Protein Secondary Structure Prediction with Multi-Class Support Vector Machines

Yann Guermeur

LORIA - CNRS

http://www.loria.fr/~guermeur

Summer School NN2008

July 4, 2008

Overview

Protein secondary structure prediction

- Different levels of structural organization of the proteins
- A problem of central importance in structural biology
- Different measures of prediction accuracy

State of the art

- Choice of the predictors
- Building blocks and architecture of the main prediction methods

Overview

Implementation of multi-class SVMs

- Models implemented
- Training algorithm
- Dedicated RBF kernel
- Computation of the weighting vector θ
- Experimental results

Conclusions and future work

Basic notions about proteins

Definition

- Proteins: macromolecules made up of amino acids
- 20 amino acids, each of them represented by a letter (A, R, N, D, C, E, ...)

Hierarchical description of the conformation

- Primary structure (sequence of amino acids) $\Longleftarrow$ sequencing
- Secondary structure (sequence of structural elements) $\Longleftarrow$ circular dichroism
- Tertiary structure (three-dimensional structure) $\Longleftarrow$ X-ray, NMR
- ...

Sequence or primary structure ($1.6 \cdot 10^6$ **known sequences)**

```
MEEKLKKAKIIFVVGGPGSGKGTQCEKIVQKYGYTHLSTC...
```

Secondary structure

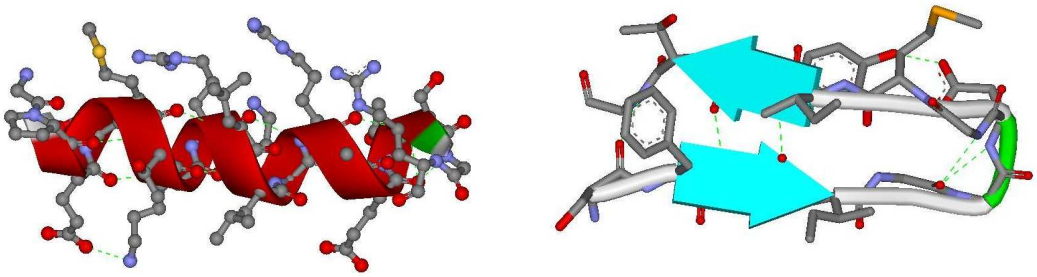
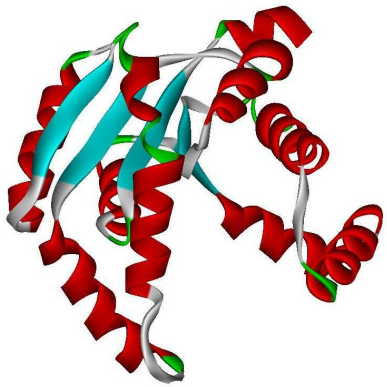


Figure 1: Periodic structural elements: α helix (left) and β strands (right)

Tertiary structure ($2.7 \cdot 10^4$ **known 3D structures)**



A problem of central importance in structural biology

Biological context Functional exploitation of the data generated by the large-scale sequencing projects: rests on the availability of the 3D structure of the proteins.

1. Massive arrival of protein sequences (exponential growth of the databases)

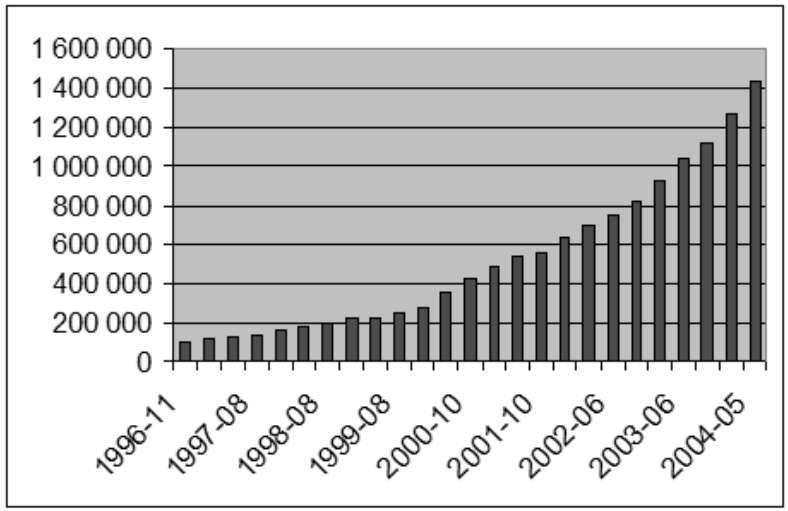


Figure 2: Growth of the international bank TREMBL from 1996 until 2005

2. Experimental determination of the 3D structure: highly labour-intensive task. . . when it can be done $\Longrightarrow$ Necessity to switch from a biochemical approach to a predictive approach

Different measures of prediction accuracy

Q_3: recognition rate at the residue level

Pearson's/Matthews' correlation coefficients (Matthews, 1975)

$$C_i = \frac{p_i n_i - u_i o_i}{\sqrt{(p_i + u_i)(p_i + o_i)(n_i + u_i)(n_i + o_i)}}$$

Root mean square deviation (r.m.s.d.)

$$\sigma_i = \sqrt{\frac{1}{n_s} \sum_{j=1}^{j=n_s} (obs_{ij} - pred_{ij})^2}$$

Sov coefficients (Rost *et al.*, 1994; Zemla *et al.*, 1999)

$$Sov(\delta) = \frac{1}{n} \sum_{S_1} \left\{ \frac{1}{n_{S_2}} \sum_{S_2 / S_1 \cap S_2 \neq \emptyset} \frac{\min(end(S_1), end(S_2)) - \max(beg(S_1), beg(S_2)) + 1 + \delta}{\max(end(S_1), end(S_2)) - \min(beg(S_1), beg(S_2)) + 1} len(S_1) \right\}$$

Choice of the predictors

Local approach of the prediction

- Basic principle: use of a window sliding on the sequence
- Incorporation of physico-chemical information (hydrophobicity, charge and bulk of the residues...)

Exploiting evolutionary information: processing multiple sequence alignments

- Computation of sequence profiles (Rost & Sander, 1993; Jones, 1999;...)
- Combination of the predictions performed independently for each of the sequences of an alignment (Riis & Krogh, 1996)

Building blocks and architecture of the main prediction methods

Main models used

- Neural networks: MLPs (Qian & Sejnowski, 1988), BRNNs (Baldi *et al.*, 1999)
- Hidden Markov models (Asai *et al.*, 1993; Martin *et al.*, 2005)
- Bi-class support vector machines (Hua & Sun, 2001) and M-SVMs (Guermeur, 2000)

Basic architecture of a prediction method

- Two-level prediction: a structure-to-structure module post-processes the output of a sequence-to-structure module (Qian & Sejnowski, 1988 $\longrightarrow$)
- Use of ensemble methods involving up to hundreds of basic classifiers (Rost & Sander, 1993; Petersen *et al.*, 2000)
- Hierarchical architecture involving discriminant and generative models (Guermeur, 1997)

Three M-SVMs with different statistical properties

General formulation of the training algorithm

Problem 1

$$\min_{h\in\mathcal{H}}\left\{\phi_{M\text{-}SVM}\left(\left(\ell_{M\text{-}SVM}\left(y_i,h(x_i)\right)\right)_{1\leq i\leq m}\right)+\lambda\|\bar{h}\|_{\bar{\mathcal{H}}}^2\right\}$$
$$s.t.\ \sum_{k=1}^{Q}h_k=0$$

1. M-SVM of Weston and Watkins: $\begin{cases}\ell_{\mathrm{WW}}(y,h(x))=\sum_{k\neq y}\left(1-h_y(x)+h_k(x)\right)_+\\ \phi_{\mathrm{WW}}(t)=\|t\|_1\end{cases}$

2. M-SVM of Lee, Lin and Wahba: $\begin{cases}\ell_{\mathrm{LLW}}\left(y,h(x)\right)=\sum_{k\neq y}\left(h_k(x)+\frac{1}{Q-1}\right)_+\\ \phi_{\mathrm{LLW}}=\phi_{\mathrm{WW}}\end{cases}$

3. M-SVM^2: $\begin{cases}\ell_{\text{M-SVM}^2}=\ell_{\mathrm{LLW}}\\ \phi_{\text{M-SVM}^2}(t)=t^TM_tt=\sum_{i=1}^m\sum_{j=1}^m\sum_{k=1}^Q\sum_{l=1}^Q\left(\delta_{k,l}-\frac{1}{Q}\right)\delta_{i,j}t_{ik}t_{jl}\end{cases}$

Frank-Wolfe algorithm (1956)

Problem 2 (General formulation of the problem considered)

$$\min_t f(t)$$

$$s.t. \begin{cases} At = b \\ t \geq 0 \end{cases}$$

Two-step iterative method generating a sequence of feasible points $\left(t^{(n)}\right)$

(1) Solve the linear programming problem $LP\left(t^{(n)}\right)$ given by:

Problem 3

$$\min_u \left\{ \nabla f \left(t^{(n)}\right)^T u \right\}$$

s.t. constraints of Problem 2

(2) $u^{(n)}$: optimal solution of $LP(t^{(n)})$. $t^{(n+1)}$: chosen so as to minimize f on $\left[t^{(n)}, u^{(n)}\right]$.

Frank-Wolfe algorithm applied to the M-SVM of Weston and Watkins

Expression of the LP problem

$\beta = (\beta_{ik})_{1\leq i\leq m, 1\leq k\leq Q}$, $(\beta_{iy_i})_{1\leq i\leq m} = 0$

Problem 4 (Computation of $\beta^{(n)}$)

$$\min_{\beta} \left\{ {\alpha^{(n)}}^T H_{WW} \beta - 1_{Qm}^T \beta \right\}$$

$$s.t. \begin{cases} 0 \leq \beta_{ik} \leq C, & (1 \leq i \leq m),\ (1 \leq k \neq y_i \leq Q) \\ \sum_{i=1}^{m} \sum_{l=1}^{Q} \left(\delta_{y_i,k} - \delta_{k,l}\right) \beta_{il} = 0, & (1 \leq k \leq Q-1) \end{cases}$$

Coefficient of the optimal convex combination

$$\gamma^{(n)} = \underset{\gamma\in[0,1]}{\operatorname{argmin}} J_d \left((1-\gamma)\alpha^{(n)} + \gamma\beta^{(n)} \right)$$

$$\gamma^{(n)} = \min \left\{ -\frac{\nabla J_d(\alpha^{(n)})^T \left\{\beta^{(n)} - \alpha^{(n)}\right\}}{\left\{\beta^{(n)} - \alpha^{(n)}\right\}^T H_{\mathrm{WW}} \left\{\beta^{(n)} - \alpha^{(n)}\right\}},\ 1 \right\}$$

Remark 1 *Our implementation incorporates a* decomposition method.

RBF kernel for protein sequence processing

Analytical expression (primary structure only)

$\mathbf{x} = (x_i)_{-n \leq i \leq n}$: vector coding a polypeptide (content of a window of size $2n+1$)

$$\kappa_{\theta,D}(\mathbf{x}, \mathbf{x}') = \exp\left(-\sum_{i=-n}^{n} \theta_i^2 \|x_i - x_i'\|^2\right)$$

Extension for multiple alignment processing

Straightforward: $\mathbf{x}$ replaced with $\tilde{\mathbf{x}} = (\tilde{x}_i)_{-n \leq i \leq n}$ such that $\tilde{x}_i = \sum_{j=1}^{22} \theta_{ij} a_j$

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

Taking into account the substitutions (matrix A)

Several standard similarity matrices S

	G	P	D	E	A	N	Q	S	T	K	R	H	V	I	M	C	L	F	Y	W
G	2																			
P	1	3																		
D	0	0	2																	
E	0	-1	1	2																
A	0	-1	0	1	2															
N	0	0	1	0	0	3														
Q	0	0	0	1	0	1	2													
S	0	0	0	0	1	0	0	2												
T	0	0	0	0	0	0	0	0	2											
K	0	0	0	0	0	1	0	0	0	2										
R	0	0	0	0	0	0	0	0	0	1	2									
H	0	0	0	0	0	0	0	0	0	0	0	2								
V	-1	-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	2							
I	-1	-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	1	2						
M	-1	-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	0	0	2					
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2				
L	-1	-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	1	0	2	0	2			
F	-1	-1	-1	-1	0	-1	-1	-1	0	-1	-1	-1	0	1	0	-1	0	2		
Y	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	-1	0	1	2	
W	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	0	0	0	0	-1	0	0	0	2

Figure 3: Secondary structure similarity matrix (Levin *et al.*, 1986)

Approximating S with a Gram matrix

- $A = (a_{ij}) \in \mathcal{M}_{22,22}(\mathbb{R})$: (implicit) representations of the amino acids
- $G = AA^T$: matrix of dot products = symmetric positive semidefinite approximation of S

Let the diagonalization of S be given by:

$$S = PDP^{-1} = PDP^T$$

(P is orthogonal since S is symmetric).

Then

$$AA^T = PD_+P^T$$

where D_+ is derived from D by setting to 0 the negative eigenvalues.

This leads to

$$A = P\sqrt{D_+}.$$

Kernel alignment

Definition 1 (Kernel alignment, Cristianini *et al.*, 2002) *Let κ and κ' be two measurable kernel functions defined on $\mathcal{T} \times \mathcal{T}$, where the space $\mathcal{T}$ is endowed with a probability measure $P_{\mathcal{T}}$. The* alignment between κ and κ', $A(\kappa, \kappa')$, *is defined as follows:*

$$A(\kappa, \kappa') = \frac{\langle \kappa, \kappa' \rangle}{\|\kappa\| \|\kappa'\|} = \frac{\int_{\mathcal{T}^2} \kappa(t,t')\kappa'(t,t') dP_{\mathcal{T}}(t) dP_{\mathcal{T}}(t')}{\sqrt{\int_{\mathcal{T}^2} \kappa(t,t')^2 dP_{\mathcal{T}}(t) dP_{\mathcal{T}}(t')} \sqrt{\int_{\mathcal{T}^2} \kappa'(t,t')^2 dP_{\mathcal{T}}(t) dP_{\mathcal{T}}(t')}}.$$

Definition 2 (Empirical kernel alignment, Cristianini *et al.*, 2002) *$\mathcal{T}$, κ and κ' being defined as above, let $T^n = (T_i)_{1 \leq i \leq n}$ be a n-sample of independent random variables distributed according to $P_{\mathcal{T}}$. The* empirical alignment of κ and κ' with respect to T^n *is the quantity:*

$$\hat{A}_{T^n}(G, G') = \frac{\langle G, G' \rangle_F}{\|G\|_F \|G'\|_F}$$

where G and G' respectively denote the Gram matrices associated with κ and κ', computed on T^n, and $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product between matrices, so that $\langle G, G' \rangle_F = \sum_{i=1}^n \sum_{j=1}^n \kappa(T_i, T_j) \kappa'(T_i, T_j)$. $\| \cdot \|_F$ represents the corresponding norm.

Kernel-target alignment

Tuning parameter θ using kernel-target alignment

The strategy to tune kernel parameters based on the principle of kernel alignment 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 $z^m = \{(x_i, y_i) : 1 \leq i \leq m\}$, choose θ^* satisfying:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}} \hat{A}_{z^m}(G_\theta, G_t),$$

where G_θ is the Gram matrix associated with the pair (κ_θ, z^m), G_t being the Gram matrix associated with the pair (κ_t, z^m).

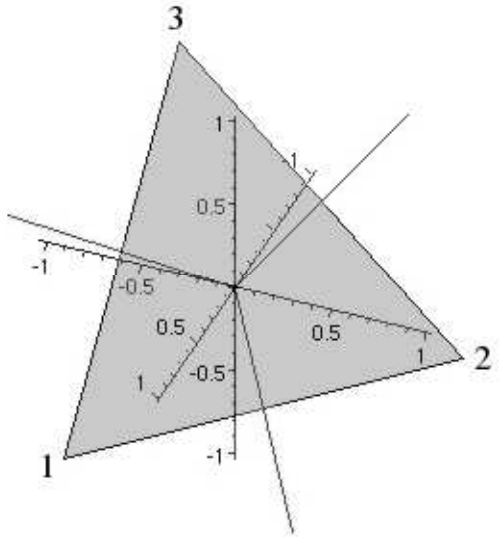
Choice of the target kernel

Bi-class case (Cristianini *et al.*, 2002)

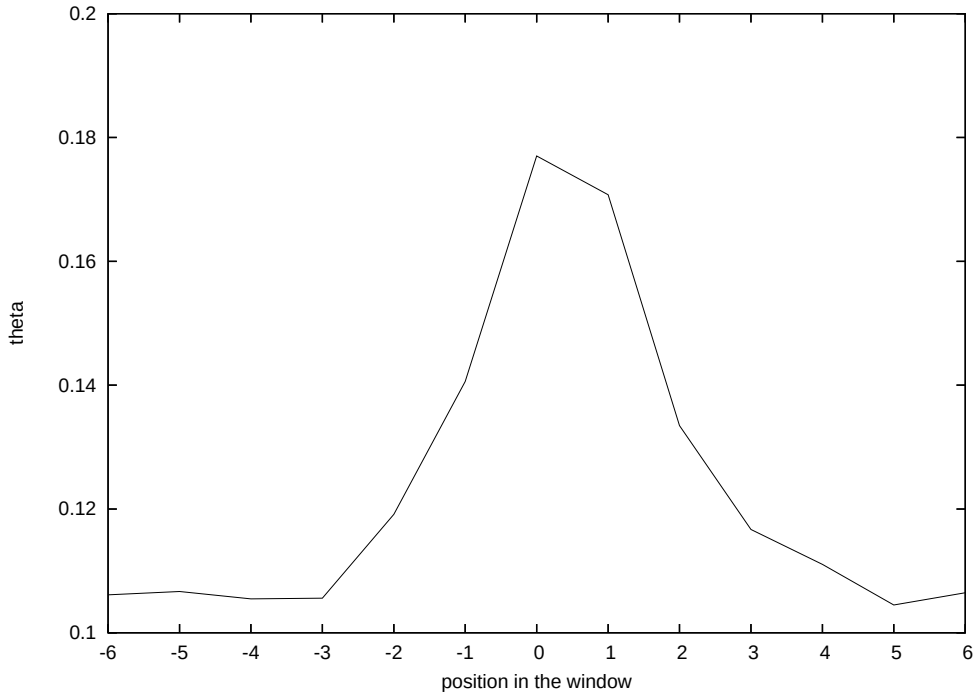
$$\forall\left((x,y),(x',y')\right)\in\left(\mathcal{X}\times\mathcal{Y}\right)^2,\ \kappa_t(x,x')=yy'$$

Multi-class case (Vert, 2002)

$$\forall\left((x,y),(x',y')\right)\in\left(\mathcal{X}\times\mathcal{Y}\right)^2,\ \kappa_t(x,x')=\left(-\frac{1}{Q-1}\right)^{1-\delta_{y,y'}}$$



Vector θ obtained



Training algorithm: stochastic gradient descent. Let $G'_{\theta_k,D} = \left(\frac{\partial}{\partial\theta_k} k_{\theta,D}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right)$.

$$\forall k \in [\![-n, n]\!], \quad \frac{\partial}{\partial\theta_k}\hat{A}_{z^m}(G_{\theta,D}, G_t) = \frac{\langle G'_{\theta_k,D}, G_t\rangle_F}{\|G_{\theta,D}\|_F\|G_t\|_F} - \frac{\langle G_{\theta,D}, G_t\rangle_F\langle G_{\theta,D}, G'_{\theta_k,D}\rangle_F}{\|G_{\theta,D}\|_F^3\|G_t\|_F}$$

Experimental results

Data set: P1096 (sequence identity $< 30\%$). Size of the sliding window: 13. 5-fold cross-validation.

	MLP	M-SVM WW	M-SVM LLW	M-SVM^2
Q_3	66.0	66.9	66.7	66.7
C_α	0.50	0.52	0.51	0.51
C_β	0.41	0.42	0.40	0.41
C_c	0.45	0.46	0.46	0.46
Sov	55.7	56.0	56.2	56.1
Sov_α	57.7	59.5	62.2	60.1
Sov_β	49.4	51.7	46.7	51.2
Sov_c	57.8	58.4	58.7	58.0

Table 1: Prediction accuracy of a MLP and three M-SVMs measured on the base P1096 (268575 residues)

Conclusions and future work

Conclusions

- Incorporating SVMs and M-SVMs in the secondary structure prediction methods should improve the prediction accuracy.
- This task raises interesting problems for "kernel designers".
- Future should belong to hybrid methods integrating discriminant and generative models.

Future work

- Applying ensemble methods to combine several M-SVMs
- Applying M-SVMs to multiple alignments
- Post-processing the output of the M-SVMs with Hidden Markov Models (IHMM...)

Modular and hierarchical approach of the prediction