Supervised Selective Kernel Fusion for Membrane Protein Prediction

Alexander Tatarchuk[1], Valentina Sulimova[2], Vadim Mottl[1], and David Windridge[3]

[1] Computing Center of the Russian Academy of Sciences,Moscow, Russia
[2] Tula State University, Tula, Russia
[3] Centre for Vision, Speech and Signal Processing,
University of Surrey, Guildford, UK
aitech@yandex.ru, vsulimova@yandex.ru, vmottl@yandex.ru,
D.Windridge@surrey.ac.uk

Abstract. Membrane protein prediction is a significant classification problem, requiring the integration of data derived from different sources such as protein sequences, gene expression, protein interactions etc. A generalized probabilistic approach for combining different data sources via supervised selective kernel fusion was proposed in our previous papers. It includes, as particular cases, SVM, Lasso SVM, Elastic Net SVM and others. In this paper we apply a further instantiation of this approach, the *Supervised Selective Support Kernel SVM* and demonstrate that the proposed approach achieves the top-rank position among the selective kernel fusion variants on benchmark data for membrane protein prediction. The method differs from the previous approaches in that it naturally derives a subset of "support kernels" (analogous to support objects within SVMs), thereby allowing the memory-efficient exclusion of significant numbers of irrelevant kernel matrixes from a decision rule in a manner particularly suited to membrane protein prediction.

Keywords: Multiple Kernel Learning, SVM, supervised selectivity, support kernels, membrane protein prediction

1 Introduction

Membrane proteins carry out a variety of crucial functions in cells, such as removing polluting hydrophobic molecules; transporting undesired molecules, such as drugs, out of the cell; sending signals concerning events occurring outside the cell across the membrane into the cell in order that proper action can be taken such as e.g. the starting or stopping of cell division etc. Consequently, membrane protein prediction, i.e. the classification of proteins as either a membrane or non-membrane is a medically important problem, and the subject of much research [1],[2],[3].

This is a typical pattern recognition problem in that the most informative individual feature (in this case typically amino acid sequence data) does not provide the full story. Additional feature information can be derived from a number

of other sources, such as gene expression data, protein-protein interactions and so on. All these data sources contain different and at least partly independent information about membrane protein prediction task. Consequently, there is a natural desire to incorporate them into a combined prediction rule to decrease prediction-errors.

If the data consisted of vectorized features then this act of combination would constitute a trivial matter of appending feature spaces. However, this is generally not the case for gene-based problems, where data may, for instance, consist only of pairwise comparisons. The most appropriate way for integrating heterogeneous data with a wide variety of gene representations (in this case, amino acid and gene sequences, feature vectors, graphs and so on) thus consists in embedding data objects into representation-specific hypothetical linear spaces via kernel functions and constructing the decision function at the combined space. (A kernel function is any real-valued symmetric function of two-arguments, which forms a semidefinite matrix for any finite collection of objects [4],[5]). In particular, there are a number of approaches in the literature for introducing kernel functions into biomolecular data (cf [4]).

Any kernel function embeds a set of objects into some linear space and plays the role of inner product within it [4],[5]. This fact allows us to employ the kernel-based interpretation of the Support Vector Machine (SVM) method, which was originally designed for linear feature space [6] and is one of the most convenient and effective instruments for the binary classification of objects, forming an optimal linear separating hyperplane from specific "support" training examples.

Mercer Kernels further have the property that linear combinations are also Mercer, meaning that kernel combination is straightforward. There have thus been a number of attempts at combine kernel functions for biomolecular data analysis, the simplest approach being an unweighted sum of kernels. Different linear (or even non-linear) combinations with fixed or heuristically-chosen weights have also been considered; however, overall performance is generally poor.

The most general method of kernel fusion is the approach of Lanckriet et al. [8] which seeks to directly solve for the optimal linear combination of kernels and gives rise to a quadratically-constrained algorithm for determining the non-negative adaptive weights of kernel matrices. The respective kernel combination is incorporated into a decision rule with each kernel's influences on the decision proportional to its weight.

A number of authors have carried this work further in various ways, generalizing the approach to problems other than classification [9],[10], working on algorithmic improvements [11],[12], or deriving theoretical variations, applying different restrictions for weights [13] and making certain theoretical extensions, e.g. weighting not only kernels but also features [14],[15]. These variants typically perform well in constrained scenarios, and where the data are initially represented by feature vectors. However, they tend not to out-perform [8] on real protein data.

Furthermore, most of existing multiple kernel learning methods share a common disadvantage - the absence of a mechanism for supervising so-called "sparse-

ness" of the obtained vector of kernel weights. In the genetic arena, the obtained vector of weights is frequently too sparse, with many informative kernels excluded from the decision rule, with the resulting loss the decision quality.

Only a few methods are explicitly oriented towards elimination of this disadvantage and obtaining non-sparse decisions [17],[18] (more advanced versions utilize a *supervised* sparseness parameter [19],[20],[21],[22]). We refer to this property as "*selectivity*", because it defines an algorithm's ability to select kernels most useful to the classification task at hand. A generalized probabilistic approach for supervised selective kernel fusion was proposed by the authors in [21],[22] and includes, as particular cases, such familiar approaches as the classical SVM [6], Lasso SVM [23], Elastic Net SVM [24] and others.

In this paper we apply a further particular case of this approach, called Supervised Selective Support Kernel SVM (SKSVM), initially proposed in [22] to the membrane protein prediction problem.

We will demonstrate that the proposed approach achieves the top-ranked position among the selective kernel fusion variants on benchmark data set for membrane protein prediction. Uniquely, the proposed approach has the very significant qualitative advantage over the other methods of explicitly indicating a discrete subset of support kernels within the combination, in contrast to the other methods that assign some positive (even if small) weight to *each* kernel, requiring significantly greater memory overhead.

2 Generalized Probabilistic Formulation of the Multiple Kernel Two-Class Recognition Problem

Let $\{(\omega_j, y_j),\ j=1,...,N\}$ be the training set of real-world objects $\omega_j \in \Omega$ (for example, proteins) and $y_j = y(\omega_j) \in \{-1, 1\}$ defines its class-membership. Let also n similarity functions $K_i(\omega', \omega''), \omega', \omega'' \in \Omega, i = 1, ..., n$ be defined, each of which forms a positive semidefinite matrix $\{K_i(\omega_j, \omega_k)\}$ for any finite set of objects $\{\omega_j, \omega_k \in \Omega, j, k = 1, ..., S\}$ and is hence a kernel function [5].

Each kernel function $K_i(\omega', \omega''), i = 1, ..., n$ embeds the set of objects Ω into some hypothetical linear space $\mathbb{X}_i$ by a hypothetical mapping $x_i = x_i(\omega) \in \mathbb{X}_i, \omega \in \Omega$, and plays the role of inner product within it $K_i(\omega', \omega'') = < x_i(\omega'), x_i(\omega'') >$: $\mathbb{X}_i \times \mathbb{X}_i \to \mathbb{R}$.

For combination using several kernels we here utilize the generalized probabilistic formulation of the SVM, which was proposed in [18, 20, 21] as an instrument for making Bayesian decisions on the discriminant hyperplane $\sum_{i=1}^{n} K_i(a_i, \omega) + b \gtrless 0$ within the Cartesian product of the kernel-induced hypothetical linear spaces $\boldsymbol{a} = (a_1, ..., a_n) \in \mathbb{X}$, $b \in \mathbb{R}$.

The main idea of the proposed probabilistic formulation consists in assuming a specific system of probabilistic assumptions regarding the two distribution densities of hypothetical feature vectors for the two classes: $\varphi(\boldsymbol{x}|y=+1)$ and $\varphi(\boldsymbol{x}|y=-1)$, defined by the (as yet) undetermined hyperplane in the combined linear space $\boldsymbol{x} = (x_1, ..., x_n) \in \mathbb{X} = \mathbb{X}_1 \times ... \times \mathbb{X}_n$ under certain *a priori* probabilistic assumptions.

Let $\boldsymbol{a}^T\boldsymbol{x} + b \gtrless 0$ be some hyperplane with the direction element $\boldsymbol{a} \in \mathbb{X}$ and the bias $b \in \mathbb{R}$. Associated with it are two parametric families of conditional distributions of object densities:

$$\varphi(\boldsymbol{x}|\boldsymbol{a}, b, y; c) = const \begin{cases} 1, y(\boldsymbol{a}^T\boldsymbol{x} + b) \geqslant 1, \\ exp\left[-c\left(1 - y(\boldsymbol{a}^T\boldsymbol{x} + b)\right)\right], \quad y\left(\boldsymbol{a}^T\boldsymbol{x} + b\right) < 1. \end{cases} \tag{1}$$

We assume that the random vectors of two classes are distributed substantially within their respective subspaces $\boldsymbol{a}^T\boldsymbol{x} + b > 0$ and $\boldsymbol{a}^T\boldsymbol{x} + b < 0$; the parameter c regulates the extent to which this assumption holds. (Note that fact that the uniform distribution in the upper row of (1) implies an infinite area does not lead to mathematical contradiction, since it participates only in the Bayes' formula[25]).

Suppose the training set $\{(\boldsymbol{x}_j, y_j),\ j=1, ..., N\}, \boldsymbol{x}_j \in \mathbb{X} = \mathbb{X}_1 \times ... \times \mathbb{X}_n, y_j = \pm 1$ has been obtained. Then the conditional distribution of the whole training set is

$$\Phi(X|Y, \boldsymbol{a}, b; c) = \prod\nolimits_{j=1}^{N} \varphi(\boldsymbol{x}_j|\boldsymbol{a}, b, y_j; c). \tag{2}$$

The second key assumption in the proposed probabilistic model is the assumption of a joint *a priori* distribution $\Psi(\boldsymbol{a}, b)$ of parameters $(\boldsymbol{a}, b)$ defining the separating hyperplane. Assume that we have no any *a priori* preferences about b. We then have that:

$$\Psi(\boldsymbol{a}, b) \propto \Psi(\boldsymbol{a}). \tag{3}$$

The *a posteriori* distribution density $P(\boldsymbol{a}, b|X, Y; c)$ of parameters $(\boldsymbol{a}, b)$ with respect to the training set (X, Y) is then defined by Bayes' formula:

$$P(\boldsymbol{a}, b|X, Y; c) = \frac{\Psi(\boldsymbol{a}, b)\Phi(X|Y, \boldsymbol{a}, b; c)}{const} \propto \Psi(\boldsymbol{a}, b)\Phi(X|Y, \boldsymbol{a}, b). \tag{4}$$

Understanding the training problem as that of maximizing this *a posteriori* distribution density in the space of model parameters $(\boldsymbol{a}, b)$ leads to the criterion:

$$(\hat{\boldsymbol{a}}, \hat{b}|X, Y; c) = \underset{\boldsymbol{a}\in\mathbb{X}, b\in\mathbb{R}}{\operatorname{argmax}} \left[\ln \Psi(\boldsymbol{a}, b) + \ln \Phi(X|Y, \boldsymbol{a}, b; c)\right] \tag{5}$$

Theorem 1. *The training criterion (5) for distributional family (1) and* a-priori *distribution of hyperplane parameters (3) is equivalent to the problem of minimization of the criterion $J(\boldsymbol{a}, b, \boldsymbol{\delta}|c)$ in a convex set defined by linear inequality constraints for training objects:*

$$\begin{cases} -\ln \Psi(a_1, ..., a_n) + c\sum\limits_{j=1}^{N} \delta_j \to \min\left(a_i \in \mathbb{X}_i, b \in \mathbb{R}, \delta_j \in \mathbb{R}\right), \\ y_j \left(\sum\limits_{i=1}^{n} < a_i, x_i(\omega_j) > + b\right) \geqslant 1 - \delta_j,\ \delta_j \geqslant 0,\ j = 1, ..., N. \end{cases} \tag{6}$$

Kernelizing criterion (6) yields the form:

$$\begin{cases} -\ln \Psi(a_1, ..., a_n) + c \sum_{j=1}^{N} \delta_j \rightarrow \min\big(a_i \in \mathbb{X}_i, b \in \mathbb{R}, \delta_j \in \mathbb{R}\big), \\ y_j \left(\sum_{i=1}^{n} K_i(a_i, \omega_j) + b \right) \geqslant 1 - \delta_j, \ \delta_j \geqslant 0, \ j = 1, ..., N. \end{cases} \tag{7}$$

Each specific choice of *a priori* distribution density $\Psi(a_1, ..., a_n)$ expresses a specific *a priori* preference about the hyperplane orientation, and endows training criterion (7) with the ability to select informative kernel-representations (and suppress redundant ones).

In particular, a number of well-known SVM-based training criteria can be obtained form the proposed probabilistic approach, for example, the traditional SVM, Lasso SVM and Elastic Net SVM, differing from one another in the regularization function, which has the form, respectively: $\sum_{i=1}^{n} K_i(a_i, a_i)$, $\sum_{i=1}^{n} \sqrt{K_i(a_i, a_i)}$ and $\sum_{i=1}^{n} K_i(a_i, a_i) + \mu \sum_{i=1}^{n} \sqrt{K_i(a_i, a_i)}$.

3 Supervised Selective Support Kernel SVM (SKSVM)

We apply here a very specific case of the general problem formulation (7), one which was initially proposed in [22]. The *a priori* density of orientation distributions is represented here as composite of the Laplace distribution, while the norms of the components are not less than some given threshold $\sum_{i=1}^{n} \sqrt{K_i(a_i, a_i)} \leq \mu$, and the Gaussian distribution when the norms are greater than the given threshold
$\sum_{i=1}^{n} \sqrt{K_i(a_i, a_i)} > \mu$:

$$\begin{aligned} &\psi(a_i|\mu) \propto \exp(-q(a_i|\mu)), \\ &q(a_i|\mu) = \begin{cases} 2\mu \sum_{i=1}^{n} \sqrt{K_i(a_i, a_i)}, & \sum_{i=1}^{n} \sqrt{K_i(a_i, a_i)} \leq \mu, \\ \mu^2 + \sum_{i=1}^{n} K_i(a_i, a_i), & \sum_{i=1}^{n} \sqrt{K_i(a_i, a_i)} > \mu. \end{cases} \end{aligned} \tag{8}$$

The *a priori* assumption of (8) along with the generalized training criterion (7) together define a training optimization problem of the form:

$$\begin{cases} J_{SKSVM}(a_1, ..., a_n, b, \delta_1, ..., \delta_N \,|\, c, \mu) = \\ \sum_{i=1}^{n} q(a_i \,|\, \mu) + c \sum_{j=1}^{N} \delta_j \rightarrow \min\big(a_i \in \mathbb{X}_i, \ b \in \mathbb{R}, \ \delta_j \in \mathbb{R}\big), \\ q(a_i \,|\, \mu) = \begin{cases} 2\mu \sqrt{K_i(a_i, a_i)} & \text{if } \sqrt{K_i(a_i, a_i)} \leqslant \mu, \\ \mu^2 + K_i(a_i, a_i) & \text{if } \sqrt{K_i(a_i, a_i)} > \mu, \end{cases} \\ y_j \left(\sum_{i=1}^{n} K_i(a_i, x_{ij}) + b \right) \geqslant 1 - \delta_j, \delta_j \geqslant 0, j = 1, ..., N. \end{cases} \tag{9}$$

The proposed training criterion (9) is thus a generalized version of the classical SVM that implements the principle of*kernel selection*. We hence refer to the threshold $0 \leqslant \mu < \infty$ in(1) as a "selectivity" parameter because it regulates the ability of the criterion to enact kernel selection. When $\mu = 0 \Rightarrow$

$q(a_i|\mu) = K_i(a_i, a_i)$ the criterion (7) is equivalent to the kernel-based SVM criterion with the minimum ability to kernel selection. At the same time, values $\mu \gg 0 \Rightarrow q(a_i|\mu) = 2\mu\sqrt{K_i(a_i, a_i)}$ are equivalent to the Lasso SVM with increasing selectivity as μ is increased with respect to the parameter c (until full suppression of all kernels occurs).

Moreover, this criterion, in contrast to other criteria for kernel fusion, explicitly partitions the entire set into two subsets (as is shown in the next section); "support" kernels (which occur in the resulting discriminant hyperplane) and excluded kernels. The proposed approach is hence termed the *Supervised Selective Support Kernel SVM (SKSVM).*

The approach to solving problem (9) is set out the following two theorems; more detailed description can be found at [22].

Theorem 2. *The decision implicit in problem (9) is equivalent to the decision* $(\hat{\xi}_i \geqslant 0,\ i \in I = \{1, ..., n\},\ \hat{\lambda}_j \geqslant 0, j = 1, ..., N)$ *of the dual problem*

$$\begin{cases} L\left(\lambda_1, ..., \lambda_N | c, \mu\right) = \sum\limits_{j=1}^{N} \lambda_j - \sum\limits_{i \in I} (1/2)\xi_i \to \max(\lambda_1, ..., \lambda_N), \\ \xi_i \geq 0,\ \ \xi_i \geq \sum\limits_{j=1}^{N} \sum\limits_{l=1}^{N} y_j y_l K_i(\omega_j, \omega_l)\lambda_j \lambda_l - \mu^2,\ \ i \in I = \{1, ..., n\}, \\ \sum\limits_{j=1}^{N} y_j \lambda_j = 0,\ \ 0 \leq \lambda_j \leq (c/2),\ j = 1, ..., N. \end{cases} \tag{10}$$

and is expressed at the form

$$\begin{cases} \hat{a}_i = \sum\limits_{j:\hat{\lambda}_j > 0} y_j \hat{\lambda}_j x_i(\omega_j),\ i \in I^+ = \left\{ i \in I : \sum\limits_{j=1}^{N} \sum\limits_{l=1}^{N} y_j y_l K_i(\omega_j, \omega_l)\hat{\lambda}_j \hat{\lambda}_l - \mu^2 > 0 \right\}. \\ \hat{a}_i = \hat{\eta}_i \sum\limits_{j:\hat{\lambda}_j > 0} y_j \hat{\lambda}_j x_i(\omega_j), i \in I^0 = \left\{ i \in I : \sum\limits_{j=1}^{N} \sum\limits_{l=1}^{N} y_j y_l K_i(\omega_j, \omega_l)\hat{\lambda}_j \hat{\lambda}_l - \mu = 0 \right\}, \\ \hat{a}_i = 0, \qquad\qquad i \in I^- = \left\{ i \in I : \sum\limits_{j=1}^{N} \sum\limits_{l=1}^{N} y_j y_l K_i(\omega_j, \omega_l)\hat{\lambda}_j \hat{\lambda}_l - \mu^2 < 0 \right\}, \end{cases} \tag{11}$$

4 The resulting discriminant hyperplane and support kernels

Assume the dual optimization problem (10) has been solved. Only the Lagrange multipliers $\lambda_1 \geqslant 0, ..., \lambda_N \geqslant 0$ are of interest. In accordance with (11), the solution arrived at partitions the set of all kernels $I = \{1, ..., n\}$ into three subsets:

$$I^+ = \left\{ i \in I : \sum_{j=1}^{N} \sum_{l=1}^{N} y_j y_l K_i(x_{ij}, x_{il}) \lambda_j \lambda_l > \mu^2 \right\},$$
$$I^0 = \left\{ i \in I : \sum_{j=1}^{N} \sum_{l=1}^{N} y_j y_l K_i(x_{ij}, x_{il}) \lambda_j \lambda_l = \mu^2 \right\}, \quad (12)$$
$$I^- = \left\{ i \in I : \sum_{j=1}^{N} \sum_{l=1}^{N} y_j y_l K_i(x_{ij}, x_{il}) \lambda_j \lambda_l < \mu^2 \right\}.$$

Theorem 3. *The optimal discriminant hyperplane defined by the solution of the Supervised Selective Support SVM training problem (9) has the form*

$$\sum_{j:\lambda_j>0} y_j \lambda_j \left(\sum_{i \in I^+} K_i(\omega_j, \omega) + \sum_{i \in I^0} \eta_i K_i(\omega_j, \omega) \right) + b \gtrless 0, \quad (13)$$

where the numerical parameters $\{0 \leqslant \eta_i \leqslant 1,\ i \in I^0;\ b\}$ *are solutions of the linear programming problem:*

$$\begin{cases} 2\mu^2 \sum\limits_{i \in I^0} \eta_i + c \sum\limits_{j=1}^{n} \delta_j \to \min(\eta_i, i \in I^0;\ b;\ \delta_1, \ldots, \delta_N), \\ \sum\limits_{i \in I^0} \left(\sum\limits_{l=1}^{N} y_j y_l K_i(\omega_j, \omega_l) \lambda_l \right) \eta_i + y_j b + \delta_j \geqslant 1 - \sum\limits_{i \in I^+} \sum\limits_{l=1}^{N} y_j y_l K_i(\omega_j, \omega_l) \lambda_l, \\ \delta_j \geqslant 0,\ j = 1, \ldots, N, \quad 0 \leqslant \eta_i \leqslant 1,\ i \in I^0. \end{cases} \quad (14)$$

5 The subset of support kernels

The solution $(\hat{\eta}_i, i \in I^0;\ \hat{b};\ \hat{\delta}_1, \ldots, \hat{\delta}_N)$ of the linear programming problem (14) is completely defined by the training set (X, Y). As is seen from criterion (14), some of coefficients $(\hat{\eta}_i, i \in I^0)$ may equal zero if the respective constraints $0 \leqslant \eta_i$ are active at the solution point.

However, it can be shown that, if all the linear spaces $\mathbb{X}_i$ are finite-dimensional and if the training set is considered as randomly-selected points defined by a continuous probability distribution, then the inequalities $\hat{\eta}_i > 0$ are almost certainly met for all $i \in I^0$.

This means that, without any loss of generality, the constraints $\{0 \leqslant \eta_i \leqslant 1,\ i \in I^0\}$ may be omitted in (14), and, yet, all kernels $i \in I^0$ will occur in the discriminant hyperplane (13) with nonzero weights. It is hence natural (by analogy with the notion of support objects) to call the subset $I_{supp} = I^+ \cup I^0 \subseteq I$ the set of support kernels.

The structure of the subsets of kernels (12) explicitly reveals how the subset of support kernels I_{supp} is affected by the parameter μ in the training criterion (9). Thus, if $\mu = 0$, the set of evident support kernels $I^+ \subseteq I$ coincides with the entire set $I = \{1, \ldots, n\}$. In this particular case, the function $q(a_i \mid \mu)$ in (9) is quadratic $q(a_i \mid \mu) = const + K_i(a_i, a_i)$ for all $a_i \in \mathbb{X}_i$, and the training criterion does not differ from the usual SVM without selectivity properties; all of the initial kernels are support kernels because they all occur in the resulting decision rule.

As μ grows, increasing numbers of kernels appear in the set I^- of nonsupport kernels (12), and, correspondingly, the set of support kernels $I_{supp} = I^+ \cup I^0$ gets smaller. At the asymptote, the selectivity parameter $\mu \to \infty$ forces all kernels into I^-, such that no support kernels remain at all: $I_{supp} = \varnothing$.

6 Adjusting the Selectivity Parameter

The selectivity parameter $0 \leqslant \mu < \infty$ is thus a structural parameter of the Supervised Selective Support Kernel SVM training criterion that determines a sequence of nested classes of training-set models whose dimensionality diminishes as μ grows, starting from the usual SVM model when $\mu = 0$. As it is not determined *a priori*, at present, the most effective method for choosing the value of the structural parameter is via cross-Validation, directly estimating the generalization performance of the training method.

7 Experimental Design

7.1 Membrane Proteins Data Set

To evaluate the proposed approach as a method for membrane protein prediction we use the same data set as Lanckriet et all. (described in [8]). We thus use as a gold standard the annotations provided by the Munich Information Center for Protein Sequences Comprehensive Yeast Genome Database (CYGD) [26]. The CYGD assigns subcellular locations to 2318 yeast proteins, of which 497 belong to various membrane protein classes. The remaining approximately 4000 yeast proteins have uncertain location and are therefore not used in these experiments.

7.2 Kernel Functions for Membrane Proteins

For the membrane protein prediction we evaluate seven kernel matrices derived from three different types of data: four from the primary protein sequence, two from proteinprotein interaction data, and one from mRNA expression data collected by Lanckriet et all. [8]. (All of these kernel matrices, along with the data from which they were generated are available at *noble.gs.washington.edu/proj/sdp-svm*).

The first two kernel matrices (K_{SW} and K_B) are based on the pairwise sequence alignment algorithms SmithWaterman local alignment (SW) and BLAST (B).

The third kernel (K_{Pfam}) was also derived from protein sequences, but was obtained using hidden Markov models (HMMs) on the Pfam database.

The fourth kernel (K_{FFT}) uses hydropathy profiles, generated from the Kyte-Doolittle index and characterized by alternations of hydrophobic and hydrophilic aminoacids regions which are sufficiently conserved for membrane proteins. The frequency content of the hydropathy profiles, estimated by a FFT procedure, was utilized as a feature vector and used for forming the Gaussian (radial) kernel.

The next two kernels - the linear kernel (K_{Li}) and the diffusion kernel (K_D) are constructed from information about medium- and high-confidence protein-protein interactions from a database of known interactions, which is presented as a matrix [2318x2318] of ones (for pairs of interacted proteins) and zeros (for pairs of non-interacted proteins).

The linear kernel (K_{Li}) matrix is derived from protein feature vectors (i.e. via the inner-product of protein feature pairs).

The diffusion kernel (K_D) considers the interaction-matrix as a graph, in which the nodes corresponded to proteins and the edges to the interactions between them. The diffusion kernel function then measures the similarity of two nodes of the graph based on a randomwalk distance, i.e. such that nodes that are connected by shorter paths (or by many paths) are considered more similar.

Finally, the seventh kernel (K_E) is a radial kernel constructed on the basis of 441-element feature vectors obtained entirely from microarray gene expression measurements. Though gene expression information is not expected to be particularly correlated with any one membrane protein, it is not possible to exclude this kernel *a priori*.

Additionally, five random kernels ($K_{Rnd1}, ..., K_{Rnd5}$) were computed on the basis of 100-length feature vectors, randomly generated without taking into account labeling information about the classes of the proteins. These non-informative kernels were introduced in order to check the ability of the proposed procedure to eliminate non-useful information.

7.3 Experimental setup

The full set of 2318 proteins (497 membrane proteins and 1821 non-membrane proteins) was randomly split 30 times into training and test sets in the proportion 80:20. As a result, each training set contained 397 membrane proteins and 1456 non-membrane proteins. Each of the test sets contain, respectively, 100 membrane proteins and 365 non-membrane proteins.

For each of 30 training sets obtained we derive 20 different decision rules for membrane protein prediction:

1. For each of 7 informative and 5 random kernels the traditional SVM training procedure was performed separately;
2. SVM classification on the unweighted sum of all 12 kernels was also applied;
3. For all 12 kernels, the proposed Selective Supervised Selective Kernel SVM was performed 6 times with 6 different values of the selectivity-parameter;
4. The optimal decision rule was selected for the proposed method via 5-fold cross-validation.

As a pre-processing step each kernel matrix was centered and normalized to be a unit diagonal matrix.

The quality of each decision was estimated via the ROC-score using the hyperplane bias b to vary sensitivity.

7.4 Results and Discussion

The average of 30 ROC-scores, obtained for each of 20 training conditions listed in the previous section, are presented in table 1.

Table 1. Results of membrane protein prediction

Kernels	algorithm	μ	ROC-score	Kernels	algorithm	μ	ROC-score
K_B	SVM	-	0,825± 0,032	K_{Rnd4}	SVM	-	0,521± 0,029
K_{SW}	SVM	-	0,809± 0,027	K_{Rnd5}	SVM	-	0,509± 0,029
K_{Pham}	SVM	-	0,859± 0,022	All 12	SVM	-	0,881± 0,014
K_{FTT}	SVM	-	0,776± 0,014	All 12	SKSVM	0	0,881± 0,014
K_{Li}	SVM	-	0,634± 0,042	All 12	SKSVM	1	0,881± 0,015
K_D	SVM	-	0,638± 0,037	All 12	SKSVM	5	0,909± 0,014
K_E	SVM	-	0,752± 0,022	All 12	SKSVM	7.5	**0,917± 0,015**
K_{Rnd1}	SVM	-	0,510± 0,029	All 12	SKSVM	10	0,916± 0,015
K_{Rnd2}	SVM	-	0,517± 0,028	All 12	SKSVM	15	0,904± 0,015
K_{Rnd3}	SVM	-	0,515± 0,030	All 12	SKSVM	optimal	**0,918± 0,016**

As we can see from table 1, the results of the proposed supervised selective support kernel SVM outperform those obtained for each of 12 kernels individually, and also those of the unweighted kernel sum with SVM training. The result obtained at the zero-selectivity level is exactly equal to the result obtained for the unweighted kernel sum (and which supports the theoretical results above).

Moreover, it may be seen that practically all reasonable values of the selectivity-parameter provide good results. The performance obtained using the optimal selectivity value selected via 5-fold cross-validation for each of 30 training sets individually only slightly outperforms the best result obtained using fixed selectivity-levels. This implies that the same selectivity-level is near optimal across the range of training sets (though of course a fixed selectivity level may be not appropriate for different tasks, for example, for recognition different classes of proteins).

The reported results of membrane protein prediction obtained by another multiple kernel learning techniques [8], [14], [16] for the same data set lie in the range [0.87-0.917]. The proposed approach therefore achieves the top-ranked position of the methods reported in the literature. It should be noted, furthermore, that the proposed approach has the unique qualitative advantage of clearly-delineating the subset of support kernels that participate in the decision rule, being thereby directly scientifically interpretable, and potentially assisting with further experimental hypothesis generation.

To demonstrate this delineation of support kernels for one of 30 training sets, table 2 contains the results of the partitionings of the full set of 12 kernels into three subsets: 1) the subset of kernels I^-, which were classified by the algorithm as non-supported, and which are not weighted or included in the decision rule; 2) the subset of kernels I^+ having unit weight and 3) the subset of kernels I^0 having a weight between 0 and 1.

Only the kernels of subsets I^+ and I^0 are support kernels and participate in the decision rule.

Table 2. Kernel fusion results for different selectivity values μ: subsets of non-support (I^-) kernels and support (I^+ and I^0) kernels with their weights

μ	K_B	K_{SW}	K_{Pham}	K_{FTT}	K_{Li}	K_D	K_E	K_{Rnd1}	K_{Rnd2}	K_{Rnd3}	K_{Rnd4}	K_{Rnd5}	ROC
0	I^+	I^+	I^+	I^+	I^+	I^+	I^+	I^+	I^+	I^+	I^+	I^+	
	1	1	1	1	1	1	1	1	1	1	1	1	0.877
5	I^+	I^+	I^+	I^+	I^0	I^+	I^+	I^0	I^0	I^0	I^0	I^0	
	1	1	1	1	0.87	1	1	0.26	0.26	0.14	0.36	0.30	0.907
7.5	I^0	I^+	I^0	I^0	I^-	I^+	I^0	I^-	I^-	I^-	I^-	I^-	
	0.70	1.00	0.88	0.81	-	1.00	0.99	-	-	-	-	-	0.919
10	I^0	I^+	I^0	I^0	I^-	I^+	I^0	I^-	I^-	I^-	I^-	I^-	
	0.34	1	0.63	0.56	-	1	0.72	-	-	-	-	-	0.913
15	I^-	I^0	I^0	I^0	I^-	I^0	I^0	I^-	I^-	I^-	I^-	I^-	
	-	0.95	0.14	0.10	-	0.94	0.11	-	-	-	-	-	0.889

As we can see from table 2, the highest selectivity value excludes the random kernels from the set of support kernels entirely. Also, the interaction-based linear kernel K_{Li} was excluded in the most cases. Thus, only half (6 of 12) of the full kernel set are support kernels in this example, saving on memory requirements.

It is thus, in sum, this particular feature of the proposed method makes it preferable to other multi-kernel methods within the literature, which generally assign positive weight to all kernels.

8 Acknowledgements

We would like to acknowledge support from grants of the Russian Foundation for Basic Research 11-07-00409, 11-07-00728, 14-07-00661, and from UK EPSRC grant EP/F069626/1 (ACASVA).

References

1. Krogh, A., Larsson, B., von Heijne, G., Sonnhammer, E.L.L.: Predicting Transmembrane Protein Topology with a Hidden Markov Model: Application to Complete Genomes. J. Mol. Biol. 305, 567–580 (2001)
2. Chen, C.P., Rost B.: State-of-the-art in Membrane Protein Prediction. Applied Bioinformatics 1, 2135 (2002)
3. Gao, F.P., Cross, T.A.: Recent developments in membrane-protein structural genomics. Genome Biology 6:244 (2005)
4. Schölkopf, B., Tsuda K., Vert, J.-P. eds.: Kernel Methods in Computational Biology. MIT Press (2004)
5. Hofmann, T., Schölkopf B., Smola, A. J.: Kernel methods in machine learning. Ann. Statist. Volume 36, Number 3, 1171-1220 (2008)

6. Vapnik V.: Statistical Learning Theory. John-Wiley and Sons, Inc. (1998)
7. Pavlidis, P., Weston, J., Cai, J., Grundy, W.N.: Gene functional classification from heterogeneous data. In: Proceedings of the 5th Annual International Conference on Computational Molecular Biology. 242–248 (2001)
8. Lanckriet, G. et al.: A statistical framework for genomic data fusion. Bioinformatics, 20, 2626-2635 (2004)
9. Ong, C.S. et al.: Learning the kernel with hyperkernels. J. Mach. Learn. Res., 6, 1043–1071 (2005)
10. Bie, T. et all.:Kernel-based data fusion for gene prioritization. Bioinformatics, Vol. 23. 125–132 (2007)
11. Bach, F.R. et al.: Multiple kernel learning, conic duality, and the SMO algorithm. In Proceedings of the Twenty-first International Conference on Machine Learning (ICML04), Banff, Canada: Omnipress (2004)
12. Sonnenburg S., G. Rätsch, C. Schäfer, and B. Schölkopf. Large Scale Multiple Kernel Learning. Journal of Machine Learning Research, 7:1531–1565 (2006)
13. Hu, M., Chen, Y., Kwok, J.T.-Y.: Building sparse multiple-kernel SVM classi?ers, IEEE Transactions on Neural Networks 20 (5) 827–839 (2009)
14. Gönen, M., Alpayd, E.: Multiple Kernel Machines Using Localized Kernels. Proc. of PRIB (2009)
15. Gönen, M., Alpayd, E.: Localized algorithms for multiple kernel learning. Pattern Recognition 46, 795–807 (2013)
16. Liao, L.: Data Fusion with Optimized Block Kernels in LS-SVM for Protein Classification. Engineering, 5, 233–236 (2013)
17. Cortes, C., Mohri, M., Rostamizadeh, A.: Learning non-linear combinations of kernels. In Y. Bengio et al., editors, Advances in Neural Information Processing Systems 22, 396–404 (2009)
18. Mottl, V., Tatarchuk, A., Sulimova, V., Krasotkina, O., Seredin, O.: Combining pattern recognition modalities at the sensor level via kernel fusion. Proc. IW on MCS (2007)
19. Kloft, M., Brefeld, U., Sonnenburg, S. et al.: Efficient and accurate lp-norm multiple kernel learning . In Y. Bengio et al. editors, Advances in Neural Information Processing Systems 22, 997–1005. MIT Press, (2009).
20. Tatarchuk, A., Mottl, V., Eliseyev, A., Windridge, D.: Selectivity supervision in combining pattern-recognition modalities by feature- and kernel-selective Support Vector Machines. Proc. ICPR (2008)
21. Tatarchuk, A., Sulimova, V., Windridge, D., Mottl, V., Lange, M. : Supervised selective combining pattern recognition modalities and its application to signature verification by fusing on-line and off-line kernels. Proc. IW on MCS (2009)
22. Tatarchuk, A., Urlov, E., Mottl, V., Windridge, D.: A support kernel machine for supervised selective combining of diverse pattern-recognition modalities. In: El Gayar, N., Kittler, J., Roli, F. (eds.) MCS (2010)
23. Bradley P., Mangasarian O.: Feature selection via concave minimization and support vector machines. In International Conference on Machine Learning (1998)
24. Wang, L., Zhu, J., Zou, H.: The doubly regularized support vector machine. Statistica Sinica, 01/2006; 16:589–615 (2006)
25. De Groot, M.H.: Optimal Statistical Decisions. Wiley Classics Library (2004)
26. Mewes, H. W., Frishman, D., Gruber, C., Geier, B., Haase, D., Kaps, A., Lemcke K., Mannhaupt, G., Pfei?er, F., Schüller, C., et al.: MIPS: a database for genomes and protein sequences Nucleic Acids Research 28, 37–40 (2000)