



Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/damPattern analysis for the prediction of fungal pro-peptide cleavage sites[☆]S. Özögür-Akyüz^{a,*}, J. Shawe-Taylor^b, G.-W. Weber^a, Z.B. Ögel^c^a Institute of Applied Mathematics, Middle East Technical University, 06531 Ankara, Turkey^b Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, United Kingdom^c Department of Food Engineering, Middle East Technical University, 06531 Ankara, Turkey

ARTICLE INFO

Article history:

Received 19 January 2007

Received in revised form 18 June 2007

Accepted 11 June 2008

Available online xxxx

Keywords:

Computational biology

Machine learning

Support vector machines

Continuous optimization

LibSVM

Pro-peptide cleavage site

ABSTRACT

Support vector machines (SVMs) have many applications in investigating biological data from gene expression arrays to understanding EEG signals of sleep stages. In this paper, we have developed an application that will support the prediction of the pro-peptide cleavage site of fungal extracellular proteins which display mostly a monobasic or dibasic processing site. Many of the secretory proteins and peptides are synthesized as inactive precursors and they become active after post-translational processing. A collection of fungal pro-protein sequences are used as a training data set. A specifically designed kernel is expressed as an application of the well-known Gaussian kernel via feature spaces defined for our problem. Rather than fixing the kernel parameters with cross validation or other methods, we introduce a novel approach that simultaneously performs model selection together with the test of accuracy and testing confidence levels. This leads us to higher accuracy at significantly reduced training times. The results of the server ProP1.0 which predicts pro-peptide cleavage sites are compared with the results of this study. A similar mathematical approach may be adapted to pro-peptide prediction in other eukaryotes.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

There is a growing interest in the application of machine learning techniques together with optimization to real world applications such as biological problems, engineering problems etc. This paper is devoted to solving one of the important problems in peptide biology, namely predicting pro-peptide cleavage site for a given amino acid sequence of a protein by using an SVM (support vector machine) [5] which is introduced by a novel *confidence level* model selection algorithm. There have been many studies on predicting peptide regions such as signal peptide [4,6,7,10,12,21,32], pro-peptide [13] solved with neural networks method [28] using classical model selection methods such as cross validation (CV) [15].

In this study, we have developed an efficient and novel model selection algorithm embedded in a classical SVM to predict pro-peptide cleavage sites in *filamentus fungi*. Prediction results of the confidence level by an SVM are compared with the results achieved by the pro-peptide prediction tool ProP1.0 [13]. **ProP1.0** is a bioinformatics tool which predicts pro-peptide cleavage sites on a furin specific based network and general PC network separately using a **neural network**. ProP1.0 consists of 227 proteins of all eukaryotes including those of humans and animals. The data set is presented to the neural network by sparsely encoded moving windows. The output of neural network is assessed by a threshold of 0.5 to determine the potential pro-peptide cleavage site.

[☆] This study is supported by PASCAL (Pattern Analysis, Statistical Modelling and Computational Learning) and Institute of Applied Mathematics, METU, Turkey.

* Corresponding author. Fax: +90 216 483 9550.

E-mail addresses: sozogur@metu.edu.tr (S. Özögür-Akyüz), jst@cs.ucl.ac.uk (J. Shawe-Taylor), gweber@metu.edu.tr (G.-W. Weber), zogel@metu.edu.tr (Z.B. Ögel).

This study concentrates more on fungal proteins due to the industrial importance of these organisms in heterologous protein production, including those of humans. The data set is collected from largely non-homologous fungal proteins consisting of 72 sequences. Our prediction tool, *confidence level SVM* is fed with both binary input vectors and the substitution matrix PAM250 separately and results are reported for both. The sequences are given to the learning machine by encoded sliding windows through each sequence. Each protein is tested with different training sets. Rather than splitting the data set into groups we have used a different strategy that enables us to use the whole data for both training and testing. This is explained in detail in the next section. The construction of the data set from non-homologous sequences is justified by using ClustalW to construct a phylogenetic tree which is through multiple sequence alignment.

1.1. N-terminal pro-peptides of fungal secreted proteins

There is growing interest to the proteolytic processing events in cell biology due to recent findings related to their critical functions in apoptosis [33], in triggering human diseases such as Alzheimer's [14] as well as their well-known role in cell trafficking [30]. Most of the proteolytic processing events take place at the N-terminus of proteins. Among these processes, signal peptide cleavage is perhaps the most well-known, and software programmes are established by which eukaryotic and prokaryotic signal peptides and cleavage sites can be predicted with high precision such as the SignalP Server [21]. Relatively less is known about the function and mechanism of the post-translational removal of pro-peptides which exist not only in secreted proteins but also in some of the proteins that do not pass through the endoplasmic reticulum.

Fungi are ideal model systems for the study of eukaryotic molecular mechanisms due to their relative simplicity. In general, filamentous fungi are more closely related to higher eukaryotes than the yeasts. Filamentous fungi have also attracted much attention due to their importance as heterologous expression systems [19] although some yeasts such as *Pichia pastoris* are also promising as heterologous hosts. Particularly *Aspergillus* and *Trichoderma* species find wide use as industrial protein factories. While large amounts of fungal proteins are heterologously produced in efficient and safe hosts such as *Aspergillus sojae* [23], still more progress is necessary to enhance the heterologous production of mammalian proteins [17]. This requires more in-depth understanding of the events taking place during the secretion process.

At the N-terminus, transient peptides may consist of only a signal peptide or may also contain one or more additional peptides. Here, signal peptide sequences were not considered, however, only proteins with a predicted signal peptide were selected from the NCBI Genbank database. In general, a single additional peptide is called a pro-peptide. If there are two additional peptides, they are called *pre-peptide* and *pro-peptide*, respectively.

1.2. Functions of pro-peptides

The pro-peptides have been implicated in a number of cellular processes including their role as an intracellular chaperone [25], in proper folding, in subcellular sorting and in keeping proteins in an inactive configuration. The pro-peptide is removed upon or before departure from the secretory pathway by maturases [3] that reside either in the late stage of the Golgi, the secretory vesicles or are extracellularly anchored to the cytoplasmic membrane with a GPI-anchor [29]. The processing of most of the pro-peptides occurs at either a monobasic or a dibasic cleavage site [3]. Dibasic cleavage is directed by the kexin family of endoproteases whereas monobasic cleavage is conducted in yeast by the yapsin family of endoproteases [16] and by the furin-type of proteases in *Trichoderma* [24]. A significant group of proteins, including mainly the proteases, is processed by autocatalytic cleavage [22].

1.3. Dibasic processing

Multiple Sequence Alignment results show that fungal pro-peptides of secreted proteins are cleaved following a dibasic site. In the majority of dibasic processing sites, cleavage takes place following a "Lys-Arg" pair whereas "Lys-Lys" and "Arg-Arg" pairs are less frequently encountered [22].

1.4. Monobasic processing

A remarkable number of filamentous fungal extracellular proteins possess a monobasic cleavage site at their leader-mature protein junction. Considering the putative pro-peptides of proteins that are subject to monobasic processing, a common sequence motif does not exist, with the exception of a proline that is consistently present and frequently adjacent to a Leu or Ile. The fact that the pro-peptides contain both hydrophilic and hydrophobic residues and the absence of sequence homology could either indicate processing by different proteases or the importance of conformational determinants for cleavage; in the latter case the presence of a proline may be highly significant. In filamentous fungi there are no examples where proline is present immediately before or after the basic residue at the cleavage site. Nevertheless, since the role of proline is suggested to be at the level of three-dimensional structure, rather than the primary sequence [22], a similar function can still be attributed to the proline residues within the structure of pro-peptides of filamentous fungi where monobasic processing takes place.

2. Materials and methods

The data set is collected from the NCBI databank based on fungal proteins which are publicly available.¹ 72 fungal sequences are selected among non-homologous protein families. This is one of the reasons for the small number of sequences contained. To reduce further redundancy in the data set and prevent the training and testing from being homologous, we made a phylogenetic tree analysis based on multiple sequence alignment by ClustalW. There, a tree many individual main branches resulted (data not shown) indicating that the selected proteins are not homologous. In our learning process by SVM, we chose symmetric windows around possible cleavage sites, where the window length varies between 11 to 21 and the results indicates that the optimum window length lies between 13 and 19. The best accuracy results are found with window length chosen as 15. These parameters can vary according to the type of data set and the kind of problem.

To see the discriminative motifs existing in the sequences, we used *MEME* software. This yielded the motif **KR**. To check this result, *Multiple Sequence Alignment*, MLA with the package *ClustalW* is applied to the data set which confirms this observation. The motif **KR** gives us a clue for the preparation of the input sequences for the SVM. With MLA, most of the cleavage site patterns are in the form of either **K**, **R** or **KR**. Therefore, it is sensible to train the SVM restricted to inputs with **K** or **R** residues.

2.1. Input and output for the SVM

There are different ways to represent *text based* data when introducing the data to a learning algorithm. In bioinformatics, these data can be amino acid (a.a.) sequences, DNA sequences, etc. The most popular method of encoding amino acid sequences into numerical values is given by binary vectors [2]. However, this ignores the *context* information. There has been a lot of research on encoding amino acids to give each individual amino acid a numerical value regarding the biochemical and physiochemical properties [18]. One of the most powerful substitution matrices is PAM250 matrix due to its property of preserving mutations of the sequences. In this study, two types of encoding are considered, namely, a binary encoding matrix and the PAM250 substitution matrix. Note that, encoding a.a. by substitution matrices are needed for the input vectors for the SVM. Thus, the windows of a.a. sequences are presented to the SVM with the numerical values corresponding to the input vectors.

There are many similarity matrices developed according to different similarity approaches and gap penalties given between two amino acids. Dayhoff et al. [11] created a table where they aligned the proteins in several families of proteins and constructed phylogenetic trees for each family [11]. The resulting similarity table presents relative frequencies with which amino acids replace each other in a short evolutionary period since each phylogenetic tree was checked for the substitutions found on each branch. The traditional Dayhoff PAM250 matrix assumes the occurrence of 250 point mutations per 100 amino acids or 300 nucleotides in the gene [20].

PAM matrices are theoretically more advantageous than the others. They arise from Dayhoff's method [11] which is based on observed evolutionary mutations. Hence, they preserve information given by the processes that generate the mutations. Statistically, PAM matrices and other log-odds matrices are the most accurate description of the changes in the amino acid composition after a given number of mutations. Details about the formulation of log odds matrices and PAM matrices can be found in [1,11].

Since we have 20 amino acids, we have entries in a 20×20 PAM250 matrix. Each amino acid is represented by a 20 dimensional vector corresponding to the entries in a column of the PAM250 matrix. If there is a sequence of n amino acids then we will have an $n \times 20$ dimensional real-valued vector as input.

2.2. Sliding window approach for constructing a test set

The *sliding window approach* is a method to construct the training and test set with a previously chosen window size. Training windows are chosen from the neighborhood of the potential cleavage sites in such a way that the cleavage sites are at the centre of the window. For example, if we have a window size of 11, then the considered cleavage site is between the 5th and the 6th position of the window. In this way, each sequence contributes one positive window. For the negative class three windows are chosen from each sequence by selecting positions which have residues **K** or **R** at their centre. Here, windows are chosen as symmetric in all cases. A test sequence is constructed by sliding the window through the whole sequence as illustrated in Fig. 1. In our case, all the sequences have at least one **K** or **R** which are the motifs that we learned from ClustalW through Multiple Sequence Alignment. Sliding windows through the whole sequence generate many test windows, i.e., test inputs. Furthermore, the cleavage window(s) in the test sequence are going to be labeled as a positive class from the output of SVM and the others as a negative class. It is clear that restricting the windows by including to those that have **K** or **R** at their center will decrease the number of test examples and, hence, makes it easier to select the positive one(s) (cleavage window(s)) when compared to the high number of windows for a particular test sequence. In other words, if we call the set of all sliding windows S and choose a special subset $A \subseteq S$ which depends on motifs known in advance from a bioinformatics tool, then searching a cleavage window(s) among A will be easier than searching from the bigger set S for a

¹ <http://www3.iam.metu.edu.tr/iam/images/1/1a/Datasetsureyya.pdf>.

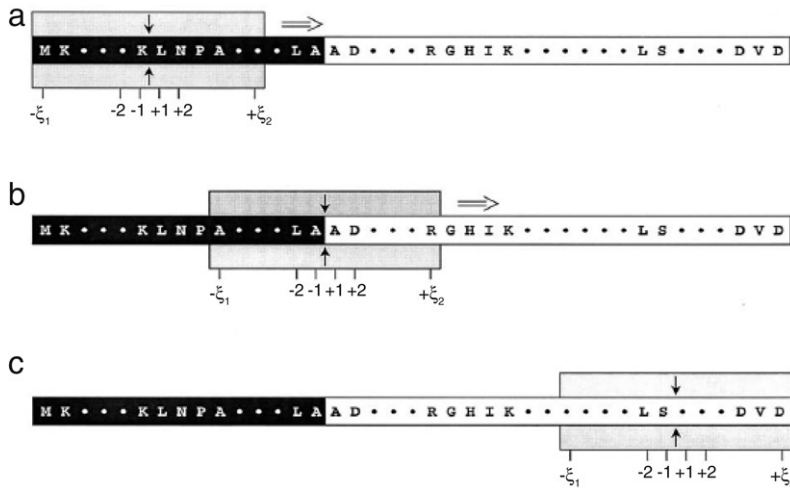


Fig. 1. Black parts denote pro-peptide region and white parts stand for the mature part of the protein [9]: (a) The window does not containing the cleavage site. (b) The window contains the cleavage site at its center. (c) The window does not contain the cleavage site.

particular test sequence. If the set A is empty, i.e., $A = \emptyset$, the set S which contains all possible windows of the particular test sequence can be used as test examples. In our special data set on fungal proteins, the subset A of S is nonempty, i.e., $A \neq \emptyset$. Moreover, the cardinality of A is always greater than 3, i.e., $|A| \geq 4$.

Our data set comprises 72 proteins and, hence, 72 amino acid sequences, each giving rise to be one positive window and three negative windows. So, 72 sequences are used for both training and testing using the *leave one out* principle that leaves each sequence in turn as testing while using the remaining 71 for training. In this way, we have trained using 71 sequences and have tested 1 sequence 72 times. The accuracy is calculated as the total number of correct predictions over the 72 sequences.

2.3. Kernel definitions

For string data, SVM can make use of string kernels which are described in [27]. These types of kernels can be used in text mining, DNA sequences and protein sequences. Since measuring the similarity between the windows of sequences is one of the most crucial items, a novel kernel function is defined with an explicit mapping Φ which measures the similarity of windows by counting the number of matching sequences in a neighbourhood of the potential cleavage site and it is shown to be a *Gaussian kernel* in a Proposition 2.1. Thus, the calculation of very high dimensional vectors Φ , is avoided by the use of a Gaussian kernel.

In our first method, which is explained in Section 2.1 with PAM250 matrices, we choose the Gaussian kernel while using the LibSVM package [8]. In order to motivate our second choice of kernel, we consider counting the number of matching sequences between two windows.

Let us regard windows to be sequences of amino acids indexed by $\{1, 2, \dots, n\}$. Moreover, let the feature space F_r be indexed by pairs (s, \mathbf{i}) , where s is a sequence of r amino acids and $\mathbf{i} = (i_1, i_2, \dots, i_r)$ a tuple of r distinct indices, $i_j \in \{1, 2, \dots, n\}$. We define the mapping $\Phi : w \mapsto (\phi_1(w), \phi_2(w), \dots, \phi_r(w), \dots) \in \prod_{r \geq 1} F_r$ by

$$\phi_r(w)_{(s, \mathbf{i})} = \begin{cases} \alpha^{r/2}, & \text{if } w_{\mathbf{i}} = s \\ 0, & \text{otherwise,} \end{cases} \tag{2.1}$$

where $w_{\mathbf{i}} = s$ means $w_{i_j} = s_j$ ($j = 1, 2, \dots, r$) and $\alpha \in \mathbb{R}, \alpha > 0$.

The feature space in which the learning will be conducted is $\prod_{r \geq 1} F_r$. It is worth nothing that this is a very high dimensional space. For example, F_5 has dimension

$$20^5 \binom{n}{5},$$

though, clearly, for $r > n$ all $\binom{n}{r}$ become 0. So, for any fixed n , the effective dimension is finite. The feature space makes it possible for the learning to assign weights for each pattern of positions and corresponding choice of amino acids at those positions.

This choice of feature space ensures the learning can readily identify the salient patterns that indicate the presence of a cleavage site. Naturally, it will not be practical to compute this feature vector explicitly. Now, we show in the next proposition that there is an efficient method of computing the kernel corresponding to the feature map Φ . This opens the way for us to learn in this feature space by using the kernel methods approach introduced above.

Let us consider (with a slight abuse of notation) the feature vector

$$\Phi(w) = [u_{w_1}, u_{w_2}, \dots, u_{w_n}]^T, \tag{2.2}$$

where u_a is defined as $u_a = [0 \dots 010 \dots 0]_{1 \times 20}$ with the value 1 in the position corresponding to the amino acids.

Proposition 2.1. *Using the defined notation above, we have for all windows v, w of size n :*

$$\begin{aligned} \kappa(v, w) &= \langle \Phi(v), \Phi(w) \rangle \\ &= (1 + \alpha)^n \exp\left(-\frac{\|\Phi(v) - \Phi(w)\|_2^2 \ln(1 + \alpha)}{2}\right). \end{aligned} \tag{2.3}$$

Proof. If we fix a number r of matches, we compare two windows by counting the number of r tuples of position that contain the identical set of amino acids. If the number of positions where the sequences of two windows agree is m , then the number of r tuples is given by the binomial coefficient $\binom{m}{r}$. This is the inner product associated with the high dimensional representation ϕ_r . Let us denote this kernel by $\kappa_r(v, w) = \alpha^r \langle \phi_r(v), \phi_r(w) \rangle$. Observe that by using a combination of these kernels, we can create our measure of similarity:

$$\kappa(v, w) = \sum_{r=0}^{\infty} \kappa_r(v, w) = \sum_{r=0}^{\infty} \alpha^r \binom{m}{r} = \langle \Phi(v), \Phi(w) \rangle.$$

Here, $m := \#\{i : v_i = w_i, i = 1, 2, \dots, l\}$, which we will denote by $\#[v == w]$, gives the number of positions in which the two sequences agree.

Therefore, from the Binomial Theorem, we have

$$\kappa(v, w) = (1 + \alpha)^{\#[v == w]}. \tag{2.4}$$

We note that $\langle \Phi(v), \Phi(w) \rangle = \#[v == w]$, while $\|\Phi(v)\|_2^2 = n$.

Letting $m = \#[v == w]$, we have

$$\begin{aligned} \kappa(v, w) &= (1 + \alpha)^m \\ &= \exp\left[m \ln(1 + \alpha) - \|\phi(v)\|_2^2 \ln(1 + \alpha)/2 - \|\phi(w)\|_2^2 \ln(1 + \alpha)/2\right] (1 + \alpha)^n \\ &= (1 + \alpha)^n \exp\left(-\frac{\|\phi(v) - \phi(w)\|_2^2 \ln(1 + \alpha)}{2}\right). \end{aligned}$$

Hence, the kernel turns out to be

$$\kappa(v, w) = (1 + \alpha)^n \exp\left(-\frac{\|\Phi(v) - \Phi(w)\|_2^2 \ln(1 + \alpha)}{2}\right)$$

as required. □

Remark 2.1. We note that, Eq. (2.3) is a *scaled Gaussian kernel* with kernel width $\sigma = \sqrt{\frac{1}{\ln(1+\alpha)}}$ by the definition of the Gaussian kernel over the features $\Phi(\cdot)$. We again consider both normalized and unnormalized versions of the features $\Phi(\cdot)$, though this only affects the scaling of the kernel width since $\|\Phi(v)\|_2^2 = \text{constant}$.

3. Model selection procedure

The definition of the kernel and the SVM algorithm both involve an additional parameter vector (C_+, C_-, σ) , the parameters C_+ and C_- for the SVM and the kernel width σ for the Gaussian kernels. The usual way to set these parameters is using cross-validation [15]. This assesses the quality of different parameter settings by dividing the training data into m groups. It then leaves out one group in turn to train the classifier with a range of possible values for the parameters and uses the group left out as a test set. The average accuracy for each parameter setting over all m test groups is then used to select the parameter settings. We used this approach where we took $m = 71$, i.e., we performed a subround of “leave one out” error estimation on each training set in order to select the parameters to use training for the set of 71 sequences before testing on 72nd left out sequence. Note that this is only *leave one out* at the level of sequences, since each sequence corresponds to 4 windows, one of which is positive.

Our second method of model selection is a novel approach for problems in which each test involves multiple inputs, but with the additional information that only one is positive: in our case, there are many windows, but only one is a cleavage site. Rather than to pre-select the parameters, we train the SVM on all the training data (other than the single test sequence) with all the parameter settings. For each SVM we compute the real-valued outputs, for all the windows arising from the sequence. We define the confidence of the classifier as the difference between the maximal output and the second largest;

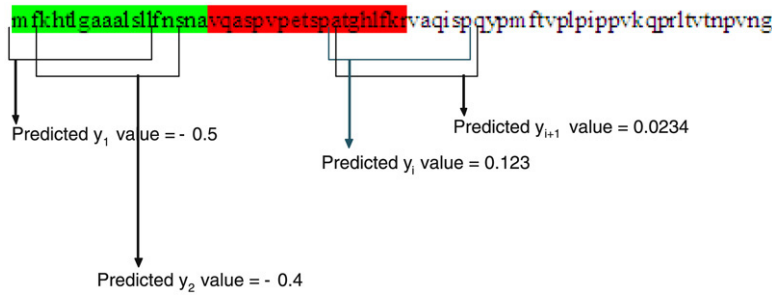


Fig. 2. Different real-valued outputs from the SVM. The confidence level is the highest difference between two maximum positive outputs.

Table 1 Accuracy results of SVM

| Input type | Cross validation (%) | Confidence level (%) |
|--|----------------------|----------------------|
| Normalized data encoded by binary vec. | 44 | 76 |
| Not normalized data encoded by binary vec. | 47 | 75 |
| Normalized data encoded by PAM250 | 37 | 58 |
| Not normalized data encoded by PAM250 | 33 | 56 |

Table 2 Average training time for the SVM for one of the 72 test sequence

| Input type | Cross Validation (s) | Confidence Level (s) |
|--|----------------------|----------------------|
| Normalized data encoded by binary vec. | 151 | 4 |
| Not normalized data encoded by binary vec. | 163 | 5 |
| Normalized data encoded by PAM250 | 1312 | 23 |
| Not normalized data encoded by PAM250 | 1924 | 34 |

see Fig. 2. Now, we select the parameter settings for which the confidence is largest and identify the window with maximal output as the cleavage site. It should be stated that not every test sequence has to have a cleavage site. It corresponds to having all test window outputs being negative. In such cases, our algorithm outputs that these sequences have no cleavage site sequences have not a cleavage site.

The model selection method involves choosing among a number of support vector machines with different parameter settings for the kernel and regularisation. The question of consistency of support vector machines has been studied by a number of authors. For example, Steinwart [31] shows a dependence on the choice of kernel and regularisation parameters, so that a priori consistency is not guaranteed for a fixed value of the regularisation parameter. It is an interesting question whether our method can choose from an appropriate sequence of regularisation parameters to ensure consistency without the need for handcrafted choices. This question is, however, beyond the scope of the current paper.

4. Results and discussion

Our data set consists of 72 sequences from fungal proteins selected among non-homologous proteins with known pro-peptide regions, determined by N-terminal amino acid sequencing. This has limited the number of available protein sequences but, is expected to have enhanced accuracy. We initialize the parameter C_+ from 0.5 and increased it by the factor of 2 for 6 iterations for both the confidence level method and cross validation. For each value of C_+ , C_- was initialized to $C_+/4$ and increased by a factor of 2 for 4 iterations. Likewise, we initialized σ to 2^{-8} and multiply a factor of 2 for 6 iterations. Accuracy results are given in Table 1. We compared our results with the ProP1.0 server [13] and the full 71 cross validation. As it can be seen in Table 1, the best accuracy is achieved with the model selection method proposed in this study by confidence level with SVM and our second approach with normalized binary inputs. We tested our data set on ProP1.0 server and it gave 61% accuracy on the 72 test sequences. Our novel approach improved on the accuracy of ProP1.0 server [13] by 15%, although our training data set is 3 times smaller than that used in the neural network approach described in [13] which used 227. Furthermore, parameter selection with the confidence level gives higher accuracy than cross validation.

When we compare our confidence level based approach with cross validation, we see from Table 2 that the computational complexity of training times of the confidence level method is significantly shorter than cross validation. Here, we show the average of the elapsed time in training each leave one out phase, i.e., the results in Table 2 give the approximate time in seconds per test sequence in training. As can be seen from Tables 1 and 2, the best method both in training and accuracy is the confidence level with binary inputs.

5. Conclusion and perspectives

Our paper has considered the problem of identifying the cleavage site for fungal pro-peptides which can be extended in general to eukaryotic proteins. This task has previously been tackled by a neural network. We presented a kernel based solution with two novel features: A *kernel* specifically defined for the task enabling the learning to take place using linear functions in a very high dimensional feature space; and the implementation of *model selection* at the test point evaluation phase, rather than by cross validation. Both of these innovations lead to a significant improvement in classification accuracy on a real world data set as well as giving results that are an improvement over the earlier approaches. It would be interesting to apply the kernel introduced here to other sequence analysis tasks. The approach to model selection is interesting in that it gives improved performance with very significantly reduced training times. We believe that this approach should be evaluated more widely on standard evaluation tasks. We also believe that using results similar to those of [26], the approach can be placed on a sound theoretical footing.

References

- [1] S.F. Altschul, Amino acid substitution matrices from an information theoretic perspective, *Journal of Molecular Biology* 219 (1991) 555–665.
- [2] V. Atalay, R. Cetin-Atalay, Implicit motif distribution based hybrid computational kernel for sequence classification, *Bioinformatics* 21 (8) (2005) 1429–1436.
- [3] D. Baker, A.K. Shiau, D.A. Agard, The role of pro regions in protein folding, *Current Opinion in Cell Biology* 5 (6) (1993) 966–970 (Review).
- [4] J.D. Bendtsen, H. Nielsen, G. Heijne, S. Brunak, Improved prediction of signal peptides: SignalP 3.0, *Journal of Molecular Biology* 340 (2004) 783–795.
- [5] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-Based Learning Methods*, Cambridge University Press, 2000.
- [6] Y.-D. Cai, X.-J. Liub, X.-B. Xu, K.-C. Chou, Prediction of protein structural classes by support vector machines, *Computers and Chemistry* 26 (2002) 293–296.
- [7] Y.-D. Cai, S.-L. Lin, K.-C. Chou, Support vector machines for prediction of protein signal sequences and their cleavage sites, *Peptides* 24 (2003) 159–161.
- [8] 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>.
- [9] K.-C. Chou, Prediction of signal peptides using scaled window, *Peptides* 22 (2001) 1973–1979.
- [10] M.G. Claros, S. Brunak, G. von Heijne, Prediction of N-terminal protein sorting signals, *Current Opinion in Structural Biology* 7 (1997) 394–398.
- [11] M.O. Dayhoff, R.M. Schwartz, B.C. Orcutt, A model of evolutionary change in proteins, in: Dayhoff M.O. (Ed.), *Atlas of Protein Sequence and Structure*, vol. 5 (3), National Biomedical Research Foundation, Washington, 1978, pp. 345–352.
- [12] A. Dubey, M.J. Realff, J.H. Lee, A.S. Bommarius, Support vector machines for learning to identify the critical positions of a protein, *Journal of Theoretical Biology* 234 (2005) 351–361.
- [13] P. Duckert, S. Brunak, N. Blom, Prediction of proprotein convertase cleavage sites, *Protein Engineering, Design and Selection* 17 (1) (2004) 107–112.
- [14] G. Evin, A. Zhu, R.M.D. Holsinger, C.L. Masters, Q.-X. Li, Proteolytic processing of the Alzheimers's disease amyloid precursor protein in brain and platelets, *Journal of Neuroscience Research* 74 (2003) 386–392.
- [15] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning - Data Mining, Inference and Prediction*, in: Springer Series in Statistics, 2001.
- [16] R. Jalving, *Proteolytic processing in the secretory pathways of Aspergillus niger*, Ph.D. Thesis, Wageningen University, 2005.
- [17] D.J. Jeenes, D.A. Mackenzie, I.N. Roberts, D.B. Archer, Heterologous protein production by filamentous fungi, *Biotechnology and Genetic Engineering Reviews* 9 (1991) 327–367.
- [18] S. Kawashima, H. Ogata, M. Kanehisa, AAindex: Amino acid index database, *Nucleic Acids Research* 27 (1999) 368–369.
- [19] K.M.H. Nevalainen, V.S.J. Te'o, P.L. Bergquist, Heterologous protein expression in filamentous fungi, *Trends in Biotechnology* 23 (9) (2005) 468–474.
- [20] H. Nicholas, A. Ropelewski, Sequence analysis: Which scoring method should I use? http://www.psc.edu/research/biomed/homologous/scoring_primer.html.
- [21] H. Nielsen, J. Engelbrecht, S. Brunak, G. von Heijne, Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites, *Protein Engineering* 10 (1997) 1–6.
- [22] Z.B. Ögel, *Molecular analysis of a fungal galactose oxidase gene*, Ph.D. Thesis, Univ. of Leeds, Leeds, UK, 1993.
- [23] P.J. Punt, N. van Biezen, A. Conesa, A. Albers, J. Mangnus, C. van den Hondel, Filamentous fungi as cell factories for heterologous protein production, *Trends in Biotechnology* 20 (5) (2002) 200–2006.
- [24] P.J. Punt, A. Drint-Kuijvenhoven, B.C. Lokman, J.A. Spencer, D. Jeenes, D.A. Archer, C.A. van den Hondel, The role of the *Aspergillus niger* furin-type protease gene in processing of fungal proproteins and fusion proteins. Evidence for alternative processing of recombinant (fusion-) proteins, *Journal of Biotechnology* 5 106 (1) (2003) 23–32.
- [25] U.P. Shinde, J.J. Liu, M. Inouye, Protein memory through altered folding mediated by intramolecular chaperones, *Nature* 389 (6650) (1997) 520–522.
- [26] J. Shawe-Taylor, Classification Accuracy based on observed margin, *Algorithmica* 22 (1998) 157–172.
- [27] J. Shawe-Taylor, N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004, pp. 344–396.
- [28] J. Shawe-Taylor, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, 2001 (Review of Anthony, Martin; Bartlett, Peter L.).
- [29] R.G. Spiro, Protein glycosylation: Nature, distribution, enzymatic formation, and disease implications of glycopeptide bonds, *Glycobiology* 12 (4) (2002) 43R–56R (Review).
- [30] J.Y. Springael, E. Nikko, B. André, A.M. Marini, Yeast Npi3/Bro1 is involved in ubiquitin-dependent control of permease trafficking, *FEBS Letters* 517 (2002) 103–109.
- [31] I. Steinwart, On the influence of the kernel on the consistency of support vector machines, *Journal of Machine Learning Research* 2 (2001) 67–93.
- [32] G. Von Heijne, A new method for predicting signal sequence cleavage sites, *Nucleic Acids Research* 14 (1) (1986) 4683–4690.
- [33] A. Weidemann, K. Paliga, U. Dürrwang, F.B.M. Reinhard, O. Schuckert, G. Evin, C.L. Masters, Proteolytic processing of the alzheimers disease amyloid precursor protein within its cytoplasmic domain by caspase-like proteases, *The Journal of Biological Chemistry* 274 9 (26) (1999) 5823–5829.