BIOINFORMATICS

Vol. 18 no. 8 2002
Pages 1091–1101

Probabilistic alignment of motifs with sequences

Pedro Gonnet[1,*] *and Frédérique Lisacek*[1,2]

[1]*GeneBio S.A., 25. Av. de Champel, 1206 Geneva, Switzerland*
and [2]*Laboratoire Geńome et Informatique, Tour Evry 2, 91034 Evry Cedex France,*

Received on November 20, 2001; revised on January 28, 2002; accepted on February 15, 2002

ABSTRACT

Motivation: Motif detection is an important component of the classification and annotation of protein sequences. A method for aligning motifs with an amino acid sequence is introduced. The motifs can be described by the secondary (i.e. functional, biophysical, etc...) characteristics of a signal or pattern to be detected. The results produced are based on the statistical relevance of the alignment. The method was targeted to avoid the problems (i.e. overfitting, biological interpretation and mathematical soundness) encountered in other methods currently available.

Results: The method was tested on lipoprotein signals in *B. subtilis* yielding stable results. The results of signal prediction were consistent with other methods where literature was available.

Availability: An implementation of the motif alignment, refining and bootstrapping is available for public use online at http://www.expasy.org/tools/patoseq/.

Contact: pedro.gonnet@genebio.com

1 INTRODUCTION

Regions of shared similarity between sequences, i.e. *motifs*, are representative of protein families and domains. A range of sequence analysis methods are used, e.g. for annotation purposes, to detect known motifs in newly sequenced and translated DNA. To perform the detection, motifs are converted into motif descriptors. In principle, constraints governing a biological process should be reflected in these descriptors. In that sense, motif descriptors are an abstract representation of the underlying mechanism in which proteins bearing the motif are involved. For example, secreted proteins are characterized by a signal peptide. This peptide is in turn characterized by amino acids whose presence can be rationalized with respect to the targeting role of the signal.

In practice, the constraints that are identified in a molecular process influence the choice of the descriptors, namely an alphabet and a set of regularities or rules. In proteins, constraints are positional and most detection methods rely on rules governing the presence/absence of a particular amino acid at a particular position. Such rules are either generated automatically or in-built in the description. Two categories of descriptors are usually distinguished: *deterministic*, such as consensus sequences or patterns, or *probabilistic*, such as frequency vectors or matrices.

The PROSITE (Bairoch, 1991) database associated with the ScanPROSITE program is the oldest reference for general protein motif detection. Early PROSITE motif descriptors were regular expressions over the alphabet of the amino acids and a wild card `x` and categorized as deterministic. In this case, rules are an integral part of the description and detection is based on a pattern matching procedure, yielding a purely qualitative result, i.e. exact match or no match. The absence of quantifiable evaluation led to defining motifs in terms of frequency vectors in later versions of PROSITE (Hofmann *et al.*, 1999) where a probabilistic result is returned for each match. This shift from deterministic to probabilistic highlights the importance of the scoring functions associated with the detection method. Other probabilistic approaches, such as neural networks or Hidden Markov models, include a training phase during which implicit rules are automatically generated. These algorithms discriminate quite well between the positive and negative training sets, but the resulting scoring scheme cannot be interpreted biologically (Baldi and Brunak, 1998). This is partly due to too much emphasis put on amino acid positional constraints. Consequently, most efforts have been invested into refining score calculations given descriptors defined over the alphabet of amino acids. Varying the alphabet describing motifs has rarely been investigated, though some examples can be found (Gascuel and Danchin, 1986; Brendel and Karlin, 1989).

More recently, alternative ways of representation have been suggested for signal peptides (Tjalsma *et al.*, 2000) or C-terminal glycosylated phosphatidylinositol (GPI) anchoring signals (Eisenhaber *et al.*, 1999).

In Tjalsma *et al.* (2000), motifs are described not only in terms of specific sites characterized by residue frequency vectors, but also as a combination of distinct features such as charge, hydrophobicity, etc... The approach is formal-

*To whom correspondence should be addressed.

ized further in Eisenhaber *et al.* (1999) where domains are also defined using physico-chemical features. Frequency vectors derived from a training set are calculated. These frequency vectors and a set of functions weighting the distinct features are the core of a matching and scoring procedure.

The matching procedure and the score function as the two main components of detection methods: a region in a protein is matched with a pattern or a profile and scored. Regularities are positional constraints whether automatically or manually derived. As a result, instances of a positive or negative set are necessarily aligned prior to processing. Matching and scoring are thus considered as independent components of detection. In the method presented here, we first suggest to change the alphabet used for describing motifs and include partial information on positional constraints in the descriptors. Doing so the matching and the scoring procedures may depend on each other. Binding scoring calculations to matching is meant to refine and stabilize scores. A protein sequence is compared to a *description* of a protein sequence, and the score is the maximized probability of the sequence fitting the description—much in the same way as in sequence alignment and scoring in (Dayhoff *et al.*, 1979) where the probability of two sequences having a common ancestor is maximized. This motif alignment method is illustrated with the detection of bacterial lipoprotein signals.

2 MOTIF DESCRIPTION

A motif consists of a sequence of tokens each describing the characteristics of one or more amino acids. The different tokens and their syntax are summarized in Table 1.

Each token and sequence character can be given a weight by prefixing it with a numerical value. Values prefixed to `A`, `_` and `*` tokens will be ignored.

For example, the initial motif used for identification of lipoproteins in *B. subtilis* is written as:

```
M[p,4:2][h,12:5]{}_{}C{}*
```

It is interpreted as starting with a Methionine residue, containing a positively charged sequence of 4 amino acids, a hydrophobic helix of 12 amino acids, a stretch of three amino acids defined by two signatures around a random residue, a fixed Cysteine residue, and a signature for the residue following the Cysteine.

3 MOTIF SCORING

A motif can be aligned to a sequence the same way two sequences can be aligned with each other. However, one motif token may encompass more than one amino acid. No token or symbol can be inserted or deleted.

Once a sequence is aligned with a motif, the score of the alignment can be determined as the sum of the partial scores of the aligned tokens:

$$score(t) = \sum_i score(t_i)$$

where t is a token sequence and $score(t_i)$ is the partial score for the alignment of the token t_i. Each aligned token has a given length and content. For sequence tokens, the partial score is composed of the sum of the score for its length and the scores of the characters included in it.

For consistency, all partial scoring functions are purposefully in the same units of measurement. Each score is expressed in terms of a probability, namely the probability of the sequence matching a given token. The logarithm of this probability is used as the score:

$$score(t_i) = \log(P^+(t_i))$$

where $P^+(t_i)$ is the probability of the t_i matching the sequence where it is aligned. A $P^+(t)$ is then determined for each token type.

Since we are using logarithms, the sum of the partial scores represents the total probability of the motif adhering to the sequence:

$$e^{score(t)} = \prod_i P^+(t_i)$$

which is what we will use as the total score of the alignment.

The probability function $P^+(t)$ of the space (`_`), any (`*`) and fixed amino acid (`A`) tokens, is always one, since these

Table 1. Token syntax and meaning

`A`	A fixed amino acid (anchor).
`a`	A variable amino acid (distance measure used: Dayhoff_{250}).
`{a=0.1,g=0.9}`	A signature (frequency vector). An empty signature will be interpreted as the natural frequencies of the amino acids. The values given are normalized. Entries with no frequency value are given the value `1`.
`[x,10:1]`	A sequence of length 10 with variance 1 and *x* being either *p* or *n* for positive or negative charge, *o* or *y* for hydrophobicity or hydrophilicity, *h* for a trans-membrane helix, *a* or *b* for an amphipatic alpha helix or beta sheet, a signature for a frequency vector or * for any sequence (only length is matched).
`_`	Any single amino acid.
`*`	Any sequence of amino acids.

tokens always match (in fact, for the fixed amino acid, this is a constraint). Their score is therefore always 0.

The functions used for scoring the other tokens are described in the following sections.

3.1 Signature scoring

A signature represents an amino acid distribution. The probability of an amino acid belonging to the given signature or to a random distribution (natural amino acid frequency), can be calculated as follows:

$$P_s^+(aa) = \frac{P_s(aa)}{P_s(aa) + P_n(aa)}$$

where $P_s(aa)$ is the frequency of the amino acid aa given by the signature and $P_n(aa)$ its natural frequency. For a signature within a sequence token, the total score $score_s(s)$ is then:

$$score_s(s) = \frac{1}{n} \sum_i^n \log\left(P_s^+(s_i)\right)$$

which is the geometric mean of the probabilities for each position.

To avoid over-penalization while training a motif, a noise value ϵ can be added to allow for imprecision in the signature definition:

$$P^+(aa) = \frac{P_s(aa) + \epsilon P_n(aa)}{P_s(aa) + P_n(aa)}$$

3.2 Length scoring

Given a target length μ for a sequence token, it is assumed that the observed lengths will be Poisson-distributed around this value. Assuming that the lengths of random sequence tokens are evenly distributed over a range $[a, b]$, the probability of the length l belonging to the Poisson distribution around μ is:

$$P_l^+(l) = \frac{\Pi_\mu(l)}{\Pi_\mu(l) + \frac{1}{b-a}}$$

where $\Pi_\mu(l)$ is the Poisson probability function at l.

The range $[a, b]$ is given by either a range restriction (by appending `(a,b)` to the sequence token) or the range optimized over during alignment.

Since for the Poisson distribution $\mu = \sigma$, the supplied σ of the distribution is ignored.

3.3 Charge and hydrophobicity scoring

Charge and hydrophobicity are slightly more difficult to evaluate than the length, since scoring relies on a more abstract notion of *high* or *low* charge or hydrophobicity.

In general, the measure of charge on a sequence s is defined as

$$ch(s) = \sum_i ch(s_i)$$

where s_i is the ith amino acid in the sequence s. The same equation can be used for the measure of hydrophobicity by replacing the function $ch(s_i)$ with $h(s_i)$, reflecting the hydrophobicity index (Kawashima *et al.*, 1999) of the residue s_i.

In the implementation, the average charge and hydrophobicity and its standard deviation for a single amino acid is calculated using the natural amino acid frequencies from SWISS-PROT (Bairoch and Apweiler, 2000):

$$\mu_{ch} = \sum f_{aa} ch(aa)$$
$$\sigma_{ch}^2 = \sum f_{aa} (ch(aa) - \mu_{ch})^2$$

where f_{aa} is the natural frequency of the amino acid aa. The distribution of the charge and hydrophobicity measures for a sequence s of length n are therefore given by the normal distributions:

$$ch(s) = \mathcal{N}(n\mu_{ch}, n\sigma_{ch}),\ h(s) = \mathcal{N}(n\mu_h, n\sigma_h)$$

where n is the number of residues in s.

Given these distributions, $P^+(t)$ is the probability of the charge or hydrophobicity measure being lower than the observed measure. For positive charges and hydrophobicity, this is the cumulative density function (CDF):

$$\begin{aligned} P_{ch}^+(s) &= CDF(\mathcal{N}(n\mu_{ch}, n\sigma_{ch}), ch(s)) \\ P_h^+(s) &= CDF(\mathcal{N}(n\mu_h, n\sigma_h), h(s)) \end{aligned}$$

The probabilities for negative charges and hydrophilicity are provided by the complement. The scoring function for hydrophobic helices is the same as that for hydrophobic sequences, except that it uses a helix-specific hydrophobicity index.

3.4 Amphipatic alpha helix and beta sheet scoring

As seen with the charge and hydrophobicity scoring, the distribution of a numerical function measuring a character can be converted into a probability of a high or low occurrence of that character.

For the scoring of amphipatic alpha helices, the following function is used:

$$score_\alpha(s) = \sum_i h(s_i) \cos(100(i-1) + \Delta)$$

Likewise, for beta sheets:

$$score_\beta(s) = \sum_i h(s_i) \cos(180(i-1) + \Delta)$$

where $h(aa)$ is the relative hydrophobicity index for the amino acid aa normalized around 0.

The scoring function can be interpreted as follows: the hydrophobicity index of each residue is weighted

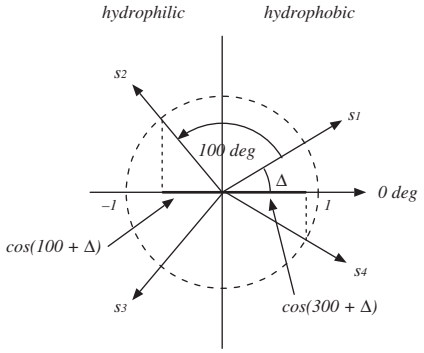

Fig. 1. α-helix score calculation: the hydrophobicity index of each residue s_i is multiplied by the cosine of its angle to the border normal plus an offset Δ.

according to its angle relative to the hydrophobic side. If a hydrophobic residue lands on the hydrophilic side, its weight will be negative. The weight of residues diminishes toward 0 as they approach the hydrophobic/hydrophilic border.

Δ is the angle of the normal to the hydrophobic/hydrophilic border relative to the first residue in the sequence and is chosen so as to maximize the score:

$$\Delta = -\arctan\left(\frac{\sum_i h(s_i)\sin(100(i-1))}{\sum_i h(s_i)\cos(100(i-1))}\right)$$

The scoring functions described above are non-linear and the distribution of the result values cannot be derived analytically as with the charge and hydrophobicity measures. However, a good measure for these distributions can be obtained by generating and evaluating a large number of random sequences and fitting these results to a distribution for any given sequence length n.

$P^+(\alpha)$ is then calculated from:

$$P^+_\alpha(s) = CDF(\mathcal{N}(\mu_{\alpha n}, \sigma_{\alpha n}), score_\alpha(s))$$

where $\mu_{\alpha n}$ and $\sigma_{\alpha n}$ are the parameters of the distribution derived for $score_\alpha(s)$ for sequences s of length n. Likewise, for beta sheets:

$$P^+_\beta(s) = CDF(\mathcal{N}(\mu_{\beta n}, \sigma_{\beta n}), score_\beta(s))$$

In general, any function describing a characteristic for a single residue or subsequence, provided the distribution of its results for random sequences can be modeled, can be adapted to a motif character.

3.5 Weighting

Some tokens, or characters within sequence tokens, might be more important than others. A weight for each partial score is introduced and reflected in the following alignment score:

$$score(s) = \sum_i w_i score(t_i)$$

which is expanded to:

$$score(s) = \sum_i \log(P^+(t_i)^{w_i})$$

The weighted probabilities can be interpreted as follows: given two events e_1 and e_2 with probabilities P_1 and P_2 with weights w_1 and w_2, the weighted probabilities $P_1^{w_1}$ and $P_2^{w_2}$ reflect the event e_1 being observed $\frac{w_1}{w_2}$ times more often than the event e_2. An event that occurs more often than others will therefore contribute more to the total score.

Although multiplying the weights by a constant factor will not change the classification results, they do however change the interpretation of the result. If the sum of the weights w_i is chosen such that it is the number of scoring tokens in the motif, the weighted score will reflect the total probability of the sequence matching all the tokens.

If the weights are chosen to sum 1, the weighted score becomes the geometric average of the weighted partial scores. This is the weighting normalization used for scoring by our algorithm, since it allows us to compare the scores for alignments of unequal motif length directly.

4 MOTIF ALIGNMENT

The motif is aligned to a sequence with dynamic programming in a way similar to classic sequence alignment.

The main difference to classic sequence alignment lies in the fact that one motif token can (or must) match more than one amino acid. Moreover, most of the scoring functions for sequence characters are not additive, yielding a somewhat higher algorithmic complexity.

Let M be the alignment matrix and $M[i, j]$ be the best alignment score for the sequence up to the amino acid at position $i-1$ and the motif up to token j. The first row is initialized with

$$M[i,0] = \begin{cases} 0 & i = 1 \\ -\infty & \text{otherwise} \end{cases}$$

For a token t_j being a fixed amino acid `A`, we then get:

$$M[i,j] = \begin{cases} M[i-1, j-1] & \text{if } s_{i-1} = \texttt{A} \\ -\infty & \text{otherwise} \end{cases}$$

Likewise, for a space token (_), we get:

$$M[i,j] = \begin{cases} M[i-1, j-1] & \text{if } i > 1 \\ -\infty & \text{otherwise} \end{cases}$$

and for the any-token (∗):

$$M[i, j] = \max_{k=1..i} M[k, j-1]$$

Note that the any-token is the only token that may have length 0.

Variable amino acid tokens (a) and signature tokens ({}) are aligned the same way as for classic sequence alignment:

$$M[i, j] = \begin{cases} M[i-1, j-1] + score_j(s_i) & i > 1 \\ -\infty & \text{otherwise} \end{cases}$$

where $score_j(aa)$ is partial scoring function for either signatures or variable amino acids.

So far there is no added complexity. This is, however, no longer the case with the alignment of the sequence tokens. The partial score for a sequence token over a partial sequence s is:

$$score_{seq}(s) = score_l(|s|) + \sum_c score_c(s)$$

or with weights:

$$score_{seq}(s) = w_l score_l(|s|) + \sum_c w_c score_c(s)$$

where the values of c are the different characters in the sequence token. And therefore for the alignment:

$$M[i, j] = \max_{k=1..i-1} score_{seq}(s_{k..i})$$

where $s_{k..i}$ is the subsequence from position k to position i in the sequence s. Consequently, the number of operations required to align a sequence token with a sequence of length n is no longer $\in O(n)$, as with the other tokens, but $\in O(n^2)$. For a motif with m tokens, the total number of operations is then $\in O(mn^2)$, compared to $O(n^2)$ for classical sequence alignments.

Considering that m is small compared to n, the increase in complexity may seem reasonably manageable. In the case of motif alignment, the constant factor for each operation consist of evaluating exponential and trigonometric functions. In classical sequence alignment, it is merely the cost of a table lookup, which is far less computationally expensive.

5 MOTIF REFINEMENT

For a given motif, the optimal weights, lengths and signatures are usually calculated using a training set of known positives and negatives. However, before discussing the optimization of each parameter, the results of a classification over a training set must be evaluated.

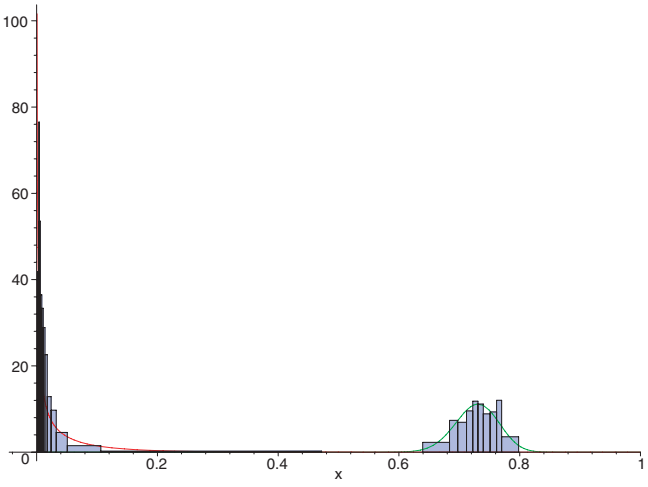

Fig. 2. Histograms for the scores of lipoprotein prediction over the entire *B. subtilis* genome with the fitted beta-distributions after bootstrapping.

5.1 Classification of results

S is the set of all the sequences in a given organism or a consistent collection. It can be partitioned into two subsets, S^+ and S^-, containing those sequences which should match the motif, and those who should not.

The positive and negative training sets, S_t^+ and S_t^- are selected randomly from S^+ and S^- respectively. Despite the possible lack of explicit criteria, a presumption on the partition of S can be made allowing S_t^+ and S_t^- as sets of sequences which *supposedly* belong to S^+ and S^- respectively. It should be noted that the training set constitutes the hypothesis—being that the sequences are unbiased and randomly selected—on which we build the classification.

Since we will be refining the motif to better describe the classification, much care must be taken to select S_t^+ and S_t^- according to criteria independent of the motif we are looking for. Otherwise we will be introducing a bias, therefore only reinforcing our initial assumptions.

If S_t^+ and S_t^- are chosen randomly from S^+ and S^-, then the distribution of the alignment scores should reflect the distributions for the yet unknown sets S^+ and S^- (Figure 2). The alignment scores, being probabilities, are assumed to fit a beta-distribution.

Given a cutoff value c, the discernibility of this value is the probability of classifying the sequence correctly, assuming the scores for S^+ and S^- are distributed as $\mathcal{B}(\mu_t^+, \sigma_t^+)$ and $\mathcal{B}(\mu_t^-, \sigma_t^-)$:

$$disc(c) = \frac{|S^-|}{|S|} CDF(\mathcal{B}(\mu_t^-, \sigma_t^-), c) +$$

$$\frac{|S^+|}{|S|}(1 - CDF(\mathcal{B}(\mu_t^+, \sigma_t^+), c))$$

assuming $\mu_t^+ > \mu_t^-$. $|S^+|$ and $|S^+|$ are initially assumed to be equal. The sizes of the expected sets of false positives and false negatives, F^+ and F^- respectively, can then also be calculated with:

$$E(|F^+|) = |S^-|(1 - CDF(\mathcal{B}(\mu_t^-, \sigma_t^-), c))$$
$$E(|F^-|) = |S^+|CDF(\mathcal{B}(\mu_t^+, \sigma_t^+), c)$$

for a given cutoff c.

Finally, the maximum discernibility over $\mathcal{B}(\mu_t^+, \sigma_t^+)$ and $\mathcal{B}(\mu_t^-, \sigma_t^-)$ defines the classification score:

$$score = \max_c disc(c)$$

The cutoff value c is then used for classifying the sequences in S. If the parameters of the two distributions for the positive and negative scores are known, the relative probability of belonging to one distribution or another can also be calculated.

5.2 Length and signature refinement

When all sequences in S_t^+ are aligned, the length of each sequence token is adjusted to maximize the alignment score with:

$$\mu_{new} = \arg\max_{\mu} \left[\sum_{s \in S_t^+} score_l(s) \right]$$

where $score_l(s)$ is the length score for the given token and μ over the sequence s.

Since the signatures should represent the frequency of the amino acids in the positive sequences, and S_t^+ was chosen randomly from S^+, frequencies are recalculated from the observed frequencies in S_t^+.

It should be noted that refining signatures in this way restricts the motif to allow only residues which were matched in the training set. Therefore, an ill-defined or biased training set can, through signature refinement, greatly influence the outcome of predictions.

5.3 Weight refinement

Sequence classification is scored as a function of the overlap between the distributions $\mathcal{B}(\mu_t^+, \sigma_t^+)$ and $\mathcal{B}(\mu_t^-, \sigma_t^-)$. The weights are therefore optimized to tighten the standard deviations of the two score distributions, thereby increasing discernibility, as opposed to minimizing the number of mis-classifications in S_t^+ and S_t^-, as is done in most of the literature on classification problems.

The alignment can be approximated as a linear system of equations:

$$\mathbf{Ax} = \mathbf{b}$$

$$\begin{bmatrix} \frac{1}{m_+} & \frac{s_{1,1}}{m_+} & \cdots & \frac{s_{1,n}}{m_+} \\ \vdots & & & \\ \frac{1}{m_+} & \frac{s_{m_+,1}}{m_+} & \cdots & \frac{s_{m_+,n}}{m_+} \\ \frac{1}{m_-} & \frac{s_{m_++1,1}}{m_-} & \cdots & \frac{s_{m_++1,n}}{m_-} \\ \vdots & & & \\ \frac{1}{m_-} & \frac{s_{m_++m_-,1}}{m_-} & \cdots & \frac{s_{m_++m_-,n}}{m_-} \end{bmatrix} \times \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \frac{\mu_+}{m_+} \\ \vdots \\ \frac{\mu_+}{m_+} \\ \frac{\mu_-}{m_-} \\ \vdots \\ \frac{\mu_-}{m_-} \end{bmatrix}$$

Fig. 3. The system of equations for the partial scores of S_t^+ and S_t^- over n tokens for the target scores μ_+ and μ_-.

where each row of the matrix **A** represents the partial scores of one sequence, **x** the token weights and **b** a vector containing the targeted μ (i.e. the average alignment score for a given set). The norm of the residue **r** given by:

$$\mathbf{Ax} - \mathbf{b} = \mathbf{r}$$

is the standard deviation σ^2 of the observed *normally distributed* results around μ. If $\|\mathbf{r}\|_2$ is minimized over **x**, the weights that give the tightest σ^2 are selected.

This problem can easily be expanded for more than one distribution by weighting each line in **A** and **b** with m^{-1}, where m is the number of equations for a given target μ ind **b**. The result is then optimal for the given targets.

The only problem remaining is the choice of adequate target μ_+ and μ_- values. If, however, a constant vector to **A** and a new weight w_0 are added, the same residue $\|\mathbf{r}\|_2$ (and therefore σ^2) is obtained for any pair of μ_+ and μ_- with constant difference $\mu_+ - \mu_-$ (for consistency, the values are taken from the distribution of the scores for S_t^+ and S_t^-) and the same weight vector **x** (albeit a constant factor) for any pair of μ_+ and μ_- with constant $\frac{\mu_+}{\mu_-}$.

The system of equations can be seen in Figure 3.

After minimizing the residual, the weights are then reapplied to the respective tokens. w_0 is ignored, since it has no effect on the classification score $\max_c disc(c)$ (a constant shift of scores does not change $\mu_+ - \mu_-$ or the variances).

It should be noted, at this point, that we are minimizing the amount of overlap between the normal distribution of the sums of the partial scores of each sequence, and not the overlap of the Beta-distributions of the exponential of that sum. The results of the optimization are therefore only an *approximation* of the optimal weights.

5.4 Alternative weight refinement

Although the method mentioned above will find the best parameter weighting for any given alignment, it does not easily converge toward a global minimum for bad initial

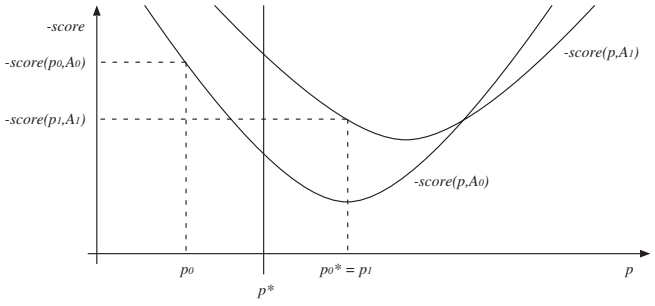

Fig. 4. The best parameter p for one alignment is not always valid after realignment (for $p > p^*$).

conditions (i.e. the initial motif). Therefore, an alternate, entropy-based weight refinement is introduced.

The distribution of the partial scores for the positive and negative training sets, $\mathcal{B}(\mu_i^+, \sigma_i^+)$ and $\mathcal{B}(\mu_i^-, \sigma_i^-)$ can be calculated for each token or sequence character t_i. From these distributions the weight is then:

$$w_i = -\log\left(1 - disc(\mathcal{B}(\mu_i^+, \sigma_i^+), \mathcal{B}(\mu_i^-, \sigma_i^-))\right)$$

where $disc()$ is the discernibility equation from the result classification. If any parameter achieves perfect discernibility, it gets a score of ∞, since it would perfectly classify the training set regardless of the other parameters.

When close to a local maximum, this weighting method does not converge as fast as the least squares method does. However, in the early stages of refinement, it moves toward the global maximum very quickly.

5.5 Adaptive parameter adjustment

Any adjustment to the motif parameters is likely to change the partial scores in the alignments and therefore the alignments themselves. As a result, the parameters have to be changed adaptively.

To optimize a parameter p (either a weight or a length) over an alignment A. The parameter is initialized at p_0, resulting in the alignment A_0 and the alignment score $score(p_0, A_0)$.

After refinement, the new optimal value p_0^* for the original parameter (relabeled p_1), which is used to create a new alignment A_1 which in turn gives a new score $score(p_1, A_1)$. If, as shown in Figure 4, the alignment switches from A_0 to A_1 at p^*, and $score(p_1, A_1) < score(p*, A_0)$ then an optimal solution, namely that at $p > p^*$, is missed.

To avoid getting stuck, our parameter p can be updated adaptively according to an adaptivity coefficient a:

$$p_{n+1} = \frac{p_n(a-1) + p_n^*}{a}$$

For $a = 1$, the new parameter p_{n+1} is set to p_n^*, for $a = 2$ the distance between p_n and p_n^* is divided by 2, and so forth. The higher the value for a, the closer p^* is approximated. A high value for a, however, also reduces exploration of the score surface and converges rather slowly. It is therefore recommended to start with $a = 1$ and to increase a whenever realignment does not improve the score.

It should also be noted that changes in the parameters cause changes in the alignment. As a result, parameters can only be optimized within a single alignment, with no guarantee of actually finding the optimal parameters, and therefore, the optimal classification.

5.6 Refining over a training Set

The motif is refined iteratively, one set of parameters (signatures, lengths and weights) at a time, and keeping the refinements only if the classification score increases. The refinement procedure is detailed in Algorithm 1.

The main loop performs one refinement of the signatures, lengths and weights per pass, keeping the resulting refined motif only if the classification score is improved. If no improvement is achieved at the end of a pass, the adaptivity coefficient a is increased. a is set back to 1 every time the motif is changed.

The loop continues until no improvement is made and the adaptivity coefficient has reached some maximum value a_{max}.

5.7 Bootstrapping

Once a motif has been refined over a training set, it can be applied to the test data, resulting in the hypothetical partition into S_h^+ and S_h^-. If we assume—or have good reason to believe that—the partition represents the characteristics we want to classify by, then we can re-refine our motif over these two sets. This process can be repeated until a stable motif and stable sets S_h^+ and S_h^- are achieved.

The idea behind bootstrapping is rather simple: assuming a classification is given by the presence of one of the two characters A and B. If S_t^+ contains only members with A, the motif will be refined toward this character only, and after classification, the sequences with A and AB will be predicted. If bootstrapping is performed—and the character B is a good discriminator—the sequences containing B will appear as false positives, provided they do not represent a too large part of S and we are relying on a classification method not based on the number of false positives/negatives.

6 RESULTS

The method was tested with a motif characterizing lipid-anchored proteins (lipoproteins) in *Bacillus subtilis*. The results of the prediction were then compared to the

Algorithm 1 Motif Refinement

Require: Training sets S_t^+ and S_t^-, motif m, maximum adaptivity a_{max}
$A = align(m, S_t)$ $\{S_t = S_t^+ \cup S_t^-\}$
$a = 1$ {initial adaptivity}
$s_{old} = s = score(A)$ {classification score}
while $s < 1$ **do**

{refine signatures}
$m_{new} = refine\_signatures(m, A, a)$
$A_{new} = align(m_{new}, S_t)$
$s_{new} = score(A_{new})$
if $s_{new} > s$ **then**
$m = m_{new}, A = A_{new}, s = s_{new}, a = 1$
end if

{refine lengths}
$m_{new} = refine\_lengths(m, A, a)$
$A_{new} = align(m_{new}, S_t)$
$s_{new} = score(A_{new})$
if $s_{new} > s$ **then**
$m = m_{new}, A = A_{new}, s = s_{new}, a = 1$
end if

{refine weights}
$m_{new} = refine\_weights(m, A, a)$
$A_{new} = align(m_{new}, S_t)$
$s_{new} = score(A_{new})$
if $s_{new} > s$ **then**
$m = m_{new}, A = A_{new}, s = s_{new}, a = 1$
end if

{need we continue?}
if $s = s_{old}$ **then**
if $a < a_{max}$ **then**
$a = 2a$
else
break
end if
else
$s_{old} = s$
end if

end while

sequences selected in Tjalsma *et al.* (2000) and the sequences identified by the PROSITE (Hofmann *et al.*, 1999) entry PS00013.

6.1 Selection of training sets

Although *B. subtilis* is considered as a model organism, rather few experimentally confirmed lipoproteins are known. Out of all the *B. subtilis* entries in SWISS-PROT (Bairoch and Apweiler, 2000), only 39 contain the keyword 'Lipoprotein'. Of these 39, 17 are hypothetical sequences. Out of the remaining 22, only 3 (OPPA_BACSU, SLP_BACSU and QOX2_BACSU) have confirmed cleavage sites (not marked as 'Putative', nor 'Potential' nor 'By Similarity').

The positive training set was defined as the 22 non-hypothetical SWISS-PROT (release 40) entries with the keyword 'Lipoprotein'.

All other *B. subtilis* entries in SWISS-PROT cannot be included in the negative training set since the absence of annotation cannot be equated to the absence of cleavage site. To avoid the unintentional inclusion of false-positives as much as possible, only non-hypothetical sequences not containing the keyword 'Signal'—1300 entries in all—were considered.

6.2 Initial motif

According to Tjalsma *et al.* (2000), the lipoprotein signal in *Bacillus subtilis* corresponds to:

```
M [p,4:1](2,10) [h{},12:1](8,25)
{} _ {} C {} *
```

The motif can be read as starting with a Methionine residue (`M`), followed by a positively charged region of length 2 to 10 (`[p,4:1](2,10)`), followed by a hydrophobic trans-membrane helix of length 8 to 25 (`[h{},12:1](8,25)`), followed by a residue characterized by a frequency vector ({}), followed by any single residue (_), followed by another frequency vector ({}), a fixed Cysteine residue (`C`), a final frequency vector ({}) and the rest of the sequence (`*`).

The hydrophobic trans-membrane helix is characterized by the symbol `h` which is nothing other than the hydrophobicity function mentioned earlier, yet with hydrophobicity indices specific to trans-membrane helices (Kawashima *et al.*, 1999). Since this character alone is not sufficient to detect trans-membrane segments faithfully, a frequency vector was added, which drastically increases detection capabilities.

6.3 Motif refinement

A first refinement was started with the initial motif described above. After 20 rounds, using $\epsilon = 10^{-5}$ for profile scoring, a classification score of 99.960% with a cutoff value of 54.990% was achieved.

All of the sequences in the positive training set were classified correctly. Two sequences from the negative training set, PBPC_BACSU (score 77.833%) and GERM_BACSU (score 65.529%) were misclassified. These two sequences scored exceptionally high—the next-highest score in the negative set was 51.097%. Examination of annotations did not show any conclusive information as to the precise role of the corresponding proteins however both are expected to interact with the membrane: PBPC_BACSU as a penicillin-binding protein

and GERM_BACSU as playing a putative role in peptidoglycan synthesis during sporulation (Moszer *et al.*, 1995). Consequently, the two sequences were excluded from the negative training set.

A second run was performed with the new negative training set and a classification score of 99.999% was reached with a cutoff value of 49.054% after 21 rounds. The lowest score in the positive training set was 61.707% and the highest in the negative set 21.756%. The distributions of the alignment scores in shown in Figure 2.

6.4 Jackknife testing

To validate the classification results, further training was undertaken with all possible subsets of the initial positive training set with one sequence removed. The removed sequences were then classified according to the newly generated motif. All but two sequences were correctly classified as lipoproteins. GERD_BACSU, when removed from the training set, was misclassified, likewise for LPLA_BACSU, since both these sequences are unique examples of Valine and Isoleucine respectively in position C-3.

6.5 Bootstrap and prediction

To improve the quality of prediction of the refined motif, bootstrapping was performed over all possible *Bacillus subtilis* sequences. To that end, the 4106 sequences of the SubtiList database (Moszer *et al.*, 1995) were filtered to select the 866 containing the required Methionine (position 1) and Cysteine (position between 15 and 40) residues.

With the larger positive training set, a frequency vector was added to the second token (positively charged region).

After 2 iterations, the bootstrap converged resulting in 65 predicted lipoproteins with a classification score of 99.996% and a cutoff at 61.468% (highest scoring negative: AMYC_BACSU, 47.699%). The refined motif is thus:

```
M 0.509[1.078p0.684{A=1.9,R=13.4,N=2.3,Q=1.4,
H=2.3,I=4.6,L=6.0,K=51.9,M=1.9,F=2.3,S=3.2,
T=3.2,W=0.5,Y=2.8,V=2.3},4.0:1.0](2,10)
0.865[1.155h1.837{A=13.4,N=0.1,C=2.7,Q=0.3,
G=4.0,I=11.0,L=29.5,M=6.8,F=9.0,P=1.0,S=5.8,
T=5.3,W=1.0,Y=0.1,V=10.0},12.0:1.0](8,25)
0.589{A=4.6,I=6.2,L=78.5,V=10.8} -
0.854{A=47.7,G=52.3} C 0.635{A=4.6,G=44.6,
S=36.9,T=6.2,W=7.7} *
```

The predicted lipoprotein sequences and their motif alignments are listed in Table 2.

6.6 Comparison to PROSITE motif PS00013

The PROSITE (Bairoch, 1991) pattern database contains an entry (PS00013) describing prokaryotic lipoproteins. The pattern, in the PROSITE syntax, is:

Table 2. Prediction results for *B. subtilis* lipoproteins with the signal peptides color-coded according to their motif alignment. Sequences marked with a [1] were used as the positive training set, those marked with [2] were not detected by PROSITE pattern PS00013 and those marked with [3] were not present in Tjalsma *et al.* (2000)

ID	Score	Motif Alignment
LytA[1]	79.92613	M KK FIALLFFILL L S G C G VNS...
YtkA[2]	79.07044	M KK MLVVLLFSALL L N G C G SGE...
SsuA[1]	77.55045	M KK GLIVLVAVIFL L A G C G ANG...
YjhA	77.16516	M KK VLLLLFVLTIGLA L S A C S QSS...
YqiX	76.69412	M KK WLLLLVAACITFA L T A C G SSN...
YfjL	76.64932	M KK LVFGLLAIV L F G C G LYI...
GerAC[1]	76.40709	M KIR ILCMFICTLL L S G C W DSE...
YckK	76.37906	M KK ALLALFMVVSIAA L A A C G AGN...
OppA[1]	76.25113	M KKR WSIVTLMLIFTLV L S A C G FGG...
GlnH	76.09330	M KK IFSLALISLFAVIL L A A C G SKG...
AppA[1]	76.01213	M KRRK TALMMLSVLMVLAIF L S A C S GSK...
YdhF	75.92298	M RR ILSILVFAIM L A G C S SNA...
RbsB[1]	75.64796	M KK AVSVILTLSLFL L T A C S LEP...
YxeM	75.39458	M KMKK WTVLVVAALLAV L S A C G NGN...
YddJ[3]	75.12139	M KN LFIFLSLMMMFV L T A C G GSK...
YvrC	75.11922	M KKR AGIWAALLLAAVM L A G C G NPA...
GerKC[1,2]	75.10202	M VRK CLLAVLMLLSVIV L P G C W DKR...
MsmE[2]	74.98970	M KH TFVLFLSLILLV L P G C S AEK...
YncB	74.88917	M KK ILISMIAIVLSIT L A A C G SNH...
YxeB	74.79640	M KKN ILLVGMLVLLLMF V S A C S GTA...
PbpC	74.60154	M LKK CILLVFLCVGLIG L I G C S KTD...
AraN	74.43983	M KK MTVCFLVLMMLLTLV I A G C S AEK...
YerH	74.15124	M KK TLALAATAAVLM L S A C S SGF...
YclQ	74.10789	M KK FALLFIALVTAVV I S A C G NQS...
OpuCC[1]	73.87573	M TKIK WLGAFALVFVML L G G C S LPG...
YokF	73.78656	M KK VLLGFAAFTLSLS L A A C S SND...
YurO	73.73391	M KK MLLFLIIAAVSMLT I A G C S SQS...
YqgG	73.57028	M KKNK LVLMLLMAAFMMI A A A C G NAG...
YtmK	73.33743	M KTK TAFMAILFSLITV L S A C G AGS...
CccB	73.29458	M KSK LSILMIGFALSVL L A A C G SND...
Med[1,3]	73.06732	M ITR LVMIFSVLLL L S G C G QTP...
YutC	72.98184	M KR TAVSLCLLTGL L S G C G GAG...
SpoIIIJ[1]	72.87161	M LLKRR IGLLLSMVGVFML L A G C S SVK...
GerBC[1]	72.43715	M KTASK FSVMFFMLLA L C G C W DVK...
YojM	72.34194	M HR LLLLMMLTALG V A G C G QKK...
YfkR[3]	72.15590	M KKTIYK CVLPLLICIL L T G C W DRT...
YqiH	72.12229	M KQ TVLLLFTALF L S G C S VAS...
FeuA[1]	72.04993	M KK ISLTLLILLLALT A A A C G SKN...
CtaC[1]	72.03121	M VKHWR LILLLALVPLL L S G C G KPF...
YcdH	71.61163	M FKK WSGLFVIAACFLL V A A C G NSS...
YciB	71.58876	M KL SLFIIAVLMPVIL L S A C S DHA...
YpmR	71.45023	M KLR IFSIMASLILL L T A C T SIR...
OpuBC[1]	71.24445	M KRKYLK LMIGLALAATLT L S G C S LPG...
YvfK	71.08584	M KMAKK CSVFMLCAAVSLS L A A C G PKE...
PrsA[1]	71.06691	M KK IAIAAITATSILA L S A C S SGD...
Slp[1]	70.87476	M RYR AVFPMLIIVFA L S G C T LST...
YvgL	70.60146	M FKKY SIFIAALTAFLL V A G C S SNQ...
YscB[3]	70.59775	M NK LIQLALFFTLM L T G C S NSS...
YpmQ	70.44568	M KVIK GLTAGLIFLF L C A C G GQQ...
YxkH	70.42447	M KR LFLSIFLLGSCLA L A A C A DQE...
OpuAC[1]	70.39505	M LKK IIGIGVSAMLALS L A A C G SEN...
FhuD[1]	69.67573	M THIYKK LGAAFFALLLIAA L A A C G NNS...
YtgA	69.43152	M RQ GLMAAVLFATFA L T G C G TDS...
YndF[2]	68.84606	M KSKLKRQ LPAMVIVCLLMIC V T G C W SSR...
DppE[1]	68.51677	M KRVKK LWGMGLALGLSFA L M G C T ANE...
YfmC	68.30495	M RTYSNK LIAIMSVLLLACLI V S G C S SSQ...
LplA[1]	67.97288	M KIRMRKK WMALPLAAMM I A G C S HSE...
YckB	67.68721	M KSFMHSK AVIFSFTMAFFLI L A A C S GKN...
YvdG	67.12830	M VLLKK GFAILAASFLAIG L A A C S SSK...
GerM	67.04662	M LKK GPAVIGATCLTSALL L S G C G LFQ...
YdeJ	66.73849	M KKRRK ICYCNTALLLMIL L A G C T DSK...
YybP[3]	66.02677	M KI ILTVLAGVGLLS A G G C G MLD...
GerD[1]	65.88020	M SKAK TLLMSCFLLLS V T A C A PKD...
YlaJ	65.12331	M RI LFIIIQLTLI L S A C A YQQ...
QoxA[1]	64.57110	M IFLFRALK PLLVLALLTVVFV L G G C S NAS...

```
{DERK}(6)-[LIVMFWSTAG](2)-[LIVMFYSTAGCQ]-[AGS]-C
```

which can be interpreted as a sequence of non-charged residues of length 6, two hydrophobic residue, one non-polar residue, one residue of either A, G or S and a final Cysteine residue. Furthermore, to be accepted, the final Cysteine residue must lie between positions 15 and 35 and there must be a positively charged residue within the first 7 residues of the sequence.

In SWISS-PROT release 40, only 37 *B. subtilis* entries are annotated as containing this pattern, although it is detected in 97 sequences. Within the annotated sequences, we find the following cases:

```
YhcN: M FGKK QVLASVLLIPLL M T G C GVA ...
YbbD: M RPVF PLILSAVL F L S C FFG ...
```

YhcN, contains a Methionine residue at the position C-3. In comparison with the residues observable at this position in positive instances (A, I, L and V), Methionine has no common characteristic.

YbbD, although being annotated as a lipoprotein in SWISS-PROT, has highly unlikely residues C-1 and C-3 and two unexpected Proline residues in the positively charged and hydrophobic regions.

Finally, there are 4 sequences detected by our method (YtkA, GerKC, MsmE and YndF, see Table 2) which are not detected by PROSITE.

The INTERPRO (Apweiler *et al.*, 2001) database was searched as well, but no additional information could be extracted.

6.7 Comparison to results by Tjalsma *et al.*

In their paper (Tjalsma *et al.*, 2000), Tjalsma and co-workers looked at peptide-dependent transport in *Bacillus subtilis* and identified 114 probable lipoproteins. The detection was a simple sequence similarity search, which results were filtered by hand using characteristic regions in a similar way the motif is described in the present paper.

The motif defined herein detected 6 lipoproteins missing in Tjalsma *et al.* (2000) despite the loose criteria applied in the filtering step (see Table 2).

7 DISCUSSION

As seen from the results, the motif alignment algorithm presented here discriminates well between positive and negative training sets. The jackknife-test confirms there is no over-fitting.

Table 2 shows that the lipoprotein signals do not align well one to another. The length variation of the hydrophobic core and the variable number of charged residues are not quite suited to the definition of a consensus pattern. As seen above, the fluctuation of the Cysteine residue between position 15 and 40 imposes the introduction of quite a number of gaps in the multiple alignment of all signals. A regular expression can be used to express these loose positional constraints but the definition of a profile remains difficult. A similar situation is dealt with in the SignalP program (Nielsen *et al.*, 1997) while training several independent neural networks (NN) and allowing each amino acid to be weighted. However no explicit information on the relative contribution of the various parts of the signal is generated.

Furthermore, in contrast to NN and hidden Markov models (HMM) methods, the biological interpretation of the results is maintained throughout the training process and in the refined motif, since the motifs are based strictly on secondary characteristics. The level of abstraction from the residues themselves in the motif definition (i.e. through the use of characteristics such as relative charge and hydrophobicity) also greatly reduces the risk of residue-specific over-fitting.

A similar, yet more abstract approach is taken in Gannai *et al.* (2001), where an algebra for defining features for machine learning is presented: the main construct of this algebra are *views*, which represent functions on a sequence. The functions are constructed from predefined parameterized *view operators*, i.e. the subsequence operator $\mathcal{S}_{i,j}$, the indexing operator $\mathcal{I}_i$ and the pattern matching operator $\mathcal{P}_{i,A}$. Once a set of views has been defined for a given pattern or signal, it is then optimized over its parameters using the Mathews correlation coefficient (Mathews, 1975) as a metric. The view operators used in the prediction, however, appear again too position specific and may lead to over-training.

In most methods mentioned above, the emphasis is put on positional constraints on amino acids. The introduction of physico-chemical descriptors is recent and was initiated in the composite prediction functions defined in Eisenhaber *et al.* (1999), though the method presented here is statistically sounder. In Eisenhaber *et al.* (1999) the functions are based on the distribution of the observed feature values but not on their probabilities of occurring randomly. Consequently, the underlying statistics focus on converging toward the positive case whereas our method is set to discriminate from the average case. Finally, Eisenhaber *et al.* (1999) lacks the flexibility introduced by combining the scoring function and the signal alignment.

In general, though, it is quite difficult to compare prediction methods qualitatively. Although much effort was invested into detection and prediction methods themselves, less is done on the analysis of the results produced. The question as to which classification score should be used (Precision/Recall, Mathews Correlation Coefficient, distribution overlap, etc. . .) has not been subject to much discussion. Consequently, authors are tempted to choose the scoring method to best suit their results, and not according to any qualitative argument. A good discussion on the topic is given in Wootton (1997).

In the present paper, we attempted to quantify how a sequence can fit a motif description. An alignment maximizing the alignment score is selected from all possible alignments through dynamic programming. The selection is therefore not rule-based and does not refer to an underlying grammar although a syntax is defined for the motifs.

8 PERSPECTIVES

8.1 Motif prediction

Throughout this paper, we have only discussed aligning sequences with known motifs. Although motif prediction cannot be done by alignments alone, the nature of the motifs—a linear chain of tokens with variable non-discrete parameters—makes them quite appealing for genetic programming, allowing point mutations (variation of a numerical parameter), indel events (insertion or deletion of a token) and crossovers between motifs.

8.2 Alternative classification

There is a wealth of literature available on the topic of classification. Some of the better known classification schemes which optimize the weights in a linear sum include the Logistics Function (Jordan, 1995), Fisher Linear Discriminant (Fisher, 1936), Cross Entropy and LinearClassify (Gonnet, 2001).

It should be noted that none of the above mentioned methods, including the one used in the algorithm, take into account that the motif will have to be realigned with the sequences.

ACKNOWLEDGEMENTS

We would like to thank the reviewers for their helpful comments as well as Alexandre Gattiker from the Swiss Institute of Bioinformatics for his work on the web interface of the program.

REFERENCES

Apweiler,R., Attwood,T.K., Bairoch,A., Bateman,A., Birney,E., Biswas,M., Bucher,P., Cerutti,L., Corpet,F., Croning,M.D.R. *et al.* (2001) InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Res.*, **29**, 37–40.

Bairoch,A. (1991) PROSITE: a dictionary of site and patterns in proteins. *Nucleic Acids Res.*, **19**, 2241–2245.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.*, **28**, 45–48.

Baldi,P. and Brunak,S. (1998) *Bioinformatics: the Machine Learning Approach*. MIT Press, Cambridge, MA.

Brendel,V. and Karlin,S. (1989) Association of charge clusters with functional domains of cellular transcription factors. *Proc. Natl Acad. Sci. USA*, **15**, 5698–5702.

Dayhoff,M.O., Schwartz,R.M. and Orcutt,B.C. (1979) A model of evolutionary change in proteins. In Dayhoff,M.O. (ed.), *Atlas of Protein Structure*. National Biomedical Reasearch Foundataion, Silver Spring, MD, **5(Suppl. 3)**, pp. 345–352.

Eisenhaber,B., Bork,P. and Eisenhaber,F. (1999) Prediction of potential GPI-modification sites in proprotein sequences. *J. Mol. Biol.*, **292**, 741–758.

Fisher,R.A. (1936) The use of multiple measures in taxonomic problems. *Ann. Eugenics*, **7**, 179–188.

Gannai,H., Tamada,Y., Maruyama,O., Nakai,K. and Miyano,S. (2001) Views: fundamental building blocks in the process of knowledge discovery. *Proceedings of the 14th International FLAIRS Conference*. AAAI Press, Menlo Park, CA, pp. 233–238.

Gascuel,O. and Danchin,A. (1986) Protein export in prokaryotes and eukaryotes: indications of a difference in the mechanism of exportation. *J. Mol. Evol.*, **24**, 130–142.

Gonnet,G. (2001) LinearClassify, As implemented and described in the Darwin system for computational biology, Swiss Federal Institute of Technology (ETH) Zurich, http://cbrg.ethz.ch/Darwin.

Hofmann,K., Bucher,P., Falquet,L. and Bairoch,A. (1999) The PROSITE database, its status in 1999. *Nucleic Acids Res.*, **27**, 215–219.

Jordan,M.I. (1995) Why the Logistic Function? A Tutorial Discussion on Probabilities and Neural Networks, Technical Report 9503, Massachusetts Institute of Technology, Computational Cognitive Science.

Kawashima,S., Ogata,H. and Kanehisa,M. (1999) AAindex: amino acid index database. *Nucleic Acids Res.*, **27**, 368–369.

Mathews,B.W. (1975) Comparison of predicted an observed secondary structure of t4 phage lysozyme. *Biochim. Biophys. Acta*, **405**, 442–451.

Moszer,I., Glaser,P. and Danchin,A. (1995) SubtiList: a relational database for the *Bacillus subtilis* genome. *Microbiology*, **141**, 261–268.

Nielsen,H., Engelbrecht,J., Brunak,S. and von Heijne,G. (1997) Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Eng.*, **10**, 1–6.

Tjalsma,H., Bolhuis,A., Jongbloed,H.D.H., Bron,S. and Maarten van Dijl,J. (2000) Signal peptide-dependent protein transport ind *Bacillus subtilis*: a genome-based survey of the secretome. *Microbiology and Molecular Biology Reviews*, **64**, 515–574.

Wootton,J.C. (1997) Evaluating the effectiveness of sequence analysis algorithms using measures of relevant information. *Computers Chem.*, **21**, 191–202.