Pattern Recognition 43 (2010) 3891–3899

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Peptide classification using optimal and information theoretic syntactic modeling

E. Aygün [a,*], B.J. Oommen [b,1], Z. Cataltepe [a]

[a] Computer Engineering Department, Istanbul Technical University, Maslak, Istanbul 34469, Turkey
[b] School of Computer Science, Carleton University, Ottawa, Ontario, Canada K1S 5B6

ARTICLE INFO

Article history:
Received 25 February 2009
Received in revised form
24 February 2010
Accepted 18 May 2010

Keywords:
Biological sequence analysis
Optimal and information theoretic syntactic classification
Peptide classification
Sequence processing
Syntactic pattern recognition

ABSTRACT

We consider the problem of classifying peptides using the information residing in their syntactic representations. This problem, which has been studied for more than a decade, has typically been investigated using distance-based metrics that involve the edit operations required in the peptide comparisons. In this paper, we shall demonstrate that the Optimal and Information Theoretic (OIT) model of Oommen and Kashyap [22] applicable for syntactic pattern recognition can be used to tackle peptide classification problem. We advocate that one can model the differences between compared strings as a mutation model consisting of random substitutions, insertions and deletions obeying the OIT model. Thus, in this paper, we show that the probability measure obtained from the OIT model can be perceived as a sequence similarity metric, using which a support vector machine (SVM)-based peptide classifier can be devised. The classifier, which we have built has been tested for eight different substitution matrices and for two different data sets, namely, the HIV-1 Protease cleavage sites and the T-cell epitopes. The results show that the OIT model performs significantly better than the one which uses a Needleman–Wunsch sequence alignment score, it is less sensitive to the substitution matrix than the other methods compared, and that when combined with a SVM, is among the best peptide classification methods available.

1. Introduction

Peptides are relatively short amino acid sequences that occur either as separate molecules or as subsequences of proteins. Apart from their significance in analyzing proteins, peptides themselves may have various distinct chemical structures that are *themselves* related to different molecular functions. These functions, such as cleavage or binding, while being interesting in their own right, have also been shown to be important in areas such as biology, medicine, drug design, disease pathology, and nanotechnology [31,28,11,30,27]. Indeed, for more than a decade, researchers have sought computational techniques to rapidly identify peptides that are known to be, or can be, related to certain molecular functions.

The research in peptide classification is not new—indeed, a host of techniques have been proposed for *in silico* peptide classification. In 1998, Cai and Chou [3], presented one of the pioneering works in this area. They classified 8-residue peptides and used artificial neural networks with 20 input nodes per residue, thus involving a total of 160 input nodes. In their work, each amino acid was encoded using 20 bits so that the 20 amino acids were encoded as $A=100\ldots00$, $B=010\ldots00,\ldots,Y=000\ldots01$. Similarly, Zhao et al. in [37] mapped the amino acid sequences of peptides directly into feature vectors and fed them into a support vector machine (SVM). They, however, represented the amino acids by a set (more specifically, 10) of their biophysical properties, such as hydrophobicity or beta-structure preference, instead of an orthonormal representation, as advocated in [3]. By resorting to such a representation, they were eventually able to reduce the dimensionality of the input space by 50%. To further increase the information density of input vectors, the authors of [34] used bio-basis artificial neural networks, which are a revision of radial-basis function networks, that use biological similarities rather than spatial distances. This work was subsequently enhanced by Trudgian and Yang in [35] by optimizing the substitution matrices that are used to compute the latter biological similarities. Kim et al. [16] followed a rule-based approach to achieve results which were interpretable. It should be mentioned that there were also earlier studies based on the properties of quantitative matrices [24], binding motifs [25] and hidden Markov models [21], which should really be treated as precursors to the results cited above. The differences between our results and those which use Hidden Markov Models (HMMs) will be clarified presently.

A completely different sequence representation technique was introduced in the area of protein fold recognition by Liao and Noble in [20]. The authors of [20] represented protein sequences

[*] Corresponding author.
E-mail addresses: eser.aygun@itu.edu.tr (E. Aygün), oommen@scs.carleton.ca (B.J. Oommen), cataltepe@itu.edu.tr (Z. Cataltepe).
[1] This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway.

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.05.022

by their pairwise biological similarities, which were measured by ordinary sequence alignment algorithms. Subsequently, by considering these similarities as feature vectors, relatively simple classifiers were trained and successfully utilized for classifying and discriminating between different protein folds [26,13].

Probably one of the more fascinating ways of combining "state-of-the-art" metrics and techniques is found in the work of Li and Jiang in 2005 [19]. The impressive facet of this research [19] is that it combines SVMs with non-traditional sequence similarity measures. Indeed, rather than using sequence similarity measures in their virgin form, or invoking basic algebraic kernels, they advocated the use of *edit kernels*, which are first of all, based on the edit distances between sequences, and further, where the concept of the edit kernel was defined as a family of functions of the form

$$K(x,y) = e^{-\gamma \cdot edit(x,y)},$$

where $edit(x,y)$ is the edit distance between the sequences x and y, and where γ is a parameter used to scale the values in order to make the kernel matrix positive definite [19]. A qualitative comparison between our work and the work of [19] will be given presently.

The primary intention in this present study is to use a SVM-based classifier in achieving the classification and discrimination. However, rather than the use of distances, we shall advocate the use of a rigorous probabilistic model, namely one which has been proven to be both optimal and to attain the information theoretic bound. Indeed, in this study, we combine the strategy of Liao and Noble [20] (i.e., to use pairwise SVM classifiers) with a probabilistic similarity metric, and to successfully classify peptides. Observe that, instead of resorting to the alignment scores, we quantify the similarity by means of their optimal and information theoretic (OIT) garbling probabilities as described in [22]. The latter OIT garbling probability is the probability of obtaining a sequence Y from a sequence U based on the OIT mutation model, whose properties will be clarified later. One clear difference between the alignment scores and OIT garbling probabilities is that whereas an alignment score considers only the shortest path between two sequences, the OIT garbling probabilities covers all possible paths. Furthermore, since it assigns a probability mass to every possible path (i.e., possible garbling operations), it unarguably contains more information about the similarity between the two sequences.

It is now relevant to highlight the difference between our present work and the results of [19]. The crucial difference between the latter methodology and ours is that, first of all, the OIT model is capable of considering the assigned (associated) probability mass for every possible edit path, which the work of [19] is incapable of doing. That being understood, secondly, we do not use the OIT model to compute a complete pairwise similarity matrix of instances and use it as a kernel. Rather, we use our total-probability similarity measure to build a feature matrix that holds the similarities between the instances and some predefined set of representative sequences. Subsequently, we feed this feature matrix into a classical linear kernel SVM. Thus, from an overall perspective, apart from the fundamental advantages of using the OIT model over edit distances, our approach has two main advantages over Li and Jiang's: (i) in our approach, the number of computations grows only linearly with the number of instances, rather than quadratically and (ii) our approach does not intrinsically depend on SVMs at all *per se*, as one could rather have used the same feature matrix in conjunction with a completely different type of classifier to invoke the corresponding training and testing modules. Readers interested in sequence-based kernels should also take a look at the use of *spectrum kernels* advocated by [14,17,18]. Since, as explained above, these are not directly related to our work, in the interest of brevity, these are not addressed here in any more detail.

It is pertinent to also mention that a similar transition probability measurement based on HMMs was earlier proposed by Bucher and Hofman in [2]. Indeed, since then, HMM-based similarity metrics have been used in many biological applications [10,15,16,32]. The difference between our work and the ones which use HMMs is the following: Whenever a system models the garbling mechanism using a HMM, it implicitly assumes that the probability of inserting a sub-string with k elements is distributed as a mixture of Geometric distributions [22]. Indeed, such a model is incapable of capturing arbitrary non-Geometric-based distributions. The OIT model, however, permits mutation models with arbitrary insertion probability distributions such as the Poisson distribution or the binomial distribution, or for that matter, any parametric or non-parametric distribution. Thus, we argue that the superiorities of an OIT-based mechanism, listed later, have motivated us to use them for peptide classification. The entire problem of using the OIT model to quantify the similarity between biological compounds other than peptides, and subsequently classify them, is still open. We believe that this will be an extremely rewarding exercise, which could lead to a host of future research avenues.

What then are the advantages of the OIT model, which renders it superior to the "distance-based" approaches? We clarify this by perceiving the model causing the mutations as a "channel" through which the original string is transmitted, the output of which is the garbled string containing substitution, insertion and deletion (SID) errors. Thus, throughout this paper, for the sake of simplicity, we shall use the terms "model", "channel" and "generator" interchangeably. Using the notation that U is the input to the channel (string generator) and that Y is its random output, we list below the novel, salient features of the OIT model, Π^*, which "distance-based" approaches do not possess [22]:

1. Π^* is functionally complete because it comprehensively considers all the ways by which U can be mutated into Y using the three elementary SID operations. We shall show that whereas the number of ways by which U can be transformed into Y is a combinatorially "explosive" large number, each of these events is assigned a valid probability measure, and the sum of these measures over all the possible transformations is exactly unity, rendering it stochastically consistent.
2. The distributions involved for the various garbling operations in Π^* can be completely arbitrary. These constitute the parameters of the generator (model) which are not merely real numbers, but arbitrary distributions, giving the practitioner much more freedom to model the biological differences between U and Y.
3. The model Π^* even captures the scenarios in which the probability of a particular string U being transformed into another string Y, is arbitrarily small, which is not possible with "distance-based" approaches because the latter render many inter-string distances to be identical.
4. For a given U, the length of Y is a random variable whose distribution does not necessarily have to be a mixture of geometric distributions.
5. If the input U is itself an element of a dictionary, and the OIT channel is used to model the noisy channel, the technique for computing the probability $Pr[Y|U]$ can be utilized in a Bayesian way to compute the *a posteriori* probabilities, and thus yield an optimal, minimum probability of error pattern classification rule. In a non-Bayesian approach, this would be a maximum likelihood pattern classification rule.

6. Most importantly, in both the Bayesian and non-Bayesian approaches, the OIT model actually attains the information theoretic bound for recognition accuracy when compared with all the other models which have the same underlying garbling philosophy.

We have tested our solution, which involves the combination of the SVM-pairwise and the OIT model, on two peptide classification problems, namely the HIV-1 Protease cleavage site, and the T-cell epitope prediction problems. Both of these problems are closely related to pharmacological research work that has been the focus of a variety of computational approaches [3,16,34,35,37]. The results, which we present in a subsequent section, indicate that our solution paradigm leads to an extremely good classification performance for both problems.

The rest of the paper is organized as follows. In Section 2 we first briefly explain the OIT model, including here only the relevant particulars that are required for this present paper. In Section 3, we then present the methodology and explain how we have used it in classification of peptides. Section 4.2 contains the outcomes of the experiments conducted, and it also includes a discussion and interpretation, and a comparison of our results to the previous work. Section 5 concludes the paper and proposes the avenues for future work.

2. Modeling—the string generation process

We now describe the model by which a string Y is generated given an input string $U \in A^*$, where A is the alphabet under consideration, and ξ and λ are the input and output null symbols, respectively.

First of all, we assume that the model utilizes a probability distribution G over the set of positive integers. The random variable in this case is referred to as Z, and is the number of insertions that are performed in the mutating process. G is called the *quantified* insertion distribution, and in the most general case, can be conditioned on the input string U. The quantity $G(z|U)$ is the probability that Z=z given that U is the input word. Thus, G has to satisfy the following constraint:

$$\sum_{z \geq 0} G(z|U) = 1. \qquad (1)$$

Examples of the distribution G are the Poisson and the geometric distributions whose parameters depend on the word or the length of the input word. However, the distributions can be arbitrarily general.

The second distribution that the model utilizes is the probability distribution Q over the alphabet under consideration. Q is called the *qualified* insertion distribution. The quantity $Q(a)$ is the probability that $a \in A$ will be the inserted symbol conditioned on the fact that an insertion operation is to be performed. Note that Q has to satisfy the following constraint:

$$\sum_{a \in A} Q(a) = 1. \qquad (2)$$

Apart from G and Q, another distribution that the model utilizes is a probability distribution S over $A \times (A \cup \{\lambda\})$, where λ is the output null symbol. S is called the substitution and deletion distribution. The quantity $S(b|a)$ is the conditional probability that the given symbol $a \in A$ in the input string is mutated by a stochastic substitution or deletion—in which case it will be transformed into a symbol $b \in (A \cup \{\lambda\})$. Hence, $S(c|a)$ is the conditional probability of $a \in A$ being substituted for by $c \in A$, and analogously, $S(\lambda|a)$ is the conditional probability of $a \in A$ being deleted. Observe that S has to satisfy the following constraint for all $a \in A$:

$$\sum_{b \in (A \cup \{\lambda\})} S(b|a) = 1. \qquad (3)$$

Using the above distributions we now informally describe the OIT model for the garbling mechanism (or equivalently, the noisy string generation process). Let $|U| = N$. Using the distribution G, the generator[2] first randomly determines the number of symbols to be inserted. Let Z be random variable denoting the number of symbols that are to be inserted in the mutation. Based on the output of the random number generator, let us assume that Z takes the value z. The algorithm then determines the position of the insertions among the individual symbols of U. This is done by randomly generating an input edit sequence $U' \in (A \cup \{\xi\})^*$. We assume that the $\binom{N+z}{z}$ possible strings are equally likely.

Note that the positions of the symbol ξ in U' represents the positions where symbols will be inserted into U. The non-ξ symbols in U' are now substituted for or deleted using the distribution S. Finally, the occurrences of ξ are transformed independently into the individual symbols of the alphabet using the distribution Q.

This defines the model completely. An example that will help clarify the OIT garbling channel follows.

Example 1. Let U = “string” . Let the number of insertions based on the distribution G, be 2. The positions of the two insertions are now randomly chosen out of the 28 possible positions. Let us suppose the resultant string is $U' =$ “ striξngξ” . The non-ξ symbols of U' are now randomly substituted for or deleted using the distribution S. Let us suppose that ‘s’ gets transformed to ‘s’, ‘t’ gets transformed to ‘e’, ‘r’ gets transformed to ‘t’, ‘i’ became ‘u’, ‘n’ is deleted, and ‘g’ is substituted for by ‘f’. The new string that is to be operated on is thus $U' =$ “ setu$\xi f \xi$” . Finally, the ξ’s in U' are now transformed into the symbols of the alphabet A using the distribution Q. Let us suppose the first ξ gets changed into a ‘p’ and the second ξ gets transformed into an ‘o’. The final garbled version of U is thus $Y =$ “ setupfo” .

The process followed by the model is formally given as **Algorithm Generate_String** below. A graphical display of the channel modeling the garbling process is shown in Fig. 1. The theoretical properties of the OIT model are found in [22], and omitted here in the interest of brevity.

Algorithm 1. Generate_String

```
Input: The word $U$ and the distributions $G$, $Q$ and $S$.
Output: A random string $Y$ which garbles $U$ with random SID
mutations as per the OIT Model.
Method:
1: Using $G$ randomly determine $z$, the number of symbols to
   be inserted in $U$.
2: Randomly generate an input edit sequence $U' \in (A \cup \{\xi\})^*$ by
   randomly determining the positions of the insertions
   among the individual symbols of $U$.
3: Randomly independently substitute or delete the non-$\xi$
   symbols in $U'$ using $S$.
4: Randomly independently transform the occurrences of $\xi$
   into symbols of $A$ using $Q$.
5: return $Y$ as the final string obtained after the above
   operations.
End Algorithm Generate_String
```

[2] We assume that the user is capable of generating non-uniform random variables having the respective distributions G, Q and S, T. An excellent treatise on the subject is the one due to Devroye [8].

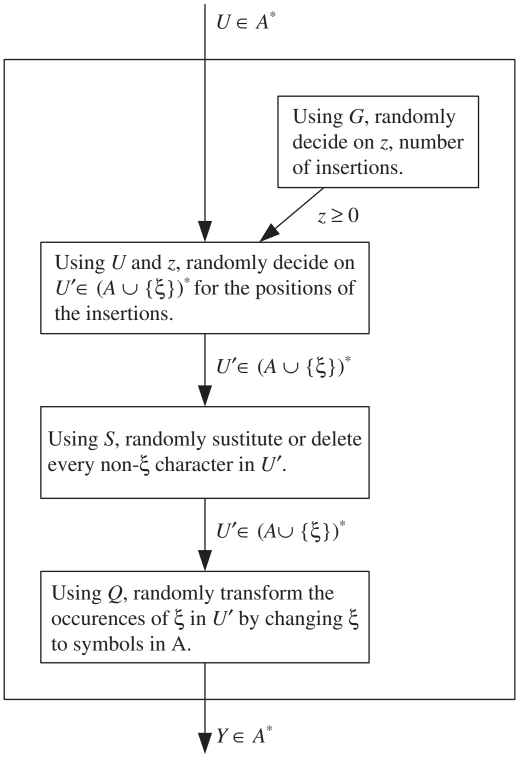
$U \in A^*$
Using G, randomly decide on z, number of insertions.
$z \geq 0$
Using U and z, randomly decide on $U' \in (A \cup \{\xi\})^*$ for the positions of the insertions.
$U' \in (A \cup \{\xi\})^*$
Using S, randomly sustitute or delete every non-ξ character in U'.
$U' \in (A \cup \{\xi\})^*$
Using Q, randomly transform the occurences of ξ in U' by changing ξ to symbols in A.
$Y \in A^*$

Fig. 1. A pictorial representation for the OIT model due to Oommen and Kashyap [22]. The input to the channel is the string U, and the output is the random string Y.

3. Proposed methodology

In this section, we provide the explicit details of the syntactic probabilities of the OIT model, and also explain the way by which we utilize it together with the SVM-pairwise scheme for peptide classification.

For a mutation consisting of random SID operations as per the OIT model, Oommen and Kashyap [22] have derived the syntactic probability of obtaining the sequence $Y = y_1 y_2 \ldots y_M$, from the sequence $U = u_1 u_2 \ldots u_N$ as

$$P(Y|U) = \sum_{z=\max(0,M-N)}^{M} \frac{G(z)N!z!}{(N+z)!} \sum_{U'} \sum_{Y'} \prod_{i=1}^{N+z} p(y'_i|u'_i), \tag{4}$$

where $G(z)$ is the probability of inserting z elements into U, and $p(y'_i|u'_i)$ is the probability of substituting the symbol element u'_i with the symbol element y'_i. Observe that in the above,

$$u'_i = \xi \Rightarrow y'_i \neq \lambda \quad \text{and} \quad y'_i = \lambda \Rightarrow u'_i \neq \xi.$$

The sum over the strings $U' = u'_1 u'_2 \ldots u'_{N+z}$ and $Y' = y'_1 y'_2 \ldots y'_{N+z}$ (of the same length), represent the sum over all possible pairs of strings U' and Y' of equal length $N+z$, generated by inserting ξ's into random positions in string U, and λ's into random positions in strings Y, respectively, and which are to represent the insertion and the deletion operations, respectively. Although this requires a summation over a combinatorially large number of elements (represented by U' and Y'), Oommen and Kashyap [22] have shown that this can be computed[3] in an extremely efficient manner in cubic time, i.e., with complexity $O(M \cdot N \cdot \min\{M,N\})$.

We now consider how the OIT model can be utilized for the particular problem at hand. The reader will observe that the OIT model essentially requires three "parameters" namely, S for the Substitution/Deletion probabilities, Q, for the insertion distribution, and G. With this as the background, we list the issues crucial to our solution:

1. The input and output alphabets in our application domain consist of 20 amino acids and one gap element, which for the input strings is the null symbol, ξ, representing an inserted element, and for output strings is the null symbol, λ, representing a deleted element.
2. The substitution of an amino acid with another corresponds to a series of mutations in the biological context. Based on this premise, we have computed our substitution probabilities on the mutation probability matrix refereed to as `PAM1` derived by the authors of [7]. `PAM1` is a 20 × 20 matrix, `M`, where each cell $\mathtt{m}_{ij}$ corresponds to the probability of replacing amino acid i with amino acid j after 1% of the amino acids are replaced. Indeed, it is possible to generate matrices for a series of longer mutations using successive multiplications of `PAM1`, and thus, for example, `PAM250` is equal to `PAM249` × `PAM1` [7].
3. The first major deviation from the traditional `PAM` matrices involves the operation of deletion. Observe that `PAM` matrices generally do not specify deletion probabilities for amino acids. As opposed to this, the OIT model of [22] suggests that an element can be deleted (substituted by λ) as well as substituted by another element. In this vein, we advocate that the matrix `PAM1` be extended by appending another column for λ, where the value Δ is assigned to the deletion probabilities of amino acids, and where each row is normalized to satisfy the probability constraint:
$$\sum_{y \in A \cup \{\lambda\}} p(y|u) = 1, \tag{5}$$
where A is the set of all amino acids, and u is the amino acid corresponding to the row.
4. There is no standard method of determining the deletion probabilities of amino acids. Comparing the widely used gap penalties as per [33] to the *log-odd* `PAM` matrices, we opted to use $\Delta = 0.0001$. The question of how to optimally determine Δ is open, and we are currently considering how it can be obtained from a training phase using known Input/Output string patterns.
5. The second major deviation from utilizing the traditional `PAM` matrices involves the operation of insertion. As in the case of deletion, we propose to extend the new `PAM` matrix by appending a row for ξ and assigned to $p(y|\xi)$ (i.e. the probability that a newly inserted amino acid is y) the relative frequency of observing y, $f(y)$. In our experiments, the relative frequencies were computed in a maximum likelihood manner by evaluating the limit of the `PAM`n matrix as n goes to infinity, i.e., as each row of the limiting matrix converges to $f(y)$. Finally, the remaining cell of our extended `PAM` matrix, $p(\lambda|\xi)$, is, by definition, equal to zero. The resulting matrix has been referred to as the `OIT_PAM` matrix, and is a 21 × 21 matrix. Table 1 gives a typical `OIT_PAM` matrix for the amino acid application domain. Observe that as in the case of the traditional `PAM` matrices, it is possible to derive higher order `OIT_PAM` matrices for longer mutation sequences by multiplying `OIT_PAM1` by itself. In our work, we have experimented with `OIT_PAM` matrices of different orders to observe the effect of different assumptions that concern evolutionary distances.

[3] Based on the work of [22], we have programmed our own toolkit to efficiently compute the syntactic probabilities between two arbitrary sequences, and adapted the tools to the particular application domain. We are willing to provide this tool to other researchers who are interested in collaborating with us on the use of these techniques and the OIT model for other bioinformatics applications.

Table 1
The `Log-OIT_PAM1` matrix used for the OIT model.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	λ
A	−0.01	−9.21	−7.82	−7.42	−9.21	−8.11	−6.91	−6.17	−9.21	−8.52	−8.11	−8.52	−9.21	−9.21	−6.65	−5.88	−6.12	−36.04	−9.21	−6.65	−9.21
R	−8.52	−0.01	−9.21	−36.04	−9.21	−7.01	−36.04	−9.21	−7.13	−8.52	−9.21	−5.60	−9.21	−9.21	−7.60	−6.81	−8.52	−8.52	−36.04	−8.52	−9.21
N	−7.01	−9.21	−0.02	−5.47	−36.04	−7.82	−7.26	−6.73	−6.32	−8.11	−8.11	−5.99	−36.04	−9.21	−8.52	−5.68	−6.65	−36.04	−8.11	−9.21	−9.21
D	−6.91	−36.04	−5.63	−0.01	−36.04	−7.60	−5.19	−6.81	−8.11	−9.21	−36.04	−7.42	−36.04	−36.04	−9.21	−7.26	−7.82	−36.04	−36.04	−9.21	−9.21
C	−8.11	−9.21	−36.04	−36.04	0.00	−36.04	−36.04	−9.21	−9.21	−8.52	−36.04	−36.04	−36.04	−36.04	−9.21	−6.81	−9.21	−36.04	−8.11	−8.11	−9.21
Q	−7.13	−6.91	−7.82	−7.42	−36.04	−0.01	−5.66	−8.11	−6.21	−9.21	−7.42	−6.73	−8.52	−36.04	−7.13	−7.82	−8.11	−36.04	−36.04	−8.52	−9.21
E	−6.38	−36.04	−7.42	−5.24	−36.04	−5.91	−0.01	−7.26	−9.21	−8.52	−9.21	−7.26	−36.04	−36.04	−8.11	−7.42	−8.52	−36.04	−9.21	−8.52	−9.21
G	−6.17	−36.04	−7.42	−7.42	−36.04	−9.21	−7.82	−0.01	−36.04	−36.04	−9.21	−8.52	−36.04	−9.21	−8.52	−6.44	−8.52	−36.04	−36.04	−8.11	−9.21
H	−8.52	−6.91	−6.17	−7.82	−9.21	−6.07	−8.52	−9.21	−0.01	−36.04	−7.82	−8.52	−36.04	−8.52	−7.60	−8.52	−9.21	−36.04	−7.82	−8.11	−9.21
I	−7.42	−8.11	−8.11	−9.21	−9.21	−9.21	−8.11	−36.04	−36.04	−0.01	−6.12	−7.82	−7.60	−7.13	−9.21	−8.52	−6.81	−36.04	−9.21	−5.17	−9.21
L	−7.82	−9.21	−9.21	−36.04	−36.04	−8.11	−9.21	−9.21	−9.21	−7.01	−0.01	−9.21	−7.13	−7.42	−8.52	−9.21	−8.52	−36.04	−9.21	−6.81	−9.21
K	−8.52	−6.27	−6.65	−8.11	−36.04	−7.42	−7.82	−8.52	−9.21	−8.52	−8.52	−0.01	−7.82	−36.04	−8.52	−7.26	−7.13	−36.04	−36.04	−9.21	−9.21
M	−7.42	−7.82	−36.04	−36.04	−36.04	−7.82	−9.21	−9.21	−36.04	−6.73	−5.40	−6.21	−0.01	−7.82	−9.21	−7.82	−7.42	−36.04	−36.04	−6.38	−9.21
F	−8.52	−9.21	−9.21	−36.04	−36.04	−36.04	−36.04	−9.21	−8.52	−7.26	−6.65	−36.04	−9.21	−0.01	−9.21	−8.11	−9.21	−9.21	−6.17	−9.21	−9.21
P	−6.12	−7.82	−8.52	−9.21	−9.21	−7.42	−8.11	−8.11	−8.11	−36.04	−8.11	−8.11	−36.04	−36.04	−0.01	−6.38	−7.60	−36.04	−36.04	−8.11	−9.21
S	−5.66	−7.42	−6.21	−7.60	−7.60	−8.52	−7.82	−6.17	−9.21	−9.21	−9.21	−7.13	−9.21	−8.52	−6.73	−0.02	−5.74	−9.21	−9.21	−8.52	−9.21
T	−5.74	−9.21	−7.01	−8.11	−9.21	−8.52	−8.52	−8.11	−9.21	−7.26	−8.11	−6.81	−8.52	−9.21	−7.82	−5.57	−0.01	−36.04	−9.21	−6.91	−9.21
W	−36.04	−7.13	−9.21	−36.04	−36.04	−36.04	−36.04	−36.04	−9.21	−36.04	−7.82	−36.04	−36.04	−8.11	−36.04	−7.60	−36.04	0.00	−8.52	−36.04	−9.21
Y	−8.52	−36.04	−7.82	−36.04	−8.11	−36.04	−9.21	−36.04	−7.82	−9.21	−8.52	−9.21	−36.04	−5.88	−36.04	−8.52	−8.52	−9.21	−0.01	−8.52	−9.21
V	−6.32	−9.21	−9.21	−9.21	−8.52	−9.21	−8.52	−7.60	−9.21	−5.71	−6.50	−9.21	−7.82	−36.04	−8.52	−8.52	−7.01	−36.04	−9.21	−0.01	−9.21
ξ	−2.43	−3.21	−3.20	−3.04	−3.43	−3.27	−2.99	−2.41	−3.42	−3.33	−2.46	−2.54	−4.21	−3.19	−2.96	−2.66	−2.84	−4.68	−3.45	−2.76	$-\infty$

Each element $\mathtt{M}_{i,j}$ is equal to the logarithm of the probability associated with the event of replacing the *i*th element with *j*th element. The symbols ξ and λ represent the insertion and deletion of an element of the alphabet, respectively. More details of this are found in Section 3.

6. The final parameter of the OIT model involves the Quantified Insertion distribution, $G(z)$, which specifies the probability that the number of insertions during the mutation is z. In our experiments, we have assumed that the probability of inserting an amino acid during a single PAM mutation is equal to the deletion probability of an amino acid, Δ. This assumption leads to the conclusion that for longer mutation series, the insertion distribution converges to a Poisson distribution such that

$$G(z) = \mathtt{Poisson}(z; n\Delta) = \frac{(n\Delta)^z e^{-\Delta n}}{z},$$

   where n is the number of PAMs (i.e. the length of the mutation series). In other words, we have currently used $\mathtt{Poisson}(z; n\Delta)$ as the insertion distribution whenever we use `OIT_PAM`*n* as the substitution probability matrix.
7. Using the OIT model and the parameters assigned as described above, a classification methodology based on the SVM-pairwise scheme proposed by Liao et al. [20] was devised. This will be explained in the next subsection.

Having explained how the OIT-based scheme works, we shall now also present the results obtained from our experiments.

4. Experiments and discussions

At the very outset, before we explain the experimental set-up and the results obtained, it is pertinent to emphasize the fact that our goal is not to compete with *complicated* pattern recognition techniques involving spectrum kernels etc. Rather, as the reader will observe, we have provided a new probabilistically consistent model and a sequence similarity metric, which have been proven to attain the corresponding information theoretic bound. Thus, from a mere theoretical perspective, we submit that our contribution involves the application of this model and the corresponding metric to the problem at hand. However, what is more impressive is the fact that it is, indeed, so successful—it can reach (and surpass) the state-of-the-art methods even with a simple classifier such as the linear SVM.

4.1. Experimental setup

In our experiments, we used two peptide classification data sets that are generally accepted as benchmark sets. The first one, referred to as HIV-754, was produced for the HIV-1 Protease cleavage site prediction problem by Kim et al. in [16]. It as an enhanced version of Cai and Chou's HIV-362 data set [3], and it contains 754 *8*-residue peptides with 396 positives and 358 negatives. The second data set, referred to as TCL-203, was produced for the T-cell epitope prediction problem by Zhao et al. in [37], and it contains 203 *10*-residue peptides of which 36 were positives and 167 were negatives.

In the first suite of experiments, we experimented with three different configurations:

1. The linear SVM with the OIT features.
2. The linear SVM with the Needleman–Wunsch (NW) alignment score-based features, and
3. The bio-basis function neural networks (BBFNN) of Thomson et al. [34].

As mentioned earlier, our SVM classification methodology was based on the SVM-pairwise scheme proposed by Liao and Noble [20] to detect remote evolutionary relationships between proteins. According to our scheme, m representative peptides were chosen *a priori* from the training set. Subsequently, for each instance, an *m*-dimensional vector of scores was computed by comparing the instance to the representatives, thus resulting in a maximum likelihood classifier. Our representatives were chosen to be the positive training instances. We also used the corresponding NW features in addition to the OIT, because the NW methodology is a commonly used sequence comparison method for peptide classification (see, for example, [23]).

It is well-known that operating on the logarithms of probabilities to improve numerical stability is a common procedure. This is true for our situation too. But apart from this, as a computational convenience, we have used the logarithm of the OIT probability as the measure of the similarity. This is because of the fact that the logarithm is a monotonic function, and furthermore, it turns out (we omit the algebraic details here in the interest of not unnecessarily complicating issues) that these logarithms can be computed more efficiently than the original OIT probabilities while traversing the 3-dimensional trellis.

The BBFNN, however, is a drastically different approach to the peptide classification problem than our SVM-based scheme. It is, in principle, similar to a radial-basis function neural network [5], with the difference being that instead of using similarities in a real-valued space, it uses sequence similarities, which have clear and straightforward biological significances. BBFNNs have been successfully applied to many biological problems including the detection of natively disordered regions in proteins [36], the identification of protein phosphorylation sites [1], the HIV-1 Protease cleavage site prediction [34,35] and the T-cell epitope prediction [35]. Indeed, it would be fair to consider the BBFNN as a state-of-the-art methodology, and thus we believe that a positive comparison with the BBFNN is definitely indicative of the advantages of our proposed scheme.

For each configuration, we used eight different substitution matrices with mutation lengths 10, 50, 100, 200, 250, 300, 400 and 500. In the testing phase, we estimated the performance of different methods by means of a cross-validation process. To do this, we divided the HIV-754 data set into 10 partitions and the TCL-203 data set, which is rather small, into five partitions as was done in [16,37], respectively. We also ensured the preservation of the ratio of positive and negative instances across the partitions. All the classification and performance estimations were performed on the Mathworks MATLAB [12] system with the help of PRTools 4.1, the pattern recognition toolbox [9], and LIBSVM 2.88, a library of support vector machine software modules [4].

4.2. Experimental results and discussions

We tested the three above-mentioned configurations for eight different substitution matrices on the two data sets. In each case, we recorded the area under the ROC (AUC), the accuracy (Acc), the sensitivity (Sens) and the positive predictive value (PPV). Tables 2 and 3 show the averaged values of these measurements for the HIV-754 and the TCL-203 data sets, respectively. In addition to these, the behaviors of the configurations for different score matrices can be seen in Figs. 2 and 3. These two figures display how the AUCs and their 95% confidence intervals vary as the assumption of the mutation length increases from 10 to 500 PAMs.

As one can observe from Tables 2 and 3, the OIT-based scheme generally yields results which are superior to both the NW-based scheme and the BBFNN for all substitution matrices, and with respect to *any* performance metric. In some cases the superiority is categorically marked—for example, whereas the best accuracy of OIT is 91.3% (for 250 PAMs in HIV-754 data set), the corresponding accuracy of the NW and the BBFNN are 86.3% and 84.1%, respectively. Tables 4 and 5 record the *t*-test results

Table 2
The performance measurements for the *HIV* data set using OIT, NW and BBFNN.

	PAM	10	50	100	200	250	300	400	500
OIT	AUC	0.948	0.962	0.968	0.969	0.965	0.965	0.958	0.949
	Acc	0.881	0.902	0.917	0.911	0.913	0.911	0.901	0.893
	Sens	0.863	0.891	0.897	0.877	0.874	0.863	0.849	0.830
	PPV	0.884	0.904	0.927	0.932	0.938	0.948	0.937	0.938
NW	AUC	0.881	0.890	0.890	0.897	0.891	0.890	0.892	0.832
	Acc	0.841	0.850	0.853	0.857	0.863	0.854	0.852	0.806
	Sens	0.476	0.522	0.542	0.583	0.592	0.577	0.602	0.461
	PPV	0.586	0.606	0.621	0.625	0.648	0.597	0.599	0.465
BBFNN	AUC	0.707	0.901	0.959	0.907	0.891	0.883	0.764	0.682
	Acc	0.702	0.856	0.903	0.863	0.841	0.850	0.744	0.672
	Sens	0.615	0.815	0.852	0.838	0.788	0.813	0.681	0.587
	PPV	0.721	0.872	0.941	0.874	0.875	0.847	0.756	0.631

The highest AUC values are underlined.

Table 3
The performance measurements for the *TCL* data set using OIT, NW and BBFNN.

	PAM	10	50	100	200	250	300	400	500
OIT	AUC	0.918	0.937	0.943	0.947	0.944	0.945	0.939	0.936
	Acc	0.852	0.872	0.882	0.897	0.902	0.887	0.887	0.882
	Sens	0.922	0.934	0.929	0.940	0.946	0.940	0.946	0.929
	PPV	0.901	0.912	0.928	0.935	0.936	0.924	0.919	0.928
NW	AUC	0.883	0.892	0.889	0.889	0.885	0.895	0.904	0.819
	Acc	0.837	0.842	0.847	0.853	0.853	0.852	0.867	0.793
	Sens	0.928	0.922	0.922	0.905	0.893	0.916	0.911	0.881
	PPV	0.882	0.891	0.895	0.917	0.927	0.905	0.928	0.871
BBFNN	AUC	0.719	0.857	0.910	0.934	0.856	0.917	0.865	0.767
	Acc	0.779	0.876	0.896	0.916	0.866	0.891	0.852	0.720
	Sens	0.839	0.946	0.958	0.965	0.940	0.964	0.940	0.726
	PPV	0.886	0.909	0.921	0.937	0.903	0.910	0.888	0.919

The highest AUC values are underlined.

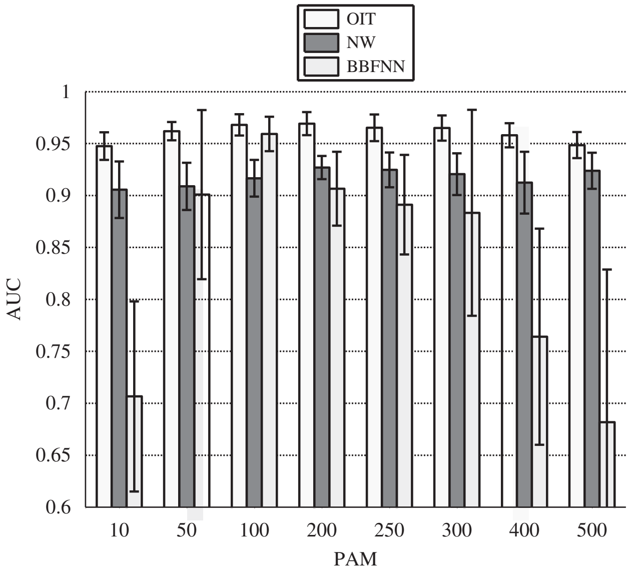

Fig. 2. The behavior of OIT, NW and BBFNN on the HIV-754 data set when the mutation length assumption changes between 10 and 500 PAMs. The error bars display the respective 95% confidence intervals.

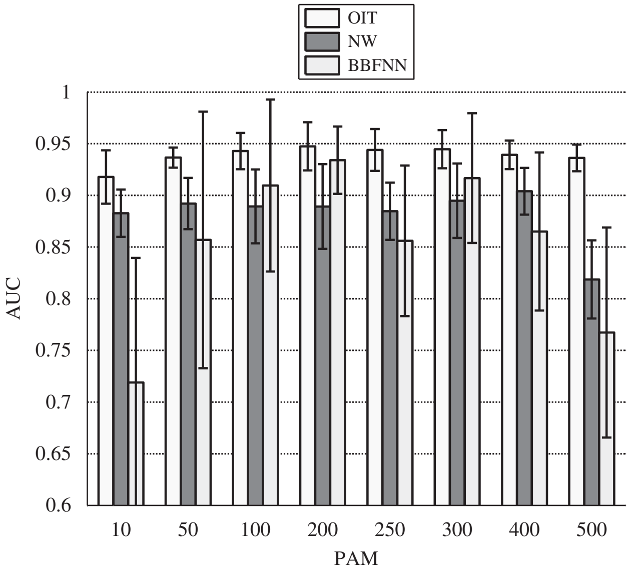

Fig. 3. The behavior of OIT, NW and BBFNN on the TCL-203 data set when the mutation length assumption changes between 10 and 500 PAMs. The error bars display the respective 95% confidence intervals.

Table 4
The *t*-test results for the 5% significance level comparing the AUC values of the OIT-based and NW-based schemes.

PAM	HIV-754		TCL-203	
	AUC(OIT) > AUC(NW)	*p*-Value	AUC(OIT) > AUC(NW)	*p*-Value
10	**Yes**	0.013	**Yes**	0.018
50	**Yes**	0.001	**Yes**	0.025
100	**Yes**	< 0.001	**Yes**	0.047
200	**Yes**	< 0.001	**Yes**	0.014
250	**Yes**	< 0.001	**Yes**	< 0.001
300	**Yes**	< 0.001	**Yes**	0.015
400	**Yes**	0.012	**Yes**	0.001
500	**Yes**	0.014	**Yes**	0.001

Table 5
The *t*-test results for the 5% significance level comparing the AUC values of the OIT-based scheme and BBFNN.

PAM	HIV-754		TCL-203	
	AUC(OIT) > AUC(BBFNN)	*p*-Value	AUC(OIT) > AUC(BBFNN)	*p*-Value
10	**Yes**	< 0.001	**Yes**	0.023
50	No	0.083	No	0.150
100	No	0.146	No	0.261
200	**Yes**	0.002	No	0.269
250	**Yes**	0.005	No	0.053
300	No	0.079	No	0.228
400	**Yes**	0.004	No	0.075
500	**Yes**	0.003	**Yes**	0.012

that validate the superiority of the OIT over both the NW and the BBFNN.

Another interesting observation is that whereas the performance of the BBFNN depends strongly on the substitution matrix, the NW's performance displays only a marginal dependence, while the performance of the OIT is almost independent of the substitution matrix. Therefore, even though the results seem to indicate that the BBFNN has the potential of possibly attaining the level of the OIT, it is clear that one has to carefully choose or optimize the substitution matrix, both of which are exhausting and computationally intensive processes.

The reader will also observe that for the HIV-754 data set, all of the three configurations attained their highest performances between the 100 and 200 PAM settings. For the TCL-203 data set, however, the NW prefers the `PAM400` parameters. The reader should also note that the 95% confidence intervals are generally wider for the TCL-203 data set than they are for the HIV-754 data set. We believe that this is because the cross-validation was performed through a five-fold strategy on the former, and through a 10-fold strategy on the latter.

4.3. *Comparison using the HIVcleave toolkit*

To further demonstrate the significance of our results, we have also taken the steps to compare our results with HIVcleave [29], which is a fairly well-known online tool for HIV-1 Protease cleavage site prediction. To place the latter in the right context, we mention that HIVcleave is primarily based on the works of Chou [6] and the discriminant function algorithm. To quantify the performance of HIVcleave, we fed all the peptides in the HIV-754 data set one by one into HIVcleave and recorded the scores generated. Having obtained these, we subsequently were able to measure an accuracy and an AUC value. The results obtained were quite conclusive: The accuracy and the AUC values for HIVcleave on HIV-754 data set is measured to be 0.833 and 0.899, respectively, which are even less than the minimum values measured for the OIT. Indeed, we can conclusively state that the OIT-based scheme attains AUCs which lead to 5.5–7.8% higher AUC values than HIVcleave for any substitution matrix.

4.4. *Comparison with literature*

Our experimental setup for the HIV-754 data set is compatible with the one in [16], where the authors provide the accuracy values for 10 different classifier and feature set combinations. The OIT-based scheme outperforms nine of them, while only the Gaussian SVM with orthogonal coding (i.e., with 8 × 20 binary features for each instance) is reported to have a marginally higher average accuracy value. However, it is impossible to decide if the superiorities are significant or not, as the authors have not provided the standard deviations. Similarly, our experimental setup for the TCL-203 data set is compatible with the one in [37]. Considering the fact that the authors of [37] have provided

Table 6
Comparison of various results on the HIV-1 Protease cleavage site prediction problem.

	Data set	AUC	Acc
OIT	HIV-754	0.968 (*0.028*)	0.917 (*0.018*)
NW	HIV-754	0.927 (*0.018*)	0.857 (*0.035*)
HIVcleave[a] [29]	HIV-754	0.899[b] (N/A)	0.833 (N/A)
GSVM[c] [16]	HIV-754		0.926[d]
BPNN[e] [3]	HIV-362[f]		0.921[g] (N/A)
BBFNN[h] [35]	HIV-362	0.910 (*0.050*)	0.858 (*0.049*)
EBBN[i] [35]	HIV-362	0.950 (*0.050*)	0.907 (*0.065*)

Numbers inside the parentheses indicate the reported standard deviation.

[a] HIVcleave is mostly based on [3].
[b] Tested with the whole data set.
[c] Gaussian SVM.
[d] Authors have not provided the standard deviation.
[e] Back-propagation neural network.
[f] The precursor of the HIV-754 data set.
[g] Authors have not performed a cross-validation.
[h] Bio-basis function neural network.
[i] Evolutionary bio-basis network.

Table 7
Comparison of various results on the T-cell epitope prediction problem.

	Data set	AUC	Acc
OIT	TCL-203	0.944 (*0.023*)	0.902 (*0.024*)
NW	TCL-203	0.904 (*0.026*)	0.867 (*0.042*)
LSVM[a] [37]	TCL-203	0.919[b]	
BBFNN[c] [35]	TCL-203	0.930 (*0.040*)	0.891 (*0.045*)
EBBN[d] [35]	TCL-203	0.910 (*0.100*)	0.884 (*0.085*)

Numbers inside the parentheses indicate the reported standard deviation.

[a] Linear SVM.
[b] Authors have not provided the standard deviation.
[c] Bio-basis function neural network.
[d] Evolutionary bio-basis network.

sensitivity, PPV and AUC values for seven different classifiers, we believe that it is noteworthy that the OIT-based scheme outperformed all of them.

There are many other works that use the HIV-754, TCL-203 or HIV-362 (the precursor of HIV-754) data sets. For the sake of completeness, we compiled the results we have obtained in this work and the results reported in the literature in Tables 6 and 7. The superiority of the OIT-based scheme is conclusive!

5. Conclusions and future work

In this paper, we have considered the problem of classifying peptides using syntactic pattern recognition methodologies. This problem has typically been tackled using distance-based metrics that involve the traditional edit operations of substitution, insertion and deletion (SID) required when the string representations of the respective peptides are compared. In this paper we have considered how the pattern recognition can be achieved by using the optimal and information theoretic (OIT) model of Oommen and Kashyap [22]. We have shown that one can model the differences between the compared strings as a mutation model consisting of random SID operations which obeys a OIT model. Consequently, by using the probability measure obtained from the OIT model as a pairwise similarity metric, we have devised a support vector machine (SVM)-based peptide classifier. The classifier has been tested for eight different substitution matrices and for two different data sets, namely, the HIV-1 protease cleavage sites and the T-cell epitopes, and the results obtained categorically demonstrate that the OIT model performs significantly better than the one which uses a Needleman–Wunsch sequence alignment score, and that when combined with a SVM, is among the best peptide classification methods available. Last but not least, the OIT is very robust regarding to the similarity matrices, which is shown to not be the case for the bio-basis function neural networks.

There are numerous avenues for future research. First of all, we believe that the entire concept of using the OIT model of Oommen and Kashyap [22] for other bioinformatics applications will be very interesting. The software to compute the OIT similarities between given sets of sequences is available from the corresponding author. More importantly, though, the reader will observe that we have, in this paper, merely used the probabilities for the PAM matrices as those that are already reported in the literature. However, the question of *training* the classifier to get the best (maximum likelihood or Bayesian) PAM matrix based on the training data is open. Finally, currently, as far as we know, the use of syntactical probabilities for peptide and other bioinformatics pattern recognition problems, has been limited to a *global* sequence analysis. In the future, we foresee that methods that involve a local version of such probabilistic methods could be more powerful, especially for the classification of proteins.

The final open issue concerns the scalability of our solution, which was raised by one anonymous referee. It is, indeed, true that in the experimental section, we tested the method only on short peptides. We believe, though, that our method would also scale to the sequences which contain several 100 proteins. However, the real problem is to *effectively* and *quickly* achieve the computation of the corresponding probabilities. While it is true that we are able to do precise computations on long sequences, nevertheless, as it stands now, the cubic time complexity of the OIT algorithm is a significant hurdle—implying that we cannot do this fast enough. Of course, this assumes that a global (as opposed to a local) inter-protein comparison is meaningful in the PR problem domain. In other words, at present, we can easily handle the case when we encounter *more* sequences, but the question of managing *longer* sequences in real-time is open-ended.

Acknowledgments

The first and third authors, Aygün and Cataltepe, were supported by TÜBİTAK (The Scientific and Technological Research Council of Turkey) Research Project EEEAG 105E164. The second author, Oommen, was partially supported by NSERC (The Natural Sciences and Engineering Research Council of Canada). Authors would also like to thank the anonymous referees whose comments greatly helped to improve the quality of this paper.

References

[1] E. Berry, A. Dalby, Z. Yang, Reduced bio basis function neural network for identification of protein phosphorylation sites: comparison with pattern recognition algorithms, Computational Biology and Chemistry 28 (1) (2004) 75–85.
[2] P. Bucher, K. Hofmann, A sequence similarity search algorithm based on a probabilistic interpretation of an alignment scoring system, in: Proceedings of the Conference on Intelligent Systems for Molecular Biology, 1996, pp. 44–51.
[3] Y.D. Cai, K.C. Chou, Artificial neural network model for predicting HIV protease cleavage sites in protein, Advances in Engineering Software 29 (2) (1998) 119–128.
[4] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at: 〈http://www.csie.ntu.edu.tw/~cjlin/libsvm〉.
[5] S. Chen, C. Cowan, P. Grant, et al., Orthogonal least squares learning algorithm for radial basis function networks, IEEE Transactions on Neural Networks 2 (2) (1991) 302–309.
[6] K. Chou, Prediction of human immunodeficiency virus protease cleavage sites in proteins, Analytical Biochemistry 233 (1) (1996) 1–14.
[7] M. Dayhoff, R. Schwartz, B. Orcutt, A model of evolutionary change in proteins, Atlas of Protein Sequence and Structure 5 (Suppl. 3) (1978) 345–352.
[8] L. Devroye, Non-uniform Random Variate Generation, Springer-Verlag, New York, 1986.
[9] R.P.W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D.M.J. Tax, PRTools, a Matlab Toolbox for Pattern Recognition, Delft University of Technology, 2004.
[10] S.R. Eddy, Multiple alignment using hidden Markov models, in: Proceedings of the International Conference on Intelligent Systems for Molecular Biology, vol. 3, 1995, pp. 114–120.
[11] I. Gozes, O. Perl, E. Giladi, A. Davidson, O. Ashur-Fabian, S. Rubinraut, M. Fridkin, Mapping the active site in vasoactive intestinal peptide to a core of four amino acids: neuroprotective drug design, Proceedings of the National Academy of Sciences USA 96 (7) (1999) 4143–4148.
[12] M.R. Guide, The MathWorks, Inc., Natick, MA, 1998.
[13] Y. Hou, W. Hsu, M. Lee, C. Bystroff, Efficient remote homology detection using local structure, Bioinformatics 19 (17) (2003) 2294–2301.
[14] E. Ie, J. Weston, W. Noble, C. Leslie, Multi-class protein fold recognition using adaptive codes, in: Proceedings of the 22nd International Conference on Machine LearningACM, New York, 2005, p. 336.
[15] K. Karplus, C. Barrett, R. Hughey, Hidden Markov models for detecting remote protein homologies, Bioinformatics 14 (10) (1998) 846–856.
[16] H. Kim, Y. Zhang, Y.S. Heo, H.B. Oh, S.S. Chen, Specificity rule discovery in HIV-1 protease cleavage site analysis, Computational Biology and Chemistry 32 (1) (2008) 71–78.

[17] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, C. Leslie, Profile-based string kernels for remote homology detection and motif extraction, Journal of Bioinformatics and Computational Biology 3 (3) (2005) 527–550.

[18] C. Leslie, E. Eskin, W. Noble, The spectrum kernel: a string kernel for SVM protein classification, in: Proceedings of the Pacific Symposium on Biocomputing, vol. 7, 2002, pp. 566–575.

[19] H. Li, T. Jiang, A class of edit kernels for SVMs to predict translation initiation sites in eukaryotic mRNAs, Journal of Computational Biology 12 (6) (2005) 702–718.

[20] L. Liao, W.S. Noble, Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships, Journal of Computational Biology 10 (6) (2003) 857–868.

[21] H. Mamitsuka, Predicting peptides that bind to MHC molecules using supervised learning of hidden Markov models, Proteins 33 (4) (1998) 460–474.

[22] B.J. Oommen, R.L. Kashyap, A formal theory for optimal and information theoretic syntactic pattern recognition, Pattern Recognition 31 (8) (1998) 1159–1177.

[23] E.E. Oren, C. Tamerler, D. Sahin, M. Hnilova, U.O.S. Seker, M. Sarikaya, R. Samudrala, A novel knowledge-based approach to design inorganic-binding peptides, Bioinformatics 23 (21) (2007) 2816–2822.

[24] K.C. Parker, M.A. Bednarek, J.E. Coligan, Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide sidechains, Journal of Immunology 152 (1) (1994) 163–175.

[25] H.G. Rammensee, T. Friede, S. Stevanoviic, MHC ligands and peptide motifs: first listing, Immunogenetics 41 (4) (1995) 178–228.

[26] H. Rangwala, G. Karypis, Profile-based direct kernels for remote homology detection and fold recognition, Bioinformatics 21 (23) (2005) 4239–4247.

[27] M. Sarikaya, C. Tamerler, A. Jen, K. Schulten, F. Baneyx, Molecular biomimetics: nanotechnology through biology, Nature Materials 2 (9) (2003) 577–585.

[28] G. Selivanova, V. Iotsova, I. Okan, M. Fritsche, M. Stroem, B. Groner, R. Grafstroem, K. Wiman, Restoration of the growth suppression function of mutant p53 by a synthetic peptide derived from the p53 C-terminal domain, Nature Medicine 3 (1997) 632–638.

[29] H. Shen, K. Chou, HIVcleave: a web-server for predicting human immunodeficiency virus protease cleavage sites in proteins, Analytical Biochemistry 375 (2) (2008) 388–390.

[30] E. Sigurdsson, H. Scholtzova, P. Mehta, B. Frangione, T. Wisniewski, Immunization with a nontoxic/nonfibrillar amyloid-beta homologous peptide reduces Alzheimer's disease-associated pathology in transgenic mice, American Journal of Pathology 159 (2) (2001) 439–447.

[31] J. Sloan-Lancaster, P. Allen, Altered peptide ligand-induced partial T cell activation: molecular mechanisms and role in T cell biology, Annual Reviews in Immunology 14 (1) (1996) 1–27.

[32] J. Soding, Protein homology detection by HMM–HMM comparison, Bioinformatics 21 (7) (2005) 951–960.

[33] T.A. Tatusova, T.L. Madden, BLAST 2 sequences, a new tool for comparing protein and nucleotide sequences, FEMS Microbiology Letters 174 (2) (1999) 247–250.

[34] R. Thomson, T.C. Hodgman, Z.R. Yang, A.K. Doyle, Characterizing proteolytic cleavage site activity using bio-basis function neural networks, Bioinformatics 19 (14) (2003) 1741–1747.

[35] D.C. Trudgian, Z.R. Yang, Substitution matrix optimisation for peptide classification, in: Lecture Notes in Computer Science, vol. 4447, 2007, p. 291.

[36] Z. Yang, R. Thomson, P. McNeil, R. Esnouf, RONN: the bio-basis function neural network technique applied to the detection of natively disordered regions in proteins, Bioinformatics 21 (16) (2005) 3369.

[37] Y. Zhao, C. Pinilla, D. Valmori, R. Martin, R. Simon, Application of support vector machines for T-cell epitopes prediction, Bioinformatics 19 (15) (2003) 1978–1984.

About the Author—ESER AYGÜN was born on 11 August 1983 in Izmir, Turkey. After graduating from Izmir Scientific High School in 2001, he obtained his B.S. degree from Computer Engineering Department of Istanbul Technical University in 2006. He then worked on pattern recognition and bioinformatics in Computer Engineering Program of Istanbul Technical University until he completed his M.S. Thesis on January 2009. Currently, he is a Ph.D. candidate in Computer Engineering Program of Istanbul Technical University and he is working as a part-time Research Assistant. He published several papers and posters on document classification, pattern recognition, bioinformatics and complex networks.

About the Author—JOHN OOMMEN was born in Coonoor, India on 9 September 1953. He obtained his B.Tech. degree from the Indian Institute of Technology, Madras, India in 1975. He obtained his M.E. from the Indian Institute of Science in Bangalore, India in 1977. He then went on for his M.S. and Ph.D. which he obtained from Purdue University, in West Lafayettte, Indiana in 1979 and 1982, respectively. He joined the School of Computer Science at Carleton University in Ottawa, Canada, in the 1981–1982 academic year. He is still at Carleton and holds the rank of a Full Professor. Since July 2006, he has been awarded the honorary rank of Chancellor's Professor, which is a lifetime award from Carleton University. His research interests include automata learning, adaptive data structures, statistical and syntactic pattern recognition, stochastic algorithms and partitioning algorithms. He is the author of more than 300 refereed journal and conference publications, and is a Fellow of the IEEE and the IAPR. Dr. Oommen has served on the Editorial Board of the journals IEEE Transactions on Systems, Man and Cybernetics, and Pattern Recognition.

About the Author—ZEHRA CATALTEPE is an Associate Professor at Computer Engineering Department of Istanbul Technical University. Her research interests are machine learning theory and applications, especially in bioinformatics, social network analysis, web/document mining, and music recognition and recommendation. She got her Ph.D. degree from Caltech in Computer Science in 1998 and her B.S. degree from Bilkent University, Ankara, in 1992 where she graduated first in her class. She worked at Bell Labs as a Postdoc and then at StreamCenter Inc. and Siemens Corporate Research as Researcher after she got her Ph.D. She has six U.S. patents and is the author of more than 35 publications in the form of book chapters, journal and refereed conference papers.