Truncation of Protein Sequences for Fast Profile Alignment with Application to Subcellular Localization

Man-Wai Mak and Wei Wang
Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hung Hom, Hong Kong SAR
Email: enmwmak@polyu.edu.hk

Sun-Yuan Kung
Dept. of Electrical Engineering
Princeton University
New Jersey, USA
kung@princeton.edu

***Abstract*—We have recently found that the computation time of homology-based subcellular localization can be substantially reduced by aligning profiles up to the cleavage site positions of signal peptides, mitochondrial targeting peptides, and chloroplast transit peptides [1]. While the method can reduce the profile alignment time by as much as 20 folds, it cannot reduce the computation time spent on creating the profiles. In this paper, we propose a new approach that can reduce both the profile creation time and profile alignment time. In the new approach, instead of cutting the profiles, we shorten the sequences by cutting them at the cleavage site locations. The shortened sequences are then presented to PSI-BLAST to compute the profiles. Experimental results and analysis of profile-alignment score matrices suggest that both profile creation time and profile alignment time can be reduced without sacrificing subcellular localization accuracy. Once a pairwise profile-alignment score matrix has been obtained, a one-vs-rest SVM classifier can be trained. To further reduce the training and recognition time of the classifier, we propose a perturbation discriminant analysis (PDA) technique. It was found that PDA enjoys a short training time as compared to the conventional SVM.**

***Keywords*-Subcellular localization; cleavage sites prediction; profiles alignment; protein sequences; kernel discriminant analysis; SVM.**

I. Introduction

A. Motivation of Subcellular Localization Prediction

Proteins must be transported to the correct organelles of a cell and folded into correct 3-D structures to properly perform their functions. Therefore, knowing the subcellular localization is one step towards understanding its functions. The determination of this information by experimental means is often time-consuming and laborious. Given the large number of un-annotated sequences from genome projects, it is imperative to develop efficient and reliable computation techniques for annotating biological sequences.

In recent years, impressive progress has been made in the computational prediction of subcellular localization. A number of approaches have also been proposed in the literature. These methods can be generally divided into four categories, including predictions based on sorting signals [2], [3], [4], [5], [6], [7], global sequence properties [8], [9], [10], [11], homology [12], [13], [14] and other information in addition to sequences [15], [16]. Methods based on sorting signals are very fast, but they typically suffer from low prediction accuracy. Homology-based methods are more accurate, but they are very slow. Therefore, *fast* and *reliable* predictions of subcellular localization still remain a challenge.

B. Approaches to Subcellular Localization Prediction

Signal-based methods predict the localization via the recognition of N-terminal sorting signals in amino acid sequences. PSORT, proposed by Nakai in 1991 [3], is one of the early predictors that use sorting signals for protein's subcellular localization. PSORT and its extensions – WoLF PSORT [4], [5] – derive features such as amino acid compositions and the presence of sequence motifs for localization prediction. In the late 90's, researchers started to investigate the application of neural networks [17] to recognize the sorting signals. In a neural network, patterns are presented to the input layer of artificial neurons, with each neuron implementing a nonlinear function of the weighted sum of the inputs. Because amino acid sequences are of variable length, the input to the neural network is extracted from a short window sliding over the amino acid sequence. TargetP [18], [19] is a well-known predictor that uses neural networks.

Another type of approaches relies on the fact that proteins of different organelles have different global properties such as amino-acid composition. Based on amino-acid composition and residue-pair frequencies, Nakashima and Nishikawa [11] developed a predictor that can discriminate between soluble intracellular and extracellular proteins. Another popular predictor based on amino acid composition is SubLoc [8]. In SubLoc, a query sequence is converted to 20-dim amino-acid composition vector for classification by SVMs. Recently, Xu et al. [20] proposed a semi-supervised learning technique (a kind of transductive learning) that makes use of unlabelled test data to boost the classification performance of SVMs. One limitation of composition-based methods is that information about the sequence order is not easy to represent. Some authors proposed using amino-acid pair compositions (dipeptide) [21], [10], [9] and pseudo amino-acid compositions [22] to enrich the representation power of the extracted vectors.

The homology-based methods use the query sequence to

This work was in part supported by The Hong Kong Research Grant Council, Grant No. PolyU5251/08E and PolyU5264/09E.

search protein databases for homologs [12], [13] and predict the subcellular location of the query sequence as the one to which the homologs belong. This kind of method can achieve very high accuracy when homologs of experimentally verified sequences can be found in the database search [23]. A number of homology-based predictors have been proposed. For example, Proteome Analyst [24] uses the presence or absence of the tokens from certain fields of the homologous sequences in the Swiss-Prot database as a means to compute features for classification. In Kim et al. [25], an unknown protein sequence is aligned with every training sequences (with known subcellular locations) to create a feature vector for classification. Mak et al. [14] proposed a predictor called PairProSVM that uses profile alignment to detect weak similarity between protein sequences. Given a query sequence, a profile is obtained from PSI-BLAST search [26]. The profile is then aligned with every training profile to form a score vector for classification by SVMs.

Some predictors not only use amino acid sequences as input but also require extra information such as lexical context in database entries [15] or Gene Ontology entries [16] as input. Although studies have shown that this type of method can outperform sequence-based methods, the performance has only been measured on data sets where all sequences have the required additional information. Thus, the applicability is limited.

C. Limitations of Existing Approaches

Among all the methods mentioned above, the signal-based and homology-based methods have attracted a great deal of attention, primarily because of their biological plausibility and robustness in predicting newly discovered sequences. Comparing these two approaches, the signal-based methods seem to be more direct, because they determine the localization from the sequence segments that contain the localization information. However, this type of method is typically limited to the prediction of a few subcellular locations only. For example, the popular TargetP [6], [7] can only detect three localizations: chloroplast, mitochondria, and secretory pathway signal peptide. The homology-based methods, on the other hands, can in theory predict as many localizations as available in the training data. The downside, however, is that the whole sequence is used for the homology search or pairwise alignment, without considering the fact that some segments of the sequence are more important or contain more information than the others. Moreover, the computation requirement will be excessive for long sequences. The problem will become intractable for database annotation where tens of thousands of proteins are involved.

D. Our Proposal for Addressing the Limitations

Our earlier report [1] has demonstrated that computation time of subcellular localization based on profile alignment SVMs can be substantially reduced by aligning profiles up to the cleavage site positions of signal peptides, mitochondrial targeting peptides, and chloroplast transit peptides. Although 20-fold reduction in total computation time (including alignment, training and recognition time) has been achieved, the method fails to reduce the profile creation time, which will become a substantial part of the total computation time when the database becomes large. In this paper, we propose a new approach that can reduce both the profile creation time and profile alignment time. In the new approach, instead of cutting the profiles, we shorten the sequences by cutting them at the cleavage site locations. The shortened sequences are then presented to PSI-BLAST to compute the profiles. To further reduce the training and recognition time of the classifier, we propose replacing the SVMs by kernel perturbation discriminants.

II. Kernel Discriminant Analysis

This section derives the formulation of kernel discriminant analysis and explains how it can be applied to multi-class problems such as subcellular localization. The key idea lies on the equivalency between the optimal projection vectors in the Hilbert space, spectral space and empirical space. A more in-depth treatment can be found in [28].

A. Input, Hilbert, Spectral, and Empirical Spaces

Denote the mapping from an input space $\mathcal{X}$ into a Hilbert space $\mathcal{H}$ as:

$$\overrightarrow{\boldsymbol{\phi}} : \mathcal{X} \to \mathcal{H} \quad \text{such that} \quad \boldsymbol{x} \mapsto \overrightarrow{\boldsymbol{\phi}}(\boldsymbol{x}).$$

In bioinformatics, $\mathcal{X}$ is a vectorial space for microarray data and a sequence space for DNA or protein sequences. Given a training dataset $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ in $\mathcal{X}$ and a kernel function $K(\boldsymbol{x}, \boldsymbol{y})$, an object can be represented by a vector of similarity with respect to all of the training objects [29]:

$$\overrightarrow{\boldsymbol{k}}(\boldsymbol{x}) \equiv [K(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, K(\boldsymbol{x}_N, \boldsymbol{x})]^T.$$

This N-dim space, denoted by $\mathcal{K}$, will be named empirical space. The associate kernel matrix is defined as

$$\boldsymbol{K} = \left[\overrightarrow{\boldsymbol{k}}(\boldsymbol{x}_1), \ldots, \overrightarrow{\boldsymbol{k}}(\boldsymbol{x}_N)\right].$$

The construction of the empirical space for vectorial and non-vectorial data are quite different. For the former, the elements of $\boldsymbol{K}$ are a simple function of the corresponding pair of vectors in $\mathcal{X}$. For the latter, the elements in $\boldsymbol{K}$ are similarities between the corresponding pairs of objects.

The kernel matrix $\boldsymbol{K}$ can be factorized with respect to the basis functions in $\mathcal{H}$: $\boldsymbol{K} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}$, where $\boldsymbol{\Phi} = [\overrightarrow{\boldsymbol{\phi}}(\boldsymbol{x}_1), \ldots, \overrightarrow{\boldsymbol{\phi}}(\boldsymbol{x}_N)]$. Alternatively, it can be factorized via spectral decomposition: $\boldsymbol{K} = \boldsymbol{U}^T\boldsymbol{\Lambda}\boldsymbol{U} = \boldsymbol{U}^T\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U} = (\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U})^T(\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U}) = \boldsymbol{E}^T\boldsymbol{E}$, where $\boldsymbol{E} = \boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U}$.

Denote the i-th row of $\boldsymbol{E}$ as $\boldsymbol{e}^{(i)} = \left[e^{(i)}(\boldsymbol{x}_1), \ldots, e^{(i)}(\boldsymbol{x}_N)\right]$. Because $\boldsymbol{E}\boldsymbol{E}^T = \boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{\Lambda}^{\frac{1}{2}} = \boldsymbol{\Lambda}$, the rows of $\boldsymbol{E}$ exhibit a vital orthogonality property:

$$\boldsymbol{e}^{(i)}\boldsymbol{e}^{(j)T} = \begin{cases} 0 & \text{if } i \neq j \\ \lambda_i & \text{if } i = j, \end{cases}$$

where λ_i is the i-th element of the diagonal of $\boldsymbol{\Lambda}$.

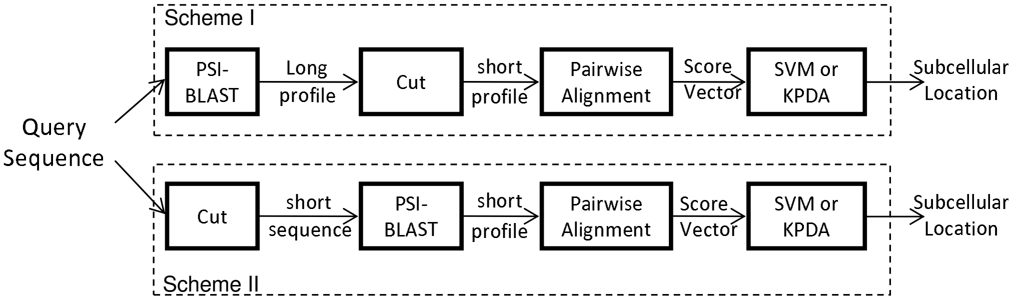


Figure 1: Two schemes for reducing the computation of the subcellular localization process. In Scheme I, a full-length query sequence is presented to PSI-BLAST for computing a full-length profile; then the profile is truncated at the predicted cleavage site. The truncated profile is then aligned with all of the truncated training profiles to produce a profile-alignment score vector for classification. In Scheme II, the query sequence is truncated at the predicted cleavage site before inputting to PSI-BLAST for computing the profile. The cleavage sites are predicted by CSitePred [27] or TargetP [6].

For any positive-definite kernel function $K(\boldsymbol{x},\boldsymbol{y})$ and training dataset $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$ in $\mathcal{X}$, there exists a (nonlinear) mapping from the original input space $\mathcal{X}$ to an N-dim spectral space $\mathcal{E}$:[1]

$$\overrightarrow{\boldsymbol{e}}:\mathcal{X}\rightarrow\mathcal{E}\quad\text{such that}\quad \boldsymbol{x}\mapsto\overrightarrow{\boldsymbol{e}}(\boldsymbol{x})\equiv\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{U}\overrightarrow{\boldsymbol{k}}(\boldsymbol{x}).$$

Many kernel-based machine learning problems involve finding optimal projection vectors in $\mathcal{H}$, $\mathcal{E}$, and $\mathcal{K}$, which will be respectively denoted as $\boldsymbol{w}$, $\boldsymbol{v}$, and $\boldsymbol{a}$. It can be shown [28] that the projection vectors are linearly related as follows:

$$\boldsymbol{w}^T\overrightarrow{\boldsymbol{\phi}}(\boldsymbol{x})=\boldsymbol{v}^T\overrightarrow{\boldsymbol{e}}(\boldsymbol{x})=\boldsymbol{a}^T\overrightarrow{\boldsymbol{k}}(\boldsymbol{x}),\tag{1}$$

where we have used the relationships $\boldsymbol{w}=\boldsymbol{\Phi}\boldsymbol{a}$ and $\boldsymbol{v}=\boldsymbol{E}\boldsymbol{a}$.

B. Orthogonal Hyperplane Principle (OHP)

Assume that the dimension of $\mathcal{H}$ is M and that the training data in $\mathcal{H}$ are mass-centered. When $M\geq N$, all of the N training vectors $\{\overrightarrow{\boldsymbol{\phi}}(\boldsymbol{x}_i);i=1,\ldots,N\}$ will fall on an $(M-1)$-dim *data hyperplane*. Mathematically, the data-hyperplane is represented by its normal vector $\boldsymbol{p}$ such that $\boldsymbol{\Phi}^T\boldsymbol{p}=\boldsymbol{1}$. The optimal decision-hyperplane in $\mathcal{H}$ (represented by $\boldsymbol{w}$) must be orthogonal to the data-hyperplane:

$$\boldsymbol{w}^T\boldsymbol{p}=0\quad\Rightarrow\quad\boldsymbol{a}^T\boldsymbol{\Phi}^T\boldsymbol{p}=0\quad\Rightarrow\quad\boldsymbol{a}^T\boldsymbol{1}=0.$$

C. Kernel Fisher Discriminant Analysis (KFDA)

The objective of KFDA [30] is to determine an optimal discriminant function (linearly) expressed in the Hilbert space $\mathcal{H}$:

$$f(\boldsymbol{x})=\boldsymbol{w}^T\overrightarrow{\boldsymbol{\phi}}(\boldsymbol{x})+b,$$

where b is a bias to account for the fact that training data may not be mass-centered. The discriminant function may be equivalently expressed in the N-dim spectral space $\mathcal{E}$:

$$f(\boldsymbol{x})=\boldsymbol{v}^T\overrightarrow{\boldsymbol{e}}(\boldsymbol{x})+b.$$

The finite-dimensional space $\mathcal{E}$ facilitates our analysis and design of optimal classifiers. In fact, the optimal projection vector $\boldsymbol{v}_{\text{opt}}$ in $\mathcal{E}$ can be obtained by applying conventional FDA to the column vectors $\{\overrightarrow{\boldsymbol{e}}(\boldsymbol{x}_i)\}$. To derive the objective function of KFDA, let us define

$$\boldsymbol{d}=\frac{2}{d_++d_-}\left(d_+\boldsymbol{1}_+-d_-\boldsymbol{1}_-\right),\tag{2}$$

where $d_+=\sqrt{\frac{N_-}{NN_+}}$ and $d_-=\sqrt{\frac{N_+}{NN_-}}$; $\boldsymbol{1}_+$ and $\boldsymbol{1}_-$ contain 1's in entries corresponding to Classes $\mathcal{C}_+$ and $\mathcal{C}_-$, respectively, and 0's otherwise; and N_+ and N_- are the number of training samples in classes $\mathcal{C}_+$ and $\mathcal{C}_-$, respectively. It can be shown that the objective function of KFDA is:

$$J_{\text{KFDA}}(\boldsymbol{v})=\frac{\boldsymbol{v}^T\boldsymbol{S}_b^{\mathcal{E}}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{S}_w^{\mathcal{E}}\boldsymbol{v}}=\frac{\boldsymbol{v}^T\boldsymbol{E}\boldsymbol{d}\boldsymbol{d}^T\boldsymbol{E}^T\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{E}\left(\boldsymbol{I}-\frac{\boldsymbol{1}\boldsymbol{1}^T}{N}\right)\boldsymbol{E}^T\boldsymbol{v}},\tag{3}$$

where $\boldsymbol{1}$ is an N-dim vector with all elements equal to 1 and $\boldsymbol{S}_b^{\mathcal{E}}=\boldsymbol{E}\boldsymbol{d}\boldsymbol{d}^T\boldsymbol{E}^T$ and $\boldsymbol{S}_w^{\mathcal{E}}=\boldsymbol{E}(\boldsymbol{I}-\frac{\boldsymbol{1}\boldsymbol{1}^T}{N})\boldsymbol{E}^T$ are between-class and within-class covariance matrices in $\mathcal{E}$ space, respectively.

D. Perturbational Discriminant Analysis (PDA)

The FDA and KFDA are based on the assumption that the observed data are perfectly measured. It is however crucial to take into account the inevitable perturbation of training data. For the purpose of designing practical classifiers, we can adopt the following perturbational discriminant analysis (PDA).

It is assumed that the observed data is contaminated by additive white noise in the spectral space. Denote the center-adjusted matrix of $\boldsymbol{E}$ as $\bar{\boldsymbol{E}}$ and the uncorrelated noise as $\boldsymbol{N}$, then the perturbed scattered matrix is

$$(\bar{\boldsymbol{E}}+\boldsymbol{N})(\bar{\boldsymbol{E}}+\boldsymbol{N})^T\approx\bar{\boldsymbol{E}}\bar{\boldsymbol{E}}^T+\rho\boldsymbol{I}=\boldsymbol{E}\left(\boldsymbol{I}-\frac{\boldsymbol{1}\boldsymbol{1}^T}{N}\right)\boldsymbol{E}^T+\rho\boldsymbol{I},$$

where ρ is a parameter representing the noise level. Its value can sometimes be empirically estimated if the domain knowledge is well established a priori. Under the perturbation analysis, the kernel Fisher score in Eq. 3 is modified to the

[1] $\boldsymbol{K}=\boldsymbol{E}^{\mathsf{T}}\boldsymbol{E}$, i.e., $\boldsymbol{E}=(\boldsymbol{E}^{\mathsf{T}})^{-1}\boldsymbol{K}=(\boldsymbol{U}^{\mathsf{T}}\boldsymbol{\Lambda}^{\frac{1}{2}})^{-1}\boldsymbol{K}=\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{U}\boldsymbol{K}$. Therefore, $\overrightarrow{\boldsymbol{e}}(\boldsymbol{x}_i)=\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{U}\overrightarrow{\boldsymbol{k}}(\boldsymbol{x}_i)$, $i=1,\ldots,N$.

following perturbed variant:

$$J_{\text{PDA}}(\boldsymbol{v}) = \frac{\boldsymbol{v}^T \boldsymbol{E}\boldsymbol{d}\boldsymbol{d}^T \boldsymbol{E}^T \boldsymbol{v}}{\boldsymbol{v}^T \left[\boldsymbol{E}\left(\boldsymbol{I} - \frac{\boldsymbol{1}\boldsymbol{1}^T}{N}\right)\boldsymbol{E}^T + \rho \boldsymbol{I}\right]\boldsymbol{v}}. \qquad (4)$$

By taking the derivative of $J_{\text{PDA}}(\boldsymbol{v})$ with respect to $\boldsymbol{v}$, the optimal solution to Eq. 4 can be obtained as:

$$\boldsymbol{v}_{\text{opt}} = \left[\boldsymbol{E}\left(\boldsymbol{I} - \frac{\boldsymbol{1}\boldsymbol{1}^T}{N}\right)\boldsymbol{E}^T + \rho \boldsymbol{I}\right]^{-1} \boldsymbol{E}\boldsymbol{d},$$

and using the Sherman-Morrison-Woodbury identity it can be shown that

$$\boldsymbol{v}_{\text{opt}} = \left(\boldsymbol{E}\boldsymbol{E}^T + \rho \boldsymbol{I}\right)^{-1} \boldsymbol{E}(\boldsymbol{d} - \eta \boldsymbol{1}) = (\boldsymbol{\Lambda} + \rho \boldsymbol{I})^{-1} \boldsymbol{E}(\boldsymbol{d} - \eta \boldsymbol{1}) \qquad (5)$$

where η is a scalar whose value can be determined through the optimal solution in $\mathcal{K}$ space as follows.

Recall from Eq. 1 that dot-products in the three spaces are equivalent. Therefore, the discriminant function in $\mathcal{K}$ space can be written as:

$$f(\boldsymbol{x}) = \boldsymbol{a}^T \overrightarrow{\boldsymbol{k}}(\boldsymbol{x}) + b. \qquad (6)$$

Given the optimal solution $\boldsymbol{v}_{\text{opt}}$ in the $\mathcal{E}$ space, the corresponding optimal solution in the $\mathcal{K}$ space is[2]

$$\begin{aligned} \boldsymbol{a}_{\text{opt}} &= \boldsymbol{E}^{-1}\boldsymbol{v}_{\text{opt}} \\ &= \boldsymbol{U}^T \boldsymbol{\Lambda}^{-\frac{1}{2}} (\boldsymbol{\Lambda} + \rho \boldsymbol{I})^{-1} \boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{U}(\boldsymbol{d} - \eta \boldsymbol{1}) \qquad (7) \\ &= (\boldsymbol{K} + \rho \boldsymbol{I})^{-1} (\boldsymbol{d} - \eta \boldsymbol{1}), \end{aligned}$$

where we have used $\boldsymbol{K} = \boldsymbol{U}^T \boldsymbol{\Lambda} \boldsymbol{U}$ and $\boldsymbol{E} = \boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{U}$. Note that unlike Eq. 5, Eq. 7 does not require spectral decomposition, thus offering a fast close-form solution. Now using the orthogonal hyperplanes principle (Section II-B), we have

$$\begin{aligned} \boldsymbol{a}_{\text{opt}}^T \boldsymbol{1} &= (\boldsymbol{d}^T - \eta \boldsymbol{1}^T)(\boldsymbol{K} + \rho \boldsymbol{I})^{-1} \boldsymbol{1} = 0 \\ \Rightarrow \qquad \eta &= \frac{\boldsymbol{d}^T (\boldsymbol{K} + \rho \boldsymbol{I})^{-1} \boldsymbol{1}}{\boldsymbol{1}^T (\boldsymbol{K} + \rho \boldsymbol{I})^{-1} \boldsymbol{1}}. \end{aligned} \qquad (8)$$

The value of b can be obtained by using the relationship [28]:[3] $(\boldsymbol{y} - b\boldsymbol{1}) = (\boldsymbol{d} - \eta \boldsymbol{1})$, which gives

$$b = y_i - (d_i - \eta) \quad \text{for any } i = 1, \ldots, N. \qquad (9)$$

E. Application of PDA to Multi-Class Problems

A C-class problem can be formulated as C binary classification problems in which each problem is solved by a one-versus-rest binary classifier. Here, we propose two approaches to applying PDA to solve multi-class problems.

1) One-vs-Rest PDA Classifier: Given the training samples of C classes, we train C PDA score functions as follows:

$$f_i(\boldsymbol{x}) = \boldsymbol{a}_i^T \overrightarrow{\boldsymbol{k}}(\boldsymbol{x}) + b_i, \qquad i = 1, \ldots, C,$$

where $\boldsymbol{a}_i$ and b_i are obtained by using Eq. 7 and Eq. 9, respectively. Then, given a test sample $\boldsymbol{x}$, the class label is obtained by

$$l = \arg\max_i f_i(\boldsymbol{x}).$$

Class Index	Subcellular Location	Number of Proteins
1	Extracellular	693
2	Mitochondria	167
3	Chloroplast	74
4	Others(Cytoplasm/Nucleus)	1617
		2552(total)

Table I: Breakdown of eukaryotic dataset derived from the Swiss-Prot database (release 57.5).

2) Cascaded Fusion of PDA and SVM: Because of the dependence in $d_i, i = 1, \ldots, C$, the rank of matrix $[\boldsymbol{d}_1, \ldots, \boldsymbol{d}_C]$ is $C - 1$. Therefore, there are $C - 1$ independent sets of PDA parameters:

$$\begin{aligned} \hat{\boldsymbol{A}} &= [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{C-1}] \\ &= (\boldsymbol{K} + \rho \boldsymbol{I})^{-1}([\boldsymbol{d}_1, \ldots, \boldsymbol{d}_{C-1}] - \boldsymbol{1}[\eta_1, \ldots, \eta_{C-1}]). \end{aligned}$$

During recognition, an unknown sample $\boldsymbol{x}$ is projected onto a $(C-1)$-dim PDA space spanned by $[\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{C-1}]$ using

$$\boldsymbol{g}(\boldsymbol{x}) = \hat{\boldsymbol{A}}^T \boldsymbol{k}(\boldsymbol{x}) + [b_1, \ldots, b_{C-1}]^T.$$

Then, $\boldsymbol{g}(\boldsymbol{x})$ is classified by one-vs-rest SVMs. In the sequel, we refer to this cascaded fusion as PDAproj+SVM.

III. Experiments

A. Data Set Construction

Protein sequences with experimentally annotated subcellular locations were collected from the Swiss-Prot Release 57.5 according to the following criteria.

1) Only the entries of Eukaryotic species are included, which are annotated with "Eukaryota" in the OC (Organism Classification) fields in Swiss-Prot.
2) A large amount of sequences in Swiss-Prot are annotated with ambiguous words, such as "probable", "by similarity" and "potential". These entries were excluded because of the lack of experimental evidence.
3) Sequences annotated with "fragment" were excluded.
4) Sequences that have 25% or higher sequence identity to any other sequences are excluded.
5) For signal peptides, mitochondria, and chloroplast, only sequences with experimentally annotated cleavage sites are included.

Table I shows the breakdown of the dataset.

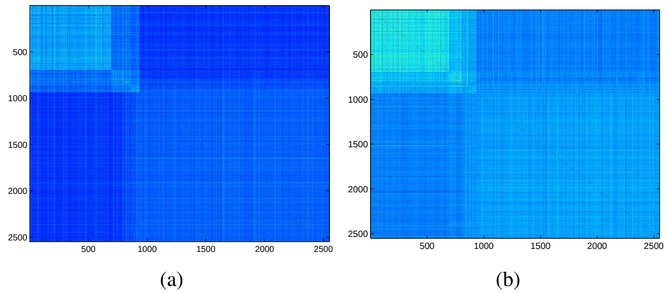
B. Assessment of the Prediction Results

We used 5-fold cross validation to evaluate the performance. The overall prediction accuracy, the accuracy for each subcellular location, and the Matthew's correlation coefficient (MCC) [31] were used to quantify the prediction performance. MCC allows us to overcome the shortcoming of accuracy on unbalanced data [31].

We used TargetP and a CRF-based predictor (CSitePred) [27] for cleavage site prediction and SVM [14] and PDA for classification. We measured the computation time on a Core(TM)2 Duo 3.16GHz CPU running Matlab and SVM-light. The computation time was divided into profile creation time, alignment time, classifier training time, and classification time.

[2] Eq. 1 suggests that $\boldsymbol{a}^T \boldsymbol{K} = \boldsymbol{v}^T \boldsymbol{E}$. Therefore, we have $\boldsymbol{a}^T = \boldsymbol{v}^T \boldsymbol{E} \boldsymbol{K}^{-1} = \boldsymbol{v}^T \boldsymbol{E}(\boldsymbol{E}^T \boldsymbol{E})^{-1} = \boldsymbol{v}^T \boldsymbol{E}^{-T}$, which suggests that $\boldsymbol{a} = \boldsymbol{E}^{-1}\boldsymbol{v}$.

[3] Note that our definition of $\boldsymbol{d}$ in Eq. 2 and that of [28] differ by a proportional constant.



(a) (b)

Figure 2: Profile-alignment score matrices produced by (a) Scheme I and (b) Scheme II in Fig. 1.

Scheme	Input to PSI-BLAST	Profile Creation Time (sec.)	Subcellular Localization Accuracy
I	Full-length sequences	30.5	91.69%
II	Sequences truncated at predicted cleavage sites	4.7	91.45%

Table II: Average computation time to create a profile by PSI-BLAST using sequences of different length as input. In Scheme I, full-length sequences were presented to PSI-BLAST and the resulting profiles were truncated at the predicted cleavage sites. In Scheme II, truncation was applied to the sequences before presenting to PSI-BLAST. In both cases, CRFs (CSitePred) were used to predict the cleavage sites.

IV. Results and Discussions

A. Comparing Profile Creation Schemes

Fig. 2 shows the score matrices obtained by the two profile creation schemes (see Fig. 1). The figure shows that the two alignment score matrices exhibit a similar pattern, suggesting that classifiers based on these matrices will produce similar classification accuracy. This argument is confirmed by Table II, which shows that cutting the sequences at cleavage sites before inputting to PSI-BLAST can reduce the profile creation time by 6 times without significant reduction in subcellular localization accuracy.

B. SVM versus PDA

Table III shows that the training time of PDA and PDAproj+SVM are only one-fifth of that of SVM. However, the accuracy of PDA and PDAproj+SVM are lower than that of SVM.

C. Compared with State-of-the-Art Predictors

We compared the accuracy of our cascaded fusion method with SubLoc[8] and TargetP by presenting the sequences to their webservers. The results suggest that the overall accuracy of our method is 5.2% higher than that of TargetP and is significantly better than that of SubLoc. Our method outperforms TargetP in Ext and Cyt/Nuc prediction while performing worse than TargetP in predicting Mit and Chl.

Classification Method	Training Time (sec.)	Classification Time (sec.)	SubLoc Acc.
SVM	51.4	0.7	91.45%
PDA	9.9	1.9	90.24%
PDAproj+SVM	8.9	0.1	89.97%

Table III: The computation time and performance of different classifiers in the subcellular localization task. The classification time is the time to classify a profile-alignment score vector with dimension equal to the number of training vectors. The training time is time required to train a classifier, given a profile-alignment score matrix. In PDAproj+SVM, PDA was applied to project the samples in the input space to a $(C-1)$-dim space ($C = 4$ here); the projected vectors were then classified by RBF-SVMs.

V. Conclusions

This paper has demonstrated that homology-based subcellular localization can be speeded up by reducing the length of the query amino acid sequences. Because shortening an amino acid sequence will inevitably throw away some information in the sequence, it is imperative to determine the best truncation positions. This paper shows that these positions can be determined by cleavage site predictors such as TargetP and CSitePred. The paper also shows that as far as localization accuracy is concerned, it does not matter whether we truncate the sequences or truncate the profiles. However, truncating the sequence has computation advantage because this strategy can save the profile creation time by as much as 6 folds.

References

[1] W. Wang, M. W. Mak, and S. Y. Kung, “Speeding up subcellular localization by extracting informative regions of protein sequences for profile alignment,” in *Proc. Computational Intelligence in Bioinformatics and Computational Biology*, Montreal, May 2010, pp. 147–154.

[2] G. von Heijne, “A new method for predicting signal sequence cleavage sites,” *Nucleic Acids Research*, vol. 14, no. 11, pp. 4683–4690, 1986.

[3] K. Nakai and M. Kanehisa, “Expert system for predicting protein localization sites in gram-negative bacteria,” *Proteins: Structure, Function, and Genetics*, vol. 11, no. 2, pp. 95–110, 1991.

[4] P. Horton, K. J. Park, T. Obayashi, and K. Nakai, “Protein subcellular localization prediction with WoLF PSORT,” in *Proc. 4th Annual Asia Pacific Bioinformatics Conference (APBC06)*, 2006, pp. 39–48.

[5] P. Horton, K. Park, T. Obayashi, N. Fujita, H. Harada, C. Adams-Collier, and K. Nakai, “WoLF PSORT: protein localization predictor,” *Nucleic acids research*, vol. 35, no. Web Server issue, pp. 585–587, 2007.

[6] O. Emanuelsson, H. Nielsen, S. Brunak, and G. von Heijne, “Predicting subcellular localization of proteins based on their N-terminal amino acid sequence,” *J. Mol. Biol.*, vol. 300, no. 4, pp. 1005–1016, 2000.

[7] O. Emanuelsson, S. Brunak, G. von Heijne, and H. Nielsen, “Locating proteins in the cell using TargetP, SignalP, and

Row	Cleavage Site Predictor	Localization Predictor	Classification Accuracy (%)					Matthew's correlation coefficient (MCC)				
			Ext	Mit	Chl	Cyt/Nuc	Overall	Ext	Mit	Chl	Cyt/Nuc	Overall
1	—	SubLoc [8]	51.44	55.83	—	77.86	66.79	—	—	—	—	—
2	—	TargetP (P)	79.08	88.02	89.19	69.57	73.93	0.79	0.49	0.79	0.64	0.65
3	—	TargetP (N)	97.40	89.22	0.00	87.82	87.97	0.93	0.58	0.00	0.81	0.84
4	TargetP(N)	SVM	97.26	67.07	36.49	95.86	92.63	0.93	0.70	0.53	0.86	0.90
5	TargetP(N)	PDA	97.55	61.68	6.76	95.61	91.34	0.91	0.68	0.26	0.84	0.88
6	TargetP(N)	PDAproj+SVM	97.26	65.27	37.84	93.57	91.10	0.93	0.64	0.50	0.83	0.88
7	CRF	SVM	94.52	63.47	28.38	95.86	91.45	0.90	0.68	0.45	0.84	0.89
8	CRF	PDA	94.81	59.28	1.35	95.55	90.24	0.88	0.67	0.11	0.82	0.81
9	CRF	PDAproj+SVM	94.66	63.47	25.68	93.63	89.97	0.90	0.60	0.41	0.82	0.87

Table IV: Subcellular localization performance achieved by different classifiers. The second column specifies the the cleavage site predictors that were used for determining the positions at which the amino sequences were truncated. Notice that TargetP can perform both cleavage site prediction and subcellular localization. For Rows 4 and 5, TargetP was used as a cleavage site predictor, where "TargetP(P)" and "TargetP(N)" mean selecting plant or non-plant option in TargetP, respectively. For Rows 6–8 "CRF" means that conditional random fields were used for cleavage site prediction.

related tools," *Nature Protocols*, vol. 2, no. 4, pp. 953–971, 2007.

[8] S. J. Hua and Z. R. Sun, "Support vector machine approach for protein subcellular localization prediction," *Bioinformatics*, vol. 17, pp. 721–728, 2001.

[9] Y. Huang and Y. Li, "Prediction of protein subcellular locations using fuzzy k-NN method," *Bioinformatics*, vol. 20, no. 1, pp. 21–28, 2004.

[10] K. J. Park and M. Kanehisa, "Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs," *Bioinformatics*, vol. 19, no. 13, pp. 1656–1663, 2003.

[11] H. Nakashima and K. Nishikawa, "Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies," *J. Mol. Biol.*, vol. 238, pp. 54–61, 1994.

[12] R. Mott, J. Schultz, P. Bork, and C. Ponting, "Predicting protein cellular localization using a domain projection method," *Genome research*, vol. 12, no. 8, pp. 1168–1174, 2002.

[13] M. Scott, D. Thomas, and M. Hallett, "Predicting subcellular localization via protein motif co-occurrence," *Genome research*, vol. 14, no. 10a, pp. 1957–1966, 2004.

[14] M. W. Mak, J. Guo, and S. Y. Kung, "PairProSVM: Protein subcellular localization based on local pairwise profile alignment and SVM," *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, vol. 5, no. 3, pp. 416 – 422, 2008.

[15] R. Nair and B. Rost, "Inferring sub-cellular localization through automated lexical analysis," *Bioinformatics*, vol. 18, pp. S78–S76, 2002.

[16] K. Chou and H. Shen, "Recent progress in protein subcellular location prediction," *Analytical Biochemistry*, vol. 370, no. 1, pp. 1–16, 2007.

[17] P. Baldi and S. Brunak, *Bioinformatics : The Machine Learning Approach*, 2nd ed. MIT Press, 2001.

[18] H. Nielsen, J. Engelbrecht, S. Brunak, and G. von Heijne, "A neural network method for identification of prokaryotic and eukaryotic signal perptides and prediction of their cleavage sites," *Int. J. Neural Sys.*, vol. 8, pp. 581–599, 1997.

[19] H. Nielsen, J. Engelbrecht, S. Brunak, and G. Von Heijne, "Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites," *Protein Engineering Design and Selection*, vol. 10, no. 1, p. 1, 1997.

[20] Q. Xu, D. H. Hu, H. Xue, W. Yu, and Q. Yang, "Semi-supervised protein subcellular localization," *BMC Bioinformatics*, vol. 10, 2009.

[21] Z. Yuan, "Prediction of protein subcellular locations using Markov chain models," *FEBS Letters*, vol. 451, no. 1, pp. 23–26, 1999.

[22] K. C. Chou, "Prediction of protein cellular attributes using pseudo amino acid composition," *Proteins: Structure, Function, and Genetics*, vol. 43, pp. 246–255, 2001.

[23] R. Nair and B. Rost, "Sequence conserved for subcellular localization," *Protein Science*, vol. 11, pp. 2836–2847, 2002.

[24] Z. Lu, D. Szafron, R. Greiner, P. Lu, D. S. Wishart, B. Poulin, J. Anvik, C. Macdonell, and R. Eisner, "Predicting subcellular localization of proteins using machine-learned classifiers," *Bioinformatics*, vol. 20, no. 4, pp. 547–556, 2004.

[25] J. Kim, G. Raghava, S. Bang, and S. Choi, "Prediction of subcellular localization of proteins using pairwise sequence alignment and support vector machine," *Pattern Recognition Letters*, vol. 27, no. 9, pp. 996–1001, 2006.

[26] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Res.*, vol. 25, pp. 3389–3402, 1997.

[27] M. W. Mak and S. Y. Kung, "Conditional random fields for the prediction of signal peptide cleavage sites," in *Proc. ICASSP*, Taipei, April 2009, pp. 1605–1608.

[28] S. Y. Kung, "Kernel approaches to unsupervised and supervised machine learning," in *Proc. PCM*, ser. LNCS 5879, P. Muneesawang, et al., Ed. Springer-Verlag, 2009, pp. 1–32.

[29] K. Tsuda., "Support vector classifier with asymmetric kernel functions," in *Proceedings ESANN*, Brussels, 1999, pp. 183–188.

[30] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing IX*, Y. H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds., 1999, pp. 41–48.

[31] B. W. Matthews, "Comparison of predicted and observed secondary structure of T4 phage lysozyme," *Biochim. Biophys. Acta*, vol. 405, no. 2, pp. 442–451, 1975.