Bioinformatics Advance Access published March 31, 2008

Probabilistic multi-class multi-kernel learning: On protein fold recognition and remote homology detection

Theodoros Damoulas [1]* and Mark A. Girolami [1]

[1]Department of Computing Science, University of Glasgow, S. A. W. Building, G12 8QQ, UK.

Associate Editor: Prof. Thomas Lengauer

ABSTRACT

Motivation: The problems of protein fold recognition and remote homology detection have recently attracted a great deal of interest as they represent challenging multi-feature multi-class problems for which modern pattern recognition methods achieve only modest levels of performance. As with many pattern recognition problems, there are multiple feature spaces or groups of attributes available, such as global characteristics like the amino-acid composition (C), predicted secondary structure (S), hydrophobicity (H), van der Waals volume (V), polarity (P), polarizability (Z), as well as attributes derived from local sequence alignment such as the Smith-Waterman scores. This raises the need for a classification method that is able to assess the contribution of these potentially heterogeneous object descriptors while utilizing such information to improve predictive performance. To that end, we offer a single multi-class kernel machine that informatively combines the available feature groups and, as is demonstrated in this paper, is able to provide the state-of-the-art in performance accuracy on the fold recognition problem. Furthermore, the proposed approach provides some insight by assessing the significance of recently introduced protein features and string kernels. The proposed method is well-founded within a Bayesian hierarchical framework and a variational Bayes approximation is derived which allows for efficient CPU processing times.

Results: The best performance which we report on the SCOP PDB-40D benchmark data-set is a 70% accuracy by combining all the available feature groups from global protein characteristics but also including sequence-alignment features. We offer an 8% improvement on the best reported performance that combines binary SVM classifiers while at the same time reducing computational costs and assessing the predictive power of the various available features. Furthermore, we examine the performance of our methodology on the SCOP 1.53 benchmark data-set that simulates remote homology detection and examine the combination of various state-of-the-art string kernels that have recently been proposed.

Contact: theo@dcs.gla.ac.uk

1 INTRODUCTION

Much effort has been directed to the prediction of the three-dimensional structures of proteins for which no experimental structures are available (Baker and Sali, 2001). Where there is sequence similarity to proteins of known structure, a comparative matching procedure is often adopted. However, where no such sequence similarity exists, the prediction problem is formidable, not least because the overall structure may be unlike that of any protein, the structure of which has been determined.

In this context, one approach, known as the *taxonomic* approach (Ding and Dubchak, 2001; Shen and Chou, 2006), has been to divide the problem of determining the overall three-dimensional structure into that of determining its 'fold'. The term 'fold' is used to denote a particular arrangement of a specific number of secondary structure components (usually alpha-helices and beta-strands) that is the basis of the overall structure of several different proteins which may have little or no amino acid sequence similarity. The appearances of some of these arrangements have given rise to names like 'barrel', 'bundle', 'sandwich' and 'propeller', although these tend to encompass several more specific folds e.g. the TIM beta/alpha barrel and the 5-bladed beta-propeller. Hence, protein fold prediction can be seen as a challenging multiclass recognition problem where proteins are classified into folds based on their characteristics and available measurements.

Past work on the problem of predicting protein folds has employed artificial neural networks (ANNs), support vector machines (SVMs), Bayesian networks, Hidden Markov Models and k-nn classifiers (Chou and Zhang, 1995; Dubchak *et al.*, 1995; Jaakkola *et al.*, 1999; Raval *et al.*, 2002) with varying success. In Ding and Dubchak (2001) an extensive study on a publicly available data-set, consisting of 27 SCOP folds (Lo Conte *et al.*, 2000; Andreeva *et al.*, 2004), was conducted exploring the use of various multiclass adaptations of the well-known binary SVM classifier methodology. In that work, the best methodology for combining binary SVMs was identified for the particular problem giving an accuracy of 56%, and furthermore, via an extensive experimental procedure the most *predictive* protein characteristics were selected from the initial group considered. These were found to be the amino-acid composition (C), the secondary structure (S) and the hydrophobicity (H).

Recently, Shen and Chou (2006) proposed two modifications to the method of Ding and Dubchak (2001) that raised the best performance accuracy from 56% to 62.1%. Firstly, they proposed a somewhat *ad-hoc* ensemble learning approach where multiclass k-nn classifiers individually trained on each feature space (such as C or S) were later combined and secondly, they proposed the use of 4 additional feature groups to replace the amino-acid composition. These were pseudo-amino acid compositions (Chou, 2005) designed to capture sequence-order effects by using a correlation function

*to whom correspondence should be addressed

between hydrophobicity and hydrophilicity in different intervals of the protein sequence.

In the present work, we concentrate on the same benchmark dataset of Ding and Dubchak (2001) with the extra groups of features proposed by Shen and Chou (2006) and also including sequence-alignment[1] features via a *pairwise* kernel (Liao and Noble, 2003), which essentially describes the sequence based similarity of the proteins. We offer a single multi-class kernel machine able to operate on all of these groups of features simultaneously and instructively combine them. This offers a new and efficient way of incorporating multiple feature characteristics of the proteins without an increase in the number of required classifiers. In addition, we assess the importance and predictive power of the pseudo-amino acid compositions proposed by Shen and Chou (2006) together with all the other available characteristics and gain insight on the protein fold recognition problem.

Furthermore, we demonstrate the generality of our methodology in a practical setting by addressing the remote homology problem on the SCOP 1.53 data-set as previously studied and described by a large number of works, see Liao and Noble (2003); Leslie *et al.* (2004); Saigo *et al.* (2004); Lingner and Meinicke (2004) and references within, where a variety of *string* kernels in conjunction with a discriminative SVM methodology have been proposed. Following our approach we select four of these state-of-the-art string kernels and combine them into an overall composite kernel where the multinomial probit kernel machine operates.

Related methodologies on kernel machines and multiple kernel learning (MKL) includes the work by Lanckriet *et al.* (2004a,b); Lewis *et al.* (2006a,b); Sonnenburg *et al.* (2006) and references within, where semidefinite programming (SDP) or semi-infinite linear programming techniques are employed in order to minimize a loss function with respect to the kernel combination. These approaches build upon the support vector machine (SVM) methodology and formulate the kernel combination problem as a further optimization procedure. The (SDP) approach suffers from large requirements in memory and CPU time in the order $O(S^3N^2)$, where S is the number of sources and N is the number of covariates. These methods also carry the inherent drawback of SVM methodologies, namely their problematic scaling for multiclass problems as they are based on binary classifiers by nature.

An explicit multiclass classifier within the Gaussian Process methodology was introduced recently by Girolami and Zhong (2007) which enables data integration by combining the covariance functions instead of kernels. Their methodology has the drawback of employing a first order approximation for the inverse of the covariance functions. Finally, recent work by Melvin *et al.* (2007) in the context of protein classification employed adaptive codes to handle the multiclass prediction problem. However their methodology is based on learning a weighting of binary classifiers which, we argue, is not an efficient strategy especially for multiple feature space problems such as the one considered in this study.

2 APPROACH

The approach adopted is based on the motivation to reduce the number of classifiers needed for such challenging multi-class recognition problems where multiple feature sets are available, while improving performance. Combining binary classifiers as in the work by Ding and Dubchak (2001) increases heavily the computational resources needed since, e.g for the best performing all-vs-all method, we need to deploy $S \times \frac{C(C-1)}{2} = 2106$ classifiers, where S is the number of feature spaces or *sources* (only 6 in their work) and C the number of classes.

Furthermore, even when employing multiclass classifiers in an ensemble learning framework such as the one proposed by Shen and Chou (2006), we still need as many classifiers as there are available feature spaces. Considering the nature of the protein fold prediction problem, where the fold type of a protein can depend on a large number of protein characteristics and also noting that even in the *taxonomic* approach the number of fold types already approaches the thousand boundary, it is straightforward to see the need for a methodological framework that can cope with a large number of classes and can incorporate as many as there are available feature spaces while assesing their informational content.

The proposed approach, as can be seen from Figure 1, is based on the ability to embed each object description via the kernel trick (Shawe-Taylor and Cristianini, 2004) into a kernel space (Hilbert space). This produces a similarity measure between proteins in every feature space and then, having a common measure, we can combine informatively these similarities onto a composite kernel space. Hence now, a single multiclass kernel machine can operate on that composite space effectively "disregarding" the number of feature spaces used. Inference by Bayes theorem on our hierarchical multiclass model enables us to learn the significance of each source/feature space and their predictive power by the corresponding kernel weights $\boldsymbol{\beta}$, to learn the regressors and the kernel parameters without resorting to *ad-hoc* ensemble learning, combination of binary classifiers or parameter tuning.

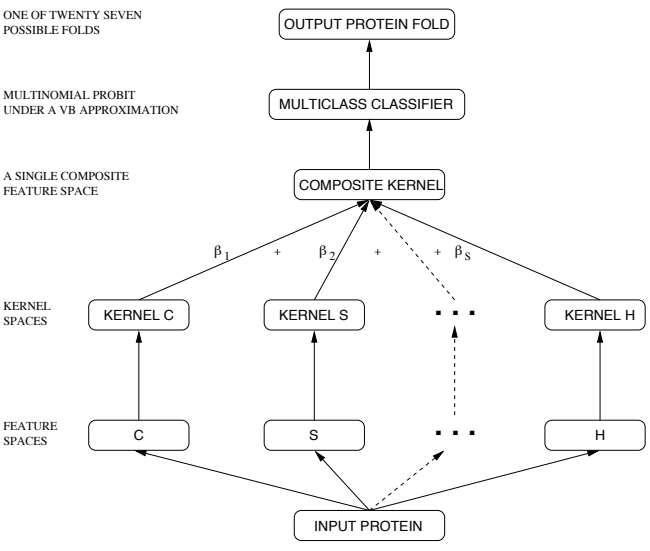

Fig. 1. Diagrammatic representation of the kernel combination methodology (VBKC) for protein fold prediction. The original feature spaces are first embedded into kernels (Hilbert spaces) and then combined into a composite kernel where the multiclass kernel machine operates on.

[1] Despite the apparent low homology data-set

3 MATERIALS AND METHODS

A. *Fold recognition*: The original dataset from Ding and Dubchak (2001) (based on SCOP PDB-40D) consists of 313 proteins for training and 385 proteins for testing with less than 35% sequence identity between any two proteins in the train and the test set. Furthermore, the extensions proposed by Shen and Chou (2006) exclude 4 proteins from the original dataset, namely proteins 2SCMC and 2GPS from the training set plus 2YHX_1 and 2YHX_2 from the test set, due to lack of sequence records.

The 27 SCOP fold types (Dubchak *et al.*, 1995) together with the original feature spaces in Ding and Dubchak (2001), the 4 proposed by Shen and Chou (2006) which describe pseudo-amino acid compositions (PseAA) estimated on different intervals of the protein sequence, and the 2 local alignment Smith-Waterman (SW) based feature spaces, with different scoring matrices, are described in Tables 1 and 2 in the *Online Supplementary Materials* (OSM hereafter).

B. *Remote homology detection* (RHD): The SCOP 1.53 benchmark dataset[2] as described in Liao and Noble (2003) is employed to simulate the RHD problem. It consists of 4,352 proteins belonging to one of 54 families and the positive training is performed on low-homologs while the positive testing on members of the same family. We consider four state-of-the-art string kernels, namely a *local alignment* (LA) kernel (Saigo *et al.*, 2004), a *mismatch* (MM) kernel (Leslie *et al.*, 2004), an *oligomer* kernel (Mono) (Lingner and Meinicke, 2004) and a *pairwise* (PW) kernel (Liao and Noble, 2003), taking the best performing case from each string kernel category as a separate informational source. We follow the above past works within the kernel machine paradigm by adding a class-dependent regularization parameter to the diagonal of the kernels to improve performance on this highly imbalanced problem.

C. *Methodology*: Consider now S feature spaces or sources of information; From each one we have input variables $\mathbf{x}_n^s$ as object descriptors such as strings or D^s- dimensional vectors for $s = 1,\ldots,S$ and corresponding multinomial target variables $t_n \in \{1,\ldots,C\}$ for $n = 1,\ldots,N$ where N is the number of observations and C the number of classes. By applying the *kernel trick* on the individual feature spaces created by the S sources we can define the $N \times N$ composite kernel as

$$\mathbf{K}^{\boldsymbol{\beta\Theta}} = \sum_{s=1}^{S} \beta_s \mathbf{K}^{s\boldsymbol{\theta}_s}$$

with each element of the matrix given as

$$K^{\boldsymbol{\beta\Theta}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{s=1}^{S} \beta_s K^{s\boldsymbol{\theta}_s}(\mathbf{x}_i^s, \mathbf{x}_j^s)$$

where $\boldsymbol{\beta}$ is an $S \times 1$ column vector, indicating each kernel's contribution and significance, and $\boldsymbol{\Theta}$ is an $S \times D^s$ matrix, describing the D^s-dimensional kernel parameters $\boldsymbol{\theta}_s$ of all the base kernels $\mathbf{K}^s$, which intuitively corresponds to the level of smoothing within each kernel. Now as we can see the mean composite kernel is a weighted summation of the base kernels, where each one describes a similarity measure between proteins based on specific features, with β_s as the corresponding weight for each one.

Following the standard approach for the multinomial probit by Albert and Chib (1993) we introduce auxiliary variables $\mathbf{Y} \in \mathcal{R}^{C\times N}$ and define the relationship between the auxiliary variable y_{cn} and the target variable t_n as

$$t_n = i \text{ if } y_{in} > y_{jn}\ \forall\, j \neq i \tag{1}$$

Now, the model response regressing on the variable y_{cn} with model parameters $\mathbf{W} \in \mathcal{R}^{C\times N}$, where w_{cn} is the weight with which data point n "votes" for class c, and assuming a standardised normal noise model as in Albert and Chib (1993) and Girolami and Rogers (2006) is given by

$$y_{cn}|\mathbf{w}_c, \mathbf{k}_n^{\boldsymbol{\beta\Theta}} \sim \mathcal{N}_{y_{cn}}\left(\mathbf{w}_c \mathbf{k}_n^{\boldsymbol{\beta\Theta}}, 1\right) \tag{2}$$

where $\mathcal{N}_x(m, v)$ denotes the normal distribution of x with mean m and variance v, $\mathbf{W}$ and $\mathbf{Y}$ are $C \times N$ matrices, $\mathbf{w}_c$ is a $1 \times N$ row vector and $\mathbf{k}_n^{\boldsymbol{\beta\Theta}}$ is an $N \times 1$ column vector from the n^{th} column of the composite kernel $\mathbf{K}^{\boldsymbol{\beta\Theta}}$. Note that $\mathbf{w}_c \mathbf{k}_n^{\boldsymbol{\beta\Theta}}$ is similar for any two data points n, n' that are similar in feature space. Hence now, the likelihood, can be expressed as the following by simply marginalizing over the auxiliary variable $\mathbf{y}_n$ and making use of relations 1 and 2:

$$\begin{aligned} P\left(t_n = i|\mathbf{W}, \mathbf{k}_n^{\boldsymbol{\beta\Theta}}\right) &= \int P(t_n = i|\mathbf{y}_n)\, P\left(\mathbf{y}_n|\mathbf{W}, \mathbf{k}_n^{\boldsymbol{\beta\Theta}}\right) d\mathbf{y}_n \\ &= \int \delta(y_{in} > y_{jn}\ \forall\, j \neq i) \prod_{c=1}^{C} \mathcal{N}_{y_{cn}}\left(\mathbf{w}_c \mathbf{k}_n^{\boldsymbol{\beta\Theta}}, 1\right) d\mathbf{y}_n \\ &= \mathcal{E}_{p(u)}\left\{\prod_{j\neq i} \Phi\left(u + (\mathbf{w}_i - \mathbf{w}_j)\, \mathbf{k}_n^{\boldsymbol{\beta\Theta}}\right)\right\} \end{aligned} \tag{3}$$

where the expectation $\mathcal{E}$ is taken with respect to the standardised normal distribution $p(u) = \mathcal{N}(0, 1)$. Hence, we can easily calculate the likelihood by averaging the quantity inside the expectation for a sufficient number of random samples of u.

The proposed graphical model as depicted in Figure 2 is completed by considering prior distributions on the model variables and, following a hierarchical approach, hyper-prior distributions on the parameters of the first. We place a product of zero mean Gaussian distributions on the regressors $\mathbf{W} \sim \prod_{c=1}^{C}\prod_{n=1}^{N} \mathcal{N}_{w_{cn}}(0, \zeta_{cn})$ with variance ζ_{cn} (described by the variable $\mathbf{Z}$ in the graph) and a gamma distribution on each scale with hyper-hyper-parameters τ, υ, reflecting our lack of prior knowledge and taking advantage of the conjugacy of these distributions.

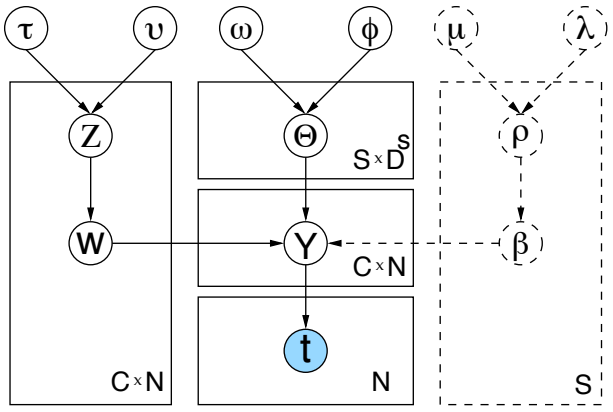

Fig. 2. Plates diagram of the model's random variables. The S plate indicated by dashed lines is omitted when a fixed summation of base kernels is employed instead of the general mean composite case.

Furthermore, we place a gamma distribution with associated hyper-hyper-parameters ω, ϕ on each kernel parameter since $\theta_{sd} \in \Re^+$. In the case of the mean composite kernel, a Dirichlet distribution with parameters $\boldsymbol{\rho}$ is placed on the combinatorial weights in order to satisfy the constraints imposed on the possible values which are defined on a simplex. A further gamma distribution is placed on each ρ_s with associated hyper-hyper-parameters μ, λ. The hyper-hyper-parameters $\boldsymbol{\Xi} = \{\tau, \upsilon, \omega, \phi, \mu, \lambda\}$ can be set by type-II maximum likelihood or set to uninformative values and the hyper and first level parameters $\boldsymbol{\Psi} = \{\mathbf{Y}, \mathbf{W}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\Theta}, \mathbf{Z}\}$ are sampled accordingly.

[2] Available from http://www.ccls.columbia.edu/compbio/svm-pairwise

It is now straightforward to see that a Gibbs sampler can be readily constructed and standard MCMC approaches (Andrieu, 2003) can be employed for Bayesian inference in our model. In this paper though we offer a variational Bayes approximation in order to achieve efficient computational processing times without loss of predictive performance.

Hence, we bound the model evidence by using an ensemble of factored posteriors to approximate the joint parameter posterior distribution. The joint likelihood of the model is defined as $p(\mathbf{t}, \mathbf{\Psi}|\mathbf{X}, \mathbf{\Xi}) = p(\mathbf{t}|\mathbf{Y})\, p(\mathbf{Y}|\mathbf{W}, \boldsymbol{\beta}, \mathbf{\Theta})\, p(\mathbf{W}|\mathbf{Z})\, p(\mathbf{Z}|\tau, \upsilon)\, p(\boldsymbol{\beta}|\boldsymbol{\rho})\, p(\mathbf{\Theta}|\omega, \phi)\, p(\boldsymbol{\rho}|\mu, \lambda)$ and the factorable ensemble approximation of the required posterior is $p(\mathbf{\Psi}|\mathbf{\Xi}, \mathbf{X}, \mathbf{t}) \approx Q(\mathbf{\Psi}) = Q(\mathbf{Y})\, Q(\mathbf{W})\, Q(\boldsymbol{\beta})\, Q(\mathbf{\Theta})\, Q(\mathbf{Z})\, Q(\boldsymbol{\rho})$. We can bound the model evidence using Jensen's inequality

$$\log p(\mathbf{t}) \geq \mathcal{E}_{Q(\mathbf{\Psi})}\{\log p(\mathbf{t}, \mathbf{\Psi}|\mathbf{\Xi})\} - \mathcal{E}_{Q(\mathbf{\Psi})}\{\log Q(\mathbf{\Psi})\} \quad (4)$$

and minimise it as usual with distributions of the form $Q(\mathbf{\Psi}_i) \propto \exp\left(\mathcal{E}_{Q(\mathbf{\Psi}_{-i})}\{\log p(\mathbf{t}, \mathbf{\Psi}|\mathbf{\Xi})\}\right)$ where $Q(\mathbf{\Psi}_{-i})$ is the factorable ensemble with the i^{th} component removed.

The resulting posterior distributions for the approximation are given below with full details of the derivations in OSM. First, the approximate posterior over the auxiliary variables is given by

$$Q(\mathbf{Y}) \propto \prod_{n=1}^{N} \delta\left(y_{i,n} > y_{k,n} \forall k \neq i\right) \delta(t_n = i)\, \mathcal{N}_{\mathbf{y}_n}\left(\widetilde{\mathbf{W}}\mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}}, \mathbf{I}\right) \quad (5)$$

which is a product of N $C-$dimensional conically truncated Gaussians. The shorthand tilde notation denotes posterior expectations in the usual manner, i.e. $\widetilde{f(\boldsymbol{\beta})} = \mathcal{E}_{Q(\boldsymbol{\beta})}\{f(\boldsymbol{\beta})\}$, and the posterior expectations for the auxiliary variable follow as

$$\widetilde{y}_{cn} = \widetilde{\mathbf{w}}_c \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}} - \frac{\mathcal{E}_{p(u)}\{\mathcal{N}_u\left(\widetilde{\mathbf{w}}_c \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}} - \widetilde{\mathbf{w}}_i \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}}, 1\right) \Phi_u^{n,i,c}\}}{\mathcal{E}_{p(u)}\{\Phi\left(u + \widetilde{\mathbf{w}}_i \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}} - \widetilde{\mathbf{w}}_c \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}}\right) \Phi_u^{n,i,c}\}} \quad (6)$$

$$\widetilde{y}_{in} = \widetilde{\mathbf{w}}_i \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}} - \left(\sum_{c \neq i} \widetilde{y}_{cn} - \widetilde{\mathbf{w}}_c \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}}\right) \quad (7)$$

where Φ is the standardized cumulative distribution function (CDF) and $\Phi_u^{n,i,c} = \prod_{j \neq i,c} \Phi\left(u + \widetilde{\mathbf{w}}_i \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}} - \widetilde{\mathbf{w}}_j \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}}\right)$. Next, the approximate posterior for the regressors can be expressed as

$$Q(\mathbf{W}) \propto \prod_{c=1}^{C} \mathcal{N}_{\mathbf{w}_c}\left(\widetilde{\mathbf{y}_c}\mathbf{K}^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}}\mathbf{V}_c, \mathbf{V}_c\right) \quad (8)$$

where the covariance is defined as

$$\mathbf{V}_c = \left(\sum_{i=1}^{S}\sum_{j=1}^{S} \widetilde{\beta_i \beta_j} \mathbf{K}^{i\tilde{\boldsymbol{\theta}}_i} \mathbf{K}^{j\tilde{\boldsymbol{\theta}}_j} + \left(\widetilde{\mathbf{Z}}_c\right)^{-1}\right)^{-1} \quad (9)$$

and $\widetilde{\mathbf{Z}}_c$ is a diagonal matrix of the expected variances $\widetilde{\zeta}_i \ldots \widetilde{\zeta}_N$ for each class. The associated posterior mean for the regressors is therefore $\widetilde{\mathbf{w}}_c = \widetilde{\mathbf{y}_c}\mathbf{K}^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}}\mathbf{V}_c$ and we can see the coupling between the auxiliary variable and regressor posterior expectation.

The approximate posterior for the variances $\mathbf{Z}$ is an updated product of inverse-gamma distributions and the posterior mean is given in the OSM or Denison *et al.* (2002). Finally, the approximate posteriors for the kernel parameters $Q(\mathbf{\Theta})$, the combinatorial weights $Q(\boldsymbol{\beta})$ and the associated hyper-prior parameters $Q(\boldsymbol{\rho})$ can be obtained by importance sampling (Andrieu, 2003) in a similar manner to Girolami and Rogers (2006) since no tractable analytical solution can be offered.

Having described the approximate posterior distributions of the parameters and hence obtained the posterior expectations we turn back to our original task of making class predictions $\mathbf{t}^*$ for N_{test} new proteins $\mathbf{X}^*$ that are represented by S different information sources $\mathbf{X}^{s*}$ embedded into Hilbert spaces as base kernels $\mathbf{K}^{*s\boldsymbol{\theta}_s, \beta_s}$ and combined into a composite *test* kernel $\mathbf{K}^{*\mathbf{\Theta}, \boldsymbol{\beta}}$. The predictive distribution for a single new protein $\mathbf{x}^*$ is given by $p(t^* = c|\mathbf{x}^*, \mathbf{X}, \mathbf{t}) = \int p(t^* = c|\mathbf{y}^*)\, p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{t})\, d\mathbf{y}^* = \int \delta_c^* p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{t})\, d\mathbf{y}^*$ which ends up, see OSM, as

$$p(t^* = c|\mathbf{x}^*, \mathbf{X}, \mathbf{t}) = E_{p(u)}\left\{\prod_{j \neq c} \Phi\left[\frac{1}{\widetilde{\nu_j^*}}\left(u\widetilde{\nu_c^*} + \widetilde{m_c^*} - \widetilde{m_j^*}\right)\right]\right\} \quad (10)$$

where, for N_{test} objects, $\widetilde{\mathbf{m}_c^*} = \widetilde{\mathbf{y}}_c \mathbf{K}\left(\mathbf{K}^*\mathbf{K}^{*T} + \mathbf{V}_c^{-1}\right)^{-1} \mathbf{K}^* \widetilde{\boldsymbol{\mathcal{V}}_c^*}$ and $\widetilde{\boldsymbol{\mathcal{V}}_c^*} = \left(\mathbf{I} + \mathbf{K}^{*T}\mathbf{V}_c\mathbf{K}^*\right)$ while we have dropped the notation for the dependance of the train $\mathbf{K}\,(N \times N)$ and test $\mathbf{K}^*\,(N \times N_{test})$ kernels on $\mathbf{\Theta}, \boldsymbol{\beta}$ for clarity.

In algorithm 1 we summarize the VB approximation in a pseudo-algorithmic fashion.

Algorithm 1 VB Multinomial Probit Composite Kernel Regression

Initialize $\mathbf{\Xi}$, sample $\mathbf{\Psi}$, create $\mathbf{K}_s|\beta_s, \theta_s$ and hence $\mathbf{K}|\boldsymbol{\beta}, \mathbf{\Theta}$
while Lower Bound changing **do**
  $\widetilde{\mathbf{w}}_c \leftarrow \widetilde{\mathbf{y}_c}\mathbf{K}\mathbf{V}_c$
  $\widetilde{y}_{cn} \leftarrow \widetilde{\mathbf{w}}_c \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}} - \frac{\mathcal{E}_{p(u)}\{\mathcal{N}_u\left(\widetilde{\mathbf{w}}_c \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}} - \widetilde{\mathbf{w}}_i \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}}, 1\right) \Phi_u^{n,i,c}\}}{\mathcal{E}_{p(u)}\{\Phi\left(u + \widetilde{\mathbf{w}}_i \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}} - \widetilde{\mathbf{w}}_c \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}}\right) \Phi_u^{n,i,c}\}}$
  $\widetilde{y}_{in} \leftarrow \widetilde{\mathbf{w}}_i \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}} - \left(\sum_{j \neq i} \widetilde{y}_{jn} - \widetilde{\mathbf{w}}_j \mathbf{k}_n^{\tilde{\boldsymbol{\beta}}\tilde{\mathbf{\Theta}}}\right)$
  $\widetilde{\zeta}_{cn}^{-1} \leftarrow \frac{2\tau+1}{2\upsilon + w_{cn}^2}$
  $\widetilde{\boldsymbol{\rho}}, \widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{\Theta}} \leftarrow \widetilde{\boldsymbol{\rho}}, \widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{\Theta}}|\widetilde{\mathbf{w}}_c, \widetilde{\mathbf{y}}_n$ by importance sampling
  Update $\mathbf{K}|\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{\Theta}}$ and $\mathbf{V}_c$
end while
Create composite test kernel $\mathbf{K}^*|\widetilde{\boldsymbol{\beta}}, \widetilde{\mathbf{\Theta}}$
$\widetilde{\boldsymbol{\mathcal{V}}_c^*} \leftarrow \left(\mathbf{I} + \mathbf{K}^{*T}\mathbf{V}_c\mathbf{K}^*\right)$
$\widetilde{\mathbf{m}_c^*} \leftarrow \widetilde{\mathbf{y}}_c \mathbf{K}\left(\mathbf{K}^*\mathbf{K}^{*T} + \mathbf{V}_c^{-1}\right)^{-1} \mathbf{K}^* \widetilde{\boldsymbol{\mathcal{V}}_c^*}$
for $n = 1$ to N_{test} **do**
  for $c = 1$ to C **do**
    for $i = 1$ to K Samples **do**
      $u_i \leftarrow \mathcal{N}(0, 1), \quad p_{cn}^i \leftarrow \prod_{j \neq c} \Phi\left[\frac{1}{\widetilde{\nu_j^*}}\left(u_i\widetilde{\nu_c^*} + \widetilde{m_c^*} - \widetilde{m_j^*}\right)\right]$
    end for
  end for
  $P(t_n^* = c|\mathbf{x}_n^*, \mathbf{X}, \mathbf{t}) = \frac{1}{K}\sum_{i=1}^{K} p_{cn}^i$
end for

4 RESULTS AND DISCUSSION

Reported results are averaged over 20 (fold recognition) and 10 (RHD) randomly initialized trials in order to obtain statistical measures of accuracy and precision. We monitor convergence via the lower bound to the marginal likelihood and convergence is assumed when there is less than 0.01% increase of the lower bound progression or when a maximum of 100 (fold recognition) and 20 (RHD) iterations have been completed. Throughout this study we have employed second order polynomial kernels for the global characteristics and inner product kernels for the local characteristics (SW) as they were found to provide a better embeding of the feature spaces. CPU times reported are for a 2 GHz Intel based PC with 2Gb RAM running Matlab codes.

A. *Fold recognition*: First we examine the performance from individual feature spaces to gain an overall understanding of their predictive abilities. This however does not draw the complete

picture as complementary information may be shared across sources achieving low performances. In Table 1 we present the mean percentage accuracy with standard deviations from our method (VBKC) together with the *best* ones reported by Ding and Dubchak (2001) on the original dataset.

Table 1. Average individual F.S percentage accuracy

Feature Space	VBKC	Ding and Dubchak
Amino Acid Composition (C)	51.2 ± 0.5	44.9
Predicted Secondary Structure (S)	38.1 ± 0.3	35.6
Hydrophobicity (H)	32.5 ± 0.4	36.5
Polarity (P)	32.2 ± 0.3	32.9
van der Waals volume (V)	32.8 ± 0.3	35
Polarizability (Z)	33.2 ± 0.4	32.9
PseAA $\lambda = 1$ (λ_1)	41.5 ± 0.5	-
PseAA $\lambda = 4$ (λ_4)	41.5 ± 0.4	-
PseAA $\lambda = 14$ (λ_{14})	38 ± 0.2	-
PseAA $\lambda = 30$ (λ_{30})	32 ± 0.2	-
SW with BLOSUM62 (SW_1)	59.8 ± 1.9	-
SW with PAM50 (SW_2)	49 ± 0.7	-

- Not employed in the Ding and Dubchak data-set.

Regarding the original features employed by Ding and Dubchak (2001) we are in agreement with their observations as the best performing feature space, seems to be the amino acid composition (C). The $\lambda = 1$ and $\lambda = 4$ PseAA achieve the second best *global* individual performance and as the "step" λ increases further, the individual performances decrease. Although according to Shen and Chou (2006) the PseAA composition "has the same form as the conventional amino acid composition, but contains much more information" it seems at this stage that none of the PseAA is as predictive as the conventional amino acid composition. Furthermore, the local characteristics (SW) surprisingly outperform every global one and SW_1 achieves a higher accuracy than the best SVM-combinations proposed by Ding and Dubchak (2001). This is because although most of the proteins have less than 35% sequence similarity, this seems to be an adequate similarity level to achieve a good accuracy.

In Table 2 we report the effect of sequentially adding the feature spaces in the order of Ding and Dubchak (2001), extending that to the addition of the PseAA compositions and finally adding the sequence similarity based features. We compare against the best performing SVM combination methodology as reported in Ding and Dubchak (2001) and the ensemble method of Shen and Chou (2006). As we can see in all the steps the proposed method outperforms the best reported accuracies and offers the current *state-of-the-art* in this data-set.

The best performances can be seen in Table 3 in comparison with the best ones reported in the cited past work. We achieve an improvement over both past methods while we employ a single multiclass kernel machine without resorting to ensemble learning techniques or combining multiple binary classifiers.

When we consider a weighted combination of the base kernels, with $\sum_{i=1}^{S} \beta_i = 1 \quad \beta_i \geq 0$ we are able to infer the significance of the corresponding feature descriptions. In Figure 3 we plot a summary of the weights over 20 runs depicting the lower quartile, median, and upper quartile values.

Table 2. Effect of F.S combination. % Accuracy reported.

Feature Spaces	VBKC	Ding & Dubchak (AvA)
C	51.2 ± 0.5	44.9
CS	55.7 ± 0.5	52.1
CSH	57.7 ± 0.6	56.0
CSHP	57.9 ± 0.9	56.5
CSHPV	58.1 ± 0.8	55.5
CSHPVZ	58.6 ± 1.1	53.9
CSHPVZλ_1	60 ± 0.8	-
CSHPVZ$\lambda_1\lambda_4$	60.8 ± 1.1	-
CSHPVZ$\lambda_1\lambda_4\lambda_{14}$	61.5 ± 1.2	-
CSHPVZ$\lambda^1\lambda^4\lambda^{14}\lambda^{30}$	62.2 ± 1.3	-
CSHPVZ$\lambda^1\lambda^4\lambda^{14}\lambda^{30}SW_1$	66.4 ± 0.8	-
CSHPVZ$\lambda^1\lambda^4\lambda^{14}\lambda^{30}SW_1SW_2$	68.1 ± 1.2	-
		Shen & Chou
SHPVZ$\lambda_1\lambda_4\lambda_{14}\lambda_{30}$	61.0 ± 1.4	62.1

- Not employed in the Ding and Dubchak dataset.

Table 3. Best single run performances (% Accuracy)

Feature Spaces	Ding & Dubchak	Shen & Chou	VBKC
CSHP	56.5	-	59.3
SHPVZ$\lambda_1\lambda_4\lambda_{14}\lambda_{30}$	-	62.1	63.5
CSHPVZ$\lambda_1\lambda_4\lambda_{14}\lambda_{30}$	-	-	63.9
CSHPVZ$\lambda^1\lambda^4\lambda^{14}\lambda^{30}SW_1SW_2$	-	-	70
No. of Classifiers	2,106	9	1

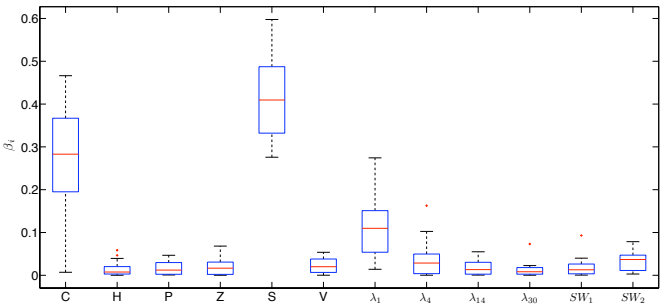

Fig. 3. Combinatorial weights when all the feature spaces are employed.

As we can observe, the amino acid composition and the secondary structure are judged as more important, followed by the PseAA $\lambda = 1$. However, it is worth noting that by taking out the amino acid composition we have only a small loss in performance as

we have seen in Table 2. These two observations suggest that the original amino acid (C) and the pseudo- ones (λ_i) carry redundant information. Furthermore, despite the individual accuracies of the SW features, they are not heavily weighted. This is because they depend solely on the sequence similarity between proteins and their quality of discriminative information is strongly related to which end of the 0-35% sequence similarity the two proteins will belong. In reality, for the real "twilight-zone" of low-homology proteins (much less than 35% similarity) such features have little effect by definition.

In Figure 4 the confusion matrix for a single run is depicted. The values on the matrix are normalized according to $R_{ij} = \frac{P_j}{N_i}$ where N_i is the total number of proteins belonging in class i and P_j is the number of these N_i proteins that were predicted to belong to class j. For example when all of the proteins in class c were predicted correctly, then $R_{cc} = 1$ and $R_{cj} = 0 \; \forall \; j \in \{1, C\}$

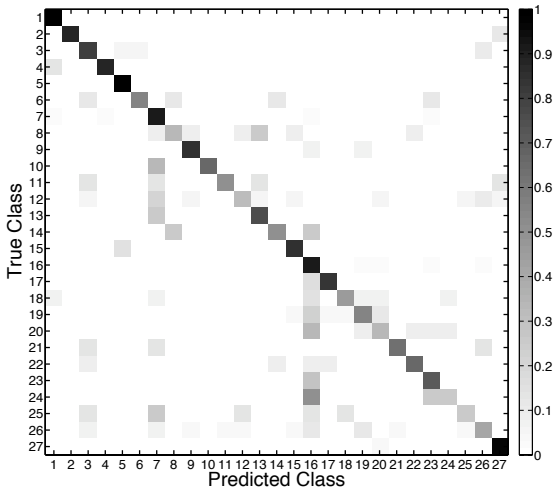

Fig. 4. Confusion matrix with each element normalized to R_{ij}

First, it is worth noting that there are two areas where consistent misclassification occurs. The first one is when proteins of class 10 to 13 (conA-like barrel, SH3-like barrel, OB, beta-trefoil) are classified as class 7 (fold: immunoglobulin like) and the second one is when proteins of class 19-20 and 24 (Rossmann fold, P-loop, periplasmic binding protein-like) are classified as class 16 (fold: TIM-barrel). Noting that folds 7 and 16 are represented by the top two largest numbers in the training set (30 and 29 proteins respectively) this seems to imply that these classes are over-represented in comparison with other folds (mean size of 10 proteins) and that features such as (pseudo- or not) amino acid composition and secondary structure offer little discriminative power on the distinction problem in these two areas.

Furthermore, besides the proteins in the fifth class (fold: 4-helical cytokines) that are correctly classified as expected by previous observations by Ding and Dubchak (2001), now the first class (fold: globin-like) is also achieving a 100% accuracy together with three more classes (7, 16, 27) (folds: immunoglobulin-like, TIM-barrel, small inhibitors) above the 90% level.

B.) *Remote homology detection*:

As a generalization of the proposed methodology to other related problem-domains we consider the *simulated* remote homology problem (RHD) as described in the works of Liao and Noble (2003); Leslie *et al.* (2004); Saigo *et al.* (2004); Lingner and Meinicke (2004). The results from the combination of the string kernels are depicted in Table 4 together with the best previously reported results within the SVM methodology. We achieve a state-of-the-art performance via the combination of the kernels and match the overall best performing SVM method outperforming other string kernels. In Fig. 5 the number of families that achieve certain ROC scores is depicted in comparison with some of the best performing methods reported in the literature.

Table 4. ROC, ROC50 and median RFP scores.

Method	Mean ROC	Mean ROC50	Mean mRFP
VBKC	0.924	0.567	0.0661
SVM (SW)	0.896	0.464	0.0837
SVM (LA)	**0.925**	**0.649**	**0.0541**
SVM (MM)	0.872	0.400	0.0837
SVM (Mono)	0.919	0.508	0.0664

Furthermore, by employing the weighted combination we infer the contribution of each string kernel and as it can be seen from Fig. 6 the Monomer (Mono) and the Local-alignment (LA) kernel are weighted most heavily as expected from Table 4 and previously reported results.

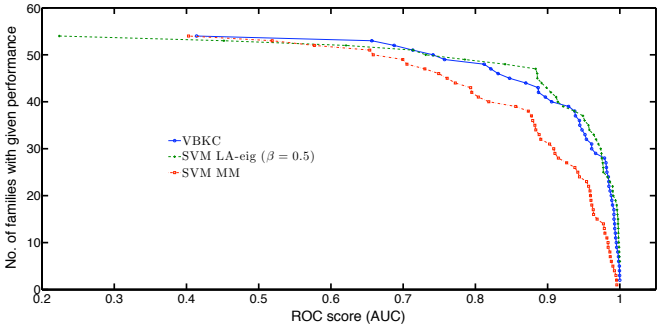

Fig. 5. ROC score (AUC) distributions for the proposed string combination method and two state-of-the-art string kernels with SVMs.

5 CONCLUSION

In this paper we offer a single probabilistic multi-class multi-kernel machine that is able to operate simultaneously in multiple feature spaces via a kernel combination methodology. Furthermore, we illustrate the capabilities of our method in a well-benchmarked dataset by Ding and Dubchak (2001) in which recent studies

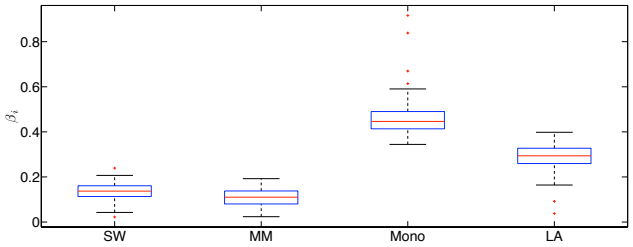

Fig. 6. Combinatorial weights when all the string kernels are employed.

(Shen and Chou, 2006) have improved predictive performance by the introduction of additional pseudo-amino acid composition feature spaces. We show that the additional feature spaces although overall improve performance by a factor of $1 - 2\%$ in reality carry non-complementary information with the original amino-acid composition. The need for such information to tackle the two misclassification patterns can also be seen in the work of (Shahbaba and Neal, 2007) where even when the problem is treated as a hierarchical classification with parent classes, the performance is not improving beyond 61.4%

Furthermore, our methodology offers a significant reduction in computational resources as it is based on a single classifier operating over a composite space which retains the dimensionality ($N \times N$) of any of the individual contributing feature spaces ($N \times N$). This, in contrast with the past work of employing thousands of binary classifiers or an ensemble of individually trained classifiers is a significant improvement. We provide, the state-of-the-art on the problem under consideration with a best performance of 70% accuracy without resorting to *ad-hoc* approaches but employing a solid Baysian formalism which enables us to infer the informative content of the feature spaces.

Finally, we extend our approach to the remote homology problem and demonstrate the generality of our approach in a practical setting by achieving a state-of-the-art performance via a combination of string kernels.

FUNDING

NCR Financial Solutions Group Ltd. Scholarship to T.D; EPSRC Advanced Research Fellowship (EP/E052029/1) to M.A.G

ACKNOWLEDGEMENT

The authors would like to acknowledge insightful discussions with Dr. David Leader and Dr. Rainer Breitling, and helpful suggestions by the anonymous reviewers. The first author would like to acknowledge the support received from the Advanced Technology & Research Group within the NCR Financial Solutions Group Ltd company and especially the help and support of Dr. Gary Ross and Dr. Chao He.

REFERENCES

Albert, J. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, **88**, 669–679.

Andreeva, A., Howorth, D., Brenner, S., Hubbard, T., Chothia, C., and Murzin, A. (2004). Scop database in 2004: refinements integrate structure and sequence family data. *Nucleic Acids Res.*, **32**, 226–229.

Andrieu, C. (2003). An introduction to MCMC for machine learning. *Machine Learning*, **50**, 5–43.

Baker, D. and Sali, A. (2001). Protein structure prediction and structural genomics. *Science*, pages 93–96.

Chou, K. (2005). Using amphiphilic pseudo-amino acid composition to predict enzyme subfamily classes. *Bioinformatics*, **21**, 10–19.

Chou, K. and Zhang, C. (1995). Prediction of protein structural classes. *Critical Revi. Biochem. Mol. Biol.*, **30**, 275–349.

Denison, D. G. T., Holmes, C. C., Mallick, B. K., and Smith, A. F. M. (2002). *Bayesian Methods for Nonlinear Classification and Regression*. Wiley Series in Probability and Statistics, West Sussex, UK.

Ding, C. and Dubchak, I. (2001). Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, **17**(4), 349–358.

Dubchak, I., Muchnik, I., Holbrook, S., and Kim, S. (1995). Prediction of protein folding class using global decsription of amino acid sequence. *Proc. Natl. Acad. Sci.*, **92**, 8700–8704.

Girolami, M. and Rogers, S. (2006). Variational Bayesian multinomial probit regression with Gaussian process priors. *Neural Computation*, **18**(8), 1790–1817.

Girolami, M. and Zhong, M. (2007). Data integration for classification problems employing Gaussian process priors. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 465–472, Cambridge, MA. MIT Press.

Jaakkola, T., Diekhans, M., and Haussler, D. (1999). Using the fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Inteligent Systems in Molecular Biology*. AAAI Press.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., and Jordan, M. I. (2004a). Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, **5**, 27–72.

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

Leslie, C. S., Eskin, E., Cohen, A., Weston, J., and Noble, W. S. (2004). Mismatch string kernels for discriminative protein classification. *Bioinformatics*, **20**(4), 467–476.

Lewis, D. P., Jebara, T., and Noble, W. S. (2006a). Nonstationary kernel combination. In *23rd International Conference on Machine Learning*.

Lewis, D. P., Jebara, T., and Noble, W. S. (2006b). Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure. *Bioinformatics*, **22**(22), 2753–2760.

Liao, L. and Noble, W. S. (2003). Combining pairwise sequnce similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *Journal of Computational Biology*, **6**(6), 857–868.

Lingner, T. and Meinicke, P. (2004). Remote homology detection based on oligomer distances. *Bioinformatics*, **22**(18), 2224–2231.

Lo Conte, L., Ailey, B., Hubbard, T., Brenner, S., Murzin, A., and Chothia, C. (2000). Scop: a structural classification of proteins database. *Nucleic Acids Res.*, **28**, 257–259.

Melvin, I., Ie, E., Weston, J., Noble, W. S., and Leslie, C. (2007). Multi-class protein classification using adaptive codes. *Journal of Machine Learning Research*, **8**, 1557–1581.

Raval, A., Ghahramani, Z., and Wild, D. L. (2002). A bayesian network model for protein fold and remote homologue recognition. *Bioinformatics*, **18**, 788–801.

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

Shahbaba, B. and Neal, R. M. (2007). Nonlinear models using dirichlet process mixtures. Technical Report 0707, Department of Statistics, University of Toronto.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, England, UK.

Shen, H.-B. and Chou, K.-C. (2006). Ensemble classifier for protein fold pattern recognition. *Bioinformatics*, **22**(14), 1717–1722.

Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, **1**, 1–18.