1 A kernel for protein secondary structure prediction

Yann Guermeur
LORIA - CNRS
Campus Scientifique, BP 239
54506 Vandœuvre-lès-Nancy cedex, France
Yann.Guermeur@loria.fr

Alain Lifchitz
LIP6 - CNRS
8, rue du Capitaine Scott
75015 Paris, France
Alain.Lifchitz@lip6.fr

Régis Vert
LRI, Bâtiment 490
Université Paris-Sud
91405 Orsay cedex, France
Regis.Vert@lri.fr

Multi-class support vector machines have already proved efficient in protein secondary structure prediction as ensemble methods, to combine the outputs of sets of classifiers based on different principles. In this chapter, their implementation as basic prediction methods, processing the primary structure or the profile of multiple alignments, is investigated. A kernel devoted to the task is introduced, which incorporates high-level pieces of knowledge. Initial experimental results illustrate the potential of this approach.

1.1 Introduction

Knowing the structure of a protein is a prerequisite to gain a thorough understanding of its function. The large-scale sequencing projects which have multiplied in the last years have produced an exploding number of protein sequences. Unfortunately, the number of known protein structures has not increased in the same proportion. Indeed, the experimental methods available to determine the 3D structure, X-ray crystallography and nuclear magnetic resonance (NMR), are highly labour intensive and do not ensure the production of the desired result (e.g., some proteins simply do not crystallize). As a consequence, predicting the tertiary structure of proteins *ab initio*, i.e., starting from their sequences, has become one of the most challenging problems in structural biology. In the sixties, Anfinsen proposed his "Thermodynamic Hypothesis" (Anfinsen et al., 1963), which implies that there is sufficient information contained in the protein sequence to guarantee correct folding from any of a large number of unfolded states. In other words, the problem of interest can theoretically be solved. However, due to its practical difficulty, highlighted for instance in (Karplus and Petsko, 1990), it is seldom tackled directly, but rather through a divide and conquer approach. In that context, a useful intermediate step consists in predicting first the secondary structure, which is a way to simplify the prediction problem by projecting the very complicated 3D structure onto one dimension, i.e., onto a string of secondary structural assignments for each residue (amino acid). Protein secondary structure refers to regular, repeated patterns of folding of the protein backbone. The two most common folding patterns are the *alpha helix* and the *beta strand*. Figure 1.1 is a schematic representation of the secondary structure of the protein G (Derrick and Wigley, 1994), which was obtained with the RASMOL software (Sayle and Milner-White, 1995). This structure has two main parts: an alpha helix, in red on the figure, and a *beta sheet* made up of four strands, in yellow. From the point of view of pattern recognition, protein secondary structure prediction can be seen as a 3-class discrimination task, which consists in assigning to each residue of a sequence its conformational state, either α-helix, β-strand or aperiodic (coil). People have started working on this problem as early as in the late sixties. Since then, almost all the main families of machine learning methods have been assessed on it. Currently, the best prediction methods are connectionist architectures (see (Rost and O'Donoghue, 1997; Baldi and Brunak, 2001; Rost, 2001) for surveys).

Although kernel methods have already found many applications in bioinformatics, as can be seen in the other chapters of this book, to the best of our knowledge, they have only been applied to protein secondary structure prediction twice. In (Hua and Sun, 2001), the authors implemented different combinations of bi-class SVMs to perform the prediction from alignment profiles generated by BLAST. In (Guermeur, 2002; Guermeur et al., 2004), different multi-class SVMs (M-SVMs) were used to combine several prediction methods, as well as the modules (BRNNs) of the current best prediction method, SSpro (Baldi et al., 1999; Pollastri et al., 2002). In both cases, the experimental results appeared promising, which was all the more satisfactory that only standard kernels were used. In this chapter, we build on these initial works, introducing a M-SVM devoted to the prediction of the secondary structure from the primary structure, or profiles of multiple alignments.

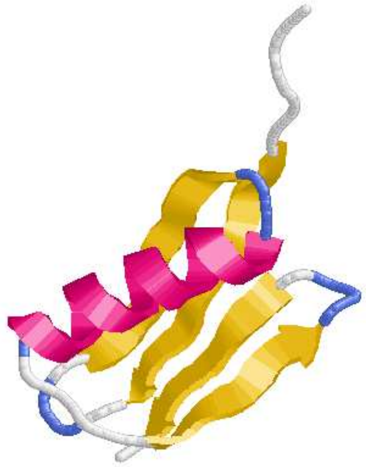

Figure 1.1 Schematic representation of the structural elements of the protein G.

Its originality rests in the nature of its kernel, designed to exploit expert knowledge on the task. The parameterization of this kernel makes use of an original extension of the principle of kernel alignment (Cristianini et al., 2001, 2002) to the multi-class case. Once more, experimental results appear promising, as they highlight the relevance of the specification performed. In short, our M-SVM, adequately dedicated, proves superior to a multi-layer perceptron in all the contexts where the latter is incorporated in the prediction methods.

The organization of the paper is as follows. Section 1.2 provides a short introduction to M-SVMs, as well as a description of the standard local approach implemented to predict the secondary structure, approach based on the use of a sliding window. The main part of our contribution, the specification and parameterization of a kernel exploiting this input, is discussed in Section 1.3. This section details our extension of the kernel alignment. Last, the resulting machine is assessed in Section 1.4, where it is compared with a multi-layer perceptron.

1.2 Multi-class SVMs for Protein Secondary Structure Prediction

This section introduces the M-SVMs, and the general principle of their implementation for protein secondary structure prediction.

1.2.1 M-SVMs

In the early days of the development of the SVM method, multi-class discrimination was implemented with bi-class machines, through decomposition schemes. The first of them was the so-called *one-against-the-rest* or *one-per-class* approach (Schölkopf et al., 1995; Vapnik, 1995). Later on came the *pairwise-coupling* decomposition scheme (Mayoraz and Alpaydin, 1998; Weston and Watkins, 1998). The first multi-class SVM, more precisely the first SVM algorithm devoted to a multivariate affine architecture, was the k-class SVM proposed independently by Vapnik and Blanz (Vapnik, 1998), Weston and Watkins (1998), and Bredensteiner and Bennett (1999), among others. Alternative possibilities were then investigated in (Crammer and Singer, 2001; Lee et al., 2001; Hsu and Lin, 2002; Guermeur, 2002). In (Guermeur et al., 2005), all these machines were endowed with a unifying theoretical framework.

All the M-SVMs share the same architecture, which corresponds, in the feature space, to a multivariate affine model. This is expressed formally below. Let $\mathcal{X}$ be the space of description (input space), $Q \geq 3$ the number of categories, k the Mercer kernel used and Φ a map into the feature space $\mathcal{F}$ induced by k. Let F be the set of vector-valued functions $f = [f_j]$, $(1 \leq j \leq Q)$, from $\mathcal{X}$ into $\mathbb{R}^Q$, computed by the M-SVMs. We have then precisely :

$$\forall \mathbf{x} \in \mathcal{X}, \ \ \forall j \in \{1, \dots, Q\}, \ \ f_j(\mathbf{x}) = \langle \mathbf{w}_j, \Phi(\mathbf{x}) \rangle + b_j$$

Thus, each category is associated with one hyperplane, and the discriminant function computed is obtained by application of the standard max rule: a pattern $\mathbf{x}$ is assigned to the category C_{j^*} satisfying $j^* = \operatorname{argmax}_j \{f_j(\mathbf{x})\}$. In its primal formulation, training thus amounts to finding the optimal values of the couples $(\mathbf{w}_j, b_j)$, $(1 \leq j \leq Q)$, for a given choice of the kernel k and the *soft margin* constant C. In the bi-class case, this choice is performed so as to maximize the (soft) margin. Strangely enough, in the first papers dealing with M-SVMs, the algorithms proposed were not related, at least explicitly, to the maximization of some notion of margin. This could be due to the fact that the standard pathways used to express the fat-shattering dimension of a SVM in terms of the constraints on the norm of $\mathbf{w}$, such as the use of a Rademacher's sequence (see for instance (Bartlett and Shawe-Taylor, 1999; Cristianini and Shawe-Taylor, 2000; Gurvits, 2001)), do not extend nicely to the multi-class case. Indeed, our efforts to endow the M-SVMs with the standard results derived in the framework of the theory of large margin classifiers (Guermeur et al., 2005) call for additional work. To the best of our knowledge, the only other study on the generalization capabilities of M-SVMs involving an extended notion of margin is reported in (Crammer and Singer, 2001). As usual, the corresponding quadratic programming (QP) problem is solved in its Wolfe dual form (Fletcher, 1989). Several algorithm can be applied to perform the optimization. Our software of the k-class SVM, used in the experiments described below, and available through the web site of kernel machines[1], implements a variant of the Frank-Wolfe algorithm (Frank and Wolfe, 1956) which includes a decomposition method (see also (Elisseeff, 2000)).

1.2.2 Selection of the predictors

The standard way to perform protein secondary structure prediction with statistical discriminant methods consists in applying a local approach. Precisely, the predictors used to predict the conformational state of a given residue are the amino acids contained in a window of fixed size centered on this residue. To code the content of each position in the window, a vector of 22 components is used. Each of the 20 first components correspond to a specific amino acid (there are 20 of them), whereas the 2 remaining ones are used to take into account unknown amino acids, usually designed by an `'X'` in the databases, as well as empty positions in the window. Empty positions occur when the window extends over the N terminus or the C terminus of the chain of interest. In short, the coding used to represent the window content is the standard orthonormal one, which induces no correlation between the symbols of the alphabet. What appears *a priori* as an advantage is an inconvenience here, as will be pointed out below. Given a window size $|W| = 2n + 1$ (typically, n will range from 5 to 10), the number of predictors is thus equal to $(2n + 1).22$, only $2n + 1$ of them being equal to 1, the rest being equal to 0. We thus end up with large but very sparse vectors. Things are different when profiles of multiple alignments are used in place of the primary structure. Special attention must be paid to the inclusion of evolutionary information in this form, since it is known to improve significantly the performance of the prediction methods (see for instance (Rost and Sander, 1993; Geourjon and Deléage, 1995)). For the sake of simplicity, details on this alternative possibility are postponed to Section 1.3.3.

1.3 Specification of the Kernel

Protein secondary structure prediction is a field that has emerged more than thirty years ago, and has been since then the subject of intensive researches. Nowadays, one cannot expect to improve over the state-of-the-art but with a discriminant method specifically designed for the task. In the case of a kernel method, this means of course designing a new kernel. The one that is introduced in this section

1. `http://www.kernel-machines.org`

rests on very simple biological considerations.

1.3.1 Shortcomings of the standard kernels

Consider the vector $\mathbf{x}$ used to predict the conformational state of a given residue. Then, according to the choice of predictors described above, $\mathbf{x} = [x_{-n}, \ldots, x_i, \ldots, x_n]^T \in \{0,1\}^{(2n+1).22}$, where x_i is the canonical coding of the amino acid which occupies the i^{th} position in the window. Consequently, the function computed by a Gaussian kernel applied on two window contents $\mathbf{x}$ and $\mathbf{x}'$ can be rewritten as :

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) = \exp\left(-\frac{(2n+1) - \sum_{i=-n}^{n} \delta_{x_i, x'_i}}{\sigma^2}\right) \tag{1.1}$$

where δ is the Kronecker symbol. The right-hand side of (1.1) highlights the fact that the kernel only depends on the Hamming distance between the two strings. This is a poor summary of the information contained in the data. Indeed, two segments corresponding to 3D-superposable parts of homolog proteins, and thus sharing the same secondary structure, can differ significantly due to two evolution phenomena, insertion/deletion and substitution. The Hamming distance is very sensitive to the first one, whereas it does not take into account the nature of the substitutions, just their number. As a consequence, one cannot expect such a combination of kernel and coding (things would be similar with the other standard kernels), to give satisfactory results for the problem of interest. This simple observation is at the origin of the work on kernel design described in the following subsections, work which makes a central use of an original extension to the multi-class case of the notion of kernel alignment.

1.3.2 Multi-class kernel alignment

1.3.2.1 Framework

Kernel alignment has been introduced in (Cristianini et al., 2001), as a mean to assess the degree of fitness of a kernel to a given learning task, and adapt in consequence the Gram matrix to increase this fitness. It is thus basically a method conceived to perform transduction, since the resulting kernel is not available in analytical form. However, we use it here with another purpose, namely to estimate some kernel parameters. The reason for this choice is the following. Consider a family of kernels, where each element is characterized by the value of a formal parameter θ belonging to a set Θ. This family, $(k_\theta)_{\theta \in \Theta}$, is supposed to be built upon some knowledge on the task of interest. In order to select a kernel function k_{θ^*} that will give good performance, a natural and practical approach consists of endowing the whole family with a measure of adequacy, and then optimizing this measure with respect to the parameter. A typical example of measure of adequacy, often used in practice, is the score given by a cross-validation procedure. However, choosing it raises two difficulties. First, this can only be done when the set Θ is finite (or was discretized). Second, it is computation consuming, since the cross-validation procedure is to be run for each value θ_i in Θ. The solution advocated in (Chapelle et al., 2002) has the same drawback, since it requires to train a SVM at each step of a gradient descent. *Kernel target alignment* is a score which does not exhibit these shortcomings. In what follows, we first define it, and then describe its use to tune a kernel with respect to some parameter.

1.3.2.2 Kernel alignment principle

Definition 1.1 Kernel alignment
Let k and k' be two measurable kernel functions defined on $\mathcal{X} \times \mathcal{X}$, where the space $\mathcal{X}$ is endowed with a probability measure P. The alignment between k and k' is defined as follows :

$$A(k,k') = \frac{\langle k,k'\rangle_2}{\|k\|_2\|k'\|_2} = \frac{\int k(\mathbf{x},\mathbf{x}')k'(\mathbf{x},\mathbf{x}')dP(\mathbf{x})dP(\mathbf{x}')}{\sqrt{\int k(\mathbf{x},\mathbf{x}')^2 dP(\mathbf{x})dP(\mathbf{x}')}\sqrt{\int k'(\mathbf{x},\mathbf{x}')^2 dP(\mathbf{x})dP(\mathbf{x}')}} \quad (1.2)$$

Definition 1.2 Empirical kernel alignment
Let k and k' be two kernel functions defined on $\mathcal{X} \times \mathcal{X}$ and consider a data set $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m) \in \mathcal{X}^m$. The empirical alignment of k with k' with respect to X is the quantity:

$$\hat{A}_X(K,K') = \frac{\langle K,K'\rangle_F}{\|K\|_F\|K'\|_F} \quad (1.3)$$

where K and K' respectively denote the kernel Gram matrices associated with k and k', computed on the sample X, $\langle .,.\rangle_F$ denotes the Frobenius inner product between matrices, so that $\langle K,K'\rangle_F = \sum_{i=1}^m \sum_{j=1}^m k(\mathbf{x}_i,\mathbf{x}_j)k'(\mathbf{x}_i,\mathbf{x}_j)$, and $\|.\|_F$ is the corresponding norm.

The alignment between two kernels k and k' should be thought of as a measure of their similarity. Roughly speaking, if k' is a well suited kernel for the problem at hand and k is well aligned with k', then k should also be a good kernel for the same problem. In practice, as the alignment is not computable (since the underlying distribution P is unknown), it is estimated empirically, with (1.3). Some concentration properties of $\hat{A}_X(K,K')$ around its expected value $A(k,k')$ were studied in (Cristianini et al., 2001).

1.3.2.3 Tuning parameter θ using kernel target alignment

Now, the strategy to tune parameters based on this measure can be summarized as follows:

(1) Select a theoretically ideal kernel k_t, hereafter called the *target kernel*, ideal in the sense that it leads to perfect classification. Practically, the Gram matrix of k_t

should be computable.

(2) Given a training set of labelled examples $Z = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m))$, choose θ^* satisfying :

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}} \hat{A}_Z(k_\theta, k_t)$$

In doing so, the conjecture is that the kernel k_{θ^*} will behave well, provided the family $(k_\theta)_{\theta \in \Theta}$ is relevant for the problem at hand. In (Cristianini et al., 2001), the authors only considered the case of dichotomies. Their ideal kernel is the obvious one, namely $k_t(\mathbf{x}, \mathbf{x}') = yy'$. Our extension to the multi-class case, based on geometrical considerations developed in (Vert, 2002), is the following one:

$$k_t(\mathbf{x}, \mathbf{x}') = \begin{cases} 1 & if\ y = y' \\ -1/(Q-1) & otherwise \end{cases}$$

This target kernel corresponds to a mapping Φ_t associating each description $\mathbf{x}$ to one of the Q vertices of a $(Q-1)$-dimensional centered simplex, according to the category to which it belongs (see Figure 1.2). This clusterisation in the feature

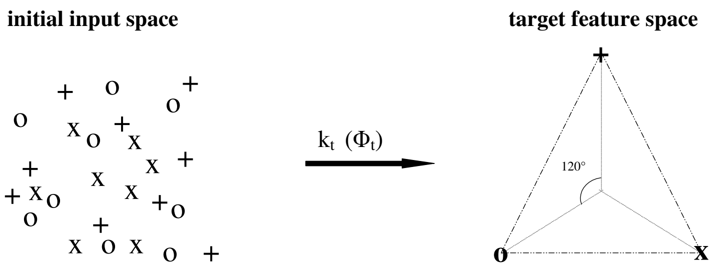

Figure 1.2 3-class problem in 2D : the target kernel performs an optimal clustering of the data in the feature space.

space is obviously the one that makes the subsequent (multi) linear separation performed by the M-SVM easiest. Note that under some regularity assumptions on k_θ, $\hat{A}_Z(k_\theta, k_t)$ is differentiable with respect to θ, and can thus be optimized using classical techniques, such as gradient descents.

1.3.3 Incorporating biological knowledge in a convolution kernel

In what follows, we adopt the terminology of (Williamson et al., 2001), where a convolution kernel is a kernel satisfying $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}', 0)$. Our goal is to take into account in such kernels two of the factors which have proved important to predict the secondary structure: the nature of the substitutions between two segments, and the relative influence of the amino acids involved as a function of their position in the window.

1.3.3.1 Dot products between amino acids

In Section 1.2.2, we pointed out the fact that the standard processing of protein sequences for secondary structure prediction involves a canonical orthonormal coding of the amino acids. This is all the more unsatisfactory that the biologists have derived well known similarity matrices for the amino acids, which differ significantly from the identity. It is the case of the PAM (Percent Accepted Mutations) (Dayhoff et al., 1978) and BLOSUM (BLOcks SUbstitution Matrix) (Henikoff and Henikoff, 1992) matrices, sometimes called substitution matrices, which are especially well suited in their log odds form. The problem raised by their use in a kernel springs from the fact that they are not symmetric positive semidefinite, and thus are not associated with an underlying dot product. To overcome this difficulty, one can think of several off-the-shelve solutions. Since the matrices are symmetric, one simple way to approximate them with a Gram matrix consists in diagonalizing them and replacing all the negative eigenvalues with 0. Another possibility consists in looking for their projection on the space of symmetric positive semidefinite matrices, the operator being associated with a matrix norm, for instance the Frobenius one. Although the projection operator will usually not be available in analytical form (the problem to be solved is a non-convex one), satisfactory estimates can result from a simple gradient descent. This descent is performed with respect to the components of the vectors representing the amino acids.

This change in the coding of the amino acids extends nicely to the case where multiple alignments are used. In that case, the profile presented as input of a connectionist classifier (see for instance (Rost and Sander, 1993; Jones, 1999; Pollastri et al., 2002)) is simply obtained by computing, for each position in the window, a weighted average of the vectors coding the amino acids present in the corresponding position of the alignment. The weight associated with a particular amino acid is its frequency of appearance in this position. In practice, let a_j, $(1 \leq j \leq 22)$, be the coding of the j^{th} amino acid (or an unknown residue, or the empty position), and θ_{ij} its frequency of appearance in the position of the alignment corresponding to the i^{th} position of the sliding window. Then the window can be represented by $\tilde{\mathbf{x}} = [\tilde{x}_{-n}, \ldots, \tilde{x}_i, \ldots, \tilde{x}_n]^T$, with $\tilde{x}_i = \sum_{j=1}^{22} \theta_{ij} a_j$. Thus, in the computation of the kernel, the dot product $\langle x_i, x_i' \rangle$ is simply replaced with:

$$\langle \tilde{x}_i, \tilde{x}_i' \rangle = \langle \sum_{j=1}^{22} \theta_{ij} a_j, \sum_{k=1}^{22} \theta_{ik}' a_k \rangle = \sum_{j=1}^{22} \sum_{k=1}^{22} \theta_{ij} \theta_{ik}' \langle a_j, a_k \rangle \tag{1.4}$$

1.3.3.2 Influence of the position in the window

As stated in Section 1.2.2, the use of the sliding window is standard in protein secondary structure prediction. Many studies have dealt with the choice of its size, or the exploitation of its content. Good illustrations are given by (Qian and Sejnowski, 1988; Zhang et al., 1992; Rost and Sander, 1993). In short, a too small window will not include enough information on the local conformation, whereas a too large one will incorporate data that risks to behave like noise. A way to overcome this difficulty consists in choosing *a priori* a large value for the size of the window, and associating each position with a weight (irrespective of the nature of the amino acid), so as to modulate its influence in the subsequent computations. This was already performed with success by different teams (Gascuel and Golmard, 1988; Guermeur, 1997). An interesting point is that these studies, although they involved very different approaches, produced similar distributions of the weights as a function of the position. This suggests that they were capable of highlighting some intrinsic property of the problem of interest. We thus decided to incorporate such a weighting in our kernel, with the values of the weights being derived through the multi-class kernel alignment.

Both parameterizations, the change in the "dot products between amino acids" and the weighting of the positions in the window, can be applied to any kind of convolution kernel. For the sake of simplicity, their incorporation is illustrated below in the case of a Gaussian kernel processing multiple alignments:

$$k_{\theta,D}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') = \exp\left(-\frac{\sum_{i=-n}^{n} \theta_i^2 (\|\tilde{x}_i\|^2 + \|\tilde{x}_i'\|^2 - 2\langle \tilde{x}_i, \tilde{x}_i' \rangle)}{2\sigma^2} \right) \tag{1.5}$$

where θ is the vector of weights and D the matrix of dot products.

1.4 Experimental Results

We have already pointed out above the difficulty to develop a state-of-the-art secondary structure prediction method. Nowadays, all the most accurate statistical methods are based on huge hierarchical and modular architectures. The best illustration of this phenomenon is given by the methods PSIPRED (Jones, 1999), SSpro2 (Pollastri et al., 2002) as well as those described in (Riis and Krogh, 1996; Petersen et al., 2000). In these architectures, which can incorporate up to several hundreds of components, the contribution of a single module (ordinarily a neural network) can hardly be assessed. From a general point of view, our goal hereafter is not to obtain recognition rates comparable with those of the above methods, but rather to highlight the fact that in each of the various contexts where a MLP can be used to perform protein secondary structure prediction (as part of a hierarchical classifier, as ensemble method, etc.), significant benefits result from replacing it with our M-SVM. In that respect, this study can be seen as the natural continuation of those reported in (Guermeur, 2002; Guermeur et al., 2004).

1.4.1 Experimental protocol

To assess our classifier, we used the set of 1096 protein sequences introduced in (Guermeur et al., 2004) under the reference P1096. This set was designed so as to meet the toughest requirements in terms of percentage of identity (see (Sander and Schneider, 1991) for details). Secondary structure assignment was performed with the DSSP program (Kabsch and Sander, 1983). This assignment is essentially based on hydrogen-bonding patterns. The reduction from 8 to 3 conformational states was derived according to the CASP method, given by: H+G $\rightarrow$ H (α-helix), E+B $\rightarrow$ E (β-strand), and all the other states in C (aperiodic or coil). This assignment is known to be somewhat harder to predict than the other ones used in the literature (see for instance (Cuff and Barton, 1999)). The PSI-BLAST alignments were compiled according to the protocol described in (Pollastri et al., 2002).

As stated in the introduction, the prediction of the secondary structure is seldom a goal in its own right. It is primarily a step towards the prediction of the tertiary structure. As a consequence, the main concern of the biologist is the recognition of all the structural elements in their order of appearance in the sequence. A small shift in the relative locations of the true and predicted structures can be tolerated, but the prediction must remain biologically plausible (no helix can be shorter than 4 residues, two periodic structures cannot be consecutive, etc.). As a consequence, the sole per residue recognition rate, hereafter denoted by Q_3, is not sufficient to characterize the quality of the prediction. To overcome this difficulty, many alternative measures of quality have been proposed. The interested reader will find in (Baldi et al., 2000) a review on the subject. In what follows, we use the three most common quality measures: the Q_3, the Pearson's/Matthews' correlation coefficients C (Matthews, 1975), and the segment overlap measure Sov (Rost et al., 1994; Zemla et al., 1999), which give complementary indications. Whereas each of the Matthews' coefficients characterizes the quality of the prediction for one particular conformational state ($\alpha/\beta/$ coil), which makes it possible, for instance, to emphasize a poor identification of the sheets, the values of the Sov coefficients give an idea of the prediction accuracy at the segment level, meeting by way of consequence one of the central requirements listed above.

1.4.2 Estimation of the parameters

The matrix of dot products between amino acids was derived from the similarity matrix introduced in (Levin et al., 1986). This choice resulted from the fact that this matrix had been specifically devised to perform secondary structure prediction based on the similarity of small peptides, i.e. on local sequence homology (see also (Levin and Garnier, 1988; Geourjon and Deléage, 1995)). In this context, it has reportedly proved superior to the Dayhoff substitution matrix. Among the two possibilities considered in Section 1.3.3.1 to generate the Gram matrix, we chose the one based on diagonalization. However, this choice was primarily made for the sake of reproducibility, since a simple gradient descent gave very similar results (Didiot, 2003). With this set of dot products at hand, the vector of weights θ could be obtained thanks to the implementation of the multi-class target alignment principle, through a stochastic gradient descent procedure (see (Vert, 2002) for details). The training set was the set of 1180 sequences used to train SSpro1 and SSpro2. This set, described in (Baldi et al., 1999; Pollastri et al., 2002), will be referred to below as P1180. This choice could be made since no sequence in this base is the homolog of a sequence of the P1096 base (see also (Guermeur et al., 2004)). Figure 1.3 illustrates the resulting values of the coefficients θ_i. This curve is

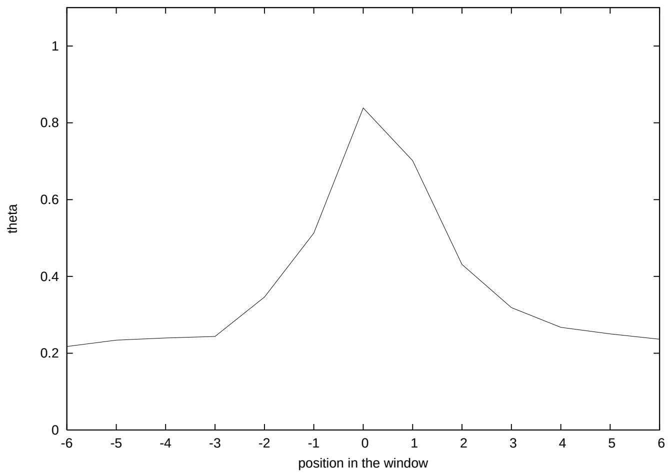

Figure 1.3 Vector θ of (1.5) maximizing the kernel target alignment.

very similar to those mentioned in Section 1.3.3.2. One of their common features is a significant dissymmetry in favour of the right-hand side context. This intriguing phenomenon, the observation of which utterly rests on statistical basis, and by no means on the incorporation of *a priori* knowledge on the task, has found no biological justification so far.

	sequences		alignments	
	MLP	M-SVM	MLP	M-SVM
Q_3	61.6	62.0	72.0	72.3
C_α	0.46	0.47	0.63	0.64
C_β	0.33	0.35	0.53	0.54
C_c	0.38	0.38	0.53	0.54
Sov	53.9	54.2	65.1	65.3
Sov_α	57.8	57.9	66.5	66.7
Sov_β	44.7	46.1	61.5	62.3
Sov_c	57.3	57.3	66.7	66.8

Table 1.1 Relative prediction accuracy of a MLP and the M-SVM on the P1096 data set.

1.4.3 Prediction from the primary structure and multiple alignments

This subsection describes the implementation and results of two experiments inspired by the pioneering works reported in (Qian and Sejnowski, 1988), and (Rost and Sander, 1993). In the first one, the M-SVM is compared with a MLP in the case where the input (vector $\mathbf{x}$) is simply given by the content of a window of size 13 sliding on the sequence. In the second one, single sequences are replaced with PSI-BLAST multiple alignments. To perform these experiments, we used twice the same procedure, a standard five-fold cross-validation (the P1096 base was devided into four sets of 219 sequences, and one set of 220 sequences). In both cases, the MLP had one hidden layer of eight units with sigmoid activation functions, and softmax output units. The parameterization of the M-SVM also remained unchanged, with the soft margin constant C being equal to 10.0, and the width of the Gaussian kernel being set to $\sigma^2 = 10.0$. Table 1.1 summarizes the results obtained.

In both configurations, the gain in recognition rate resulting from using the M-SVM in place of the MLP is statistically significant with confidence exceeding 0.95. The size of the base (255551) compensates for the small value of the increase. However, what is more promising, all measures of accuracy benefit from the change. This is particularly noticeable for the β-strands, usually the hardest conformational segments to predict. In (Hua and Sun, 2001), the authors noticed that the superiority of combinations of (bi-class) SVMs over MLPs was highlighted with the Sov coefficients. Although we were unable to duplicate their experiments and get similar results (we observed far lower Sov coefficients than they did), the same conclusion can be inferred here.

The number of dual variables of a M-SVM is equal to the number of categories minus one times the size of the training set ($(Q-1)m$). An idea of the complexity of the discriminant function it computes is given by the number of training examples for which at least one of the dual variables is different both from 0 and from C

(examples which should lie on one of the margins). In all our experiments (ten trainings of the M-SVM), the ratio of such points ranged between 25% and 30%.

1.4.4 Discussion

The results of the experiments reported above support the thesis of the superiority of our M-SVM over the standard MLP for the task at hand. However, the real touchstone to judge its usefulness is obviously the incorporation in a state-of-the-art prediction method, as we already did with SSpro2. To pave the way for this new step, one can think of implementing several straightforward improvements. A simple one is the choice of a better matrix of dot products between amino acids (matrix D). Instead of choosing, *a priori* or by any model selection method, one specific similarity matrix, one can take benefit of the fact that a convex combination of symmetric positive semidefinite matrices is still a symmetric positive semidefinite matrix. This makes it possible to select a whole set of similarity matrices, estimate them with Gram matrices, and compute the optimal combination thanks to the procedure described in Section 1.3.2.3. Another useful development is the post-processing of the conformational scores generated, to produce class posterior probability estimates. These estimates could then be used to compute the observation probability density functions of a HMM, as we did for instance in (Guermeur, 2002). Both possibilities are the subject of an ongoing work.

1.5 Conclusion and Future Work

We have described a first attempt to implement M-SVMs to perform protein secondary structure prediction from the primary structure, or profiles of multiple alignments. This study has focused on the design of a kernel incorporating high-level knowledge on the task. The process of evolution, which makes two homolog proteins differ in their sequences, while keeping similar folds, is based in two phenomena: substitution and insertion/deletion. If one can think of simple solutions to take the first phenomenon into account with a kernel, as the one we used, the second one raises more difficulties. Indeed, if substituting a multiple alignment to a single sequence provides useful additional evolutionary information, this procedure alone does not solve the problem of the comparison of two window contents by means of a dot product. What if they only differ by an insertion? A priori, a natural way to overcome this difficulty would consist in making use of the results established by Haussler (1999), or Watkins (2000), regarding dynamic alignment kernels and select, for instance, an adequately designed pair HMM (Durbin et al., 1998) to compute the kernel function. However, this solution currently remains unfeasible for such large problems as those we are interested in, due to its prohibitive CPU time requirements coming from too high computational complexity. Cheaper alternatives are thus badly needed. Well suited spectrum (Leslie and Kuang, 2003) and rational (Cortes et al., 2003) string kernels, which can extract the similarity of unequal length pairs of strings, are potentially good candidates with their efficient linear computational complexity. Their incorporation in our machine is currently under investigation.

Acknowledgements

The authors gratefully acknowledge the support of the CNRS funded "Action Spécifique : Apprentissage et Bioinformatique". They would like to thank Dr G. Pollastri for providing them with the data sets used in the experiments, and Prof. A. Ourjoumtsev and D. Eveillard for helping them with the production of Figure 1.1. Thanks are also due to Dr A. Zemla and Dr C. Venclovas for the availability of the code of the Sov measure, and to Dr F.D. Maire for carefully reading this manuscript.

References

C.B. Anfinsen, C.J. Epstein, and R.F. Goldberger. The genetic control of tertiary protein structure: studies with model systems. In *Cold Spring Harbor Symp. Quant. Biol.*, volume 28, pages 439–449, 1963.

P. Baldi and S. Brunak. *Bioinformatics: the machine learning approach.* MIT Press, Cambridge, MA, second edition, 2001.

P. Baldi, S. Brunak, Y. Chauvin, C.A. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.

P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15(11): 937–946, 1999.

P. L. Bartlett and J. Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 43–54, Cambridge, MA, 1999. MIT Press.

E.J. Bredensteiner and K.P. Bennett. Multicategory Classification by Support Vector Machines. *Computational Optimization and Applications*, 12(1/3):53–79, 1999.

O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1–3):131–159, 2002.

C. Cortes, P. Haffner, and M. Mohri. Positive definite rational kernels. In *16th Annual Conference on Learning Theory (COLT'03)*, pages 41–56, 2003.

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

N. Cristianini, A. Elisseeff, J. Shawe-Taylor, and J. Kandola. On kernel target alignment. Technical Report NC-TR-01-099, NeuroCOLT2, 2001.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, Cambridge, UK, 2000.

N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances* *in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

J.A. Cuff and G.J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins*, 34:508–519, 1999.

M.O. Dayhoff, R.M. Schwartz, and B.C. Orcutt. A model of evolutionary change in proteins. In M.O. Dayhoff, editor, *Atlas of Protein Sequence and Structure*, volume 5, pages 345–358. National Biomedical Research Foundation, Silver Spring, Washington DC, 1978.

J.P. Derrick and D.B. Wigley. The Third IgG-Binding Domain from Streptococcal Protein G: an Analysis by X-ray Crystallography of the Structure Alone and in a Complex with Fab. *J. Mol. Biol.*, 243(5):906–918, 1994.

E. Didiot. Conception et mise en œuvre de M-SVM dédiées au traitement de séquences biologiques. Master's thesis, DEA informatique de Lorraine, 2003. (in French).

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis - Probabilistic models of proteins and nucleic acids*. Cambridge University Press, Cambridge, UK, 1998.

A. Elisseeff. *Etude de la complexité et contrôle de la capacité des systèmes d'apprentissage : SVM multi-classe, réseaux de régularisation et réseaux de neurones multicouches*. PhD thesis, ENS Lyon, 2000. (in French).

R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, New York, 1989.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

O. Gascuel and J.L. Golmard. A simple method for predicting the secondary structure of globular proteins: implications and accuracy. *CABIOS*, 4(3):357–365, 1988.

C. Geourjon and G. Deléage. SOPMA: significant improvements in protein secondary structure prediction by consensus prediction from multiple alignments. *CABIOS*, 11(6):681–684, 1995.

Y. Guermeur. *Combinaison de classifieurs statistiques, application à la prédiction de la structure secondaire des protéines*. PhD thesis, Université Paris 6, 1997. (in French).

Y. Guermeur. Combining discriminant models with new multi-class SVMs. *Pattern Analysis and Applications*, 5(2):168–179, 2002.

Y. Guermeur, A. Elisseeff, and D. Zelus. A comparative study of multi-class support vector machines in the unifying framework of large margin classifiers. *Applied Stochastic Models in Business and Industry*, 21(2):199–214, 2005.

Y. Guermeur, G. Pollastri, A. Elisseeff, D. Zelus, H. Paugam-Moisy, and P. Baldi. Combining protein secondary structure prediction models with ensemble methods of optimal complexity. *Neurocomputing*, 56C:305–327, 2004.

L. Gurvits. A note on a scale-sensitive dimension of linear bounded functionals in Banach spaces. *Theoretical Computer Science*, 261(1):81–90, 2001.

D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, Departement of Computer Science, University of California at Santa Cruz, 1999.

S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Nat. Acad. Sci. USA*, 89:10915–10919, 1992.

C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Trans. on Neural Networks*, 13:415–425, 2002.

S. Hua and Z. Sun. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *Journal of Molecular Biology*, 308:397–407, 2001.

D.T. Jones. Protein Secondary Structure Prediction Based on Position-specific Scoring Matrices. *J. Mol. Biol.*, 292:195–202, 1999.

W. Kabsch and C. Sander. Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. *Biopolymers*, 22 (12):2577–2637, 1983.

M. Karplus and G.A. Petsko. Molecular dynamics simulations in biology. *Nature (London)*, 347:631–639, 1990.

Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines. Technical Report 1040, Department of Statistics, University of Madison, Wisconsin, 2001.

C. Leslie and R. Kuang. Fast kernels for inexact string matching. In *16th Annual Conference on Learning Theory (COLT'03), Kernel Workshop*, pages 114–128, 2003.

J.M. Levin and J. Garnier. Improvements in a secondary structure prediction method based on a search for local sequence homologies and its use as a model building tool. *Biochimica et Biophysica Acta*, 955:283–295, 1988.

J.M. Levin, B. Robson, and J. Garnier. An algorithm for secondary structure determination in proteins based on sequence similarity. *FEBS*, 205(2):303–308, 1986.

B.W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, 405:442–451, 1975.

E. Mayoraz and E. Alpaydin. Support Vector Machines for Multi-Class Classification. Technical Report 98-06, IDIAP, 1998.

T.N. Petersen, C. Lundegaard, M. Nielsen, H. Bohr andJ. Bohr, S. Brunak, G.P. Gippert, and O. Lund. Prediction of Protein Secondary Structure at 80% Accuracy. *PROTEINS: Structure, Function, and Genetics*, 41(1):17–20, 2000.

G. Pollastri, D. Przybylski, B. Rost, and P. Baldi. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins*, 47(2):228–235, 2002.

N. Qian and T.J. Sejnowski. Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol.*, 202:865–884, 1988.

S. Riis and A. Krogh. Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. *J. Comput. Biol.*, 3:163–183, 1996.

B. Rost. Review: Protein secondary structure prediction continues to rise. *Journal of Structural Biology*, 134(2):204–218, 2001.

B. Rost and S. O'Donoghue. Sisyphus and prediction of protein structure. *CABIOS*, 13:345–356, 1997.

B. Rost and C. Sander. Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, 232:584–599, 1993.

B. Rost, C. Sander, and R. Schneider. Redefining the Goals of Protein Secondary Structure Prediction. *J. Mol. Biol.*, 235:13–26, 1994.

C. Sander and R. Schneider. Database of homology derived protein structures and the structural meaning of sequence alignment. *Proteins*, 9:56–68, 1991.

R.A. Sayle and E.J. Milner-White. RASMOL: biomolecular graphics for all. *Trends Biochem. Sci.*, 20(9):374–376, 1995.

B. Schölkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In U. M. Fayyad and R. Uthurusamy, editors, *Proceedings, First International Conference on Knowledge Discovery & Data Mining*, Menlo Park, 1995. AAAI Press.

V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, New York, 1995. ISBN 0-387-94559-8.

V. N. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.

R. Vert. Designing a M-SVM kernel for protein secondary structure prediction. Master's thesis, DEA informatique de Lorraine, 2002.

C. Watkins. Dynamic alignment kernels. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 39–50. The MIT Press, 2000.

J. Weston and C. Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998.

R. C. Williamson, A. J. Smola, and B. Schölkopf. Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators. *IEEE Transactions on Information Theory*, 47(6):2516–2532, 2001.

A. Zemla, Č. Venclovas, K. Fidelis, and B. Rost. A modified definition of Sov, a segment-based measure for proteinsecondary structure prediction assessment. *PROTEINS: Structure, Function, and Genetics*, 34(2):220–223, 1999.

X. Zhang, J.P. Mesirov, and D.L. Waltz. Hybrid System for Protein Secondary

Structure Prediction. *J. Mol. Biol.*, 225:1049–1063, 1992.