mathcal{K}(X_i, X_j)$ is symmetric positive semidefinite. These functions are said to satisfy Mercer's conditions and are called Mercer kernels, or simply valid kernels. Both the window-based and local alignment kernels are not positive semidefinite. To overcome this problem we used the approach described in [27] to convert a symmetric function defined on the training set instances into positive definite by adding to the diagonal of the training Gram matrix a sufficiently large non-negative constant. [24]

2.4 Multiple Kernel Learning

The window-based and local-alignment based kernel matrices [24] have proven to be well optimized matrices for discrimination between the several fold classes. In this study, we integrate these different positive semidefinite kernel matrices using a convex combination technique. We use this optimization method for weighting a set of profile-based direct string kernel matrices for performing fold recognition.

Different kernels correspond to a different feature space embedding of data and capture a different similarity metric. These kernel matrices being positive semidefinite allow us to cast the problem of integrating different kernel matrices as a semi-definite programming [33] (SDP) optimization problem [16, 14, 15]. Given m kernel matrices, we would like to learn a linear weighting of the different kernel matrices, resulting in the optimal kernel matrix given by

$$K^* = \sum_{i=1}^{m} \mu_i K_i, \tag{2.3}$$

where μ_i denotes the weights learned for the different kernels and K^* denotes the optimal kernel matrix.

The optimal support values in Equation 2.2 are highly dependent on the choice of kernel matrices (which is often a black art). In the combination setting we can formulate an optimization problem by parameterizing the kernel matrix K. This is done by minimizing with respect to μ_i which gives the following optimization problem [15, 14]:

$$\begin{array}{lll} \min\limits_{\mu \in \Re^m, K^* \succeq 0} \max\limits_{\alpha} & 2\alpha^T e - \alpha^T \mathcal{D}(y) K^* \mathcal{D}(y) \alpha & \\ \mathtt{s.t.} & C \geq \alpha \geq 0, & \\ & \alpha^T y = 0, & \\ & trace(K^*) = c, & \\ & K^* = \sum_{i=1}^{m} \mu_i K_i, & \end{array} \tag{2.4}$$

where c is a constant. Equation 2.4 is a minimum maximum problem and can be rewritten as:

$$\begin{array}{lll} \min\limits_{\mu \in \Re^m, K^* \succeq 0, t} & t & \\ \mathtt{s.t.} & t \geq \max\limits_{\alpha} 2\alpha^T e - \alpha^T \mathcal{D}(y) K^* \mathcal{D}(y) \alpha, & \\ & C \geq \alpha \geq 0, & \\ & \alpha^T y = 0, & \\ & trace(K^*) = c, & \\ & K^* = \sum_{i=1}^{m} \mu_i K_i. & \end{array} \tag{2.5}$$

The Lagrangian dual of the problem to find optimal values of α and μ is given by (More details in the work by Lancreit et. al [15]):

$$\begin{array}{lll} \min\limits_{\mu \in \Re^m, K, t, \lambda, \nu, \delta} & t & \\ \mathtt{s.t.} & trace(K^*) = c & \\ & K^* = \sum_{i=1}^{m} \mu_i K_i \succeq 0 & \\ & \begin{pmatrix} \mathcal{D}(y) K \mathcal{D}(y) & e + \nu - \delta + \lambda y \\ (e + \nu - \delta + \lambda y)^T & t - 2C\delta^T e \end{pmatrix} \succeq 0 & \\ & \nu \geq 0 & \\ & \delta \geq 0 & \end{array} \tag{2.6}$$

This optimization is a well known convex optimization problem known as a semidefinite program (SDP) which can be solved using standard optimization tools [33]. These algorithms are limited by a

worse-case run time complexity of $O(n^{4.5})$ [14]. This method of integrating kernel matrices though novel, has resulted in researchers using several other forms of convex combination like second-order cone programming (SOCP) [4, 31], semi-infinite linear programming (SILP) [30, 26], relaxations leading to quadratically constrained quadratic optimization problems (QCQP) [15] and heuristic approaches [32].

We use the restriction $\mu \geq 0$, which results in a QCQP [15], leading to an efficient run time complexity, $O(n^3)$. This constraint also improves numerical stability, as well as the generalization performance [14, 15] of learned classifiers. The additional constraint $\mu \geq 0$ (steps not shown here) results in the following QCQP optimization problem [15]:

$$\begin{array}{ll} \max_{\alpha,t} & 2\alpha^T e - ct \\ \text{subject to} & t \geq \frac{1}{r_i}\alpha^T \mathcal{D}(y) K_i \mathcal{D}(y)\alpha \quad i = 1, \ldots, m \\ & \alpha^T y = 0 \\ & C \geq \alpha \geq 0 \end{array} \tag{2.7}$$

We also studied the 2-norm SVM formulation where the objective function in Equation 2.1 was modified to include a quadratic term for the variable ξ_i as

$$\min_{w,b,\xi} w^T w + C \sum_{i=1}^{n} \xi_i^2. \tag{2.8}$$

Similar to the 1-norm SVM, the Lagrangian dual optimization problem for the 2-norm SVM leads to a QCQP problem. This can be expressed as [15]:

$$\begin{array}{ll} \max\limits_{\alpha,t} & 2\alpha^T e - \frac{1}{C}\alpha^t \alpha - ct \\ \text{subject to} & t \geq \frac{1}{r_i}\alpha^T \mathcal{D}(y) K_i \mathcal{D}(y)\alpha \quad i = 1, \ldots, m \\ & \alpha^T y = 0 \\ & \alpha \geq 0 \end{array} \tag{2.9}$$

3 Numerical Experiments

We evaluated the performance of the multiple kernel learning methods using the 1-norm and 2-norm learning framework for the fold recognition problem. Specifically, we evaluated the performance of using the QCQP optimization framework [14] for integrating the three window-based and one local alignment kernels. We used the MOSEK [3] toolkit to run experiments for the QCQP problem[1]

3.1 Datasets

The dataset [25] was derived from the SCOP 1.67 [22] by selecting domains having less than 40% sequence identity. The resulting dataset consisted of 1651 protein domain sequences within 27 fold classes, split to have a training and test set size of 1307 and 344 respectively. In this setting, protein domains within the same superfamily were considered to be as positive test examples, and protein domains within the same fold but outside the superfamily were considered as positive training examples. For example, we can visually represent the setup for the fold recognition problem in terms of the test and training sets for a particular fold class (fold a.2) in Figure 1.

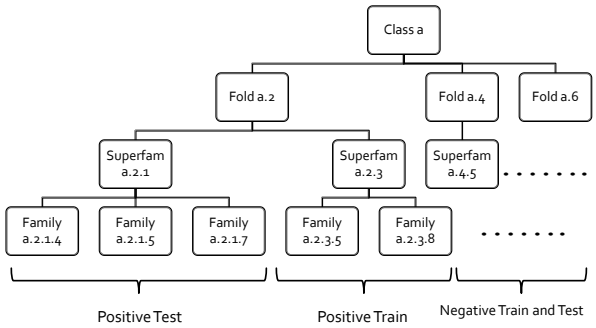


Figure 1: SCOP hierarchy tree showing the training and test instances setup for the fold recognition problem.

3.2 Kernel Functions

For purposes of the experiment we used profile-based kernel functions [24] to derive our individual kernel matrices. These kernel functions compute the similarity between a pair of sequences, using conservation information. We used the three variations of the window-based kernels (all fixed (AF) $wmer$ kernel, best fixed $wmer$ kernel (BF), best variable $wmer$ kernel (BV)) along with the smith-waterman (SW) kernel in our effort to optimally combine them.[2]

We also ran experiments using kernels derived from position independent, BLOSUM62 matrices (GSM). We also combined the position-specific and position-independent matrices to study the classification results.

3.3 Results and Discussion

We trained 27 one-versus-rest binary classifiers for fold recognition for the different kernels and their combinations. Specifically, we evaluate the QCQP optimization for integrating the different kernel matrices. The accuracy of these classifiers were evaluated using the area under the receiver operating characteristic curve (ROC), which measures the true positive rate versus the false positive rate. Specifically we computed ROC_{50} (ROC up to the first 50 false positives) measure across the 27 fold classes.

We ran a series of experiments using different combinations of kernel matrices, and the two SVM formu-

[1]The code for the 1-norm SVM was provided by Gert Lanckriet. We modified it to work within the fold recognition framework and setup the 2-norm SVM.

[2]The codes to compute these kernels are freely available at http://bioinfo.cs.umn.edu/supplements/remote-homology/.

lations. Table 1 shows the average ROC_{50} results obtained for the different experiments. To set up a base line, we computed the classification accuracy achieved by the individual matrices (i.e the AF-PSSM, BF-PSSM, BV-PSSM, SW-PSSM). We used optimized parameters for window length, gap opening, extension and shift parameters [24]. The experiment $F1$ in Table 1 denotes the classification performance for an optimal combination of the three window-based and smith-waterman based kernel matrices. In case of the 1-norm SVM framework the accuracy results improve to the third decimal place over the BV-PSSM kernel. The optimal weighting of the kernel matrices did not lead to a boost in the performance, which can also be noticed in Figure 3 where we show the ROC_{50} results for the different fold classes. The results achieved by the $F1$ combination scheme are fairly similar to the SW-PSSM and the BV-PSSM kernel matrices for the 1-norm SVM classifier.

On further analysis, the weights computed across the different kernel matrices as seen in Figure 2 shows that the combination classifier assigns higher weight to the SW-PSSM and BV-PSSM kernels, which have a better classification accuracy when compared to the BF-PSSM and AF-PSSM kernels. The fact that BV-PSSM dominates can be explained by the fact that it is more relaxed than the other two window-based kernels and the kernel learning problem setup probably handles the redundant information between the window based kernels.

We also performed similar experiments using the position independent kernel matrices (GSM). As seen previously [24], the individual GSM-based kernels were poorer in comparison to the individual PSSM-based kernels. However the optimized kernel combination (F1) learned using the optimization framework was worse for both the 1-norm and 2-norm SVM (See Table 1 and Figure 4).

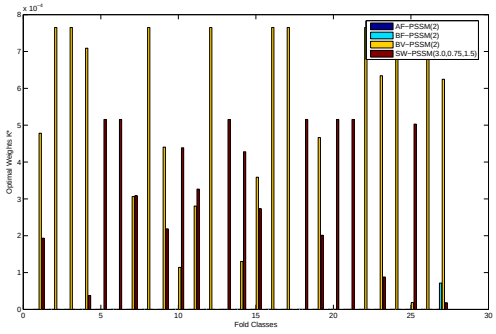
Finally, we combined all the eight kernel matrices (PSSM and GSM), shown as F2 in Table 1. This combination achieved the average ROC_{50} results of 0.351 in comparison to the average ROC_{50} score of 0.385 for the $F1$ combination scheme for the 1-norm SVM. The decrease in classifier performance was probably because of the poor performance of the GSM kernels. The input kernels were fairly similar to each other since they were derived from the same source of data. Comparing the 2-norm SVM with the 1-norm SVM, we noticed that the 2-norm SVM always showed poorer classification performance.

Table 1: Average ROC_{50} results across the 27 fold classes.

	PSSM		GSM	
Schemes	1-norm	2-norm	1-norm	2-norm
AF(2)	0.322	0.306	0.160	0.154
BF(2)	0.341	0.311	0.310	0.291
BV(2)	0.384	0.337	0.332	0.250
SW	0.380	0.356	0.261	0.249
F1	0.385	0.339	0.242	0.240
F2	0.351	0.315	-	-

PSSM and GSM denote the use of position specific scoring matrices and the global scoring matrices for the base kernels, respectively. F1 is the optimal linear weighting of AF, BF, BV and SW kernel matrices. F2 is the optimal linear weighting of eight matrices (i.e combining both sets of PSSM and GSM based matrices). AF, BF, and BV kernels use a window having $wmer = 2$. The gap-opening, extension and shift parameters are (3.0, 0.75, 1.5) and (5.0, 1.0, 0.0) for the SW-PSSM and SW-GSM kernels respectively.

Figure 2: Learned Optimal Weights for different binary classifiers (1-norm soft margin SVM)



4 Conclusion and Future Work

In this study we combined the different optimized kernel matrices using QCQP for performing discriminative learning using the 1-norm and 2-norm support vector machines.

We report a slight performance improvement when using the 1-norm SVM formulation to integrate the kernel matrices. There are a wide range of other kernel learning methods which need to be studied and analyzed in the near future. It would be interesting to test the performance of kernel integration methods using SDP [15], SCOP [4, 31] and SILP [30, 26]. We would also like to use our two-layered learning framework [25] to integrate the predictions of individual kernels.

We also realize that the different kernel matrices used in this study were of similar nature to each other. We may be adding adding redundant information while learning the optimal kernel matrix as a weighted combination of the carefully designed individuals. We would like to explore the use of kernel matrices derived from different databases or heterogeneous sources. We would also like to integrate kernel matrices derived from protein local structure prediction i.e backbone and secondary

Figure 3: Performance comparison when combining position specific based (PSSM) kernel matrices. The graph shows the number of fold classes below certain ROC_{50} values for different kernel matrices and their optimal linear combination

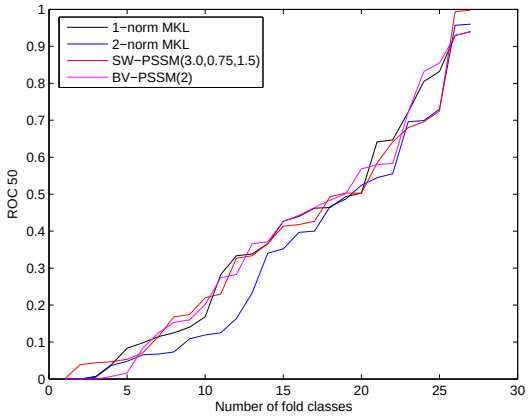
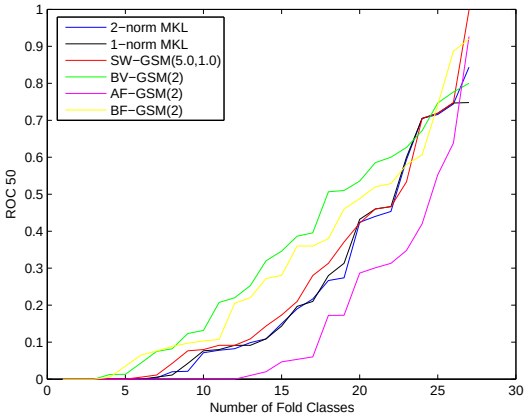


Figure 4: Performance comparison when combining position independent based (GSM) kernel matrices. The graph shows the number of fold classes below certain ROC_{50} values for different kernel matrices and their optimal linear combination



structure information. Another future study, is to explore the idea of learning a weight combination of kernel matrices for performing multi-class classification directly [6, 7, 8, 35, 1], rather than building separate one-versus-rest binary classifiers. Learning a kernel matrix for the multi-class problem would involve a larger number of constraints, but would be efficient to learn a single optimal K^*, kernel matrix rather than individual kernel matrices for each of the classification models (one per fold) as done in this study.

Acknowledgments

This project is funded by NSF project IIS 0905117 and a bioengineering startup grant provided by the Volgenau School of Information Technology and Engineering at George Mason University to Huzefa Rangwala.

References

[1] F. Aiolli and A. Sperduti. Multiclass classification with multi-prototype support vector machines. *Journal of Machine Learning Research*, 6:817–850, 2005.

[2] S. F. Altschul, L. T. Madden, A. A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–402, 1997.

[3] E. D. Anderson and A. D. Anderson. The mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In *High Performance Optimization*, pages 197–232. Kluwer Academic Publishers, 2000.

[4] F. Bach, G. R. G. Lanckriet, and M . I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. *Proceedings of the 2004 International Conference on Machine Learning*, 2004.

[5] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *Computational Learning Theory*, pages 144–152, 1992.

[6] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[7] Y. Guermeur. A simple unifying theory of multiclass support vector machines. Technical Report RR-4669, INRIA, 2002.

[8] Y. Guermeur, A. Elisseeff, and D. Zelus. A comparative study of multi-class support vector machines in the unifying framework of large margin classifiers. *Applied Stochastic Models in Business and Industry*, 21:199–214, 2005.

[9] A. Heger and L. Holm. Picasso:generating a covering set of protein family profiles. *Bioinformatics*, 17(3):272–279, 2001.

[10] Y. Hou, W. Hsu, M. L. Lee, and C. Bystroff. Efficient remote homology detection using local structure. *Bioinformatics*, 19(17):2294–2301, 2003.

[11] Y. Hou, W. Hsu, M. L. Lee, and C. Bystroff. Remote homology detection using local sequence-structure correlations. *Proteins:Structure,Function and Bioinformatics*, 57:518–530, 2004.

[12] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1):95–114, 2000.

[13] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif ex-

traction. *Computational Systems Bioinformatics*, pages 152–160, 2004.

[14] G. R. G. Lanckriet, T. D. Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.

[15] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[16] G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-based data fusion and its application to protein function prediction in yeast. In *Proceedings of the 2004 Pacific Symposium on Biocomputing*, 2004.

[17] C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for svm protein classification. *Proceedings of the Pacific Symposium on Biocomputing*, pages 564–575, 2002.

[18] C. Leslie, E. Eskin, W. S. Noble, and J. Weston. Mismatch string kernels for svm protein classification. *Advances in Neural Information Processing Systems*, 20(4):467–476, 2003.

[19] L. Liao and W. S. Noble. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *Proc. of the Intl. Conf. on Research in Computational Molecular Biology*, pages 225–232, 2002.

[20] M. Marti-Renom, M. Madhusudhan, and A. Sali. Alignment of protein sequences by their profiles. *Protein Science*, 13:1071–1087, 2004.

[21] D. Mittelman, R. Sadreyev, and N. Grishin. Probabilistic scoring measures for profile-profile comparison yield more accurate short seed alignments. *Bioinformatics*, 19(12):1531–1539, 2003.

[22] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540, 1995.

[23] C. S. Ong, A. J. Smola, and R. C. Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6:1043–1071, 2005.

[24] H. Rangwala and G. Karypis. Profile based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, 21(23):4239–4247, 2005.

[25] Huzefa Rangwala and George Karypis. Building multiclass classifiers for remote homology detection and fold recognition. *BMC Bioinformatics*, 7:455, 2006.

[26] G. Ratsch, S. Sonnenburg, and C. Schafer. Learning interpretable svms for biological sequence classification. *BMC Bioinformatics*, 7(S9), 2006.

[27] H. Saigo, J. P. Vert, N. Ueda, and T. Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689, 2004.

[28] B. Scholkopf and A. Smola. Learning with kernels. 2002.

[29] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[30] S. Sonnenburg, G. Ratsch, and C. Schafer. A general and efficient multiple kernel learning algorithm. *Proceedings of the 2005 Neural Information Processing Systems*, 2005.

[31] I. W. Tsang and J. T . Kwok. Efficient hyperkernel learning using second-order cone programming. *IEEE Transactions on Neural Networks*, 17(1):48–58, 2006.

[32] K. Tsuda, H. Shin, and B. Scholkopf. Fast protein classification with mulitple networks. *Bioinformatics*, 21:59–65, 2005.

[33] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[34] G. Wang and R. L. Dunbrack JR. Scoring profile-to-profile sequence alignments. *Protein Science*, 13:1612–1626, 2004.

[35] J. Weston and C. Watkins. Mulit-class support vector machines. Technical Report CSD-TR-89-04, Department of Computer Science, Royal Holloway, University of London, 1998.