An Automated Combination of Kernels for Predicting Protein Subcellular Localization

Alexander Zien
Friedrich Miescher Lab., Tübingen and
Fraunhofer FIRST, Berlin, Germany
alexander.zien@tuebingen.mpg.de

Cheng Soon Ong
Max Planck Inst. for Biol. Cybernetics, and
Friedrich Miescher Lab., Tübingen, Germany
chengsoon.ong@tuebingen.mpg.de

Abstract

Protein subcellular localization is a crucial ingredient to many important inferences about cellular processes, including prediction of protein function and protein interactions. We propose a new class of protein sequence kernels which considers all motifs including motifs with gaps. This class of kernels allows the inclusion of pairwise amino acid distances into their computation. We utilize an extension of the multiclass support vector machine (SVM) method which directly solves protein subcellular localization without resorting to the common approach of splitting the problem into several binary classification problems. To automatically search over families of possible amino acid motifs, we optimize over multiple kernels at the same time. We compare our automated approach to four other predictors on three different datasets, and show that we perform better than the current state of the art. Furthermore, our method provides some insights as to which features are most useful for determining subcellular localization, which are in agreement with biological reasoning.

1 Introduction

Support vector machines (SVMs) are nowadays 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 to represent prior knowledge about the similarity between the examples of the problem at hand. The kernel function implicitly maps examples from their input space $\mathcal{X}$ to a space $\mathcal{H}$ of real-valued features (e.g. $\mathcal{H} = \mathbf{R}^d$) via an associated function $\Phi : \mathcal{X} \to \mathcal{H}$. The kernel function k provides an efficient short-cut for computing dot products in $\mathcal{H}$ 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 the subsequences are single amino acids. Beyond that, [1] uses pairs of adjacent amino acids, pairs of amino acids with one position gap between them, pairs separated by two positions, and so on. A more widespread idea is to compute compositions on relevant parts of a protein separately [2, 3], for example in order to capture the statistics of signal peptides. In section 2, we define kernels that generalize these ideas. Apart from compositions, further features include the search for known motifs [4, 5] or PFAM domains [2], the use of PSI-BLAST profiles [6], and the use of PSI-BLAST similarities to other sequences [7]. In some cases, even SVMs or other classifiers are employed for feature generation or for motif detection [4, 5].

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. We instead utilize the novel multiclass multiple kernel learning (MCMKL) method [8], 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 Kernels on amino acids

First we define the kernel on individual 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. It has been shown that every sensible symmetric substitution matrix M implies a matrix R of amino acid substitution probabilities via $m_{ab} = \frac{1}{\lambda} \log \frac{r_{ab}}{q_a q_b}$. Given the constraints $q_a = \sum_b r_{ab}$ and $\sum_a \sum_b r_{ab} = 1$ and the symmetry of both M and R, R can be computed from M. We do so for BLOSUM62 as M.

The obtained R can be seen as a (complete) similarity graph between amino acids with weighted edges (with positive weights). From this we derive a positive definite kernel function k^{AA} on the amino acids by taking the graph Laplacian $k^{AA}(a, b) = \sum_c r_{ac} - r_{ab}$. Note that other choices of kernels are possible, e.g. the diffusion kernel, which is 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 k-tuples of amino acids by simply adding kernel values over the components. For $s, t \in \mathcal{A}^k$ we define $k^{AA}(s, t) = \sum_{i=1}^{k} k^{AA}(s_i, t_i)$.

2.2 Motif compositions

Previous work has shown that the amio acid composition (AAC) of a sequence is useful for classifying its subcellular localization. In subsequent work the AAC has been refined in two directions. First, instead of just considering the AAC of the entire protein sequence, it was calculated on different subsequences [2, 5]. This makes sense since 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 AAC properties (eg, pH or hydrophobicity). Second, it was noted that features corresponding to more than a single amino acid can increase the prediction performance. This seems plausible since there exist a number of (rather short) relevant motifs, e.g. the C-terminal targeting signal for microbodies (SKL) and the C-terminal endoplasmatic reticulum targeting sequence (KDEL). [1, 7] use composition of pairs of amino acids, possibly with fixed-length gaps between them. [3] consider distributions of k-length subsequences, where a reduced size alphabet is used to attenuate the combinatorial explosion of the implied feature space.

Here we carry the generalization a bit further, by allowing for patterns consisting of any number k 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. Thus each sequence is represented by a histogram of the counts of occurrences of each k-mer with the specified gap.

2.3 Motif composition kernels

After normalization, the compositions are probability distributions over discrete sets. While general purpose kernels (like the Gaussian RBF) do not take this into account, kernels specially designed for such data do exist [9]. These kernels have the added benefit of allowing us to model pairwise similarities between amino acids.

We use the Jensen-Shannon divergence kernel (corresponding to $\alpha = 1$ in [9]), which is based on a symmetric version of the Kullback-Liebler divergence. Applied to histograms on patterns of order k we have

$$k^{JS}(p, q) = \sum_{s \in \mathcal{A}^k} \sum_{t \in \mathcal{A}^k} k^{AA}(s, t) \left(p(s) \log \frac{p(s)}{p(s) + q(t)} + q(t) \log \frac{q(t)}{p(s) + q(t)} \right), \qquad (1)$$

where p and q are the histograms corresponding to two sequences, and s and t are the amino acid motifs that the distributions range over. For the amino acids kernels $k^{AA}(s, t)$ we use the summed graph Laplacian defined in Section 2.1. Note that the combinatorial explosion of possible motifs for increasing order k is no real problem: we efficiently compute the kernels by employing sparse representations, as the number of motifs with positive probability is bounded by the protein length.

3 Multiclass Multiple Kernel Learning

For the problem of protein subcellular localization, various sources of information may contribute to an accurate predictor. There exist three possible strategies of dealing with this situation:

- Concatenating all feature sets into a single feature vector to be used in a single Gaussian RBF kernel [6].
- Defining an individual kernel on each type of features, training an individual SVM on each kernel, and combining the SVM outputs or predictions, e.g. by a jury method or by another SVM [4, 7, 2, 5, 1, 3].
- Defining an individual kernel on each set of features, combining them into a single kernel (for instance, by adding them), and training a single SVM on that kernel.

For simplicity we describe all three strategies solely for the SVM setting, although the first two of them similarly apply to any other machine learning method as well. The third strategy has empirically been shown to be the most effective [10]. It allows to build complex modular kernel functions by adding several simpler ones. One difficulty with adding kernels is that doing so with uniform weights does not always yield optimal accuracy. Multiple kernel learning (MKL) is a technique for optimizing kernel weights β_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 eliminiting the corresponding kernel (by giving it zero weight) [11]. Consequently MKL can also be useful for identifying biologically relevant features [12].

In this paper we use the newly proposed multiclass extension of MKL, called MCMKL [8]. While binary SVMs have a single hyperplane normal $\mathbf{w}$ in feature space, multiclass SVMs have a different hyperplane normal $\mathbf{w}_u$ for each class u [13]. Thus a trained MCMKL classifier has a separate confidence function

$$f_u(\mathbf{x}) = \left\langle \mathbf{w}_u, \sum_p \beta_p \Phi_p(\mathbf{x}) \right\rangle = \sum_i \alpha_{iu} \sum_p \beta_p k_p(\mathbf{x}_i, \mathbf{x}) \tag{2}$$

for each class u, where the latter equality derives from the expansion of the hyperplane normals $\mathbf{w}_u = \sum_i \alpha_{iu} \Phi(\mathbf{x}_i)$ (cf. the Representer Theorem). The predicted class for a given example $\mathbf{x}$ will be chosen to maximize the confidence, i.e. $y = \arg\max_u f_u(\mathbf{x})$. For more details on this model and how it can be trained, i.e. how the values of the parameters α_{iu} and β_p can be optimized, see [8].

4 Accurate Prediction of Subcellular Localization

To predict subcellular localization, we use motif kernels up to length 5 as defined in Section 2. 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. This results in $4 \times 2^{(5-1)} = 64$ motif kernels. Note that it is not possible to evaluate the huge number of patterns available to the motif kernels with traditional methods, hence using kernels is crucial.

We augment the set of kernels available to the classifier by two small families. Using the pairwise E-value of BLAST as features, we compute a linear kernel, a Gaussian RBF kernel with width 1000, and from the logarithm of the E-value of BLAST another Gaussian kernel with width 100,000. The second additional kernel family derives from phylogenetic profiles [14]. Using the results from their webserver[1] as features, we compute a linear kernel and a Gaussian kernel of width 300. The kernel widths have been selected from a coarse grid by running MCMKL separately on each of the two kernel families. In total, we thus consider 69 candidate kernels. This renders manual selection and combination 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 apply a scaling to each kernel such that the choice $C = 1$ will at least be in a reasonable order of magnitute (cf. [8] for details). The subsequent protocol for all our experiments is as follows: (a) Ten random splits into 80% training and 20% test data are prepared. (b) 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 F1 score on the validation (hold out) part of the training set. (c) Using the selected C, we train MCMKL on the full training set and predict the labels of the test set.

4.1 Comparison on TargetP dataset

The original plant dataset of TargetP [15] is divided into five classes: chloroplast (ch), mitochondria (mi), secretory pathway (SP), cytoplasm (cy), and nucleus (nuc). However, in many reported results,

[1] `http://apropos.icmb.utexas.edu/plex/`

sel.	avg. β_k	kernel
10	26.49%	RBF on log BLAST E-value, σ=10^5
10	19.74%	RBF on BLAST E-value, σ=10^3
10	16.54%	RBF on inv phyl. profs, σ=300
10	11.19%	RBF on lin phyl. profs, σ=1
10	5.51%	motif (●,○,○,○,○) on $[1, 15]$
10	4.66%	motif (●,○,○,○,●) on $[1, 15]$
10	3.52%	motif (●,○,○,○,○) on $[1, 60]$
9	3.38%	motif (●,●,○,○,●) on $[1, 60]$
9	2.58%	motif (●,○,○,○,○) on $[1, \infty]$
5	1.32%	motif (●,○,●,○,●) on $[1, 60]$
7	1.06%	motif (●,○,○,●,○) on $[1, 15]$
7	0.93%	motif (●,●,○,○,○) on $[1, \infty]$
5	0.62%	motif (●,○,○,○,●) on $[1, \infty]$
3	0.52%	motif (●,●,●,○,●) on $[1, 60]$
2	0.41%	motif (●,○,○,●,●) on $[1, 60]$
6	0.40%	motif (●,○,●,○,○) on $[-15, \infty]$
7	0.27%	motif (●,○,○,○,○) on $[-15, \infty]$
3	0.26%	motif (●,○,●,○,●) on $[1, 15]$
2	0.18%	motif (●,○,○,●,○) on $[1, 60]$
3	0.12%	linear kernel on BLAST E-value
2	0.12%	motif (●,○,○,●,●) on $[1, 15]$
2	0.10%	motif (●,○,●,○,●) on $[-15, \infty]$
1	0.06%	motif (●,●,●,○,●) on $[-15, \infty]$
1	0.03%	motif (●,●,○,○,○) on $[1, 60]$
1	0.02%	motif (●,●,○,○,●) on $[1, 15]$

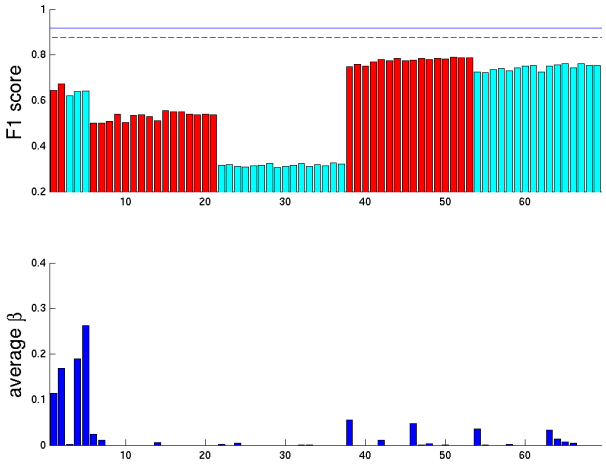

Figure 1: Kernel usage in the ten repetitions of experiments on the plant dataset. **(Left)** Kernels sorted by importance as indicated by the averaged coefficient β_k (middle column). The leftmost column states the number of repetitions in which the kernel is selected. The right column describes the kernel; for motif kernels, this includes the pattern associated with the kernel and the considered region of the protein ([1,15] and [1,60] denotes the first 15 and 60 amino acids from the N-termimus, [-15,∞] denotes the last 15 amino acids from the C-terminus, and [1,∞] denotes the whole protein sequence). **(Right)** Kernels grouped by family: the x-axis denotes the index of each of the 69 kernels. Left to right: phylogenetic tree kernels, BLAST E-value kernels, motif kernels on entire sequence, C-terminal 15 amino acids, and 15 and 60 N-terminal amino acids. The top bar plot shows the F1 score achieved when using the single kernel. The dashed black line shows the performance when all the kernels are equally weighted (after normalization of the variance in feature space). The solid blue line shows the result of our method for comparison. The bottom bar plot shows the corresponding MCMKL weights.

cy and nuc are fused into a single class "other" (OT), and hence we do the same to enable direct comparison. Overall, we obtain an average MCC of $89.1 \pm 1.2\%$, which is significantly better than the average MCC of TargetP [15] and TargetLoc [5], which obtain 79% and 85.3% respectively.

The features most often selected for classification as seen in Figure 1 are the kernels computed from BLAST E-values as well as phylogenetic information. The motif kernel (●,○,○,○,○) only measures the amino acid composition. It is reassuring to see that the amino acid composition is selected. However, observe that several long motifs, (●,○,○,○,●), (●,●,○,○,●) and (●,○,●,○,●) are selected in the N-terminus region, indicating the presence of long meaningful subsequences in that region. Also note that only a small fraction of kernels obtained a positive weight in any repetition, and that this selection is very consistent across the repetitions.

4.2 Comparison on PSORTdb dataset

PSORTb [4] claims to be the most precise bacterial localization prediction tool available. To compare our method with PSORTb, we use sequences and localizations of proteins in bacteria as obtained from PSORTdb [4]. We only consider singly localized proteins. PSORTb has the option of withholding a prediction when it is uncertain about the localization. Based on their supplementary website, we estimate the proportion of "unknown" predictions to be 13% for the Gram positive bacteria. For the performance comparison, we compute probabilistic outputs from our method by using the softmax function, that is $\hat{p}(u|\mathbf{x}) = \frac{\exp(f_u(\mathbf{x}))}{\sum_v \exp(f_v(\mathbf{x}))}$ for each example $\mathbf{x}$ and each class u, and then discard the same fraction of most uncertain predictions. The results are the mean and standard deviations of this reduced test set. For gram negative bacteria, we use only sequence motif kernels up to length 4. Overall we obtain an average F1 score on PSORT+ and PSORT- of $93.8 \pm 1.3\%$ and $96.1 \pm 0.6\%$ respectively, which is significantly better than the 90.0% and 87.5% of PSORT.

Detailed results are shown in Figure 2. Similar to the motifs selected in the plant dataset, the BLAST E-values and phylogenetic profiles are important. Note also that in both datasets, both the BLAST E-value as well as the log transformed version turns out to be useful for discrimination. This demon-

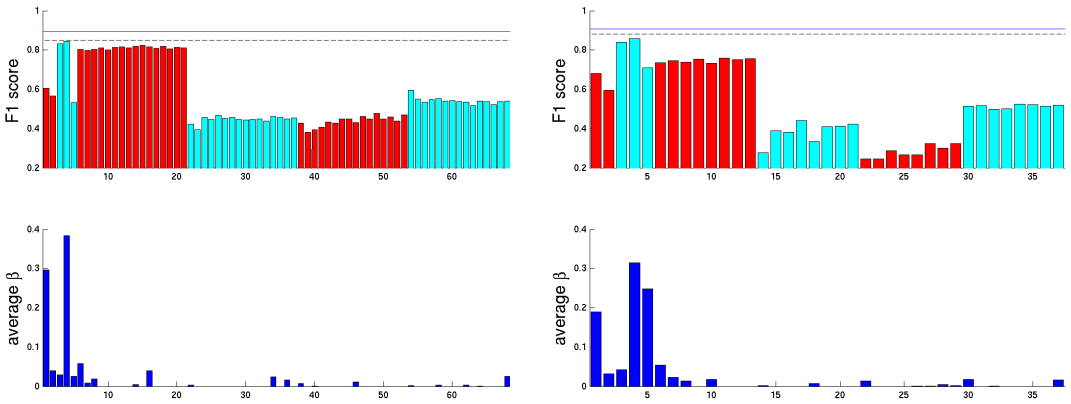

Figure 2: Kernel usage in the ten repetitions of experiments on the PSORT datasets.

strates one of the major dilemmas of using only one fixed kernel: it may be possible that some transformation of the features may improve classification accuracy. In contrast to the plant motifs, there are many more motifs selected in the C-terminus of the protein. Again, long motifs are selected in addition to the amino acid composition motif.

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 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). Selecting the relevant features is taken care of by modern machine learning methods, specifically multiclass multiple kernel learning (MCMKL); apparently very successfully. We even used MCMKL to select kernel parameters.

Figures 1 and the results for PSORT show that there is no single best kernel for all localization prediction tasks. They also show that the simple unweighted sum of all considered kernels reliably yields high accuracy. Note that this relies on the scaling of the kernels; while we use a heuristic, but justified scaling scheme, no theoretically optimal task-independent scaling methods is known. In fact, it is the goal of MCMKL to learn an optimal scaling (the values β_p), and indeed MCMKL consistently outperforms the uniform weighting: it reduces the error (1 - F1 score) by roughly 20%.

In addition to improving the accuracy, MCMKL also helps to understand the trained classifier. For example, in the plant data, the motif kernels on C-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 C-terminal chloroplast signal peptides. A promising goal for future work is determine which particular motifs are most important; in principle, this is an MKL task (cf. [12]). For the bacteria, which do not have chloroplasts 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 yield even better accuracy.

Note that the performance of the kernels taken by themselves is not a good indication of its weight in the optimized combination. For example in the plant experiments, motif kernels are individually best, 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 scaling; more machine learning research is necessary to fully understand this issue.

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). We demonstrate that this approach outperforms the current state of the art in protein subcellular localization on three datasets. Further, while the high accuracy is already achieved with

only using information from the amino acid sequence, our method also offers a principled and convenient way of integrating other data types.

As has been shown before, MKL can be used to identify individual features that are relevant to the classification. This makes the results interpretable and may aid in getting insight into biological mechanisms. 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. This may help to identify important motifs.

Finally, the MCMKL approach is very general, and could be beneficial for other multiclass bioinformatics prediction problems. This is eased by the large and increasing set of existing sequence and structure kernels. MCMKL also allows to guide the learning process with prior knowledge on the relationships of classes to each other. These exciting opportunities remain to be explored.

Acknowledgement

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] K. J. Park and M. Kanehisa. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, 19(13):1656–1663, Sep 2003.

[2] C. Guda and S. Subramaniam. TARGET: a new method for predicting protein subcellular localization in eukaryotes. *Bioinformatics*, 21(21):3963–3969, 2005.

[3] C.-S. Yu, C.-J. Lin, and J.-K. Hwang. Predicting subcellular localization of proteins for gram-negative bacteria by support vector machines based on n-peptide compositions. *Protein Science*, 13:1402–1406, 2004.

[4] J. L. Gardy, M. R. Laird, F. Chen, S. Rey, C. J. Walsh, M. Ester, and F. S. L. Brinkman. PSORTb v.2.0: expanded prediction of bacterial protein subcellular localization and insights gained from comparative proteome analysis. *Bioinfomatics*, 21:617–623, 2004.

[5] A. Höglund, P. Dönnes, T. Blum, H.-W. Adolph, and O. Kohlbacher. MultiLoc: prediction of protein subcellular localization using N-terminal targeting sequences, sequence motifs, and amino acid composition. *Bioinfomatics*, 2006.

[6] D. Xie, A. Li, M. Wang, Z. Fan, and H. Feng. LOCSVMPSI: a web server for subcellular localization of eukaryotic proteins using SVM and profile of PSI-BLAST. *Nucleic Acids Research*, 33:W105–W110, 2005.

[7] A. Garg, M. Bhasin, and G. P.S. Raghava. 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.

[8] A. Zien and C.S. Ong. Multiclass multiple kernel learning. In *International Conference on Machine Learning*, 2007.

[9] M. Hein and O. Bousquet. Hilbertian metrics and positive definite kernels on probability measures. In R. Cowell Ghahramani, Z., editor, *Proceedings of AISTATS 2005*, pages 136–143, 2005.

[10] P. Pavlidis, J. Weston, J. Cai, and W. S. Noble. Learning gene functional classifications from multiple data types. *Journal of Computational Biology*, 9(2):401–411, 2002.

[11] G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. Stafford Noble. A statistical framework for genomic data fusion. *Bioinfomatics*, 20(16):2626–2635, 2004.

[12] S. Sonnenburg, G. Rätsch, and C. Schäfer. A general and efficient multiple kernel learning algorithm. In *Neural Information Processings Systems*, 2005.

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

[14] Matteo Pellegrini, Edward M. Marcotte, Michael J. Thompson, David Eisenberg, and Todd O. Yeates. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proc. Natl. Acad. Sci.*, 96(8):4285–4288, 1999.

[15] O. Emanuelsson, H. Nielsen, S. Brunak, and G. von Heijne. Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *Journal of Molecular Biology*, 300:1005–1016, 2000.