# An Automated Combination of Kernels for Predicting Protein Subcellular Localization

Cheng Soon Ong[1,2] and Alexander Zien[1,3]

[1] Friedrich Miescher Laboratory, Tübingen, Germany
[2] Max Planck Institute for Biological Cybernetics, Tübingen, Germany
[3] Fraunhofer Institute FIRST, Berlin, Germany

**Abstract.** Protein subcellular localization is a crucial ingredient to many important inferences about cellular processes, including prediction of protein function and protein interactions. While many predictive computational tools have been proposed, they tend to have complicated architectures and require many design decisions from the developer.

Here we utilize the multiclass support vector machine (m-SVM) method to directly solve protein subcellular localization without resorting to the common approach of splitting the problem into several binary classification problems. We further propose a general class of protein sequence kernels which considers all motifs, including motifs with gaps. Instead of heuristically selecting one or a few kernels from this family, we utilize a recent extension of SVMs that optimizes over multiple kernels simultaneously. This way, we automatically search over families of possible amino acid motifs.

We compare our automated approach to three other predictors on four different datasets, and show that we perform better than the current state of the art. Further, our method provides some insights as to which sequence motifs are most useful for determining subcellular localization, which are in agreement with biological reasoning. Data files, kernel matrices and open source software are available at `http://www.fml.mpg.de/raetsch/projects/protsubloc`.

## 1 Introduction

Support vector machines (SVMs, e.g. [1]) are in widespread and highly successful use for bioinformatics tasks. One example is the prediction of the subcellular localization of proteins. SVMs exhibit very competitive classification performance, and they can conveniently be adapted to the problem at hand. This is done by designing appropriate kernel functions, which can be seen as problem-specific similarity functions between examples. The kernel function implicitly maps examples from their input space $\mathcal{X}$ to a space $\mathcal{H}$ of real-valued features (e.g. $\mathcal{H} = \mathbf{R}^d, d \in \mathbf{N} \cup \{\infty\}$) via an associated function $\Phi : \mathcal{X} \to \mathcal{H}$. The kernel function $k$ provides an efficient method for implicitly computing dot products in the feature space $\mathcal{H}$ via $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$.

Many different types of features have been used for SVM-based subcellular localization prediction. One popular class of features are compositions, i.e. histograms of subsequences. The most common choice of subsequences are single amino acids. One can also generalise this to pairs of adjacent amino acids, pairs of amino acids with

K.A. Crandall and J. Lagergren (Eds.): WABI 2008, LNBI 5251, pp. 186–197, 2008.

one position gap between them, pairs separated by two positions, and also for longer patterns [2]. A more widespread idea is to capture the statistics of signal peptides by computing compositions on relevant parts of a protein separately [3,4]. In section 2, we define kernels that generalize these ideas. Apart from compositions, further features include the search for known motifs [5,6] or PFAM domains [3], the use of PSI-BLAST profiles [7], and the use of PSI-BLAST similarities to other sequences [8]. In some cases, even SVMs or other classifiers are employed for feature generation or for motif detection [5,6].

When more than one set of features have been defined and computed, the task is to combine the evidence they yield into a single final prediction. This is often done in complex, hand-crafted architectures that frequently consist of two (or even more) layers of learning machines or decision systems. Instead, we utilize the novel multiclass multiple kernel learning (MCMKL) method [9], which optimally selects kernels from a given set and combines them into an SVM classifier (Section 3). Both are jointly applied to protein subcellular localization prediction in Section 4.

## 2 Motif Composition Kernels

### 2.1 Amino Acid Kernel and Motif Kernel

Before we consider motifs consisting of several amino acids, we define a kernel on individual amino acids (AAs). This will be useful as an ingredient to the more complex motif kernel. The AA kernel takes into account pairwise similarity of amino acids.

Let $\mathcal{A}$ be the set of 20 amino acids. A substitution matrix $M$ consists of a real-valued element $m_{ab}$ for each pair of amino acids $a$ and $b$. As substitution matrices are not in general valid kernel functions, we apply some transformations. It has been shown that every sensible substitution matrix $M$ implies a matrix $R$ of amino acid substitution probabilities via $m_{ab} = \frac{1}{\lambda} \log \frac{r_{ab}}{q_a q_b}$. Here $q_a$ is the so-called background probability of $a$, its relative frequency of appearance in any protein sequence. Given the constraints $\sum_a \sum_b r_{ab} = 1$ and $q_a = \sum_b r_{ab}$ and the symmetry of both $M$ and $R$, $R$ can be computed from $M$. We do so for the popular BLOSUM62 matrix [10] serving as $M$.

The elements of the obtained $R$, being substitution probabilities, are positive, and thus $R$ can be seen as a (complete) similarity graph between amino acids with weighted edges. From this we derive a positive definite kernel $k_1^{AA}$ on the amino acids by taking the graph Laplacian:

$$k_1^{AA}(a,b) = \sum_c r_{ac} - r_{ab} \ . \tag{1}$$

Note that other choices of kernels are possible. One alternative is the diffusion kernel, which is computed by taking the matrix exponential of a scalar multiple of $R$. In this context we prefer the graph Laplacian since it does not have any parameters to be adjusted. We extend the AA-kernel to $r$-tuples of amino acids ("motifs") by simply adding kernel values over the components. For $s, t \in \mathcal{A}^r$ we define the motif kernel

$$k_r^{AA}(s,t) = \sum_{i=1}^{r} k_1^{AA}(s_i, t_i) \ . \tag{2}$$

Note that these kernels cannot be direclty applied to variable-length protein sequences.

### 2.2 Motif Compositions

Previous work has shown that the amio acid composition (AAC) of a sequence is a useful basis for classifying its subcellular localization [11]. An advantage of this set of features is that it is robust with respect to small errors in the sequences, as may be caused by automated determination from genomic DNA. In subsequent work the AAC has been generalized in two directions.

First, instead of just considering the AAC of the entire protein sequence, it was calculated on different subsequences [3,6,12]. This is motivated by the fact that important indications of localization are not global. For example, the targeting of a protein to the mitochondrion or to the chloroplast is indicated by an N-terminal signal peptide with specific properties (for example pH or hydrophobicity) that are reflected by the AAC.

Second, it was noted that features which represent dependencies between two or more amino acids can increase the prediction performance. This seems plausible since there exist a number of (rather short) known motifs that are important for subcellular targeting. Examples include the C-terminal targeting signal for microbodies (SKL), the C-terminal endoplasmatic reticulum targeting sequence (KDEL), and the bipartite nuclear targeting sequence (which consists of five basic amino acids, R or K, in a certain arrangement). Existing prediction methods that generalize the AAC to higher order compositions do so in at least two ways: [2] and [8] use composition of pairs of amino acids, possibly with fixed-length gaps between them; [4] consider distributions of consecutive subsequences of length $r$, where a reduced size alphabet is used to avoid combinatorial explosion of the feature space for large $r$.

Here we carry the generalization a bit further, by allowing for patterns consisting of any number $r$ of amino acids in any (fixed) positional arrangement. For example, we could choose the frequencies of occurance of AA triplets with two positions gap between the first two and no gap between the second two, corresponding to a pattern (•,○,○,•,•). For any given pattern, we can compute the empirical distribution of corresponding motifs from a given AA sequence. This is a histogram of occurrences of each possible $r$-mer sequence. The example above will result in a histogram of all possible 3-mers where each sequence is represented by the counts of the occurrences of each 3-mer with the specified gap. Note that the combinatorial explosion of possible motifs for increasing order $r$ is not a real problem, because the number of motifs with positive probability is bounded by the protein length, and we employ sparse representations.

### 2.3 Motif Composition Kernels

The feature sets defined just above are histograms, and after normalization they are probability distributions over discrete sets. While we can use standard kernels (like the Gaussian RBF) on these data, this would neglect the fact that they are not arbitrary vectors, but in fact carry a special structure. Hence we use kernels that are specially designed for probability distributions [13]. These kernels have the added benefit of allowing us to model pairwise similarities between amino acids. To our knowledge, this is the first time such kernels have been applied to protein sequence analysis.

We use the Jensen-Shannon divergence kernel (corresponding to $\alpha = 1$ in [13]), which is based on a symmetric version of the Kullback-Liebler divergence of

information theory. Applied to histograms on patterns of order $r$ we have

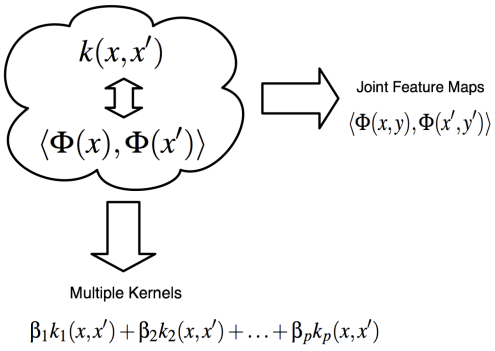
$$k_r^{JS}(p,q) = \sum_{s\in\mathcal{A}^r}\sum_{t\in\mathcal{A}^r} k_r^{AA}(s,t)\left(p(s)\log\frac{p(s)}{p(s)+q(t)} + q(t)\log\frac{q(t)}{p(s)+q(t)}\right) \quad , \tag{3}$$

where $p$ and $q$ are the $r$-mer histograms obtained from two sequences, and $s$ and $t$ run over the amino acid motifs. For this paper, we define the kernels between amino acids $k^{AA}(s,t)$ using the summed graph Laplacian defined in Equations (1) and (2).

Even using these choices, we are still left with a large number of possible patterns (as defined in Section 2.2) to consider. As linear combinations of kernels are again valid kernels [1], we could fix coefficients (e.g., uniform weights) and work with the resulting weighted sum kernel. However, it can be difficult to find optimal weights.

## 3 Multiclass Multiple Kernel Learning

Multiple kernel learning (MKL) is a technique for optimizing kernel weights $\beta_p$ in a linear combination of kernels, $k(\mathbf{x},\mathbf{x}') = \sum_p \beta_p k_p(\mathbf{x},\mathbf{x}')$. Thereby MKL is capable of detecting useless sets of features (noise) and eliminating the corresponding kernel (by giving it zero weight). Consequently MKL can be useful for identifying biologically relevant features [14,15]. In this paper we use the newly proposed multiclass extension (see Figure 1) of MKL, called multiclass multiple kernel learning, MCMKL [9].




Multiclass multiple kernel learning combines two generalizations of SVMs. The first is to learn more complex output spaces (including multiclass) than the simple binary classification task by extending the kernel to also encode classes (the horizontal arrow). The second is to simultaneously optimize over the coefficients in a linear combination of kernels, hence automatically weighting different feature spaces (the vertical arrow).

**Fig. 1.** The approach in [9] generalizes the idea of kernel machines in two directions

While binary SVMs have a single hyperplane normal $\mathbf{w}$ in feature space, multiclass SVMs (as considered here) have a different hyperplane normal $\mathbf{w}_u$ for each class $u$. Thus a trained MCMKL classifier has a separate confidence function

$$f_u(\mathbf{x}) = \left\langle \mathbf{w}_u\ , \sum_p \beta_p \Phi_p(\mathbf{x}) \right\rangle = \sum_i \alpha_{iu} \sum_p \beta_p k_p(\mathbf{x}_i,\mathbf{x}) \tag{4}$$

for each class $u$, where the latter equality derives from the expansion of the hyperplane normals $\mathbf{w}_u = \sum_i \alpha_{iu}\Phi(\mathbf{x}_i)$ (due to the Representer Theorem [1] or by Lagrange

dualization). The predicted class for a given example $\mathbf{x}$ will be chosen to maximize the confidence, that is $\hat{y}(\mathbf{x}) = \arg\max_u f_u(\mathbf{x})$.

Using an approach similar to that in [15], we convert the dual into an equivalent semi-infinite linear program (SILP) formulation by a second (partial) dualization.

$$\begin{aligned} \max_{\boldsymbol{\beta}} \theta \quad & \text{s.t.} \quad \forall \boldsymbol{\alpha} \in \mathcal{A}: \;\; \theta \leq \frac{1}{2} \sum_k \beta_k \left\| \mathbf{w}_k(\boldsymbol{\alpha}) \right\|^2 - \sum_i \alpha_{iy_i}, \\ & \textit{and} \quad \sum_{k=1}^{p} \beta_k = 1, \qquad \forall k: \;\; 0 \leq \beta_k \;\; , \end{aligned} \tag{5}$$

where

$$\mathcal{A} = \left\{ \boldsymbol{\alpha} \;\middle|\; \begin{array}{l} \forall i: 0 \leq \alpha_{iy_i} \leq C \\ \forall i: \forall u \neq y_i: \alpha_{iu} \leq 0 \\ \forall i: \sum_{u \in \mathcal{Y}} \alpha_{iu} = 0 \\ \forall u \in \mathcal{Y}: \sum_i \alpha_{iu} = 0 \end{array} \right\}$$

is the set of admissable parametrizations for the first constraint. For details on the proof see the Supplement and [9]. For a fixed $\boldsymbol{\beta}$, Equation (5) is a quadratic program (QP) which is only slightly more complicated than a standard SVM: it solves a direct multi-class SVM. Furthermore, for fixed $\boldsymbol{\alpha}$ the optimization problem in $\boldsymbol{\beta}$ is a linear program (LP). However, the constraint on $\theta$ has to hold for every suitable $\boldsymbol{\alpha}$, hence the name (refering to the infinitely many constraints).

We follow [15] and use a column generation strategy to solve (5): Solving the QP given by the constraints for a fixed $\boldsymbol{\beta}$ results in a particular $\boldsymbol{\alpha}$, which gives rise to a constraint on $\theta$ which is linear in $\boldsymbol{\beta}$. We alternate generating new constraints and solving the LP with the constraints collected so far (Figure 2). This procedure is known to converge [16,15]. For more details on this model and how it can be trained, that is how the values of the parameters $\alpha_{iu}$ and $\beta_p$ can be optimized, see [9]. Essentially the same model, though with a particular arrangement of kernels and a different optimization, is developed in [17]. Another related approach is described in [18].

## 4 Computational Experiments

To predict subcellular localization, we use motif kernels up to length 5 as defined in Section 2. Note that $20^5$ (3.2 million) different motifs of the form (•,•,•,•,•) exist; due to the Jensen-Shannon transformation (eq. 3) the feature spaces are even infinite dimensional. Apart from using the whole sequence, we compute the motif kernels on different sections of the protein sequences, namely the first 15 and 60 amino acids from the N-terminus and the 15 amino acids from the C-terminus (inspired by [6]). This results in $4 \times 2^{(5-1)} = 64$ motif kernels.

We augment the set of kernels available to the classifier by two small families based on features which have been shown to be useful for subcellular localization [19,20]. Using the pairwise E-value of BLAST as features, we compute a linear kernel, a Gaussian RBF kernel [1] with width 1000, and another Gaussian kernel with width 100000 from the logarithm of the E-value of BLAST. The second additional kernel family is derived

from phylogenetic profiles [21]. Using the results from their webserver (`http://apropos.icmb.utexas.edu/plex/`) as features, we compute a linear kernel and a Gaussian kernel of width 300. The Gaussian kernel widths were selected from a coarse grid by running MCMKL separately on each of the two kernel families; the range of the grid was inspired by the distribution of pairwise Euclidean distances of the feature vectors.

In total, we thus consider 69 candidate kernels. This renders manual selection and weighting tedious or even impossible, and thus calls for MKL. As in standard binary single-kernel SVMs, there is a parameter "C" in the MCMKL method to tune the regularization. We normalize each kernel such that setting $C = 1$ will at least be a reasonable order of magnitute (refer to [9] for details).

The subsequent protocol for all our experiments is as follows:

- Ten random splits into 80% training and 20% test data are prepared.
- For each training set, the parameter $C$ is chosen using 3-fold cross validation on the training set only. We search over a grid of values $C = \{1/27, 1/9, 1/3, 1, 3, 9, 27\}$. For all tasks, the best $C$ is chosen by maximizing the average F1 score on the validation (hold out) part of the training set. The F1 score is the harmonic mean of precision $p$ and recall $r$, $f1 = (2 * p * r)/(p + r)$. For more details, see the Supplement.
- Using the selected $C$, we train MCMKL on the full training set and predict the labels of the test set.

To compare with existing methods, we compute several different measures of performance. We assess the proposed an two different datasets, for which results with other methods are reported in the literature. The first is the data set used for training and evaluating TargetP [22]. The second dataset is a database of bacterial proteins, PSORTdb [5].

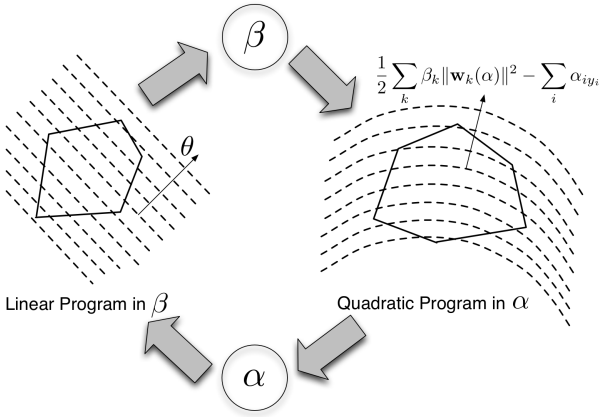



**Fig. 2.** The SILP approach to MKL training alternates between solving an LP for $\beta$ and a QP for $\alpha$ until convergence [9,15]

**Table 1.** Summary of comparative Protein Subcellular Localization results

| dataset name | performance measure | performance [%] of MCMKL | competitor | competing method, reference, year |
|---|---|---|---|---|
| TargetP plant | avg. MCC | $89.9 \pm 1.1$ | 86.0 | TargetLoc [6], 2006 |
| TargetP nonplant | avg. MCC | $89.7 \pm 0.8$ | 86.6 | TargetLoc [6], 2006 |
| PSORT+, 15.0% out | avg. Prec/Recall | 95.5 / 94.7 | 95.9 / 81.3 | PSORTb [5], 2004 |
| PSORT-, 13.3% out | avg. Prec/Recall | 96.4 / 96.3 | 95.8 / 82.6 | PSORTb [5], 2004 |
| PSORT- | avg. recall | 91.3 | 90.0 | CELLO II [20], 2006 |

A summary of the overall performance is shown in Table 1. Details of performance are available in the Supplement.[1]

### 4.1 Comparison on TargetP Dataset

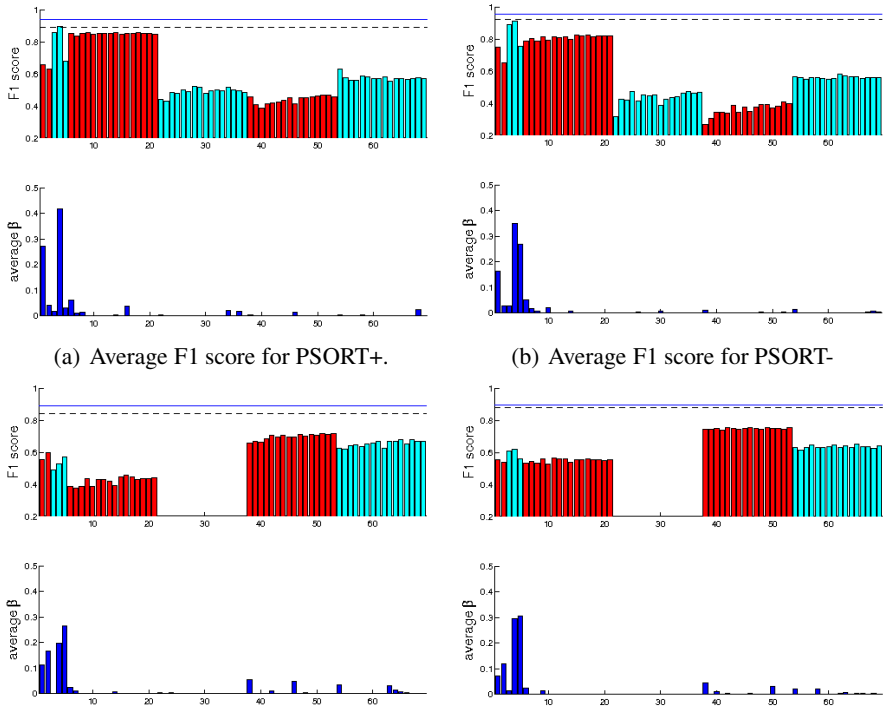
The original plant dataset of TargetP [22] is divided into five classes: chloroplast (ch), mitochondria (mi), secretory pathway (SP), cytoplasm (cy), and nucleus (nuc). However, in many reported results, cy and nuc are fused into a single class "other" (OT), and hence we do the same to enable direct comparison. Non-plant is similar, but lacks the chloroplasts. Each of the 10 random splits contains 21 cross validation optimizations (3-fold cross-validation to select from 7 values of $C$), with an additional larger optimization at the end. For the 3-class problem plant, the computation time for each split is roughly 10 hours on a 2.4Ghz AMD64 machine.

The results in [22] are reported in terms of Matthew's correlation coefficient (MCC). As can be seen in Table 1, the MCC values obtained with our proposed method are significantly better. Details for each class are shown in the Supplement. The features most often selected for classification as seen in Figures 3(c) and 3(d) are the kernels computed from BLAST E-values as well as phylogenetic information. The lists of *all* kernels selected are in the Supplement. From Table 2, we see the motif kernels which are most frequently selected. The motif kernel with pattern (•,○,○,○,○) only measures the simple amino acid composition. However, this encodes important global properties like protein mass or charge; it is thus reassuring to see that it gets selected. However, observe that several long patterns (•,○,○,○,•) and (•,•,○,○,•) are selected in the N-terminus region, indicating the presence of long meaningful subsequences in that region.

### 4.2 Comparison on PSORTdb Dataset

We also run computations on sequences and localizations of singly localized proteins in bacteria obtained from PSORTdb [5] and compare the performance to the prediction tool PSORTb v2.0 [5]. PSORTb v2.0 can withhold a prediction when it is uncertainabout the localization. From their supplementary website we estimate the

[1] The Supplement is freely available for download as `protsubloc-wabi08-supp.pdf` at `http://www.fml.tuebingen.mpg.de/raetsch/projects/protsubloc`




(a) Average F1 score for PSORT+.

(b) Average F1 score for PSORT-

(c) Average F1 score for plant

(d) Average F1 score for nonplant

**Fig. 3.** Average F1 score for various features. The horizontal axis indexes the 69 kernels. The two phylogenetic profile kernels and the three BLAST E-value kernels are on the left. The following four blocks are the motif kernels on: the whole sequence, the 15 AAs at the C-terminus, the 15 and 60 AAs at the N-terminus. In each subfigure, the bars in the lower panel display the average optimized kernel weight. The bars in the upper panels show the accuracy obtained from using each kernel just by itself. The dashed black line shows the performance when all the kernels are equally weighted (after appropriate normalization). The solid blue line shows the result of our method for comparison.

proportion of "unknown" predictions to be 15.0% for the Gram positive and 13.3% for Gram negative bacteria. We estimate probabilistic outputs from our method by applying the logistic transformation, that is $\hat{p}(y|\mathbf{x}) = \frac{\exp f_y(\mathbf{x})}{\sum_u \exp f_u(\mathbf{x})}$ to the SVM confidences. To obtain comparable figures, we discard the same fractions of most uncertain predictions, i.e. those with the lowest $\hat{p}(\hat{y}(\mathbf{x})|\mathbf{x})$. The mean and standard deviations on this reduced test set are reported in Table 1 in the comparison with PSORTb.

In 2004, PSORTb claimed to be the most precise bacterial localization prediction tool available [5]. However, as the results in Table 1 suggest, our method performs dramatically better. The performance assessments for various measures and for each class, which are reported in the Supplement, confirm this. The numbers show that our method in general matches the high precision of PSORTb and that we have extremely high recall levels, resulting in significantly better F1 scores for most localizations in

**Table 2.** Sequence motif kernels which have been selected at least 8 times out of the 10 splits for each dataset

| times selected | mean $\beta_k$ | kernel (PSORT+) |
|---|---|---|
| 10 | 6.23% | motif (•,○,○,○,○) on $[1, \infty]$ |
| 10 | 3.75% | motif (•,○,•,○,•) on $[1, \infty]$ |
| 9 | 2.24% | motif (•,○,•,•,•) on $[1, 60]$ |
| 10 | 1.32% | motif (•,○,○,○,•) on $[1, 15]$ |
| 8 | 0.53% | motif (•,○,○,○,○) on $[1, 15]$ |

| times selected | mean $\beta_k$ | kernel (plant) |
|---|---|---|
| 10 | 5.50% | motif (•,○,○,○,○) on $[1, 15]$ |
| 10 | 4.68% | motif (•,○,○,○,•) on $[1, 15]$ |
| 10 | 3.48% | motif (•,○,○,○,○) on $[1, 60]$ |
| 8 | 3.17% | motif (•,•,○,○,•) on $[1, 60]$ |
| 9 | 2.56% | motif (•,○,○,○,○) on $[1, \infty]$ |

| times selected | mean $\beta_k$ | kernel (PSORT-) |
|---|---|---|
| 10 | 5.04% | motif (•,○,○,○,○) on $[1, \infty]$ |
| 10 | 1.97% | motif (•,○,○,•,○) on $[1, \infty]$ |
| 9 | 1.57% | motif (•,•,○,○,○) on $[1, \infty]$ |
| 10 | 1.51% | motif (•,○,○,○,○) on $[1, 60]$ |
| 10 | 1.14% | motif (•,○,○,○,○) on $[1, 15]$ |
| 10 | 0.82% | motif (•,○,○,○,•) on $[-15]$ |

| times selected | mean $\beta_k$ | kernel (nonplant) |
|---|---|---|
| 9 | 4.48% | motif (•,○,○,○,○) on $[1, 15]$ |
| 10 | 3.23% | motif (•,○,○,•,•) on $[1, 15]$ |
| 9 | 2.32% | motif (•,○,○,○,○) on $[1, \infty]$ |
| 9 | 2.17% | motif (•,○,○,○,○) on $[1, 60]$ |
| 8 | 1.92% | motif (•,○,○,•,○) on $[1, 60]$ |
| 9 | 1.48% | motif (•,•,•,○,○) on $[1, \infty]$ |
| 8 | 0.94% | motif (•,○,•,○,○) on $[1, 15]$ |

both PSORT datasets. We also show in our Supplement that we are still competitive with PSORTb, even when predicting on all sequences (as opposed to withholding when unsure). Our method further compares favorably to CELLO II [20], for which results on the Gram negative bacterial protein set are published.

Similar to the motifs selected in the plant dataset, the BLAST E-values and phylogenetic profiles are important (Figure 3(a) and 3(b)). Note also that in all datasets, both the BLAST E-value as well as the log transformed version turn out to be useful for discrimination. This demonstrates one of the major dilemmas of using only one fixed kernel, as it may be possible that some transformation of the features may improve classification accuracy. Note that in Table 2 the motif (•,○,○,○,•) near the C-terminus, $[-15, \infty]$, has very little weight, and all other motifs are shorter. Indeed, in other experiments (results not shown), MCMKL with motifs up to length 4 performs equally well.

## 5 Discussion

First we note that our proposed method improves on established and state of the art methods for predicting protein subcellular localization. This is the case with respect to various figures of merit, which also demonstrates the robustness of our method. The success off our approach comes despite the fact that its design required little time and care: in contrast to complex competing methods, we only need to provide a sufficient set of candidate feature spaces (i.e. kernels) and do not have to worry about which one of them is best. In fact, Figures 3(a)-3(d) show that there is no single best kernel for all localization prediction tasks, and that kernel combinations can improve on each single kernel.

In our setting, the simple unweighted sum of all considered kernels reliably yields high accuracy. Note that this depends on the normalization of the kernels; while we use a heuristic, but justified scaling scheme, no theoretically optimal task-independent scaling method is known. However, we successfully use modern machine learning methods, specifically multiclass multiple kernel learning (MCMKL), to learn an optimal kernel weighting (the values $\beta_p$) for the given classification problem. Indeed the MCMKL reweighting consistently outperforms the plain normalization: it reduces the error (1 - score) by roughly 20%. We even used MCMKL to adjust real-valued kernel parameters by selecting them from a pre-specified coarse grid.

Note that the performance of the kernels taken by themselves is not a good indication of their weights in the optimized combination. For example in the plant experiments, motif kernels are best individually, but BLAST and phylogeny kernels obtain higher weights. We speculate that correlating information of the kernels is one reason for this: instead of choosing very similar kernels, MCMKL chooses a mixture of kernels that provide complementary information. Thus one should include as many diverse forms of information as possible. However, the weights of the kernels also depend on their prior scaling (normalization), and more machine learning research is necessary to fully understand this issue.

In addition to improving the accuracy, MCMKL also helps to understand the trained classifier. For example, in the plant data, the motif kernels on N-terminal subsequences (both length 15 and 60) provide the most informative feature spaces. The reason for this is most likely that they are best suited to detect the chloroplast and mitochondria transit peptides which are known to be in the N-terminal. For bacteria, which do not have organelles and corresponding signal peptides, the composition of the entire protein is more useful; probably because it conveys properties like hydrophobicity and charge. However, the BLAST kernels, which can pick up protein structure via remote homology, are assigned even higher weights. For the bacterial datasets, the BLAST kernels obtain more weight and perform better individually than the phylogenetic kernels. Phylogenetic profiles have only been shown to work well with organellar proteins [23], and the evidence in Figures 3(c) and 3(d) shows that they can help (slightly) for eukaryotes.

Since signal peptides are usually longer than 5 amino acids, our motifs may not capture all the information. However, one would expect that BLAST scores and phylogenetic profiles capture this information. This is reflected by the high weight given to these kernels (Figure 3). Note however that the localization signal may be distributed across the amino acid sequence, and only brought together by protein folding – the so called signal patches. If each component of the signal patches is relatively short, then this can be captured by our motif kernels.

While MKL did successfully identify relevant kernels (e.g., motif patterns), in this work we did not narrow it down to the specific features (i.e., the motifs). A promising goal for future work is to determine which particular motifs are most important; this can be modeled as an MKL task (cf. [15]). The idea of this approach is to represent the kernel by a sum of subkernels and to learn a weight (importance) for each of them. However, it seems that the imposed sparsity of the solution, while useful on the level of one kernel for each motif pattern, can be harmful at finer resolutions [15], and alternative approaches are more successful [24].

## 6 Summary and Outlook

We propose a general family of histogram-based motif kernels for amino acid sequences. We further propose to optimize over sets of kernels using a modern multiclass multiple kernel learning method, MCMKL [9]. We demonstrate that this approach outperforms the current state of the art in protein subcellular localization on four datasets. This high accuracy is already achieved with only using information from the amino acid sequence, while our method offers a principled way of integrating other data types. A promising example would be information mined from texts like paper abstracts [25]. Further, by selecting and weighting kernels, MCMKL yields interpretable results and may aid in getting insight into biological mechanisms.

Finally, the MCMKL framework [9] is very general and could be beneficial for a variety of (multiclass) bioinformatics prediction problems. For example, in this work we have only considered the case of singly located proteins, but in general proteins may exist in several possible locations in the cell. MCMKL does allow learning with multiple classes for each data point (each label $y$ would be the corresponding subset of classes), and it will be interesting to see its performance on multiply located proteins. Application to more different prediction tasks is facilitated by the large and increasing set of existing sequence and structure kernels. MCMKL also allows to guide the learning process with different types of prior knowledge, including the relationships of classes to each other (by a kernel on the classes, c.f. [9]). These exciting opportunities remain to be explored.

## Acknowledgement

We thank Alex Smola for stimulating discussions, Gunnar Rätsch for practical help with optimization with CPLEX, and Lydia Knüfing for proofreading the manuscript. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

## References

1. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press, Cambridge (2002)
2. Park, K.J., Kanehisa, M.: Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. Bioinformatics 19(13), 1656–1663 (2003)
3. Guda, C., Subramaniam, S.: TARGET: a new method for predicting protein subcellular localization in eukaryotes. Bioinformatics 21(21), 3963–3969 (2005)
4. Yu, C.-S., Lin, C.-J., Hwang, J.-K.: Predicting subcellular localization of proteins for gram-negative bacteria by support vector machines based on n-peptide compositions. Protein Science 13, 1402–1406 (2004)
5. Gardy, J.L., Laird, M.R., Chen, F., Rey, S., Walsh, C.J., Ester, M., Brinkman, F.S.L.: PSORTb v.2.0: expanded prediction of bacterial protein subcellular localization and insights gained from comparative proteome analysis. Bioinfomatics 21, 617–623 (2004)

6. Höglund, A., Dönnes, P., Blum, T., Adolph, H.-W., Kohlbacher, O.: MultiLoc: prediction of protein subcellular localization using N-terminal targeting sequences, sequence motifs, and amino acid composition. Bioinfomatics (2006)
7. Xie, D., Li, A., Wang, M., Fan, Z., Feng, H.: LOCSVMPSI: a web server for subcellular localization of eukaryotic proteins using SVM and profile of PSI-BLAST. Nucleic Acids Research 33, W105–W110 (2005)
8. Garg, A., Bhasin, M., Raghava, G.P.S.: Support vector machine-based method for subcellular localization of human proteins using amino acid composition, their order, and similarity search. The Journal of Biological Chemistry 280(15), 14427–14432 (2005)
9. Zien, A., Ong, C.S.: Multiclass multiple kernel learning. In: International Conference on Machine Learning (2007)
10. Henikoff, S., Henikoff, J.G.: Amino acid substitution matrices from protein blocks. In: Proceedings of the National Academy of Sciences, pp. 10915–10919 (1992)
11. Reinhardt, A., Hubbard, T.: Using neural networks for prediction of the subcellular location of proteins. Nucleic Acids Research 26, 2230–2236 (1998)
12. Cui, Q., Jiang, T., Liu, B., Ma, S.: Esub8: A novel tool to predict protein subcellular localizations in eukaryotic organisms. BMC Bioinformatics 5(66) (2004)
13. Hein, M., Bousquet, O.: Hilbertian metrics and positive definite kernels on probability measures. In: Cowell, R., Ghahramani, Z. (eds.) Proceedings of AISTATS 2005, pp. 136–143 (2005)
14. Lanckriet, G., De Bie, T., Cristianini, N., Jordan, M.I., Stafford Noble, W.: A statistical framework for genomic data fusion. Bioinfomatics 20(16), 2626–2635 (2004)
15. Sonnenburg, S., Rätsch, G., Schäfer, C.: A general and efficient multiple kernel learning algorithm. In: Neural Information Processings Systems (2005)
16. Hettich, R., Kortanek, K.O.: Semi-Infinite Programming: Theory, Methods, and Applications. SIAM Review 35(3), 380–429 (1993)
17. Lee, Y., Kim, Y., Lee, S., Koo, J.-Y.: Structured multicategory support vector machines with analysis of variance decomposition. Biometrika 93(3), 555–571 (2006)
18. Roth, V., Fischer, B.: Improved functional prediction of proteins by learning kernel combinations in multilabel settings. BMC Bioinformatics 8 (suppl. 2), 12 (2007)
19. Nair, R., Rost, B.: Sequence conserved for subcellular localization. Protein Science 11, 2836–2847 (2002)
20. Yu, C.-S., Chen, Y.-C., Lu, C.-H., Hwang, J.-K.: Prediction of protein subcellular localization. Proteins: Structure, Function and Bioinformatics 64(3), 643–651 (2006)
21. Pellegrini, M., Marcotte, E.M., Thompson, M.J., Eisenberg, D., Yeates, T.O.: Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. Proceedings of the National Academy of Sciences 96(8), 4285–4288 (1999)
22. Emanuelsson, O., Nielsen, H., Brunak, S., von Heijne, G.: Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. Journal of Molecular Biology 300, 1005–1016 (2000)
23. Marcotte, E.M., Xenarios, I., van der Bliek, A.M., Eisenberg, D.: Localizing proteins in the cell from their phylogenetic profiles. Proceedings of the National Academy of Sciences 97(22), 12115–12120 (2000)
24. Zien, A., Sonnenburg, S., Philips, P., Rätsch, G.: POIMS: Positional Oligomer Importance Matrices – Understanding Support Vector Machine Based Signal Detectors. In: Proceedings of the 16th International Conference on Intelligent Systems for Molecular Biology (2008)
25. Höglund, A., Blum, T., Brady, S., Dönnes, P., San Miguel, J., Rocheford, M., Kohlbacher, O., Shatkay, H.: Significantly improved prediction of subcellular localization by integrating text and protein sequence data. In: Pacific Symposium on Biocomputing, pp. 16–27 (2006)