

Performance comparison of generalized PSSM in signal peptide cleavage site and disulfide bond recognition

P. Clote*

Abstract

We generalize the familiar position-dependent position-specific score matrix (PSSM), aka weight matrix, method by considering a log-odds score for (nonadjacent) k -tuple frequencies, each k -tuple score weighted by the product of its mutual information and its statistical significance, as measured by a point estimator for the p -value of the mutual information. Performance of this new approach, along with other variants of generalized PSSM and profile methods, is measured by receiver-operating characteristic (ROC) curves for the specific problem of signal peptide cleavage site recognition. We additionally compare Vert’s recent support vector machine string kernel [29], Brown’s joint probability approximation algorithm [7, 4, 18] and the method WAM [31].

Similar algorithm comparisons are made in the case of disulfide bond recognition. While in the case of signal peptide cleavage site recognition, the monoresidue PSSM is essentially competitive, within the limits of statistical significance, even against Vert’s support vector machine kernel, diresidue and triresidue PSSM display dramatically improved performance over monoresidue PSSM for disulfide bond recognition. It follows that the choice of algorithm is heavily dependent on the data to which the classification algorithm is applied.

Introduction

One of the most important tasks in computational biology is to develop machine learning methods to solve bioinformatics *classification* problems – i.e. to determine algorithmically whether a given amino acid or nucleotide sequence belongs to a particular class of interest. Consensus sequences, profiles [14], weight matrices [27, 5, 17], neural networks [1, 24], higher-order Markov chains [3], hidden Markov models [20, 2, 1, 9], stochastic context-free grammars [10] and support vector machines [19, 29] are all examples of such techniques with such diverse applications as the detection of genes, promoters, tRNA genes, splice sites, binding sites, etc.

In this paper, we present a new method, designated WMMIP for (generalized k -tuple) *weight matrix with mutual information and p -values*, and compare the performance of our algorithm with a variety of other methods, by means of receiver-operating characteristic (ROC) curves [15].

1 Signal peptide cleavage site recognition

The 1999 Nobel Prize in Physiology or Medicine was awarded to G. Blobel for the discovery that “proteins have intrinsic signals that govern their transport and localization in the cell.” One well-

*Supported in part by Research Incentive Grant from Boston College.

Department of Biology, Boston College, Fulton Hall 410 B, Chestnut Hill MA 02467, clote@bc.edu (courtesy appointment in Computer Science)

Key words: position-specific scoring matrix, weight matrix, kernel, support vector machine (SVM), mutual information, p -value, signal peptide.

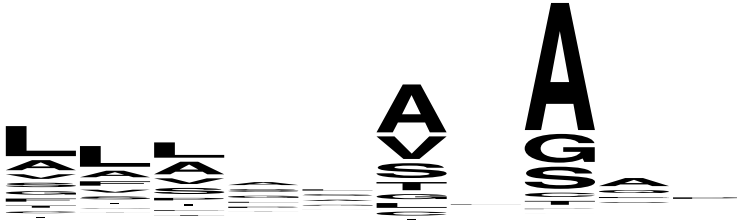


Figure 1: Logo plot, $-8, -7, \dots, -1, 1, 2$ of signal peptide cleavage sites

documented transport signal concerns the *signal peptide*, which forms the N-terminal portion of an amino acid sequence which is secreted through the cell membrane, the mature protein being formed after cleavage of the initial signal peptide. Blobel's experimental work supports the fact that the signal peptide cleavage site is determined by a signal in the amino acid sequence – c.f. Figure 1, which presents a logo plot [26] (position-specific relative entropy) for the region $-8, -7, \dots, -1, 1, 2$ preceding and succeeding the cleavage site in the signal peptide data of [28]. See [25] for a recent discussion of the role of signal peptides in the secretion mechanism.

In 1986, von Heijne [30] applied the weight matrix method to algorithmically detect signal peptide cleavage sites. In 1997 Nielsen et al. [24] applied neural networks to this recognition problem, provided a web engine for their algorithm, and equally importantly, made available the signal peptide data (consisting of an N-terminal initial segment, where the signal portion, cleavage site and initial portion of mature protein are indicated) (see [28]). In 2002, Vert [29] developed a new string kernel for the support vector machine (SVM) and compared the performance of SVM with the weight matrix method.

2 Disulfide bond recognition

A *disulfide bond* is a covalent bond between the sulfur atoms of two cysteine residues in a protein (bond length, as measured from PDB files, is around 2.04 Angstroms). Such bonds are extremely stabilizing forces for proteins, and in fact, protect certain bacteria from the otherwise lethal effect of even low concentrations of hydrogen peroxide. Two cysteine residues in a disulfide bond form a *cystine* molecule, and each partner cysteine is called a *half-cysteine*. While algorithmic prediction of whether a cysteine is a half-cysteine is reasonably good, often relying on neural networks [13], it is much more difficult to determine the cysteine partners involved in a disulfide bond. P. Fariselli and R. Casadio [12] have applied a neural net together with maximum weight matching to predict cysteine partners in disulfide bonds. (We are currently transforming the Fariselli-Casadio algorithm

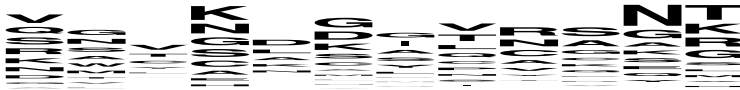


Figure 2: Logo plot, neighborhoods $(-3, -2, -1, 1, 2, 3)$ and $(-3, -2, -1, 1, 2, 3)$ of 2 cysteines in a disulfide bond

to a form suitable for comparison using ROC curves to the current approach. The results will be discussed in another paper.)

2.1 Network flow and matching

From experimental work in G. Church’s lab [6], it is clear that the assumption of positional independence required by the weight matrix method¹ is not valid – there is covariation between different (not necessarily adjacent) positions i, j . In light of this fact, it is natural to consider the log odds score using k -tuple frequencies, each tuple-dependent log odds score weighted in such a manner that higher weights are attributed for k -tuples where covariation is strong. For our application to signal peptide cleavage site determination using the data from [28], we consider only values of 2 and 3 for k , because of concern for statistical significance (collection of signal peptide cleavage site sequences is less than 1500). In the remainder of this paper, we describe our algorithm and compare Vert’s algorithm with variants of our new method, denoted WMMIP for k -tuple weight matrix using mutual information and p -values.

3 Background

Let Σ be a finite alphabet, and let Σ^* denote the set of all finite sequences or words over alphabet Σ . For applications of the method we will describe, Σ could be the set $\{A, C, G, T\}$ of DNA nucleotides, the set

$$\{A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$$

of 20 amino acids, the reduced set of Chan-Dill $\{H, P\}$, where each amino acid is designated either H (hydrophobic) or P (polar). Other alphabets considered in the literature include the following.

¹Assuming independence, WMM is the maximum likelihood estimator. See the author’s text on computational molecular biology [9] for details.

- Dill-Yue’s reduced set $\{H, P, N, X\}$, where each amino acid is designated either H (hydrophobic), P (positively charged), N (negatively charged) or X (polar but not charged), etc.
- Branden-Tooze’s reduced set $\{H, P, G, +, -\}$, where residues A,V,L,I,M,F,P are H (hydrophobic), residues S,T,Y,H,C,N,Q,W are P (polar), the unique residue G is classified as G, residues K,R are positively charged (+) and D,E are negatively charged (-).
- Mirny’s reduced set $\{H, S, P, +, -\}$, where residues A,V,L,I,M,C are H (hydrophobic), residues S,T,N,Q are P (polar), residues G,P are S (special), residues K,R are positively charged (+), D,E are negatively charged (-), and residues F,W,Y,H are R (aromatic).

In the following, we will focus exclusively on the usual set of 20 amino acids.

Some definitions. Let $|A|$ denote the number of elements in the set A . By n -mer, we mean an amino acid sequence of length n , often denoted $s = s_1 \cdots s_n$. Let P be a positive training set of $|P|$ many amino acid sequences and let N be a negative training set of $|N|$ many amino acid sequences, where each sequence in both P and N has fixed length n .

For $a \in \Sigma$, let absolute frequency $num(P, i, a)$ (resp. $num(N, i, a)$) denote the number of occurrences of a at position i in P (resp. N); i.e. $num(P, i, a)$ (resp. $num(N, i, a)$) is the number of sequences $s \in P$ (resp. $s \in N$) such that $s_i = a$. Let $f(P, i, a)$ (resp. $f(N, i, a)$, resp. $f(B, i, a)$) denote the relative (monoresidue) frequency of a at position i in positive (resp. negative resp. background) training set; i.e.

$$\begin{aligned} f(P, i, a) &= \frac{num(P, i, a)}{|P|} \\ f(N, i, a) &= \frac{num(N, i, a)}{|N|} \\ f(B, i, a) &= \frac{num(P, i, a) + num(N, i, a)}{|P| + |N|}. \end{aligned}$$

By chance, the training set might not include a particular example actually existent in nature, hence it is usual to add a *pseudocount*² $c \geq 0$, so that

$$f(P, i, a) = \frac{num(P, i, a) + c}{|P| + c \cdot |\Sigma|}$$

etc. For each $1 \leq i \leq n$, the *weight matrix*, also called *log odds scoring function*,³ $\sigma(i, a)$ is defined by

$$\sigma(i, a) = \log_2 \left(\frac{f(P, i, a)}{f(B, i, a)} \right)$$

and the score $\tau(s)$ of sequence $s = s_1 \cdots s_n \in \Sigma^*$ is the sum of the position dependent scores; i.e.

$$\tau(s) = \sum_{1 \leq i \leq n} \sigma(i, s_i).$$

These notions extend immediately to the situation of k -tuple frequencies, for any fixed $k > 1$, and indeed have been studied to a small extent by Zhang and Marr [31], whose WAM (weight array method) was considered for *contiguous* dinucleotide positions $i, i+1$ for $1 \leq i \leq n$ and applied to the

²Dirichlet prior distributions [10] constitute a more rigorous approach to the problem of addition of pseudocounts. As well, see [17] for a study of position-dependent pseudocounts.

³In the literature, weight matrix has also been called *position-specific scoring matrix* (PSSM) and *profile*.

detection of 5'-splice signals in *S. pombe*. (As discussed below, this method performs very poorly for signal peptide cleavage site detection, even when compared with the weight matrix method. We have not compared the WAM method with the other methods discussed in this paper for the detection of 5'-splice signals in *S. pombe*.) See [11] for an application of k -tuple weight matrices for functional DNA binding sites. It is worth noting that amino acid pair potentials commonly used in protein threading algorithms (pp. 223–228 see [9] for discussion) constitute essentially a certain type of diresidue weight matrix. In such knowledge-based approaches, relative frequencies $f(k, r, a, b)$ are tabulated for the number of occurrences of amino acid pair a, b occurring at linear distance k in the chain and at Euclidean distance⁴ r in a protein from an unbiased sampling of the PDB. By assuming that frequencies

$$f(k, r, a, b) = \frac{\exp\{-E(k, r, a, b)/RT\}}{Z} \quad (1)$$

follow a Boltzmann distribution, a *pseudoenergy* or *quasichemical potential* $E(k, r, a, b)$ can be approximated by assuming that the partition function $Z \approx 1$ and taking the negative log of (1). In protein threading methods, the energy of an amino acid sequence $s = s_1, \dots, s_n$ in conformation $C = p_1, \dots, p_n$, where the p_i are points in 3-dimensional Euclidean space, is defined to be

$$\begin{aligned} E(s, C) &= \sum_{1 \leq i < j \leq n} E(j - i, \text{bin}(|p_j - p_i|), a, b) / RT \\ &= \sum_{1 \leq i < j \leq n} -\ln(f(j - i, \text{bin}(|p_j - p_i|), a, b)) \\ &= -\ln \left(\prod_{1 \leq i < j \leq n} f(j - i, \text{bin}(|p_j - p_i|), a, b) \right) \end{aligned}$$

hence $E(s, C)$ is minimized exactly when $\prod_{1 \leq i < j \leq n} f(j - i, \text{bin}(|p_j - p_i|), a, b)$ is maximized. The latter expression is of course simply a variant of the diresidue weight matrix.

For clarity, we state the obvious definitions concerning k -tuple weight matrices. Fix $k \geq 1$, and let Σ^k denote the set of all k -mers $a = a_1 \cdots a_k$. Let $\binom{n}{k}$ denote the set of all *strictly increasing* k -tuples $1 \leq i_1 < i_2 < \cdots < i_k \leq n$. Note that this set obviously contains $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ many elements, thus explaining our choice of notation. Let A denote a multiple sequence alignment of words over Σ of length n . For any k -mer $\vec{a} = (a_1, \dots, a_k) \in \Sigma^k$, and increasing k -tuple $\vec{i} = (i_1, i_2, \dots, i_k) \in \binom{n}{k}$, the absolute frequency

$$\text{num}(A, \vec{i}, \vec{a})$$

is the number of sequences $s \in A$ such that $s_{i_j} = a_j$ for each $1 \leq j \leq k$. Let $f(A, \vec{i}, \vec{a})$ denote the relative k -tuple frequency of $\vec{a} = (a_1, \dots, a_k)$ at positions $\vec{i} = (i_1, i_2, \dots, i_k)$ in multiple sequence alignment A , so that allowing pseudocount $c \geq 0$,

$$f(A, \vec{i}, \vec{a}) = \frac{\text{num}(A, \vec{i}, \vec{a}) + c}{|A| + c \cdot |\Sigma|^k}$$

⁴In order to discretize the values, Euclidean distances are actually binned into a discrete number (say 20) of classes, and r is the bin number. See the author's text [9] for more details.

Let P (resp. N) be positive (resp. negative) training sets of n -mers, and define the background set $B = P \cup N$. Allowing pseudocount $c \geq 0$, we have the relative frequencies

$$\begin{aligned} f(P, \vec{i}, \vec{a}) &= \frac{\text{num}(P, \vec{i}, \vec{a}) + c}{|P| + c \cdot |\Sigma|^k} \\ f(N, \vec{i}, \vec{a}) &= \frac{\text{num}(N, \vec{i}, \vec{a})}{|N| + c \cdot |\Sigma|^k} \\ f(B, \vec{i}, \vec{a}) &= \frac{\text{num}(P, \vec{i}, \vec{a}) + \text{num}(N, \vec{i}, \vec{a})}{|P| + |N| + c \cdot |\Sigma|^k} \end{aligned}$$

For each $1 \leq i \leq n$, the k -tuple weight matrix, also called k -tuple log odds scoring function, $\sigma(\vec{i}, \vec{a})$ is defined by

$$\sigma(\vec{i}, \vec{a}) = \log_2 \left(\frac{f(P, \vec{i}, \vec{a})}{f(B, \vec{i}, \vec{a})} \right) \quad (2)$$

and the score $\tau(s)$ of sequence $s = s_1 \cdots s_n \in \Sigma^n$ is the sum of the position dependent scores; i.e.

$$\tau(s) = \sum_{\vec{i} \in \binom{[n]}{k}} \sigma(\vec{i}, \vec{a}) \quad (3)$$

where the increasing k -tuple $\vec{a} = (a_1, \dots, a_k)$ extracted from $s_1 \cdots s_n$ is defined by $a_j = s_{i_j}$, for $1 \leq j \leq k$.

Mutual information, a concept arising from Shannon's information theory (see pp. 148–152 of [16]), has been used in computational biology in many contexts. Applications of mutual information include the following.

- Determination of the covariation between nucleotide positions i, j in a multiple sequence alignment of RNA's – see p. 266 of [10].
- Edge weight assignment for the complete graph on a collection of genes, using microarray data. The graph obtained by removing those edges between genes g_i and g_j , whose mutual information is less than a fixed threshold, has been denoted *relevance network* in [8].
- Identificaton of residues which determine specificity in bacterial transcription factors [23].

Suppose that $p(a)$ denotes the position independent, relative monoresidue frequency of residue a , often called the *background* frequency. Fix $1 < k \leq n$, and let $\vec{i} = (i_1, \dots, i_k) \in \binom{[n]}{k}$. For residues $a = (a_1, \dots, a_k) \in \Sigma^k$, let $p(\vec{i}, \vec{a})$ denote the joint probability distribution that a_1, \dots, a_k occur respectively in positions i_1, \dots, i_k . The mutual information

$$M(\vec{i}) = \sum_{\vec{a} \in \Sigma^k} p(\vec{i}, \vec{a}) \cdot \log_2 \left(\frac{p(\vec{i}, \vec{a})}{p(a_1) \cdots p(a_k)} \right) \quad (4)$$

is the Kullback-Liebler distance between the joint probability distribution and the product of the background frequencies. This formula is motivated by the following. The information

$$I(a_i; a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k)$$

can be defined as the amount of surprise

$$\log_2 \left(\frac{1}{p(a_i)} \right) - \log_2 \left(\frac{1}{Pr[a_i|a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k]} \right) \quad (5)$$

from learning of a_i conditioned on already knowing $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k$. The mathematical expectation $E[I(a_i; a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k)]$ of (5) is clearly (4), hence mutual information is simply the expected information from learning a_i , given that $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k$ are known. Mutual information clearly is a measure of covariation, as it is always nonnegative, equaling 0 exactly when $p(\vec{i}, \vec{a}) = p(a_1) \cdots p(a_n)$ for all $\vec{a} = (a_1, \dots, a_k) \in \Sigma^k$.

Given a multiple sequence alignment T of words $s_1 \cdots s_n \in \Sigma^n$ which constitute the training set, for any fixed tuple $\vec{i} = (i_1, \dots, i_k) \in \binom{n}{k}$, we algorithmically determine a point estimator for the p -value of the mutual information $M(\vec{i})$ by permuting 10,000 times the entries in each of the columns i_1, \dots, i_k , each column permuted independently, and counting the fraction of times that the mutual information $M'(\vec{i})$ for the resulting permuted multiple sequence alignment exceeds the original mutual information $M(\vec{i})$. Our implementation uses

$$M(\vec{i}) = \sum_{\vec{a} \in \Sigma^k} f(\vec{i}, \vec{a}) \cdot \log_2 \left(\frac{f(\vec{i}, \vec{a})}{f(i_1, a_1) \cdots f(i_k, a_k)} \right) \quad (6)$$

The novelty of this approach, not considered in [31, 11] or elsewhere, is to incorporate a natural weighting of k -tuple log odds scores dependent on the mutual information and their p -values.

4 Methods

Our method WMIP encompasses a number of variants which we have tested, hence we introduce systematic notation for the variants. Throughout the following definitions, $s = (s_1, \dots, s_n) \in \Sigma^n$ and $\vec{i} = (i_1, \dots, i_k) \in \binom{n}{k}$, let $ss(s, \vec{i}) = (s_{i_1}, \dots, s_{i_k})$ be the subsequence extracted from s at positions in the increasing k -tuple \vec{i} . For each variant studied, we define the score $\tau(s)$ of n -mer s .

- **WM** : weight matrix method described in section 3, where WMk denotes the k -tuple weight matrix method. In this case by (3) by

$$\tau(s) = \sum_{\vec{i} \in \binom{n}{k}} \log_2 \left(\frac{f(P, \vec{i}, ss(s, \vec{i}))}{f(B, \vec{i}, ss(s, \vec{i}))} \right) \quad (7)$$

- *Profile* ignores the background frequencies; i.e. $\tau(s) = \sum_{\vec{i} \in \binom{n}{k}} \log_2 f(P, \vec{i}, ss(s, \vec{i}))$, but in practice (7) has better performance. *Profile1* is monoresidue profile, *Profile2* is diresidue profile, etc.

- **WAM** : weight array method of [31], where WAMk denotes the k -tuple weight array method. In this case,

$$\tau(s) = \sum_{1 \leq i \leq n} \log_2 f(P, i, a_i) + \sum_{1 \leq i \leq n-k+1} \log_2 f(P, (i, i+1, \dots, i+k-1), (a_i, a_{i+1}, \dots, a_{i+k-1})) \quad (8)$$

Note that (8) is simply the monoresidue weight matrix score plus the *contiguous* diresidue weight matrix score, both considered without taking into account the background frequencies.

- WMMI : weight matrix with mutual information, where k -WMMI denotes the k -tuple variant. In this case, letting $M(\vec{i})$ denote the mutual information

$$\sum_{\vec{a} \in \Sigma^k} f(\vec{i}, \vec{a}) \cdot \log_2 \left(\frac{f(\vec{i}, \vec{a})}{f(i_1, a_1) \cdots f(i_k, a_k)} \right)$$

we have

$$\tau(s) = \sum_{\vec{i} \in \binom{[n]}{k}} M(\vec{i}) \cdot \log_2 \left(\frac{f(P, \vec{i}, \text{SS}(s, \vec{i}))}{f(B, \vec{i}, \text{SS}(s, \vec{i}))} \right) \quad (9)$$

- We consider several variants of WMMIP : WMMIP-1 , WMMIP-2A , WMMIP-2B , defined in the following.
- WMMIP-1 : weight matrix with mutual information (one-sided), where k -WMMIP-1 denotes the k -tuple variant. In this case, letting $M(\vec{i})$ denote the mutual information and $P(\vec{i})$ the point estimator of the p -value for this mutual information, as obtained by simulation, the score of $s = s_1, \dots, s_n \in \Sigma^n$ is defined in by

$$\tau(s) = \sum_{\vec{i} \in \binom{[n]}{k}} M(\vec{i}) \cdot (1 - P(\vec{i})) \cdot \log_2 \left(\frac{f(P, \vec{i}, \text{SS}(s, \vec{i}))}{f(B, \vec{i}, \text{SS}(s, \vec{i}))} \right) \quad (10)$$

- WMMIP-2 : weight matrix with mutual information (two-sided), where k -WMMIP-2 denotes the k -tuple variant. In this case, let $M(\vec{i})$ denote the mutual information and $P(\vec{i})$ the point estimator of the p -value for this mutual information, as obtained by simulation. In the first variant, designated k -WMMIP-2A , $\tau(s)$ is

$$\sum_{\vec{i} \in \binom{[n]}{k}} M(\vec{i}) \cdot \left[(1 - P(\vec{i})) \cdot \log_2 \left(\frac{f(P, \vec{i}, \text{SS}(s, \vec{i}))}{f(B, \vec{i}, \text{SS}(s, \vec{i}))} \right) + P(\vec{i}) \cdot \sum_{1 \leq j \leq k} \log_2 \left(\frac{f(P, i_j, s_{i_j})}{f(B, i_j, s_{i_j})} \right) \right] \quad (11)$$

while in the second variant, designated k -WMMIP-2B , $\tau(s)$ is

$$\sum_{\vec{i} \in \binom{[n]}{k}} M(\vec{i}) \cdot (1 - P(\vec{i})) \cdot \log_2 \left(\frac{f(P, \vec{i}, \text{SS}(s, \vec{i}))}{f(B, \vec{i}, \text{SS}(s, \vec{i}))} \right) + P(\vec{i}) \cdot \sum_{1 \leq j \leq k} \log_2 \left(\frac{f(P, i_j, s_{i_j})}{f(B, i_j, s_{i_j})} \right) \quad (12)$$

- WMB : weight matrix using Brown's algorithm, where k -WMB denotes the k -tuple variant. In [4, 18], an algorithm is given to approximate a joint probability distribution guaranteed to have certain marginals.⁵ we consider the k -tuple frequencies as desired marginals distributions, and apply Brown's algorithm to compute a joint distribution having these marginals. To compute the joint probability distribution for n -mers, Brown's algorithm requires an array of size $|\Sigma|^n$. For $\Sigma = \{A, C, G, T\}$, in principle one could compute the joint probability distribution for oligonucleotides of length 10, as $4^{10} = 1048576$. In the case of Σ being the 20 letter amino acid alphabet, one can conveniently handle at most 5-mers, as $20^5 = 3200000$. In considering some of the reduced amino acid alphabets mentioned in section 3, one can

⁵There may be many joint distributions having the same given marginals; the algorithm of [4] yields the *maximum entropy* distribution, rather than the *maximum likelihood* distribution.

consider n -mers for slightly larger values of n . Let $q_P(s_1, \dots, s_n)$ resp. $q_B(s_1, \dots, s_n)$ denote the resulting joint distribution for the positive resp. background training set. Then the score of $s = s_1, \dots, s_n \in \Sigma^n$ is defined by

$$\tau(s) = \log_2 \left(\frac{q_P(s_1 \cdots s_n)}{q_B(s_1 \cdots s_n)} \right) \quad (13)$$

- *WM1posMinusNeg* is given by computing log odds scores using *monoresidue frequencies* for positive examples minus log odds scores of negative examples. *WM2posMinusNeg* resp. *WM3posMinusNeg* are similarly defined, but using diresidue resp. triresidue frequencies.
- *WM3_PLUS_WM1* adds the score obtained by method WM3 together with that of WM1. Similarly for other such examples.
- *2 - WMMIPmatch* assigns weights between all diresidue positions $1 \leq i < j \leq n$, the weight for the edge (i, j) being the mutual information at positions i, j times one minus the p -value. The implementation [22] of H. Gabow's $O(n^3)$ maximum weight matching algorithm is then applied, to consider only those dinucleotide log odds scores at positions (i, j) which are selected by the matching algorithm. This is explained more fully in the next section.

Finally, data *bis* for several cases is shown in Figure 5, thus illustrating the variability of results and limits of statistical significance. (All methods were repeated several times, but only a few data of second runs are shown.)

4.1 Network flow and matching

In the previous section, we discussed several possible weighting schemes applied to the log odds scores of k -tuple frequencies by appropriate consideration of the positional mutual information $M(\vec{i})$ and the positional point estimators $P(\vec{i})$ for the p -value of mutual information. Heuristically, such weighting schemes may remove many k -tuple positions $\vec{i} = (i_1, \dots, i_k)$ from consideration when applying the log odds score.

For diresidue scores, a natural extension of this approach is to determine a partition $C_1, \dots, C_{n/2}$ of the positions $1, \dots, n$ into disjoint classes of size 2, and to apply the log odds score only to pairs i, j in the same partition class. For instance, if $n = 4$, one might apply the diresidue log odds score only to pairs $(1, 2), (3, 4)$, or only to pairs $(1, 3), (2, 4)$, or only to pairs $(1, 4), (2, 3)$. The existence of such a partition presumes that n is even. If n is odd, then one could consider a partition of $1, \dots, n$ into one class of size 1 and $n/2$ classes of size 2, taking the monoresidue log odds score at the singleton position. Is there an efficient algorithm to determine which partition which produces the strongest log odds score for a positive test set?

Letting $P(n)$ denote the number of possible partitions of $1, \dots, n$ into disjoint $\lfloor n/2 \rfloor$ sets, each of size 2, we clearly have the recurrence relation

$$P(n) = \begin{cases} (n-1) \cdot P(n-2) & \text{if } n \text{ is even} \\ n \cdot P(n-1) & \text{if } n \text{ is odd} \end{cases}$$

with base cases $P(0) = P(1) = P(2) = 1$. It follows that for $n = 2m$ even,

$$P(n) = 1 \cdot 3 \cdot 5 \cdots (2m-1) = \frac{(2m-1)!}{2^{m-1}(m-1)!} = \Omega\left(\frac{n}{3}\right)^{m/2-1}$$

The values $P(n)$ are related to Stirling numbers of the second kind, and grow exponentially large, thus disallowing any brute force method to determine an optimal partition. Here is a table of the first few values of $P(n)$ for even n (note that if n is odd, then $P(n) = P(n + 1)$).

2	4	6	8	10	12	14	16	18	20
1	3	15	105	945	10395	135135	2027025	34459425	654729075

For an arbitrary amino acid sequence $s = s_1 \cdots s_n$ of length n , the (i, j) -score $\tau(i, j, s)$ of sequence s is $\sigma(i, j, s_i, s_j)$, and the score $\tau(s)$ of s is defined by

$$\tau(s) = \sum_{1 \leq i < j \leq n} \tau(i, j, s) = \sum_{1 \leq i < j \leq n} \sigma(i, j, s_i, s_j).$$

Let $\mu(i, j)$ denote the average log odds score over all diresidue pairs a, b in the positive test set P' ; i.e.

$$\mu(i, j) = \frac{\sum_{s \in P'} \tau(s)}{|P'|}.$$

Suppose that n is even. Consider the optimization problem of partitioning the integers $1, \dots, n$ into $n/2$ disjoint classes $C_1, \dots, C_{n/2}$, each of size 2, such that the sum of $\mu(i, j)$ over all pairs i, j belonging to the same partition class is maximized. We refer to this problem as the PARTITION OPTIMIZATION problem; i.e. given fixed training sets P, N of n -mers and test set P' of n -mers, determine a partition $C_1, \dots, C_{n/2}$ which optimizes the log odds score of P' .

Theorem 1 PARTITION OPTIMIZATION can be solved in polynomial time.

Proof PARTITION OPTIMIZATION is equivalent to the maximum weight perfect matching problem. The latter is a celebrated theorem proved by Edmonds – see pp. 358–375 of [21].

It is currently unclear whether for $k > 2$ the analogous PARTITION OPTIMIZATION problem can be solved in polynomial time.

In light of algorithms 2-WMMIP-2 and 3-WMMIP-2, it is sensible to determine, for fixed $k \geq 2$, an optimal partition of $1, \dots, n$ into disjoint classes, each of size at most k . Note that this allows classes C to have any number $|C| \in \{1, \dots, k\}$ of elements, and corresponds to the situation where an appropriate disjoint mix of mono-, di-, tri-, ..., k -tuple weight matrix entries are evaluated. Whether the corresponding PARTITION OPTIMIZATION problem can be solved in polynomial time is at present unclear. Such techniques, if efficient, could improve the performance of related weight matrix methods.

5 ROC curve comparison of methods

Receiver-operating characteristic curves (ROC) curves are one familiar means of comparing the performance of different machine learning algorithms, which numerically score sequences in a statistical classification problem.

Let X denote the space of sequences under consideration. In this paper, we consider fixed-length amino acid sequences – 10-mers in the case of signal peptide cleavage site recognition, and 12-mers in the case of disulfide bond recognition. In a classification problem, the set X is partitioned into sets P (resp. N) of positive (resp. negative) examples, with $X = P \cup N$. Five-fold cross-validation

consists of randomly partitioning P (resp. N) into five equal-size classes, repeatedly training on four out of five classes, and testing on the remaining fifth class. Thus $P = P_{\text{test}} \cup P_{\text{train}}$ and $N = N_{\text{test}} \cup N_{\text{train}}$, expected size of P_{test} (resp. N_{test}) is $|P|/5$ (resp. $|N|/5$).

Suppose that A denotes a prediction algorithm, which trains on $P_{\text{train}}, N_{\text{train}}$, and for each element x of $P_{\text{test}} \cup N_{\text{test}}$, $A(x)$ is a numerical score, where positive examples generally score higher than negative examples. See, for example, Figure 3, which presents superposed histograms of scores for positive and for negative test examples, for the diresidue weight matrix method WM2 applied to disulfide bond prediction. Given any threshold T , one could consider the classification algorithm A_T , which classified examples from $P_{\text{test}} \cup N_{\text{test}}$ as follows:

$$A_T(x) = \begin{cases} 1 & \text{if } A(x) \geq T \\ 0 & \text{else} \end{cases}$$

Algorithm A_T partitions the space $P_{\text{test}} \cup N_{\text{test}}$ into four classes.

- *True positives*: those $x \in P_{\text{test}}$ for which $A_T(x) = 1$.
- *False positives*: those $x \in N_{\text{test}}$ for which $A_T(x) = 1$.
- *True negatives*: those $x \in N_{\text{test}}$ for which $A_T(x) = 0$.
- *False negatives*: those $x \in P_{\text{test}}$ for which $A_T(x) = 0$.

The *false positive rate* of A_T is the number of false positives divided by the size of P_{test} ; similarly the *true positive rate* is the number of true positives divided by the size of P_{test} . A ROC curve is produced by graphing the true positive rate as a function of the false positive rate, in varying the choice of threshold T in algorithm A .

Disulfide bond recognition

For each disulfied bond in each protein in the PDB, we extracted a *positive example* 12-mer, consisting of residues $-3, -2, -1, +1, +2, +3$ around the cystine molecule, conserving a left-to-right amino to carboxy ordering. Similarly, for each pair of cysteines, not in a disulfide bond, we produced a *negative example* 12-mer. We then performed 5-fold cross validation studies, where the background examples were those from the positive and negative training sets. (We separately considered defining the negative examples to be neighborhoods, as previously defined, about two half-cystines which are not in a disulfide bond with each other. Data is not shown.)

Figure 3 displays the histograms of diresidue weight matrix scores for disulfide bonds (right side) with those scores for nondisulfide bonds (left side). Figure 4 shows the superposed ROC curves for disulfide bond recognition using WM1, WM2 and WM3. (We studied as well as a variant of WM2, where in dinucleotide positions (i, j) , i ranges over neighbors of the first cysteine and j is required to range over the second cysteine. Interestingly, this method did not perform as well as WM2, which indicates that in the case of disulfide bond prediction, there are pair correlations within the neighborhood of a half-cystine, which along with pair cross correlations between both neighborhoods in the cystine, aid in recognizing disulfide bonds.)

The ROC curves were produced by 5-fold cross validation on the entire protein database, rather than a nonredundant version as in PDBselect25. There was a similar dramatic improvement in performance of WM2 and WM3 over WM1 also in the case with PDBselect25 (data not shown); however, in each case, performance when training and testing on PDBselect25 was vastly inferior to performance when training and testing on the entire PDB. Our goal was to illustrate the difference in compartment of algorithms on different data sets. Since it is unclear to what extent the signal

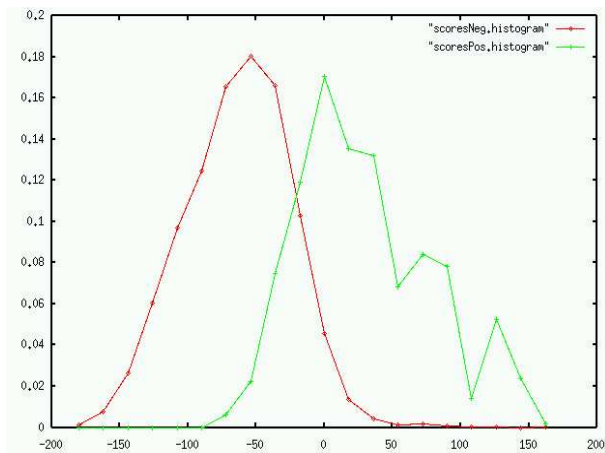


Figure 3: Histograms of positive, negative data for disulfide bonds

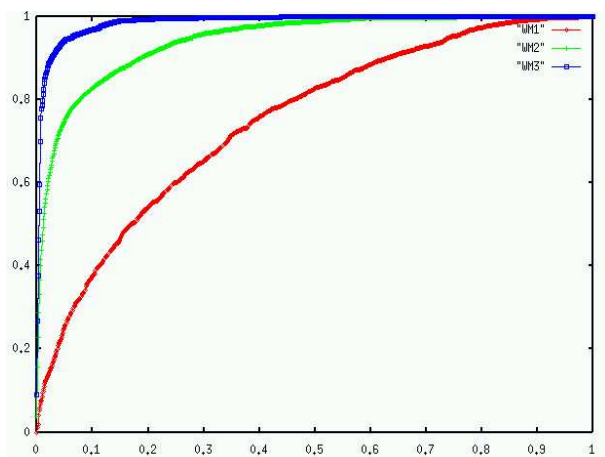


Figure 4: Disulfide bond ROC curves

peptide cleavage site data from [28] consists of sequences, which pairwise share no more than 25% homology, we have chosen to display the ROC curves from the entire PDB.

In Figure 5, we give the average improvement in true positive rate for the problem of disulfide bond recognition, when comparing a method against WM1 in the false positive range of 0% to 10%. (The approach of Fariselli-Casadio [12] does not immediately yield ROC curves. As previously mentioned, we are currently recoding and modifying the approach of Fariselli-Casadio to better compare their method with those of weight matrix variants.)

Method	Avg improvement (0-10% false pos)
WM2	34.64%
WMMI	33.92%
WMMIP1	34.32%
WM3	57.09%

Figure 5: Performance in disulfide bond recognition

Signal peptide cleavage sites

Figure 5 gives average improvement in true positive rate for signal peptide cleavage site prediction, when comparing a method against WM1 in the false positive range of 0% to 10%. Note in particular that no particular method, when its performance is averaged over the range of 0% to 10% false positives, including support vector machine kernel of [29], appears to be really significantly better than the monoresidue weight matrix method. Of particular note is the poor performance of the Profile method, where no account of background frequencies is taken, and of the WAM method of [31]. As well, note that in signal peptide cleavage site recognition, monoresidue weight matrices perform better than diresidue and triresidue weight matrices. This situation is opposite that of disulfide bond prediction.

REMARKS:

1. Additionally, average change from 0-10% false positive rate, when 5-3 weight matrix over reduced alphabet compared to Brown is: -0.52%, and at 10% false positive rate, both obtain around 60% true positive rate.
2. Average change from 0-10% false positive rate, when 5-3 weight matrix over 20 letter alphabet compared to Brown is: -5.04%, and at 10% false positive rate, weight matrix method around 70% true positive rate.
3. Not much difference between SVM and a variety of weight matrix methods, though some reasonable theoretical algorithms such as WAM perform very poorly.

6 Conclusions

The method of choice (SVM, HMM, stochastic context free grammar, neural net, weight matrix or variant) depends on particular data. In the case of disulfide bond prediction, generalizations of weight matrices perform surprisingly well (93% detection of true positives allowing up to 10 % false positives, for 5-fold cross validation on the entire PDB).

In contrast, with signal peptide cleavage site detection, weight matrix variants (and even SVM) perform around 70% detection of true positives allowing up to 10 % false positives.

In both cases, weighting by mutual information and possibly p -values of mutual information, even with the application of maximum weight matching, adds nothing to WM2 .

Method	Avg improvement (0-10% false pos)
SVM	0.47%
Profile1	-5.17%
Profile2	-14.55%
WAM2	-4.5%
WAM3	-4.99%
WM1posMinusNeg	0.27%
WM2	0.33%
WM2_PLUS_WM1	0.16%
WM2posMinusNeg	0.01%
WM3	-0.99%
WM3_PLUS_WM1	-1.21%
WM3_PLUS_WM2	-0.72%
WM3posMinusNeg	-0.81%
2-WMMI	0.37%
2-WMMIbis	0.55%
2-WMMIP1	0.22%
2-WMMIP1bis	0.15%
2-WMMIP1_PLUS_WM1	-0.05%
2-WMMIP1_PLUS_WM1bis	0.22%
2-WMMIP2a	-0.16%
2-WMMIP2abis	0.24%
2-WMMIP2b	0.46%
2-WMMIP2bbis	0.27%
2-WMML_PLUS_WM1bis	0.34%
2-WMMIPmatch	-0.88%
2-WMMIPmatchBis	-0.83%
2-WMMIPmatchBis_PLUS_WM1	-0.05%
3-WMMI	-3.8%
3-WMMIbis	-1.56%
3-WMMIP1	-1.52%
3-WMMIP1_PLUS_WM1	-1.63%
3-WMMIP2a	-3.46%
3-WMMIP2b	-3.78%

Figure 6: Performance in signal peptide cleavage site recognition

Acknowledgements

Thanks to J. Baglivo, C. Burge [7] and L. Mirny for comments. In particular, L. Mirny suggested the relevance of p -values in mutual information, and C. Burge suggested the potential applicability of Brown's algorithm.

References

- [1] P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach*. MIT Press, Cambridge, MA, 1998.
- [2] P. Baldi and Y. Chauvin. Smooth on-line learning algorithms for hidden Markov models. *Neural Computation*, 6, 1994.
- [3] M. Borodovsky and J. McIninch. Genmark: Parallel gene recognition for both DNA strands. *Computers and Chemistry*, 17(2):123–133, 1993.
- [4] D.T. Brown. A note on approximations to discrete probability distributions. *Information and Control*, 2:386–392, 1959.
- [5] P. Bucher. Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. *J. Mol. Biol.*, 212, 1990.
- [6] M.L. Bulyk, P.L.F. Johnson, and G.M. Church. Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucleic Acids Res.*, 30(5):1255–1261, 2002.
- [7] C. Burge. Maximum entropy and statistical methods for DNA sequence analysis. Personal communication, August 2 1997.
- [8] A.J. Butte and I.S. Kohane. Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements. In R.B. Altman, A.K. Dunker, L. Hunter, K. Lauderdale, and T.E. Klein, editors, *Pacific Symposium on Biocomputing 2000*, pages 418–429. World Scientific, 2000.
- [9] P. Clote and R. Backofen. *Computational Molecular Biology : An Introduction*. John Wiley & Sons, Ltd., 2000.
- [10] S. R. Eddy, G. Mitchison, and R. Durbin. Maximum discrimination hidden Markov models of sequence consensus. *J. Comp. Biol.*, 2(1):9–24, 1995.
- [11] M.P. Ponomarenko et al. Oligonucleotide frequency matrices addressed to recognizing functional DNA sites. *Bioinformatics*, 15(7/8):631–643, 1999.
- [12] P. Fariselli and R. Casadio. Prediction of disulfide connectivity in proteins. *Bioinformatics*, 17(10):957–964, 2001.
- [13] P. Fariselli, P. Riccobelli, and R. Casadio. Role of evolutionary information in predicting the disulfide-bonding state of cysteine in proteins. *Proteins*, 36:340–346, 1999.
- [14] M. Gribskov, A.D. McLachlan, and D. Eisenberg. Profile analysis: detection of distantly related proteins. *Proc. Natl. Acad. Sci. USA*, 84:4355–4358, 1987.
- [15] M. Gribskov and N.L. Robinson. The use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Computers Chem.*, 20:25–34, 1996.
- [16] R. W. Hamming. *Coding and Information Theory*. Prentice-Hall, Englewood-Cliffs, 1980.
- [17] J.G. Henikoff and S. Henikoff. Using substitution probabilities to improve position-specific scoring matrices. *Cabios*, 12(2):135–144, 2 April 1996.
- [18] C.T. Ireland and S. Kullback. Contingency tables with given marginals. *Biometrika*, 55(1):179–188, 1968.
- [19] T. Jaakkola, M. Dickhans, and D. Haussler. Using the Fisher kernel method to detect remote protein homologies. In *Intelligent Systems in Molecular Biology (ISMB'99)*, 1999.
- [20] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.*, 235:1501–1531, Feb 1994.
- [21] L. Lovász and M.D. Plummer. *Matching Theory*. North Holland Mathematics Studies **121**. Elsevier Science Publishers B.V., 1985.
- [22] H. Gabow's $O(n^3)$ maximum matching algorithm. <http://elib.zib.de/pub/Packages/mathprog/matching/weighted/>.
- [23] L.A. Mirny and M.S. Gelfand. Using orthologous and paralogous proteins to identify specificity determining residues in bacterial transcription factors. Personal communication, Feb. 10 2002.

- [24] H. Nielsen, J. Engelbrecht, S. Brunak, and G. von Heijne. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Engineering*, 10(1):1–6, 1997.
- [25] M. Paetzel and N.C.J. Strynadka. Signal peptide cleavage in the *e. coli* membrane. *CSBMCB Bulletin 2001 (Canadian Society of Biochemistry and Molecular & Cell Biology)*, 2:60–65, 2001. http://www.csbmcb.ca/Bulletin_2001/60-65.pdf.
- [26] T.D. Schneider and R.M. Stephens. *Nucleic Acids Res.*, 18:6097–6100, 1990.
- [27] R. Staden, K.F. Beal, and J.K. Bonfield. The Staden package. In S. Misener and S. Krawetz, editors, *Computer Methods in Molecular Biology*. The Humana Press Inc., Totowa, NJ 07512, 1998.
- [28] Danish Technical University. Signalp website. <http://www.cbs.dtu.dk/services/SignalP/>.
- [29] J.-P. Vert. Support vector machine prediction of signal peptide cleavage site using a new class of kernels for strings. In R.B. Altman, A.K. Dunker, L. Hunter, K. Lauderdale, and T.E. Klein, editors, *Pacific Symposium on Biocomputing 2002*, pages 649–660. World Scientific, 2002.
- [30] G. von Heijne. *Nucleic Acids Res.*, 14:4683–4690, 1986.
- [31] M.Q. Zhang and T.G. Marr. A weight array method for splicing signal analysis. *Cabios*, 9(5):499–509, 1993.