# Sequence-driven features for prediction of subcellular localization of proteins

Jong Kyoung Kim, Sung-Yang Bang, Seungjin Choi [*]

*Department of Computer Science*
*Pohang University of Science and Technology*
*San 31 Hyoja-dong, Nam-gu, Pohang 790-784, Korea*

**Abstract**

Prediction of the cellular location of a protein plays an important role in inferring the function of the protein. Feature extraction is a critical part in prediction systems, requiring raw sequence data to be transformed into appropriate numerical feature vectors while minimizing information loss. In this paper we present a method for extracting useful features from protein sequence data. The method employs local and global pairwise sequence alignment scores as well as composition-based features. Five different features are used for training support vector machines (SVMs) separately and a weighted majority voting makes a final decision. The overall prediction accuracy evaluated by the 5-fold cross-validation reached 88.53% for the eukaryotic animal data set. Comparing the prediction accuracy of various feature extraction methods, provides a biological insight into the location of targeting information. Our experimental results confirm that our feature extraction methods are very useful for predicting subcellular localization of proteins.

*Key words:* Protein sequence feature extraction, Subcellular localization prediction, Support vector machine.

## 1 Introduction

In a eukaryotic animal cell, nuclear-encoded proteins are synthesized by ribosomes in the cytosol, and delivered to their proper cellular organelles for

[*] Corresponding author. Tel.: +82-54-279-2259; Fax: +82-54-279-2299
*Email:* seungjin@postech.ac.kr (S. Choi)
*URL:* http://www.postech.ac.kr/∼seungjin (S. Choi)

the co-operational execution of a common biological function. The delivery of a newly-synthesized protein in the cytosol to an appropriate location, is referred to as *protein sorting* or *subcellular localization*. Major protein sorting processes can be divided into secretory and non-secretory pathways. In the secretory pathway, all proteins are first delivered to the endoplasmic reticulum (ER) and then transported to their associated final destinations. The delivery of proteins to the ER is determined by ER signal sequences which are generally located at the N-terminus. After translocation into the ER, most of proteins move to the Golgi complex via transport vesicles, and some proteins are delivered to the plasma membrane, lysosomes, or the extracellular matrix by further sorting. All proteins that contain no ER signal sequences are delivered through the non-secretory pathway. In this pathway, proteins with organelle-specific signal sequences are imported into mitochondrion, peroxisome, or the nucleus according to their corresponding signal sequences. Proteins having lack of any signal sequences, remain in the cytosol. The targeting information of proteins directing them to their correct cellular destinations, is stored either in the signal sequences (and additional sequences) or in the form of post-translational modifications. Proteins delivered to the ER and the mitochondrion have an N-terminal signal sequence. Proteins that are targeted to the peroxisome, have a signal sequence which is located at the N-terminus or C-terminus. Signal sequences directing proteins to the nucleus, are referred to as nuclear localization signals and are present anywhere in the protein. In the secretory pathway, proteins are sorted according to their final locations by several targeting features such as signal sequences, topogenic sequences, and post-translational modifications. The location of these features in the protein sequence cannot be restricted to the subsequences [1, 19].

Predicting the cellular location of an unknown protein plays an important role in inferring the possible function of the protein. Recently various methods have been developed to improve the prediction accuracy. This cellular location prediction, in fact, is a pattern classification problem that has been extensively studied in machine learning, pattern recognition, and statistics communities, since class labels related to cellular locations are already available in a set of training data. Various classifiers including artificial neural networks (ANN), support vector machines (SVM), and k-nearest neighbor algorithm (k-NN), were applied to this classification problem. Accurate classification requires to extract useful features from protein sequence data. Desirable feature extraction transforms the raw sequence into numerical feature vector, while minimizing information loss. In practice, the prediction accuracy is strongly affected by feature extraction methods. Most of prediction methods can be divided into two approaches, depending on their ways of feature extraction: (1) features based on protein sequence data; (2) features based on ontology data.

In the protein sequence-based approach, two popular feature extraction methods include: (1) methods involving the recognition of N-terminal signal sequences; (2) methods involving the detection of amino acids compositions from an entire sequence. The former has a strong biological implication because proteins delivered to ER, mitochondrion, or peroxisome (partially) have an N-terminal signal sequence [9, 25]. However, it is difficult to recognize underlying features from a highly diverged signal sequence, as well as to vectorize those features. The latter approach partially overcomes these difficulties, but lose the information regarding the context stored in the sequence data [2, 13, 26]. The ontology-based approach has received much attention recently because of its high prediction accuracy [3, 20]. This approach extracts the text information of homologous sequences of a query sequence by searching biological databases, in order to vectorize this information. It is not surprising that this approach leads to good performance because it utilizes various extra information derived from several sources. In addition, it cannot give biological insights on factors specifying cellular locations of proteins. Although numerous methods have been developed to improve the accuracy of subcellular localization prediction, little research was conducted for feature extraction methods relying solely on properties of amino acids sequence data.

In this paper, we present new sequence-driven feature extraction methods to predict cellular locations of proteins. To this end, we introduce feature extraction methods based on pairwise sequence alignment scores, including N-terminal profile hidden Markov model (HMM) and local/global sequence alignment. Moreover we also introduce methods based on amino acids composition to improve the prediction accuracy. Various features driven from protein sequence data, are used to train SVMs separately. For classification, we use an SVM ensemble to combine mixed types of features. Our experimental results confirm that our proposed feature extraction methods considerably improve the prediction accuracy and give a biological insight into the location of targeting information within the protein sequence.

## 2 Feature extraction

Feature extraction for prediction of subcellular localization of proteins, requires raw sequence data to be transformed into numerical feature vectors. Recent studies on feature extraction methods based on properties of amino acids sequences, are focused on the amino acids composition. Amino acids composition and subcellular localization are related [5], however composition-based methods have critical limitations in terms of their discriminative power and location coverage. Recently we proposed a feature extraction method where we used the scores of a global sequence alignment [17]. Despite its high prediction accuracy, its time complexity was relatively higher, compared to composition-based methods. Moreover its location coverage was also limited to some proteins whose signal sequences are located at the N-terminus. In order to overcome these limitations, we present three different methods which extract features from signal sequences. We also consider two composition-based methods to improve the prediction accuracy.

### *2.1 Clustering*

In our recent work [17], we used one of global sequence alignment methods, the Needleman-Wunsch algorithm [23] to compute scores between a sequence and every sequence in the training set. These scores were used to convert a protein sequence into a numerical feature vector. A drawback of this method is that the computational complexity increases dramatically as the size of the training set grows. In this paper, we select representative sequences in the training data in order to decrease the computational complexity. To this end, we carry out clustering using a constructed phylogenetic tree. The overall clustering framework is illustrated below.

First, we truncate every sequence in the training set, preserving only first 40 residues corresponding to the N-terminus. We do not use the entire sequence because it leads to very long average distance between pairs of sequences within each cluster.

Second, all sequences in the training data are grouped into clusters according to their associated class labels, so that phylogenetic trees are constructed separately for each class.

Third, we calculate the Jukes-Cantor distance $\rho_{ij}$ between each pair of sequences:

$$\rho_{ij} = -\frac{19}{20}\log\left(1 - f\frac{20}{19}\right), \tag{1}$$

where $f$ is the fraction of sites where two sequences differ after they are aligned using the Needleman-Wunsch algorithm [10, 7].

Next, we construct phylogenetic trees by the UPGMA clustering which is the unweighted pair group method using arithmetic averages. The UPGMA method is an agglomerative hierarchical clustering algorithm where each sequence is assigned to its own cluster first and then gradually these clusters are merged into larger clusters until all sequences belong to a single cluster [14]. The distance $d_{ij}$ between two clusters $C_i$ and $C_j$ is defined by the average distance between pairs of sequences from each cluster, i.e.,

$$d_{ij} = \frac{1}{|C_i||C_j|}\sum_{p\in C_i}\sum_{q\in C_j}\rho_{pq}, \tag{2}$$

where $|C_i|$ and $|C_j|$ denote the number of sequences in clusters $i$ and $j$, respectively.

The outline of the UPGMA clustering procedure is as follows. It begins by assigning each sequence to its own cluster. We define one leaf node of the tree for each sequence, placing at height 0. With this initialization, the following procedures are iterated until only one cluster remains [7]. At each iteration, we determine two clusters $i$ and $j$ for which $d_{ij}$ is minimal, defining a new cluster $k$ by $C_k = C_i \cup C_j$. The new node $k$ has child nodes $i$ and $j$, placing it at height $d_{ij}/2$. This procedure is repeated until only one cluster remains.

Clusters are determined by selecting some cluster-parent nodes in the constructed tree, considering all leaf nodes that share the same parent node to be elements in the same cluster. Selection of these cluster-parent nodes are carried out, according to their height as well as the number of their leaf nodes. The bottom-up search from leaf nodes, finds nodes their height is above a prespecified height (say, 3, in our experiments). Child nodes of these nodes are determined as cluster-parent nodes, while these cluster-parent nodes lead to separate clusters whose elements are leaf nodes sharing the same cluster-parent node. In addition, for balanced clustering, we split a cluster if the number of leaf nodes belonging to a certain cluster-parent node is too large. This can be done by its child nodes to be eligible cluster-parent nodes.

### *2.2 N-terminal profile hidden Markov model*

Our first feature extraction method which exploits the properties of sequence data, is to use N-terminal profile hidden Markov models (HMMs) that are suited for a statistical modeling of sequences [7]. Hierarchically clustered sequences (that are obtained in Sec. 2.1) undergo multiple sequence alignment for each cluster, where we used CLUSTALX 1.83 [29]. Note that the multiple sequence alignment by CLUSTALX 1.83, for each cluster, was carried out with truncated sequences (preserving first 40 residues corresponding to the N-terminus). Then, we construct profile HMMs for each cluster using HMMer 2.3.1 [8], which represent families of N-terminal sequences.

In order to transform N-terminal sequences into numerical feature vectors, we compute log-odds scores between sequences and profile HMMs. Given an N-terminal sequence $\boldsymbol{a} = a_1 a_2 \dots a_n$ of length $n$, the log-odds score $s(\boldsymbol{a}, \mathcal{M})$ between the sequence $\boldsymbol{a}$ and the profile HMM $\mathcal{M}$ is defined by

$$s(\boldsymbol{a}, \mathcal{M}) = \log_2 \frac{p(\boldsymbol{a}|\mathcal{M})}{p(\boldsymbol{a}|\mathcal{R})}, \tag{3}$$

where $p(\boldsymbol{a}|\mathcal{M})$ is the probability of the sequence $\boldsymbol{a}$ given the model $\mathcal{M}$ and $p(\boldsymbol{a}|R)$ is the probability of the sequence $\boldsymbol{a}$ given a random model $\mathcal{R}$. In the random model, $p(\boldsymbol{a}|\mathcal{R})$ is given by

$$p(\boldsymbol{a}|\mathcal{R}) = \prod_{i=1}^{n} p_{a_i}, \tag{4}$$

where $p_{a_i}$ is the probability of observing the amino acid $a_i$ in nature [7]. A $d$-dimensional feature vector $\boldsymbol{x}_t$ for the $t$th protein sequence has the form

$$\boldsymbol{x}_t = \left[x_{t1}, x_{t2}, \ldots, x_{td}\right]^\top, \tag{5}$$

where $x_{ti}$ corresponds to the log-odd score between the $t$th sequence and the $i$th profile HMM, and the superscript $\top$ denotes the matrix or vector transpose operator. Note that $d$ is associated with the total number of profile HMMs constructed.

### *2.3 N-terminal global pairwise sequence alignment*

The Needleman-Wunsch algorithm is a dynamic programming method which finds the optimal global alignment between two sequences, allowing gaps. The basic idea is to construct an optimal alignment using previous solutions for optimal alignments of smaller subsequences. Let us denote two protein sequences of length $m$ and $n$ by $\boldsymbol{a} = a_1 a_2 \ldots a_m$ and $\boldsymbol{b} = b_1 b_2 \ldots b_n$, respectively. As a shorthand notation, $a_{1:i}$ represents the initial subsequence of $\boldsymbol{a}$ up to $a_i$, i.e., $a_{1:i} = a_1 \ldots a_i$.

We construct a matrix $F \in \mathbb{R}^{(m+1)\times(n+1)}$ whose $(i, j)$-element, $F(i, j)$, is the score of the best alignment between $a_{1:i}$ and $b_{1:j}$. Initializing $F(0, 0) = 0$, the matrix $F$ is built up recursively, with filled up from top left to bottom right, through the following recursion:

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(a_i, b_j), \\ F(i-1, j) - g, \\ F(i, j-1) - g, \end{cases} \tag{6}$$

where $s(a_i, b_j)$ is the score for the match between $a_i$ and $b_j$, and $g$ is a gap penalty. In order to deal with some boundary conditions, we define $F(i, 0) = -ig$ and $F(0, j) = -jg$ for the first column and the first row, respectively. Filling in $F(i, j)$ values, we determine the final cell of the matrix $F$, $F(m, n)$, which is the best score between two sequences $\boldsymbol{a}$ and $\boldsymbol{b}$. We can find a global alignment by tracing back choices from (6) that led to the final value $F(m, n)$ [10, 7].

Scores $s(a_i, b_j)$ are obtained from a BLOSUM matrix [12] which is one of widely-used substitution matrices. BLOSUM matrices are constructed from blocks of ungapped alignments of protein families. In each block there are redundancies, hence, sequences that are sufficiently close to each other, are grouped into the same cluster. Each resulting cluster is considered as a single sequence. The closeness between sequences is determined by specifying a cut-off identity $X\%$. The frequency $A_{ab}$ of observing the amino acid pair $ab$ aligned in the same column of blocks is calculated by summing each occurrence in all the blocks. Since each cluster is considered as a single sequence, the occurrence is corrected by weighting the factor $\frac{1}{n_1 n_2}$, where $n_1$ and $n_2$ are sizes of clusters. Then, the score $s(a,b)$ is computed by

$$s(a,b) = \log\left(\frac{p_{ab}}{p_a p_b}\right), \tag{7}$$

where $p_{ab} = \frac{A_{ab}}{\sum_{cd} A_{cd}}$, $p_a = \frac{\sum_b A_{ab}}{\sum_{cd} A_{cd}}$, and $p_b = \frac{\sum_a A_{ab}}{\sum_{cd} A_{cd}}$. The log-odds score is scaled and rounded to an integer value. The BLOSUM matrix with $X = 50$, is referred to as 'BLOSUM50', which is widely-used for alignment with gaps [7].

In contrast to our earlier work [17] where all the sequences in the training set were used for pairwise sequence alignment, we select a representative sequence randomly from each cluster, in order to reduce the computational complexity. The minimal allowable length of sequences is restricted to 80, where the first residue should be methionine, implying that the first residue is translated from the start codon. Note that here we use sequences truncated after first 80 residues, while only first 40 residues are used for the case of N-terminal profile HMM. The size of the N-terminus cannot be clearly determined, therefore, we consider two conflicting properties that include the information loss as well as divergence. As the N-terminal size increases, the sequence divergence within each sequence family increases but the information loss decreases.

A $d$-dimensional feature vector $\boldsymbol{x}_t$ for the $t$th protein sequence has the form

$$\boldsymbol{x}_t = \left[x_{t1}, x_{t2}, \ldots, x_{td}\right]^\top, \tag{8}$$

where $x_{ti}$ is the score of the Needleman-Wunsch algorithm between the $t$th sequence and the $i$th representative sequence (the representative sequence randomly selected from the $i$th cluster). Note that $d$ is equal to the total number of selected representative sequences. The gap penalty was set to be $-8$ and the BLOSUM50 matrix was used as a substitution matrix.

### *2.4 Full sequence local pairwise sequence alignment*

The Smith-Waterman algorithm [28] finds the optimal alignment between subsequences of $\boldsymbol{a}$ and $\boldsymbol{b}$. This is a local alignment algorithm, which can be used to seek common patterns or domains in two sequences. The algorithm is closely related to the Needleman-Wunsch algorithm, but the main difference lies in the following recursive equation

$$F(i,j) = \max \begin{cases} 0, \\ F(i-1,j-1) + s(a_i, b_j), \\ F(i-1,j) - g, \\ F(i,j-1) - g. \end{cases} \tag{9}$$

The option 0 means that we start a new local alignment at the position. Since every element of the matrix $F$ is nonnegative, the first row and the first column are filled with 0's. To find the optimal local alignment, we first look for the highest value of $F(i,j)$, and trace back the choices of the recursion until we meet an element with value 0 [7].

So far we have assumed that signal sequences are located at the N-terminus, expecting the global alignment between two N-terminal regions. A more common case is a situation where signal sequences or targeting signals are located anywhere in the protein. In such a case, the more reasonable way of detecting the internal targeting information is to use the Smith-Waterman algorithm. Representing a protein sequence by the scores of the Smith-Waterman algorithm was successfully used in the SVM-pairwise for detecting remote structural and evolutionary relationships [18]. The general procedure for this feature extraction method is almost same as what was carried out with the the Needleman-Wunsch algorithm, except for two differences. These differences include: (1) protein sequences are not truncated (instead, full sequences are used); (2) local alignment is used, instead of global alignment.

A $d$-dimensional feature vector $\boldsymbol{x}_t$ for the $t$th protein sequence has the form

$$\boldsymbol{x}_t = [x_{t1}, x_{t2}, \dots, x_{td}]^\top , \tag{10}$$

where $x_{ti}$ is the score of the Smith-Waterman algorithm between the $t$th sequence and the $i$th representative sequence. Note that $d$ is equal to the total number of selected representative sequences. Since $d$ is equal to the total number of clusters, it is also equal to the total number of profile HMMs.

### *2.5 Full sequence dipeptide composition*

It is known that amino acids composition and subcellular localization is related [5]. However, the predictive power of the composition-based approach is not sufficient to discriminate all proteins. The dipeptide composition is an extension of amino acids composition, where we add the information on the local order of amino acids. The dipeptide means two consecutive amino acids in a protein sequence. Twenty different amino acids lead to 400 combinations of dipeptide. In practice, it is proved that the dipeptide composition has superior predictive power, compared to the amino acids composition. The compositional fraction of the $i$th dipeptide $f^{dc}(i)$ is given by

$$f^{dc}(i) = \frac{N(i)}{\sum_{j=1}^{400} N(j)}, \tag{11}$$

where $N(i)$ is the total count of the $i$th dipeptide in the protein sequence. Then, the feature vector $\boldsymbol{x}_t$ for the $t$th protein sequence is given by

$$\boldsymbol{x}_t = \left[f_t^{dc}(1), f_t^{dc}(2), \ldots, f_t^{dc}(400)\right]^\top, \tag{12}$$

where $f_t^{dc}(i)$ is the compositional fraction of the $i$th dipeptide in the $t$th protein sequence.

### *2.6 Full sequence physico-chemical properties*

Since the signal sequences are not well conserved, it is generally thought that the factors determining the cellular locations are physico-chemical properties such as hydrophobicity or the position of charged amino acids [1]. We consider 121 physico-chemical properties, the list of which is available at `http://home.postech.ac.kr/∼blkimjk/aaindex1m.txt`, in order to represent a protein sequence by a 121-dimensional feature vector based on amino acids composition. We use the AAindex database [15] to get the values of physico-chemical properties for all 20 amino acids, which are thought to be related to protein functions. To be expressed in comparable units, the values are normalized by subtracting the mean off and dividing by the standard deviation. The average value of the $i$th physico-chemical property is defined by

$$\varphi(i) = \sum_{j=1}^{20} A_i(j) f^{ac}(j), \tag{13}$$

where $A_i(j)$ is the normalized value of the $j$th amino acid of the $i$th physico-chemical property and $f^{ac}(j)$ is the compositional fraction of the $j$th amino acid. The feature vector $\boldsymbol{x}_t$ for the $t$th protein sequence is given by

$$\boldsymbol{x}_t = \left[\varphi_t(1), \varphi_t(2), \ldots, \varphi_t(121)\right]^\top, \tag{14}$$

where $\varphi_t(i)$ is the average value of the $i$th physico-chemical property in the $t$th protein sequence.

# 3 Classification

## *3.1 Support vector machine classifier*

SVM classifiers have recently been used as popular and powerful tools for classification, due to their strong theoretical origin at statistical learning theory as well as their high performance in practical applications [11, 6]. SVM classifiers are kernel-based learning algorithms, determining the optimal hyperplane decision boundary in the feature space. In kernel-based algorithms, a kernel trick leads us to process the data in a feature space without the explicit knowledge of a nonlinear mapping from the data space to a feature space. The high dimensionality of a feature space might cause the curse of dimensionality. However, the optimal separating hyperplane with a maximal margin in the feature space, can relieve this problem. In statistical learning theory, we can minimize the complexity term of the upper bound of the expected risk by maximizing the margin of the separating hyperplane. The minimization of the upper bound can be viewed as relieving the over-fitting problem [22]. The maximization of the margin can be formulated as a quadratic optimization problem so that a global solution can be easily obtained.

In the present study, we used OSU SVM Matlab toolbox 3.00 for the SVM classifier that is freely available at http://www.ece.osu.edu/∼maj/osu_svm. The prediction of subcellular localization is a multi-class classification problem, but the SVM classifier can only deal with the binary one. Therefore, we need to construct a set of binary classifiers for multi-class classification. We construct $(M-1)M/2$ binary classifiers for $M$ classes. In this pairwise classification, each possible pair of classes is considered and a test pattern is classified by the majority voting. This approach has two advantages over the *one versus the rest* method. The weak point of the latter approach is that it compares the real values in outputs of $M$ binary classifiers directly. Because each binary classifier is trained on different binary classification problems, their real values in outputs of the classifiers may not be suitable for comparison. In addition, in the *one versus the rest* approach, the numbers of positive and negative training data points are not symmetric. These two weak points can be solved by the pairwise classification [27]. The kernel function used in this study is the *radial basis function* (RBF) kernel with one parameter $\gamma$:

$$k(\boldsymbol{x}, \mathbf{y}) = \exp\left\{-\gamma \|\boldsymbol{x} - \mathbf{y}\|^2\right\}. \tag{15}$$

During the training and testing, only the RBF kernel parameter $\gamma$ and the regularization parameter $C$ were considered and the remaining parameters were kept constant.

### *3.2 Weighted majority voting*

SVM ensemble is a collection of several SVM classifiers whose individual decisions are combined in some aggregation methods. It is known that the performance of SVM ensemble is often much better than that of individual SVM classifiers, because of independently-trained SVM classifiers and their uncorrelated errors [16]. Since we train several independent SVM classifiers for each differently extracted feature vector, we need to aggregate them in an appropriate manner. The majority voting is the simplest and widely-used aggregation method.

Let $\widehat{C_k}(\boldsymbol{x})$, $k = 1, ..., K$ ($K$ is the total number of separately-trained SVM classifiers), be the class label predicted by the $k$th SVM classifier, given a feature vector $\boldsymbol{x}$. Denote by $C_m$ ($m = 1, ..., M$) class label $m$, where $M$ is the total number of class labels. For the case of a dataset with 9 classes, $M = 9$. Given a data vector $\boldsymbol{x}$, the decision of the SVM ensemble $\widehat{C}(\boldsymbol{x})$ is determined by

$$\widehat{C}(\boldsymbol{x}) = \arg\max_{m} \sum_{k=1}^{K} I_{km}, \tag{16}$$

where

$$I_{km} = \begin{cases} 1 \text{ if } \widehat{C_k}(\boldsymbol{x}) = C_m, \\ 0 \text{ otherwise.} \end{cases} \tag{17}$$

This voting scheme treats all SVM classifiers with equal weights. Prediction errors of SVM classifiers are often different, thus, it is more reasonable to give them different weights, in proportion to their prediction performance. In the weighted majority voting, the predicted class label of the SVM ensemble is given by

$$\hat{C}(\boldsymbol{x}) = \arg\max_{m} \sum_{k=1}^{K} W(k, m) I_{km}, \tag{18}$$

where $W(k, m)$ is the weight when the predicted class label of the $k$th classifier is $C_m$. The weights can be determined by calculating appropriate performance measure for each classifier. Details are illustrated in Sec. 4.3.

### *3.3 The proposed prediction system*

The overall schematic diagram of our prediction system is illustrated in Fig. 1. The prediction system consists of four steps. First, a target protein sequence is truncated after first 40 or 80 residues in order to get the N-terminus. Next, truncated sequences are transformed into two different feature vectors which are constructed by computing the scores of pairwise sequence alignments based on the profile HMMs and the Needleman-Wunsch algorithm, respectively. The full sequence is also converted into three different feature vectors that are constructed by computing the scores of the Smith-Waterman algorithm, or by calculating the compositional fractions of all dipeptides and the average values of 121 physico-chemical properties.

Representative sequences and profile HMMs are divided into two sets, one of which contains positive data and the other of which contains negative data. The positive set contains samples (or associated models) whose class labels match the target sequence, and the negative set contains the rest of samples (or models). In this way, the discriminative power of feature vectors increases, since two sets contain the information on positive examples as well as negative ones.

Five different feature vectors are fed into separate classifiers, each of which consists of $(M-1)M/2$ binary SVM classifiers for $M$ classes. In this pairwise classification, the feature vector is assigned to the class label associated with the highest value in the majority voting. Finally the weighted majority voting makes a final decision, according to five predicted class labels.

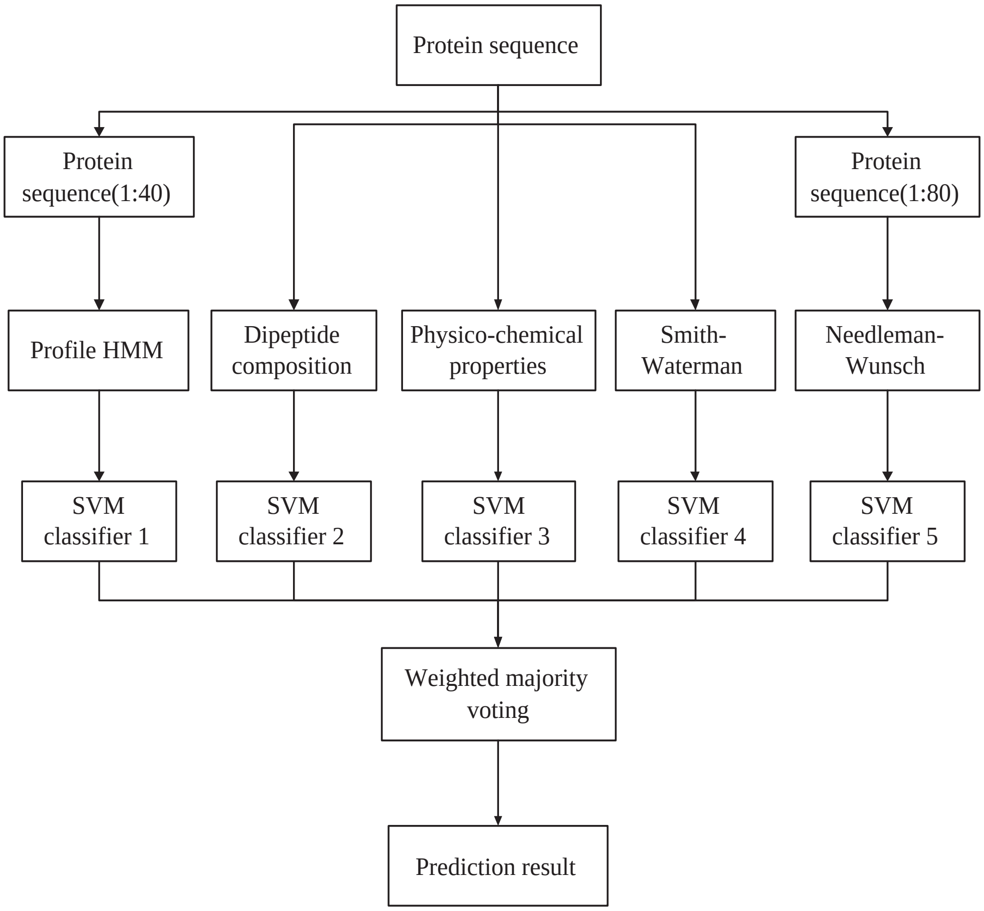

Fig. 1. The schematic diagram of the proposed prediction system is illustrated.

## 4 Numerical experiments and results

### *4.1 Data sets*

We used the animal data set generated by [20] for training and evaluating our prediction system. All sequences in the data set were extracted from SWISS-PROT release 42.7, and their cellular locations were chosen by referring to the SUBCELL field. More information on the data generation steps is available at http://www.cs.ualberta.ca/∼bioinfo/PA/Subcellular/experiments/Extract_Data_42_7.html. We excluded protein sequences containing ambiguous amino acids such as B, Z, or X. As shown in Table 1, the data set consists of 11688 eukaryotic animal proteins with 9 cellular locations: cytoplasm, ER, extracellular, golgi, lysosome, mitochondrion, nucleus, plasm membrane, and peroxisome.

Table 1
The number of proteins of each cellular locations in the data set.

| Cellular location | Number of sequences |
|---|---|
| Cytoplasm | 1945 |
| ER | 607 |
| Extracellular | 4410 |
| Golgi | 184 |
| Lysosome | 163 |
| Mitochondrion | 1220 |
| Nucleus | 2940 |
| Plasma membrane | 111 |
| Peroxisome | 108 |
| Total | 11688 |

### *4.2 Evaluation*

The performance of our prediction system was evaluated using 5-fold cross-validation. In 5-fold cross-validation, the whole data set was partitioned into five exclusive subsets, and in turn one subset was used for the test data and the remaining sets were used for the training data. To measure the performance, sensitivity, specificity, Matthew's correlation coefficient (MCC) [9, 21], and overall accuracy were calculated using the following equations:

$$\mathrm{Sensitivity}(m) = \frac{\mathrm{tp}(m)}{\mathrm{tp}(m) + \mathrm{fn}(m)}, \tag{19}$$

$$\mathrm{Specificity}(m) = \frac{\mathrm{tp}(m)}{\mathrm{tp}(m) + \mathrm{fp}(m)}, \tag{20}$$

$$\mathrm{MCC}(m) = \frac{\mathrm{tp}(m)\mathrm{tn}(m) - \mathrm{fp}(m)\mathrm{fn}(m)}{\sqrt{\mathrm{de}(m)}}, \tag{21}$$

$$\text{Overall accuracy} = \frac{\sum_{m=1}^{M} \mathrm{tp}(m)}{N}, \tag{22}$$

where

$$\mathrm{de}(m) = (\mathrm{tp}(m) + \mathrm{fn}(m))\,(\mathrm{tp}(m) + \mathrm{fp}(m))\,(\mathrm{tn}(m) + \mathrm{fp}(m))\,(\mathrm{tn}(m) + \mathrm{fn}(m))\,, \tag{23}$$

and $m$ represents the $m$th class, $N$ is the total number of sequences, $M$ is the total number of classes, tp($m$) (true positive) is the number of correctly predicted sequences of $m$th class, tn($m$) (true negative) is the number of correctly predicted sequences which is not in $m$th class, fp($m$) (false positive) is the number of over predicted sequences of $m$th class, and fn($m$) (false negative) is the number of under predicted sequences of $m$th class.

### *4.3 Results*

We applied 5-fold cross-validation to select proper values of parameters including: (1) RBF kernel width $\gamma$ and regularization parameter $C$ in the SVM classifier; (2) weights $W(k, m)$ in the SVM ensemble. For 5-fold cross-validation, we carry out clustering for each training dataset, to construct 5 different sets of profile HMMs and of representative sequences. For each feature extraction method, the SVM classifier is trained separately and optimal values of SVM parameters are selected in such a way that the SVM classifier maximize the overall accuracy for five test datasets. As performance measures, we compute sensitivity, specificity, and MCC for each SVM classifier. In the case of the weighted majority voting, we use these three measures to determine weights $W(k, m)$ in Eq. (18).

The performance of all the feature extraction methods is summarized in Table 2, 3, 4, 5, and 6. Features based on N-terminal profile HMMs ($\gamma = 0.003$ and $C = 10$) showed the overall accuracy of 83.62%. In the case of features based on the N-terminal Needleman-Wunsch algorithm ($\gamma = 2$ and $C = 100$), the prediction accuracy reached 83.25% which is slightly lower, compared to N-terminal profile HMMs. These two feature extraction methods that are specialized for extracting targeting information from N-terminal signal sequences, showed similar prediction patterns. In other words, their sensitivity for golgi, lysosome, and plasma membrane was very low, whereas they predicted extracellular, mitochondrial, and nuclear proteins with high accuracy. The sensitivity for cytoplasm, ER, and peroxisome was moderate (See Table 2 and 3).

Results with features based on full sequences, were considerably different from the N-terminal-based methods. Features based on the Smith-Waterman algorithm ($\gamma = 80$ and $C = 10$) showed the prediction accuracy of 83.23%. The overall accuracy of features based on dipeptide composition ($\gamma = 170$ and $C = 10$) and physico-chemical properties ($\gamma = 4$ and $C = 10$) reached 85.82% and 82.29%, respectively. In these three methods which are based on full sequences, the sensitivity for golgi, lysosome, and plasma membrane increased remarkably (See Table 4, 5, and 6). In addition, the specificity of these three methods was generally lower than that of the two methods based on N-terminal sequences.

To construct the SVM ensemble from the collection of separately-trained SVM classifiers, we tested four different aggregation methods in the framework of unweighted/weighted majority voting. Table 7 summarizes the overall accuracy for these four aggregation methods, including an unweighted majority voting and three weighted majority voting methods (based on sensitivity, MCC, and specificity). An shown in Table 7, the specificity-based weighted majority voting achieved the best accuracy.

We investigate the performance of various SVM ensembles that are constructed by different combinations of five separately-trained SVM classifiers. All these SVM ensembles are built using the specificity-based weighted majority voting. The first SVM ensemble (ensemble 1) was constructed by combining the three SVM classifiers based on the pairwise sequence alignment (N-terminal profile HMMs, N-terminal Needleman-Wunsch algorithm, and full sequence Smith-Waterman algorithm). The overall accuracy of the ensemble 1 reached 86.17%, which was higher than that of any individual SVM classifier. The second SVM ensemble (ensemble 2) was constructed based on dipeptide composition and physico-chemical properties. The prediction accuracy of the ensemble 2 is slightly lower than that of the ensemble 1, and even worse than that of the dipeptide composition-based method. To compare the effect of two composition-based features in the ensemble 1, we construct two new SVM ensembles (ensemble 3 and 4) that are different from the ensemble 1. As shown in Table 8, features based on dipeptide composition have more influence on the performance of the resulting SVM ensemble, compared to features based on physico-chemical properties. Finally, the SVM ensemble combining all five SVM classifiers (referred to as ensemble 5) showed the overall accuracy of 88.53%. The prediction accuracy of our SVM ensemble is nearly 10% higher than that of previous methods relying solely on amino acids sequence properties [4, 24]. However, it is less meaningful to compare the prediction accuracy directly because the datasets are different. Comparing with the ontology-based approach, whose accuracy is about 4% higher than our method, is also unfair since ontology-based methods use various extra information extracted from ontological labels [20].

Table 9 shows that the location of targeting information has a strong influence on the average sensitivity of feature extraction methods that are based on N-terminal sequences and full sequences. Proteins targeted to extracellular, mitochondrion, and nucleus were predicted by the N-terminal-based methods with higher accuracy. This result, in the case of nuclear proteins, are not well matched with the fact that the nuclear localization signals can be located anywhere. However, from this result, we may logically assume that most of nuclear localization signals are located at the N-terminus. For proteins whose targeting information is not restricted to the N-terminus, full sequence-based methods showed better sensitivity.

Table 2
Prediction performance of subcellular localization based on the N-terminal profile HMM.

| Location | Specificity | Sensitivity | MCC | Accuracy |
|---|---|---|---|---|
| Cytoplasm | 0.8005 | 0.6725 | 0.6813 | 0.8362 |
| ER | 0.8184 | 0.6458 | 0.7118 | |
| Extracellular | 0.8899 | 0.9515 | 0.8596 | |
| Golgi | 1.0000 | 0.3587 | 0.5953 | |
| Lysosome | 0.9630 | 0.3190 | 0.5508 | |
| Mitochondrion | 0.9366 | 0.8107 | 0.8553 | |
| Nucleus | 0.7464 | 0.9099 | 0.7521 | |
| Plasma membrane | 0.9500 | 0.3423 | 0.5679 | |
| Peroxisome | 0.9667 | 0.5370 | 0.7185 | |

## 5 Concluding remarks

We have presented a method for predicting cellular locations of proteins, where features associated with protein sequences were constructed by scores of pairwise sequence alignment, or by amino acids compositional information. The high prediction performance of our method was verified, using eukaryotic animal data sets, through 5-fold cross validation. The high performance mainly resulted from various types of features driven by sequences and partly came

Table 3
Prediction performance of subcellular localization based on the N-terminal Needleman-Wunsch algorithm.

| Location | Specificity | Sensitivity | MCC | Accuracy |
|---|---|---|---|---|
| Cytoplasm | 0.8092 | 0.7172 | 0.7124 | |
| ER | 0.9557 | 0.6755 | 0.7935 | |
| Extracellular | 0.8684 | 0.9544 | 0.8420 | |
| Golgi | 0.9636 | 0.2880 | 0.5230 | |
| Lysosome | 0.9583 | 0.2822 | 0.5166 | 0.8325 |
| Mitochondrion | 0.8909 | 0.7631 | 0.8029 | |
| Nucleus | 0.7538 | 0.8854 | 0.7429 | |
| Plasma membrane | 0.9394 | 0.2793 | 0.5098 | |
| Peroxisome | 0.9630 | 0.4815 | 0.6787 | |

Table 4
Prediction performance of subcellular localization based on the full sequence Smith-Waterman algorithm.

| Location | Specificity | Sensitivity | MCC | Accuracy |
|---|---|---|---|---|
| Cytoplasm | 0.7400 | 0.7522 | 0.6893 | |
| ER | 0.8608 | 0.7743 | 0.8052 | |
| Extracellular | 0.8523 | 0.9358 | 0.8134 | |
| Golgi | 0.7159 | 0.3424 | 0.4890 | |
| Lysosome | 0.9381 | 0.5583 | 0.7204 | 0.8323 |
| Mitochondrion | 0.8555 | 0.7525 | 0.7776 | |
| Nucleus | 0.8448 | 0.8500 | 0.7888 | |
| Plasma membrane | 0.8868 | 0.4234 | 0.6101 | |
| Peroxisome | 0.9259 | 0.4630 | 0.6523 | |

from a simple SVM ensemble which combined the contribution of these features.

It is expected that the discriminative power of sequence-driven features (constructed by scores of pairwise sequence alignment) increases, since they contain positive information as well as negative information. Increasing the number of representative sequences (each of which was randomly drawn from each cluster), is expected to improve the prediction accuracy. This was already confirmed in our earlier work [17] where it was observed that the prediction accuracy was highly correlated with the number of representative sequences. As

Table 5
Prediction performance of subcellular localization based on the full sequence dipeptide composition.

| Location | Specificity | Sensitivity | MCC | Accuracy |
|---|---|---|---|---|
| Cytoplasm | 0.7724 | 0.7784 | 0.7265 | |
| ER | 0.8679 | 0.8336 | 0.8414 | |
| Extracellular | 0.8737 | 0.9567 | 0.8517 | |
| Golgi | 0.8416 | 0.4620 | 0.6188 | |
| Lysosome | 0.9206 | 0.7117 | 0.8068 | 0.8582 |
| Mitochondrion | 0.8293 | 0.7943 | 0.8223 | |
| Nucleus | 0.8701 | 0.8524 | 0.8103 | |
| Plasma membrane | 0.9455 | 0.4685 | 0.6632 | |
| Peroxisome | 0.9412 | 0.5926 | 0.7448 | |

Table 6
Prediction performance of subcellular localization based on the full sequence physico-chemical properties.

| Location | Specificity | Sensitivity | MCC | Accuracy |
|---|---|---|---|---|
| Cytoplasm | 0.7338 | 0.7357 | 0.6754 | |
| ER | 0.8328 | 0.8204 | 0.8153 | |
| Extracellular | 0.8485 | 0.9222 | 0.7984 | |
| Golgi | 0.7109 | 0.4946 | 0.5866 | |
| Lysosome | 0.8133 | 0.7485 | 0.7766 | 0.8229 |
| Mitochondrion | 0.8438 | 0.7574 | 0.7736 | |
| Nucleus | 0.8376 | 0.8071 | 0.7562 | |
| Plasma membrane | 0.8254 | 0.4685 | 0.6188 | |
| Peroxisome | 0.7692 | 0.5556 | 0.6505 | |

Table 7
The performance comparison of different majority voting methods.

| Voting scheme | Overall accuracy |
|---|---|
| Unweighted majority voting | 0.8842 |
| Weighted majority voting (sensitivity) | 0.8775 |
| Weighted majority voting (MCC) | 0.8813 |
| Weighted majority voting (specificity) | 0.8853 |

the number of representative sequences increases, the representational space of protein sequences is growing so that the generalization performance of features derived from those sequences is expected to increase.

In our methods of feature extraction, we have used a biological prior knowledge on N-terminal signal sequences. We have shown that N-terminal-based features improved the prediction accuracy for proteins having N-terminal signal sequences. We have also shown that the specificity-based majority voting scheme was effective for constructing the SVM ensemble from separately-trained SVM classifiers. From the comparative study with several SVM ensembles, we have shown that the prediction performance was significantly improved by combining pairwise sequence alignment-based features with composition-based features.

Comparing the average sensitivity of the N-terminal sequence-based method and the full sequence-based method, has led us to get a biological insight into the location of the targeting information. There is still a main problem to be further explored. For proteins whose targeting information is not restricted to the N-terminus, the sensitivity is considerably low. Therefore, more study will be required to resolve this low sensitivity problem. On the other hand, our current study is expected to serve as a platform from which studies on developing feature extraction methods based on amino acids sequence properties may be undertaken with greater depth and specificity.

## 6 Acknowledgments

This work was supported by National Core Research Center for Systems Bio-Dynamics and Basic Research Fund in POSTECH.

## References

[1] B. Alberts, D. Bray, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Essential Cell Biology: An Introduction to the Molecular Biology of the Cell*. Garland, New York, 1998.

[2] M. Bhasin and G. P. Raghava. ESLpred: SVM-based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST. *Nucleic Acids Res.*, 32:W414–W419, 2004.

[3] Y. D. Cai and K. C. Chou. Predicting subcellular localization of proteins in a hybridization space. *Bioinformatics*, 20:1151–1156, 2004.

[4] Y. D. Cai, X. J. Liu, X. B. Xu, and K. C. Chou. Supprot vector machines for prediction of protein subcellular location by incorporating quasi-sequence-order effect. *J. Cell Biochem.*, 84:343–348, 2002.

[5] J. Cedano, P. Aloy, J. A. Perezpons, and E. Querol. Relation between amino acid composition and cellular location of proteins. *J. Mol. Biol.*, 266:594–600, 1997.

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge, New York, 2000.

[7] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, 1998.

[8] S. R. Eddy. Profile hidden markov models. *Bioinformatics*, 14:755–763, 1998.

[9] 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.*, 300:1005–1016, 2000.

[10] W. J. Ewens and G. R. Grant. *Statistical Methods in Bioinformatics: An Introduction*. Springer, New York, 2005.

[11] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Schölkopf. Support vector machines. *IEEE Intelligent Systems*, 13:18–28, 1998.

[12] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, 89:10915–10919, 1992.

[13] S. Hua and Z. Sun. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 17:721–728, 2001.

[14] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, New Jersey, 1988.

[15] S. Kawashima and M. Kanehisa. AAindex: amino acid index database. *Nucelic Acids Res.*, 28:374–374, 2000.

[16] H. Kim, S. Pang, H. Je, D. Kim, and S. Y. Bang. Constructing support vector machine ensemble. *Pattern Recognition*, 36:2757–2767, 2003.

[17] J. K. Kim, G. P. S. Raghava, S. Y. Bang, and S. Choi. Prediction of subcellular localization of proteins using pairwise sequence alignment and support vector machine. *Pattern Recognition Letters*, 2006. in press.

[18] L. Liao and W. S. Noble. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *J. Comput. Biol.*, 10:857–868, 2003.

[19] H. Lodish, A. Berk, P. Matsudaira, C. A. Kaiser, M. Krieger, M. P. Scott, S. L. Zipursky, and J. Darnell. *Molecular Cell Biology*. W. H. Freeman, New York, 2003.

[20] 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*, 20:547–556, 2004.

[21] B. W. Matthews. Comparison of predicted and observed secondary structure of t4 phase lysozyme. *Biochim. Biophys. Acta.*, 405:442–451, 1975.

[22] K. R. Müller, S. Mika, G. Ratsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural*

*Networks*, 12:181–202, 2001.

[23] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48:443–453, 1970.

[24] K. Park and M. Kanehisa. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, 19:1656–1663, 2003.

[25] M. Reczko and A. Hatzigerrorgiou. Prediction of the subcellular localization of eukaryotic proteins using sequence signals and composition. *Proteomics*, 4:1591–1596, 2004.

[26] A. Reinhardt and T. Hubbard. Using neural networks for prediction of the subcellular location of proteins. *Nucelic Acids Res.*, 26:2230–2236, 1998.

[27] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, 2002.

[28] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

[29] J. D. Thompson, T. J. Gibson, F. Plewniak, F. Jeanmougin, and D. G. Higgins. The CLUSTAL X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucelic Acids Res.*, 25:4876–4882, 1997.

Table 8
The performance comparison of five different SVM ensembles.

| Method | Location | Specificity | Sensitivity | MCC | Accuracy |
|---|---|---|---|---|---|
| Ensemble 1 (SW+NW+ HMM) | Cytoplasm | 0.8905 | 0.7064 | 0.7548 | 0.8617 |
| | ER | 0.9485 | 0.6985 | 0.8046 | |
| | Extracellular | 0.8717 | 0.9780 | 0.8668 | |
| | Golgi | 0.9701 | 0.3533 | 0.5817 | |
| | Lysosome | 0.9667 | 0.3558 | 0.5832 | |
| | Mitochondrion | 0.8885 | 0.8492 | 0.8516 | |
| | Nucleus | 0.8062 | 0.9184 | 0.8054 | |
| | Plasma membrane | 0.9130 | 0.3784 | 0.5853 | |
| | Peroxisome | 0.9516 | 0.5463 | 0.7190 | |
| Ensemble 2 (DC+PC) | Cytoplasm | 0.8822 | 0.6776 | 0.7318 | 0.8504 |
| | ER | 0.8270 | 0.8666 | 0.8366 | |
| | Extracellular | 0.8482 | 0.9692 | 0.8374 | |
| | Golgi | 0.8830 | 0.4511 | 0.6267 | |
| | Lysosome | 0.8872 | 0.7239 | 0.7985 | |
| | Mitochondrion | 0.8701 | 0.7631 | 0.7222 | |
| | Nucleus | 0.8311 | 0.8755 | 0.7964 | |
| | Plasma membrane | 0.9123 | 0.4685 | 0.6513 | |
| | Peroxisome | 0.9412 | 0.5926 | 0.7448 | |
| Ensemble 3 (SW+NW+ HMM+PC) | Cytoplasm | 0.8963 | 0.7064 | 0.7584 | 0.8691 |
| | ER | 0.9554 | 0.7414 | 0.8333 | |
| | Extracellular | 0.8649 | 0.9902 | 0.8711 | |
| | Golgi | 1.0000 | 0.3587 | 0.5954 | |
| | Lysosome | 0.9848 | 0.3988 | 0.6235 | |
| | Mitochondrion | 0.9181 | 0.8549 | 0.8715 | |
| | Nucleus | 0.8253 | 0.9126 | 0.8166 | |
| | Plasma membrane | 0.9423 | 0.4414 | 0.6427 | |
| | Peroxisome | 0.9531 | 0.5648 | 0.7317 | |
| Ensemble 4 (SW+NW+ HMM+DC) | Cytoplasm | 0.8822 | 0.7548 | 0.7845 | 0.8815 |
| | ER | 0.9500 | 0.7512 | 0.8367 | |
| | Extracellular | 0.8756 | 0.9907 | 0.8822 | |
| | Golgi | 1.0000 | 0.3750 | 0.6090 | |
| | Lysosome | 0.9789 | 0.5706 | 0.7446 | |
| | Mitochondrion | 0.9373 | 0.8582 | 0.8842 | |
| | Nucleus | 0.8483 | 0.9129 | 0.8345 | |
| | Plasma membrane | 0.9455 | 0.4685 | 0.6633 | |
| | Peroxisome | 0.9701 | 0.6019 | 0.7624 | |
| Ensemble 5 (SW+NW+ HMM+PC+ DC) | Cytoplasm | 0.8727 | 0.7578 | 0.7770 | 0.8853 |
| | ER | 0.9498 | 0.8105 | 0.8707 | |
| | Extracellular | 0.8844 | 0.9871 | 0.8876 | |
| | Golgi | 0.9877 | 0.4348 | 0.6519 | |
| | Lysosome | 0.9798 | 0.5951 | 0.7610 | |
| | Mitochondrion | 0.9342 | 0.8615 | 0.8844 | |
| | Nucleus | 0.8565 | 0.9116 | 0.8398 | |
| | Plasma membrane | 0.9474 | 0.4865 | 0.6767 | |
| | Peroxisome | 0.9706 | 0.6111 | 0.7684 | |

HMM: N-terminal profile HMM, NW: N-terminal Needleman-Wunsch algorithm, SW: Full sequence Smith-Waterman algorithm, PC: Full sequence physico-chemical properties, DC: Full sequence dipeptide composition

Table 9
The comparison of the average sensitivity for feature extraction methods based on N-terminal sequences and full sequences, with the targeting information.

| Location | Targeting info. | N-terminal | Full sequence |
|---|---|---|---|
| Cytoplasm | No | 0.6949 | 0.7554 |
| ER | ERS+$\alpha$ | 0.6607 | 0.8094 |
| Extracellular | ERS | 0.9530 | 0.9382 |
| Golgi | ERS+$\alpha$ | 0.3233 | 0.4330 |
| Lysosome | ERS+$\alpha$ | 0.3006 | 0.6728 |
| Mitochondrion | MS | 0.7869 | 0.7681 |
| Nucleus | NLS | 0.8977 | 0.8365 |
| Plasma membrane | ERS+$\alpha$ | 0.3108 | 0.4535 |
| Peroxisome | PS | 0.5093 | 0.5371 |

No: No targeting information, ERS: ER signal sequence (N-terminal), MS: Mitochondrion signal sequence (N-terminal), NLS: Nuclear localization signals (anywhere), PS: Peroxisome signal sequence (N-terminal or C-terminal), $\alpha$: additional targeting information, N-terminal: average sensitivity of profile HMM and Needleman-Wunsch algorithm, Full sequence: average sensitivity of dipeptide composition, physico-chemical properties, and Smith-Waterman algorithm