PROTEINS
STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

# PSLDoc: Protein subcellular localization prediction based on gapped-dipeptides and probabilistic latent semantic analysis

**Jia-Ming Chang,[1] Emily Chia-Yu Su,[2,3] Allan Lo,[2,4] Hua-Sheng Chiu,[1] Ting-Yi Sung,[1*] and Wen-Lian Hsu[1*]**

[1] Bioinformatics Lab, Institute of Information Science, Academia Sinica, Taipei, Taiwan

[2] Bioinformatics Program, Taiwan International Graduate Program, Academia Sinica, Taipei, Taiwan

[3] Institute of Bioinformatics, National Chiao Tung University, Hsinchu, Taiwan

[4] Department of Life Sciences, National Tsing Hua University, Hsinchu, Taiwan

*ABSTRACT*

***Prediction of protein subcellular localization (PSL) is important for genome annotation, protein function prediction, and drug discovery. Many computational approaches for PSL prediction based on protein sequences have been proposed in recent years for Gram-negative bacteria. We present PSLDoc, a method based on gapped-dipeptides and probabilistic latent semantic analysis (PLSA) to solve this problem. A protein is considered as a term string composed by gapped-dipeptides, which are defined as any two residues separated by one or more positions. The weighting scheme of gapped-dipeptides is calculated according to a position specific score matrix, which includes sequence evolutionary information. Then, PLSA is applied for feature reduction, and reduced vectors are input to five one-versus-rest support vector machine classifiers. The localization site with the highest probability is assigned as the final prediction. It has been reported that there is a strong correlation between sequence homology and subcellular localization (Nair and Rost, Protein Sci 2002;11:2836–2847; Yu et al., Proteins 2006;64:643–651). To properly evaluate the performance of PSLDoc, a target protein can be classified into low- or high-homology data sets. PSLDoc's overall accuracy of low- and high-homology data sets reaches 86.84% and 98.21%, respectively, and it compares favorably with that of CELLO II (Yu et al., Proteins 2006;64:643–651). In addition, we set a confidence threshold to achieve a high precision at specified levels of recall rates. When the confidence threshold is set at 0.7, PSLDoc achieves 97.89% in precision which is considerably better than that of PSORTb v.2.0 (Gardy et al., Bioinformatics 2005;21:617–623). Our approach demonstrates that the specific feature representation for proteins can be successfully applied to the prediction of protein subcellular localization and improves prediction accuracy. Besides, because of the generality of the representation, our method can be extended to eukaryotic proteomes in the future. The web server of PSLDoc is publicly available at http://bio-cluster.iis.sinica.edu.tw/~bioapp/PSLDoc/.***

Proteins 2008; 72:693–710.


***Key words:** protein subcellular localization; document classification; vector space model; gapped-dipeptides; probabilistic latent semantic analysis; support vector machines.*

## INTRODUCTION

### Protein subcellular localization prediction

#### *Background*

Predicting protein subcellular localization (PSL) is key to elucidating many biological problems, such as protein function prediction, genome annotation and drug discovery. The task is to assign a protein to one or more localization sites corresponding to the subcellular compartments based on its sequence. Recently, many prediction methods for Gram-negative bacteria have been developed using different computational techniques, including expert system,[1] *k*-nearest neighbors,[2] artificial neural networks,[3,4] support vector machines (SVM),[5,6–11] and Bayesian networks.[12–14] Among them, PSORTb v.2.0[12] (updated from PSORTb v.1.1[13]) and CELLO II[5] (updated from CELLO[11]) have been tested on a new Gram-negative bacteria data set.[15] PSORTb v.1.1,

Grant sponsor: Thematic Program of Academia Sinica and National Science Council of Taiwan; Grant numbers: AS94B003, AS95ASIA02, NSC 95-3114-P-002-005-Y.
*Correspondence to: Wen-Lian Hsu, Institute of Information Science, Academia Sinica, 128 Academia Road, Section 2, Nankang, Taipei, Taiwan, ROC.
E-mail: hsu@iis.sinica.edu.tw; or Ting-Yi Sung, Institute of Information Science, Academia Sinica, 128 Academia Road, Section 2, Nankang, Taipei, Taiwan, ROC.
E-mail: tsung@iis.sinica.edu.tw
Received 23 April 2007; Revised 7 November 2007; Accepted 30 November 2007
Published online 7 February 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/prot.21944

released in 2003, integrates homology analyses, identification of sorting signals and other motifs, and machine learning methods into an expert system based on a Bayesian network to decide the final prediction. PSORTb v.2.0, released in 2005, uses SVM as the underlying machine learning model and takes frequent subsequences occurring in proteins as input features. CELLO also uses SVM trained by multiple feature vectors derived from *n*-peptide compositions. The updated CELLO II is based on a two-level SVM system: the first-level SVM is comprised a number of SVM classifiers using different feature vectors, and each classifier generates a probability distribution of subcellular localization sites; the second-level SVM is considered as a jury SVM that yields a final probability distribution based on those generated in the previous stage and determines the final prediction as the site of the highest probability. The authors of CELLO II also classify a query protein, whose localization site is to be predicted, into low- or high-homology data sets depending on its highest pairwise sequence identity with the training data set whether it is below or above a similarity threshold of 30%. This classification of data is motivated by an observation that sequence homology and subcellular localization have strong correlation when the sequence identity is higher than 30%. Hence, they also propose a hybrid method, called HYBRID,[5] which uses the two-level SVM system for low-homology proteins and a homology search method for high-homology proteins.

### Document classification approach

In this article, we formulate PSL prediction as a document classification problem. The document classification problem is to assign an electronic document to one or more categories, based on its contents. A protein sequence can be considered as the content of a document, and localization sites are considered as categories. To predict the localization site(s) of a protein is equivalent to predicting the category (e.g., sport, politics) of a document (e.g., a piece of news). This transformation is intuitive. Document classification methods have been successfully applied in many protein classification problems, such as protein function prediction[16] and protein family classification.[17] King and Guda showed that using document classification techniques on the primary sequence can achieve good results on estimating subcellular proteomes of eukaryotes.[18]

Given a large number of documents, document classification is usually tackled by the following three steps. First, documents have to be transformed into feature vectors in which each distinct term corresponds to a feature. The value of a feature in a vector represents the weight of a term in a document. Another set of documents with known categories is used as a training set. Second, because of high-dimensional feature spaces, feature reduction is necessary before applying machine learning methods, to improve generalization accuracy[19] and to avoid overfitting.[19,20] The first two steps could be considered as *feature representation*. Finally, these reduced feature vectors are used to perform the category assignment automatically.

In this article, we propose a specific feature representation embedded in a prediction system called PSLDoc (Protein Subcellular Localization prediction based on modified Document classification method), which uses SVM as the underlying machine learning model. The design of PSLDoc's feature representation includes the following tasks: (1) define the terms of a protein; (2) design a term weighting scheme; and (3) apply a feature reduction and extraction method.

For a benchmark data set of Gram-negative bacteria,[15] PSLDoc performs better than HYBRID and PSORTb v.2.0. Our approach demonstrates that the specific feature representation for proteins can be successfully applied to PSL prediction.

## A baseline system using TFIDF

Before describing our method, we introduce a baseline system for performance comparison that uses a traditional document classification method. Salton's vector space model (VSM) is one of the most widely used methods for ad hoc retrieval in document classification.[21] Each document is represented by a feature vector (vector, in short) composed of all terms in a collection of documents, where each entry (or feature) of the vector corresponds to a term and its value is given by the weight of the term in the document.[22] The similarity between two documents $d$ and $q$, denoted by $sim(d,q)$, can be defined as the cosine of the angle between their vectors, called *cosine similarity*, as shown below:

$$sim(d,q) = \cos\left(\angle(\vec{d},\vec{q})\right) = \frac{\vec{d}\cdot\vec{q}}{|\vec{d}||\vec{q}|} \qquad (1)$$

where $\vec{d}$ denotes the vector for a document $d$. Given a collection of documents with known categories, we classify a document with unknown category (called *query document*) into the same category as the document whose cosine similarity with the query document is the largest. We refer to this prediction method as the *1-Nearest Neighboring* (1-NN) method based on cosine similarity. The advantage of the 1-NN method is that there is no training required as in general machine learning approach.

Weighting scheme, that is, determining the weight of each entry in a vector, is crucial in document classification. In this baseline system, we use *term frequency–inverse document frequency* (TFIDF) as the weighting scheme. For a term $t_i$ in a document $d$, a simple *term frequency* (TF) is the number of $t_i$'s occurrences in the

document, denoted as $n_i$. However, to prevent a bias toward longer documents, term frequency $tf(t_i,d)$ is usually normalized as follows:

$$tf(t_i, d) = \frac{n_i}{\sum_k n_k}, \quad (2)$$

where the denominator is the number of occurrences of all terms. The term frequency $tf(t_i,d)$ gives a measure of the importance of the term $t_i$ in the document $d$. The higher the term frequency, the more likely the term is a good description of the content of the document. In contrast, *inverse document frequency* (IDF) of $t_i$ is a measure of the general importance of the term. A semantically important term will often occur several times in a document if it occurs at all. However, semantically unimportant terms are spread out homogeneously over all documents. A frequently used IDF for $t_i$, $idf(t_i)$, is defined as follows:

$$idf(t_i) = \log \frac{|D|}{|(d_i \supset t_i)|}, \quad (3)$$

where $|D|$ is the number of documents in the collection, and $|(d_i \supset t_i)|$ denotes the number of documents in which $t_i$ appears. In the TFIDF scheme, the weight of the term $t_i$ in a document $d$, $W(t_i, d)$,* equals to $tf(t_i, d)$ multiplied by $idf(t_i)$.[23] The values in a vector are normalized to (0–1] by dividing the maximum value in the vector.

## METHODS

PSLDoc uses gapped-dipeptides[24] as the terms of a protein and calculates their weights according to a position specific score matrix (PSSM) instead of the TFIDF used in the baseline system. Probabilistic latent semantic analysis (PLSA) is used for feature reduction to improve learning efficiency and accuracy. The reduced feature vectors are input to five one-versus-rest (1-v-r) SVM classifiers corresponding to five localization sites. The probability estimated by a classifier can be considered as the confidence level of a target protein belonging to the corresponding localization site. The final prediction is determined to be the localization site whose corresponding classifier outputs the largest confidence score.

### Gapped-dipeptides as the terms of proteins

When considering proteins as documents, many different types of terms have been proposed, including single amino acid (AA)[3,4,7,9,25–27] as a uni-gram descriptor, and the general $n$-peptide,[11] that is, peptides of length $n$ without gaps. In particular, for $n = 2$, dipeptide (Dip) is a neighboring bi-gram descriptor. However, AA and Dip cannot represent information between two gapped amino acids. The use of $n$-peptide to capture long distance amino acid information will result in a high-dimensional vector space. For example, the feature number of a vector is 3,200,000 ($= 20^5$), when $n$ equals five. "Gapped amino acid pair" was first proposed by Park and Kanehisa[9] for protein representation. Later, Liang *et al.*[24] proposed a method based on a similar encoding scheme, called amino acid-coupling patterns, to extract the information from a protein sequence; the encoding scheme works well on distinguishing thermophilic proteins. An amino acid-coupling pattern $XdZ$ denotes the peptides of length $d + 2$ such that amino acids $X$ and $Z$ are separated by $d$ amino acids, where $d$ can be negative depending on whether the position of $X$ is closer to N-terminus or C-terminus.[24]

We adopt the same encoding scheme as in Liang *et al.* except with nonnegative $d$ as the term of a protein sequence regardless whether the pattern appears near the N-terminus or C-terminus. We call such amino acid-coupling pattern as *gapped-dipeptides*. For example, the gapped-dipeptides for $d = 0$ are dipeptide without gaps (Dip's). Given a positive integer $l$ as the upper bound of gapped distance, each protein sequence is represented by a vector in the space of gapped-dipeptides with each feature given by $XdZ$ for $0 \leq d \leq l$. The length of vectors is the number of all possible combinations of gapped-dipeptides, that is, $(l + 1) \times 20 \times 20$. For example, given $l = 10$, a protein is represented as a feature vector of 4400 ($= 11 \times 20 \times 20$) features.

### Term weighting—position specific score matrix information

#### Motivation

On the basis of the finding in a previous work that sequence identity and subcellular localizations of proteins have a strong correlation,[28] Yu *et al.*[5] proposed a homology search method for PSL prediction, which predicted the localization site of a query protein by the most similar protein among the aligned protein sequences with known localization sites generated by the global alignment program ALIGN.[29] The authors observed that, when the query protein and its most similar protein with known localization site have sequence identity over 30%, the homology search method performed very well with 97.7% accuracy. But the prediction performance dropped significantly when the sequence identity is under 20%. In this case, it would be difficult to predict the localization site of a query protein based on the sequence identity or sequence information. To overcome this difficulty, we borrow the idea from protein secondary structure prediction, in which homologous sequences are usually removed from the testing and training data sets.[30–35]

*In this paper, we use the weights of the terms in a document and in a vector, denoted by $W(t_i, d)$ and $W(t_i, \vec{d})$, interchangeably.

Most of the prediction methods address the problem of weak homology by utilizing sequence evolutionary information. One widely used representation of evolutionary information is the PSSM generated by PSI-BLAST,[36] which has been used in PSIPRED,[37] a very popular secondary structure prediction method. PSI-BLAST finds remote homologues to a query protein from a chosen sequence database (e.g., NCBI nr[38]). Instead of TFIDF based on the sequence information, our weighting scheme is based on PSSM.

### Position specific score matrix

The PSSM of a sequence $S$ of length $n$ is represented by an $n \times 20$ matrix, in which the $n$ rows correspond to the amino acid sequence of $S$ and the columns correspond to the 20 distinct amino acids. Each row of a PSSM represents the log-likelihood of the residue substitutions at the corresponding position in $S$.[36] The PSSM elements are normalized to the range from 0 to 1 using the following sigmoid function[32]:

$$f(x) = \frac{1}{1 + e^{-x}}, \quad (4)$$

where $x$ is the original PSSM value. The higher the normalized value of the residue is, the higher it is for the propensity of the residue in this position. In PSLDoc, the PSI-BLAST's parameters were set to $j = 5$ (five iterations), $e = 10^{-2}$ ($E$-value $< 0.01$), and the sequence database was NCBI nr which contains 3,747,820 sequences.

### TFPSSM weighting scheme

We design a term weighting scheme based on PSSM, denoted by TFPSSM as follows. Given a protein sequence $S$ of length $n$, any gapped-dipeptide $XdZ$ of $S$ has PSSM entries corresponding to gapped-dipeptides $S(i)dS(i + d + 1)$ for $1 \leq i \leq n - (d + 1)$, where $S(i)$ denotes the $i$th amino acid of $S$. For example, the PSSM (with original value without normalization) of the sequence MPLDLYNTLT is shown in Figure 1. From the sequence information, M2D only occurs once. However, in view of PSSM, M2D may occur in the corresponding gapped-dipeptides obtained from the sequence, that is, M2D, P2L, L2Y, D2N, L2T, Y2L, N2T. We define the weight of $XdZ$ in $S$ as

$$W(XdZ, S) = \sum_{1 \leq i \leq n-(d+1)} f(i, X) \times f(i + d + 1, Z) \quad (5)$$

where $f(i,Y)$ denotes the normalized value of the PSSM entry at the $i$th row and the column corresponding to amino acid $Y$. In the above example, the weight of M2D based on PSSM is given by $f(1,\mathrm{M}) \times f(4,\mathrm{D}) + f(2,\mathrm{M}) \times f(5,\mathrm{D}) + \cdots + f(7,\mathrm{M}) \times f(10,\mathrm{D}) = 0.99995 \times 0.04743 + 0.11920 \times 0.00247 + \cdots + 0.00669 \times 0.26894$. It is unnecessary to incorporate IDF with term weighting based on PSSM because the term occurs in all documents based on PSSM.

| | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 M | -3 | -3 | -4 | -5 | -3 | -3 | -4 | -5 | -4 | 0 | 1 | -3 | 10 | -2 | -5 | -4 | -3 | -4 | -3 | -1 |
| 2 P | 2 | -3 | -3 | -1 | -3 | -1 | -1 | -1 | -4 | -2 | -4 | -2 | -2 | -5 | 4 | 2 | 4 | -5 | -4 | -3 |
| 3 L | -4 | -5 | -6 | -6 | -4 | -3 | -5 | -6 | -5 | 3 | 5 | -5 | 4 | 0 | -5 | -5 | -3 | -4 | -3 | 2 |
| 4 D | -2 | 5 | -1 | -3 | -4 | 2 | -1 | -4 | 2 | -5 | -3 | 5 | -2 | -2 | -4 | -2 | 0 | -1 | 0 | -3 |
| 5 L | -4 | -5 | -6 | -6 | -4 | -5 | -6 | -6 | -4 | 4 | 4 | -5 | 0 | 1 | -5 | -5 | -3 | -4 | -3 | 3 |
| 6 Y | -4 | -3 | -3 | -5 | -5 | -3 | -4 | -5 | 4 | -4 | -3 | -3 | -2 | 4 | -5 | -3 | -2 | 2 | 8 | -4 |
| 7 N | -4 | -3 | 8 | 4 | -6 | -3 | -2 | -3 | -2 | -6 | -6 | -3 | -5 | -6 | -4 | -1 | -3 | -7 | -5 | -6 |
| 8 T | -2 | -3 | -1 | -3 | -1 | -3 | -3 | -4 | -3 | -4 | -4 | -1 | -4 | -4 | -4 | 4 | 6 | -5 | -4 | -2 |
| 9 L | 0 | -1 | -5 | -5 | -4 | -3 | -4 | -4 | -3 | -1 | 5 | -3 | 3 | 0 | -4 | -3 | -3 | -3 | -2 | -1 |
| 10 T | -1 | -3 | -1 | -1 | -4 | -2 | -3 | -2 | -1 | -4 | -3 | -1 | -3 | -4 | -4 | 3 | 6 | -5 | -4 | -3 |

**Figure 1**

*PSSM of the sequence MPLDLYNTL, where each entry is the original value without normalization.*

As mentioned before, each protein is represented by a vector, and each entry of the vector is given by TFPSSM of the corresponding gapped-dipeptides. Note that the values in each feature vector are normalized between 0 and 1 by dividing the maximum value in the vector.

## Feature reduction—probabilistic latent semantic analysis

### Motivation

There are some limitations of the VSM for document classification. First, the vector space is high-dimensional.[21] Training and testing have to deal with the curse of dimensionality. Second, document vectors are typically very sparse, that is, most features of a vector are zeros that are susceptible to noise,[39] and cosine similarity could be inaccurate. Finally, the inner product defining document similarity can only match occurrences of the same terms. As a result, the vector representation does not capture semantic relations between terms. Furthermore, this representation, which considers a document as a bag of words, is unable to capture phrases and semantic/syntactic regularities.

Hence, dimension reduction (feature reduction) is proposed for dealing with the above limitations. The task of dimension reduction is to map similar terms to a similar location in a low dimensional space called *latent semantic space*, which reflects semantic associations. A frequently used dimension reduction technique is Latent Semantic Analysis (LSA) (or called Latent Semantic Indexing in some papers), which uses singular value decomposition (SVD) to do data mapping.[40] The document similarity based on the inner product is computed on the latent semantic space. Empirically, there are advantages of SVD

over naive VSM. However, SVD still has the following disadvantages.[41] First, the resulting dimensions might be difficult to interpret. For instance, the size of a vector is reduced from three to two by LSA as shown below:

$$\{(\text{A0A}), (\text{A1A}), (\text{G0G})\} \rightarrow \{(1.3^{*}\text{A0A} + 0.2^{*}\text{G0G}), (\text{A1A})\}$$

The value of the first reduced feature equals 1.3 multiplied by the value of the origanl first feature plus 0.2 multiplied by the value of the original third feature. This leads to results which might be justifiable on the mathematical level, but have no interpretable meaning in the original application. Second, the probabilistic model of LSA does not match observed data.[41] Third, the reconstruction may contain negative entries, which are inappropriate as a distance function for count vectors.

### Probabilistic latent semantic analysis

Hofmann proposed probabilistic latent semantic analysis (PLSA) based on an aspect model to deal with those above disadvantages.[41] The aspect model is a latent variable model for co-occurrence data (i.e., documents and terms) that each observation is associated with an unobserved class variable $z \in Z = \{z_1, \ldots, z_K\}$. The weight of the term $w$ in a document $d$, $W(w, d)$, is considered as a joint probability $P(w, d)$ between $w$ and $d$, which is modeled by $z$, a latent variable which can be loosely thought of as a topic or a reduced feature. Thus, the joint probability $P(w, d)$ based on PLSA model is

$$P(w, d) = P(d)P(w|d), P(w|d) = \sum_{z \in Z} P(w|z)P(z|d)^{\dagger} \quad (6)$$

where $P(w|z)$ denotes the topic-conditional probability of a term conditioned on the unobserved topic, and $P(z|d)$ denotes a document-specific probability distribution over the latent variable space; that is, considering a vector $\vec{d}$ in latent variable space, $P(z|d)$ denotes the weight of the latent variable $z$ of the document $d$. Hence, a vector is mapped from the term space to latent space and its size is reduced from $|W|$ to $|Z|$.

### PLSA model fitting (training)

A PLSA model is parameterized by $P(w|z)$ and $P(z|d)$ which are estimated by fitting $P(w, d)$ to a training corpus $D$ with known $W(w, d)$. The fitting process is obtained by maximizing the log-likelihood function $L$ given below[41]:

$$L = \sum_{w \in d} \sum_{d \in D} W(w, d) \log P(w, d) \quad (7)$$

The parameters of a PLSA model, $P(w|z)$ and $P(z|d)$, are estimated using the iterative Expectation-Maximization (EM) algorithm by maximizing the log-likelihood function $L$. $P(w|z)$ and $P(z|d)$ are initialized by random values in (0,1)-range. Then, the EM procedure iterates between the E-step and the M-step. In the E-step, the probability that a term $w$ in a particular document $d$ explained by the class corresponding to $z$, is estimated as

$$P(z|w, d) = \frac{P(z, w, d)}{P(w, d)} \quad (8)$$

$$P(z, w, d) = P(d)P(z|d)P(w|z)^{\ddagger} \quad (9)$$

Using Eqs. (6), (8), and (9), we can get

$$P(z|w, d) = \frac{P(d)P(z|d)P(w|z)}{P(d)\sum_{z'} P(w|z')P(z'|d)} = \frac{P(w|z)P(z|d)}{\sum_{z'} P(w|z')P(z'|d)} \quad (10)$$

In the M-step, we calculate

$$P(w \mid z) = \frac{\sum_d W(w, d)P(z|w, d)}{\sum_{w'} \sum_d W(w', d)P(z|w', d)}$$
$$P(z \mid d) = \frac{\sum_w W(w, d)P(z|w, d)}{\sum_{z'} \sum_w W(w, d)P(z'|w, d)} \quad (11)$$

where parameters $P(w|z)$ and $P(z|d)$ are re-estimated to maximize $L$.

### PLSA model testing

After training, the estimated $P(w|z)$ parameters are used to estimate $P(z|q)$ for a new (test) document $q$ through a *folding-in* process.[41] In the folding-in process, EM procedure runs in a similar manner to the training stage. The E-step is identical but the M-step keeps all the $P(w|z)$ constant and only recalculates $P(z|q)$. Usually, a very small number of iterations of the EM algorithm are sufficient for folding-in process.

### Feature reduction by PLSA

We apply PLSA not only for feature reduction but also for gapped-dipeptide semantic relation extraction. Vectors are mapped from the gapped-dipeptide space to the latent semantic space. This will lead to improvement in learning performance and efficiency. Though it is not easy to determine an appropriate reduced feature size of PLSA, it can be approximated by the reduced feature size of LSA. To determine the reduced feature size of LSA, we calculate singular values of LSA and sort them in a decreasing order. Then, the reduced feature size of LSA equals to $n$ if the $n$-th largest singular value is close to zero.

[†]It is assumed that the distribution of terms given a class is conditionally independent of the document, that is, $P(w|z,d) = P(w|z)$.

[‡]This equation is derived from according to the Figure 1(a) of Hofmann.[39]

## The system architecture of PSLDoc

Prediction of PSL can be treated as a multiclass classification problem. For multiclass classification, the 1-v-r SVM model has demonstrated a good classification performance.[27] For each class $i$, we construct a 1-v-r ($C_i$ versus non-$C_i$) binary classifier. PSLDoc consists of five 1-v-r SVM classifiers corresponding to five localization sites in Gram-negative bacteria. Input features for all binary classifiers are the same. The SVM program LIBSVM[42] is used in PSLDoc, and it can generate probability estimates that are used for determining the confidence levels of classifications.[43] For all classifiers, we use the Radial Basis Function kernel, and tune the cost ($c$) and gamma ($\gamma$) parameters optimized by 10-fold crossvalidation on the training data set.

Given a protein, PSLDoc performs the following steps:

1. Use PSI-BLAST to generate PSSM of the protein.
2. Generate the feature vector of the protein, where each feature is defined as TFPSSM corresponding to a gapped-dipeptide.
3. Perform PLSA to generate a reduced feature vector, which will be input to each 1-v-r classifier.
4. Run five 1-v-r SVM classifiers.

In the training stage of PSLDoc, to train PLSA model with different topic sizes and the SVM classifiers, proteins with known localization sites are used to estimate $P(w|z)$ and $P(z|d)$, and the reduced vectors are used to determine the $c$ and $\gamma$ parameters of the RBF kernel of each classifier. In the testing stage of PSLDoc, Step 3 of PSLDoc performs PLSA folding-in process on trained $P(w|z)$. Step 4 of PSLDoc is performed on the trained SVM classifiers. The localization site of the protein is predicted as the class with the highest probability ($\text{prob}_i$: the confidence of the query protein predicted as class $i$; $0 \leq \text{prob}_i \leq 1$) generated from the five 1-v-r classifiers. The system architecture of PSLDoc is shown in Figure 2.

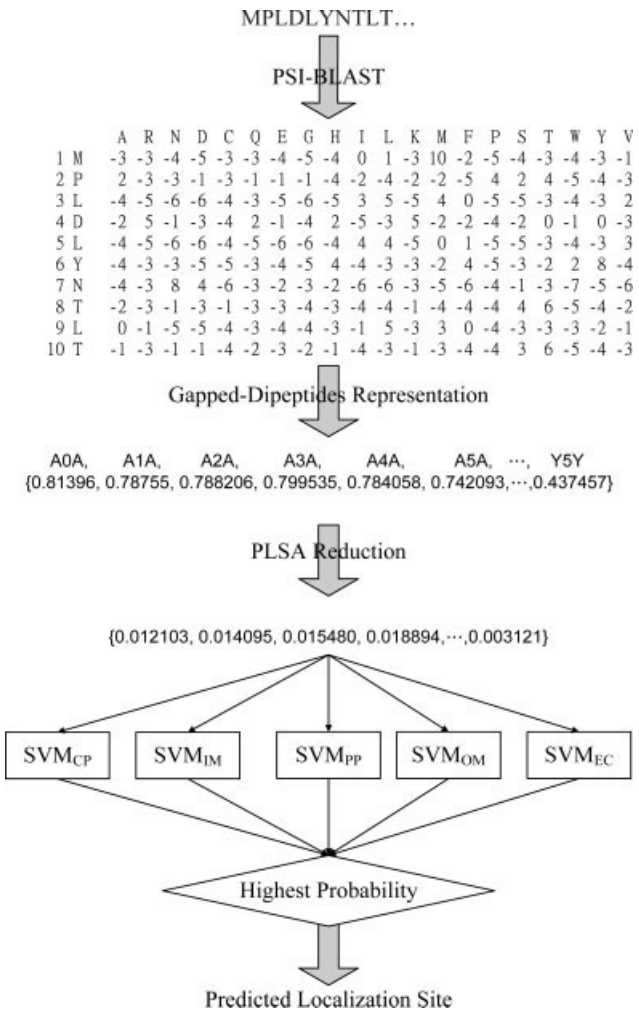


**Figure 2**

*System architecture of PSLDoc based on 1-v-r SVM models using reduced/transformed feature vectors.*

## Data sets

To evaluate the performance of PSLDoc, we utilize a benchmark data set of proteins from Gram-negative bacteria with single localization that have been used in previous works.[12,11] It consists of 1444 proteins with experimentally determined localizations, referred to as PS1444.[15] Table I lists the distribution of localization sites of the data set.

To analyze the performance of PSLDoc under the effect of sequence homology information, we further classify each protein in PS1444 into two data sets, the high- or low-homology data sets based on whether or not the protein's highest sequence identity of all-against-all alignment by ClustalW is greater than an identity threshold of 30%. The high-homology data set, referred to as PSHigh783, consists of 783 proteins and the low-homology set, referred to as PSLow661, consists of 661 proteins. The three data sets are available at http://biocluster.iis.sinica.edu.tw/~bioapp/PSLDoc/DataSet.htm.

## Evaluation measures

To evaluate the performance of our method, we follow the same measures used in previous works[1,10,11,13] for comparison with other approaches. These measures include accuracy (Acc), precision, recall, Matthew's correlation coefficient (MCC)[44] for five localization sites, and the overall accuracy defined in Eqs. (12)–(16) below:

$$\text{Acc}_i = \text{TP}_i / N_i \tag{12}$$

**Table I**
*Number of Proteins in Different Localization Sites*

| Localization sites | No. |
|---|---|
| Cytoplasmic (CP) | 278 |
| Inner membrane (IM) | 309 |
| Periplasmic (PP) | 276 |
| Outer membrane (OM) | 391 |
| Extracellular (EC) | 190 |
| All sites | 1444 |

$$\text{Precision}_i = \text{TP}_i/(\text{TP}_i + \text{FP}_i) \quad (13)$$

$$\text{Recall}_i = \text{TP}_i/(\text{TP}_i + \text{FN}_i) \quad (14)$$

$$\text{MCC}_i = \frac{(\text{TP}_i)(\text{TN}_i) - (\text{FP}_i)(\text{FN}_i)}{\sqrt{(\text{TP}_i + \text{FN}_i)(\text{TP}_i + \text{FP}_i)(\text{TN}_i + \text{FP}_i)(\text{TN}_i + \text{FN}_i)}} \quad (15)$$

$$\text{Acc} = \sum_{i=1}^{l} \text{TP}_i \Big/ \sum_{i=1}^{l} N_i, \quad (16)$$

where $l = 5$ is the number of localization sites, and $\text{TP}_i$, $\text{TN}_i$, $\text{FP}_i$, $\text{FN}_i$, and $N_i$ are the number of true positives, true negatives, false positives, false negatives, and proteins in localization site $i$, respectively. MCC considers both under- and over-predictions, and takes range from −1 to 1, where MCC = 1 indicates a perfect prediction; MCC = 0 indicates a completely random assignment; and MCC = −1 indicates a perfectly reverse correlation. The $\text{Acc}_i$ is the same as $\text{Recall}_i$ because $N_i$ equals to the sum of $\text{TP}_i$ and $\text{FN}_i$. We will use $\text{Acc}_i$ or $\text{Recall}_i$ interchangeably in the experiments depending on which method is compared.

## Five simple PSL prediction methods

To evaluate the benefit of each step in our document classification method, we propose two simple prediction methods: 1NN_TFIDF and 1NN_TFPSSM, which consist of different parts of PSLDoc. To further analyze the effect of the PSSM information generated from databases of different sizes, we propose two methods based on PSI-BLAST: 1NN_PSI-BLAST$_{ps}$ and 1NN_PSI-BLAST$_{nr}$. In addition, we also construct a homology search method, 1NN_ClustalW, which is similar to Yu *et al.*'s for comparison with PSLDoc.

### 1NN_TFIDF

1NN_TFIDF solely incorporates protein encoding scheme, the gapped-dipeptides of PSLDoc. The remaining steps are the same as the baseline system. That is, terms are weighted according to the TFIDF weighting scheme, and a query protein is predicted by 1-NN method based on cosine similarity.

### 1NN_TFPSSM

1NN_TFPSSM incorporates two parts of PSLDoc, the gapped-dipeptide encoding scheme and the TFPSSM weighting scheme. It predicts a query protein using 1-NN method based on cosine similarity.

### 1NN_PSI-BLAST$_{ps}$

1NN_PSI-BLAST$_{ps}$ performs two PSI-BLAST searches, one of which for generating a PSSM and the other for searching the most similar protein using the PSSM generated in the previous step. First, for each query protein, PSI-BLAST search is performed against the training data and its parameters are the same as those in PSLDoc. Then, 1NN_PSI-BLAST$_{ps}$ performs a one-run PSI-BLAST search (i.e., $j = 1$)[§] against the training data using the obtained PSSM.[¶] Finally, the localization site of the protein with the highest *e*-value is assigned as the predicted localization for the query protein. In a five-fold cross-validation, the PSSM information used in 1NN_PSI-BLAST$_{ps}$ is generated from a small database which consists of ∼1155 (=1444×4/5) sequences from PS1444.

### 1NN_PSI-BLAST$_{nr}$

Although 1NN_PSI-BLAST$_{ps}$ utilizes the PSSM information, the source database used is not as large as that of 1NN_TFIDF and PSLDoc. For fair comparison with 1NN_TFIDF and PSLDoc, we construct 1NN_PSI-BLAST$_{nr}$, which uses PSSM generated from the NCBI nr database. The only difference between 1NN_PSI-BLAST$_{nr}$ and 1NN_PSI-BLAST$_{ps}$ is the size of the databases searched in the first step, and the remaining steps are all the same, including the generation of PSSM, followed by performing a second PSI-BLAST search, and lastly, the prediction of the localization site of the query protein.

### 1NN_ClustalW

1NN_ClustalW differs from Yu *et al.*'s method only in the pairwise sequence alignment algorithm used, that is, ClustalW in the former and ALIGN in the latter. For a query protein, we calculate its pairwise sequence identities with the remaining proteins by performing 1-against-others pairwise sequence alignment. Then, the localization site of the query protein is predicted by the 1-NN method based on pairwise sequence identity, that is, its localization site is assigned as that of the protein whose pairwise sequence identity is highest.

[§]The parameters of *e*-vlaue are ignored because we want to find the most similar protein instead of constructing a PSSM.

[¶]Please refer to the last example on blastpgp's document for how to save a PSSM and perform PSI-BLAST search from the PSSM (http://biowulf.nih.gov/apps/blast/doc/blastpgp.html).

### Experiment design

We conduct the following experiments to evaluate the benefit of each step in our document classification model where the gapped distance upper bound, $l$, ranges from 3 to 15. We follow the same validation procedures for the performance measurement as those of the other approaches.[5,12] All experiments are carried out in five-fold cross-validation, that is, the data is equally divided into five parts. In each run, four folds are used for training and the remaining fold is used for testing. All reported results are average over the five folds. We have conducted the following six experiments:

#### *Experiment 1: comparison between 1NN_TFIDF and 1NN_TFPSSM on the PS1444, PSHigh783, and PSLow661 data sets*

The purpose of this experiment is to evaluate the benefit of using the TFPSSM weighting scheme because the simple 1NN prediction method can reflect the relation between performance and weighting schemes avoiding the effect of the prediction algorithm. The distribution of benefit among 1444 protein sequences is further analyzed by comparing their performance on PSHigh783 and PSLow661.

#### *Experiment 2: comparison among 1NN_TFPSSM, 1NN_ClustalW, 1NN_PSI-BLAST$_{ps}$, and 1NN_PSI-BLAST$_{nr}$ on the PSHigh783 and PSLow661 data sets*

To compare the effect of utilizing PSSM, we compare the performance of 1NN_TFPSSM, 1NN_ClustalW, 1NN_PSI-BLAST$_{ps}$, and 1NN_PSI-BLAST$_{nr}$. 1NN_ClustalW is based on a pairwise sequence alignment in which no PSSM information is incorporated. We further analyze the relationship between the effect of PSSM and the size of databases used in the construction of PSSM. Compared with 1NN_PSI-BLAST$_{ps}$, both 1NN_TFPSSM and 1NN_PSI-BLAST$_{nr}$ incorporate a larger database for PSSM construction. Finally, the comparison between 1NN_TFPSSM and 1NN_PSI-BLAST$_{nr}$ serves to highlight the benefit of gapped-dipeptide encoding scheme.

#### *Experiment 3: comparison between PSLDoc and PSLDoc$_{-PLSA}$ on the PS1444 data set*

PSLDoc$_{-PLSA}$ represents PSLDoc without PLSA, which simply applies SVM on the original feature vectors. The overall accuracies of PSLDoc and PSLDoc$_{-PLSA}$ are compared to evaluate the benefit of PLSA feature reduction for SVM learning.

#### *Experiment 4: comparison among PSLDoc, 1NN_TFPSSM, and 1NN_ClustalW on the PSHigh783 and PSLow661 data sets*

Using the PSHigh783 data set, we can verify whether PSLDoc can replace 1NN_ClustalW. Using PSLow661, we can investigate whether PSLDoc can improve 1NN_TFPSSM by applying PLSA and SVM classification. Hence, we could determine whether PSLDoc is suitable for both high- and low-homology data sets.

#### *Experiment 5: comparison among PSLDoc, HYBRID and PSORTb v.2.0 on the PS1444 data set*

We compare the performance of PSLDoc, HYBRID, and PSORTb v.2.0. Besides, we also assess the performance of PSLDoc using a three-way data split procedure,[45] which is commonly used in machine learning to prevent overestimation of the performance. The data set is randomly divided into three disjoint sets, that is, a training set for classifier learning, a validation set for feature selection and parameter tuning, and a test set for performance evaluation. Hence, for each run in the original five-fold cross-validation, we divide the training data set into four distinct sets: three for training, one for validation. Then, we select the gapped distance upper bound and PLSA reduced feature size based on the validation set instead of the test set. Then PSLDoc performance is evaluated under the selected parameters in the original five-fold cross-validation.

#### *Experiment 6: PSLDoc under different prediction thresholds versus PSORTb v.2.0 on the PS1444 data set*

The precision and recall of PSLDoc is evaluated under different prediction thresholds to compare with PSORTb v.2.0.

## RESULTS AND DISCUSSION

### Experimental results

#### *Experiment 1: the benefit of using the TFPSSM weighting scheme*

The overall accuracy of 1NN_TFIDF and 1NN_TFPSSM for each gapped distance are shown in Figure 3. The highest overall accuracy of 1NN_TFPSSM is 89.47% when $l$

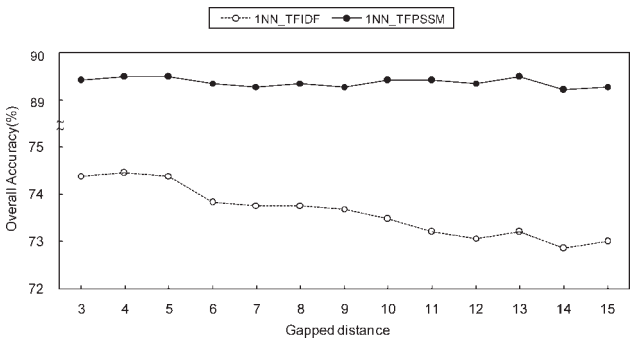


**Figure 3**

*Overall accuracy of 1NN_TFIDF and 1NN_TFPSSM with respect to maximum allowed gapped distances on the PS1444 data set.*

**Table II**
*Comparison of 1NN_TFIDF and 1NN_TFPSSM on the PSHigh783 and PSLow661 Data Sets*

| | PSHigh783 | | | | PSLow661 | | | |
|---|---|---|---|---|---|---|---|---|
| | 1NN_TFPSSM | | 1NN_TFIDF | | 1NN_TFPSSM | | 1NN_TFIDF | |
| Loc. sites | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* |
| CP | 94.20 | 0.96 | 71.01 | 0.74 | 83.25 | 0.77 | 41.15 | 0.36 |
| IM | 99.31 | 0.99 | 98.62 | 0.89 | 82.93 | 0.82 | 84.15 | 0.48 |
| PP | 95.86 | 0.94 | 86.21 | 0.89 | 74.05 | 0.63 | 38.17 | 0.46 |
| OM | 99.66 | 0.99 | 95.88 | 0.95 | 85.00 | 0.82 | 66.00 | 0.48 |
| EC | 96.99 | 0.96 | 92.48 | 0.91 | 57.89 | 0.51 | 28.07 | 0.26 |
| Overall | 97.96 | — | 91.83 | — | 79.43 | — | 53.86 | — |

equals 4, 5, and 13 and it is considerably higher than the best 1NN_TFIDF score 74.38% when $l$ equals to 4. Therefore, adopting the TFPSSM weighting scheme significantly improves the performance of 1NN_TFIDF.

The performance of 1NN_TFIDF and 1NN_TFPSSM in the high- and low-homology data sets is shown in Table II. 1NN_TFPSSM dramatically improves the performance of 1NN_TFIDF by about 26% in overall accuracy on PSLow661. Hence, the incorporation of PSSM information into the weighting scheme is useful for improving performance due to insufficient sequence information in the low-homology data set.

### *Experiment 2: the effect of incorporating PSSM information and gapped-dipeptide encoding scheme*

Table III shows the performance of 1NN_TFPSSM, 1NN_ClustalW, 1NN_PSI-BLAST$_{ps}$, and 1NN_PSI-BLAST$_{nr}$ on the PSHigh783 and PSLow661 data sets. The overall accuracy on the PSHigh783 data set is very similar for all methods. However, for the PSLow661 data set, 1NN_ClustalW, 1NN_PSI-BLAST$_{ps}$, and 1NN_PSI-BLAST$_{nr}$ attain 42.97%, 57.94%, and 66.57%, respectively, in overall accuracy. This result reveals that better performance can be achieved when a larger database is used in constructing PSSM. This also lends support to our assumption that incorporating more information into PSSM is more effective for the prediction of proteins with low sequence identity to the training set. Most notably, 1NN_TFPSSM outperforms 1NN_PSI-BLAST$_{nr}$ by 12.86% in overall accuracy. This suggests that the incorporation of PSSM based on gapped-dipeptide encoding scheme significantly improves the predictive performance, especially for proteins of low sequence identity.

### *Experiment 3: the benefit of PLSA feature reduction*

***Determine the reduced size of PLSA.*** The size of PLSA is determined by LSA singular values. Figure 4 shows the singular values in decreasing order on different gapped distances upper bound data sets.

The 40th largest singular value is close to zero in Figure 4, but in the inset the 160th largest singular value is close to zero. Hence, the reduced feature size of PLSA is set to 40, 80, and 160. However, we do not test larger PLSA reduced size or one-by-one PLSA reduced size in

**Table III**
*Comparison of 1NN_TFPSSM, 1NN_ClustalW, 1NN_PSI-BLAST$_{ps}$, and 1NN_PSI-BLAST$_{nr}$ for the PSHigh783 and PSLow661 Data Sets*

| | 1NN_TFPSSM | | 1NN_ClustalW | | 1NN_PSI-BLAST$_{ps}$ | | 1NN_PSI-BLAST$_{nr}$ | |
|---|---|---|---|---|---|---|---|---|
| Loc. sites | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* |
| PSHigh783 | | | | | | | | |
| CP | 94.20 | 0.96 | 91.3 | 0.90 | 88.41 | 0.92 | 86.96 | 0.90 |
| IM | 99.31 | 0.99 | 97.93 | 0.97 | 99.31 | 0.98 | 99.31 | 0.98 |
| PP | 95.86 | 0.94 | 93.1 | 0.93 | 93.79 | 0.93 | 92.41 | 0.91 |
| OM | 99.66 | 0.99 | 99.66 | 0.99 | 99.66 | 0.99 | 99.66 | 0.99 |
| EC | 96.99 | 0.96 | 99.25 | 0.99 | 98.50 | 0.98 | 98.50 | 0.98 |
| Overall | 97.96 | — | 97.32 | — | 97.32 | — | 96.93 | — |
| PSLow661 | | | | | | | | |
| CP | 83.25 | 0.77 | 39.23 | 0.23 | 36.84 | 0.40 | 55.50 | 0.53 |
| IM | 82.93 | 0.82 | 46.95 | 0.33 | 68.29 | 0.57 | 75.00 | 0.66 |
| PP | 74.05 | 0.63 | 41.98 | 0.44 | 59.54 | 0.51 | 64.12 | 0.54 |
| OM | 85.00 | 0.82 | 45.00 | 0.47 | 87.00 | 0.57 | 87.00 | 0.66 |
| EC | 57.89 | 0.51 | 43.86 | 0.10 | 50.88 | 0.37 | 52.63 | 0.45 |
| Overall | 79.43 | — | 42.97 | — | 57.94 | — | 66.57 | — |

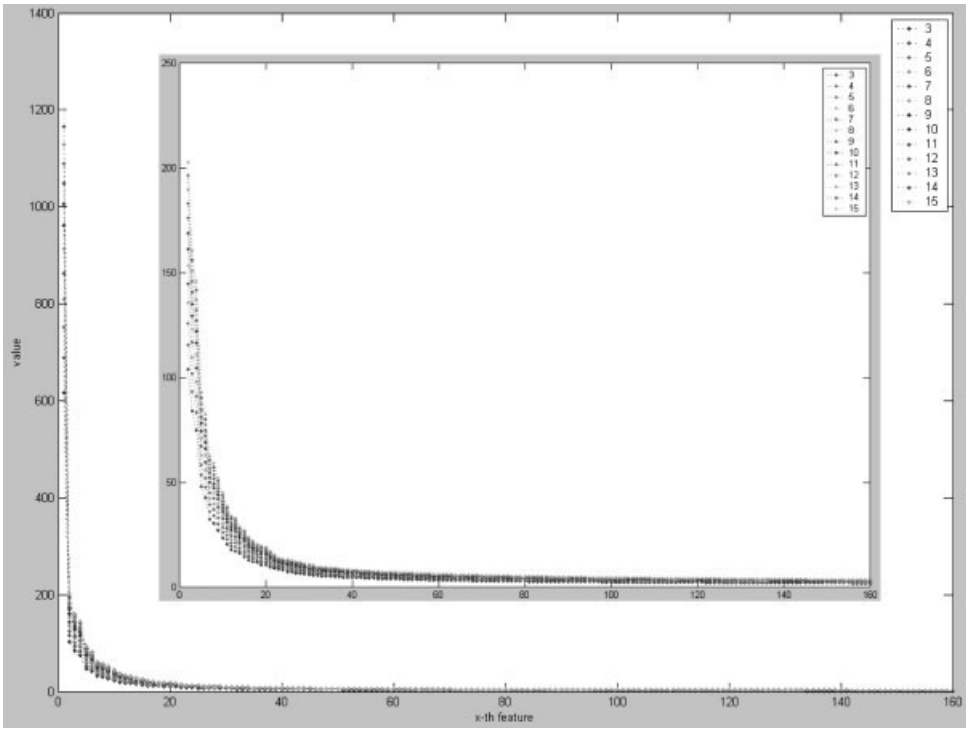


**Figure 4**

*Singular values in decreasing order of each gapped distance. The inset shows singular values without first largest one for detailed representation.*

consideration of the training efficiency and avoidance of data overfitting.

For one PLSA reduced size, the training and testing procedures of PSLDoc take 1.5 h and about 2–3 minutes for all gapped distances, respectively. However, PSLDoc$_{\text{-PLSA}}$ takes about 180 and 1.4 hours in training and testing, respectively. Figure 5 shows the performance of PSLDoc$_{\text{-PLSA}}$ and PSLDoc, where PSLDoc_F*x* denotes PSLDoc with PLSA reduced size *x*.

The highest overall accuracy among all gapped distances of PSLDoc_F40, PSLDoc_F80, and PSLDoc_F160 is 92.31%, 93.01%, and 92.52%, respectively, which is 0.83%, 1.52%, and 1.04% better than that of PSLDoc$_{\text{-PLSA}}$. Using PLSA not only improves learning efficiency but also performance. In the following experiments, PSLDoc takes the gapped distance 13 and PLSA at reduced size 80.

### *Experiment 4: the benefit of SVM and PLSA feature reduction*

Table IV shows the performance of PSLDoc, 1NN_TFPSSM, and 1NN_ClustalW on PSHigh783 and PSLow661. The overall accuracy of 1NN_ClustalW on PSHigh783 (97.32%) is very similar to that of Yu *et al.*'s (97.7%). 1NN_TFPSSM and PSLDoc perform better than 1NN_ClustalW on PSHigh783. On the other hand, PSLDoc improves 1NN_TFPSSM on PSLow661 by 7.41% because of the nonlinear SVM classification and PLSA feature reduction and extraction. This shows that PSLDoc is suitable for both the high- and low-homology data sets.

### *Experiment 5: comparison of PSLDoc, HYBRID, and PSORTb v.2.0*

Table V shows the performance of PSLDoc, HYBRID, and PSORTb v2.0 on PS1444. PSLDoc achieves the best

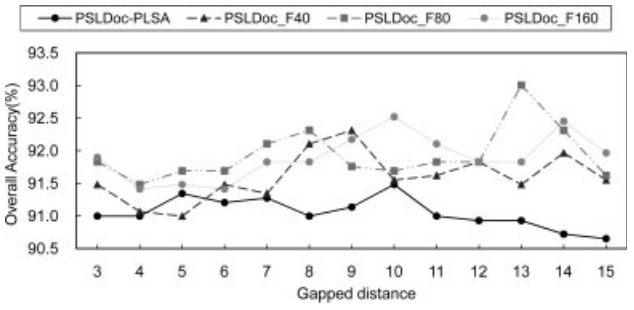


**Figure 5**

*Overall accuracy of PSLDoc_F40, PSLDoc_F80, PSLDoc_F160, and PSLDoc-$_{PLSA}$ with respect to gapped distance on the PS1444 data set.*

**Table IV**
*Comparison of PSLDoc, 1NN_TFPSSM, and 1NN_ClustalW for the PSHigh783 and PSLow661 Data Sets*

| | PSHigh783 | | | | | | PSLow661 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSLDoc | | 1NN_TFPSSM | | 1NN_ClustalW | | PSLDoc | | 1NN_TFPSSM | | 1NN_ClustalW | |
| Loc. sites | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* |
| CP | 95.65 | 0.96 | 94.20 | 0.96 | 91.3 | 0.89 | 94.74 | 0.88 | 83.25 | 0.77 | 39.23 | 0.23 |
| IM | 99.31 | 0.99 | 99.31 | 0.99 | 97.93 | 0.97 | 87.80 | 0.88 | 82.93 | 0.82 | 46.95 | 0.33 |
| PP | 95.17 | 0.94 | 95.86 | 0.94 | 93.1 | 0.93 | 82.44 | 0.78 | 74.05 | 0.63 | 41.98 | 0.44 |
| OM | 99.66 | 0.99 | 99.66 | 0.99 | 99.66 | 0.99 | 84.00 | 0.84 | 85.00 | 0.82 | 45.00 | 0.47 |
| EC | 98.50 | 0.98 | 96.99 | 0.96 | 99.25 | 0.99 | 70.18 | 0.65 | 57.89 | 0.51 | 43.86 | 0.10 |
| Overall | 98.21 | — | 97.96 | — | 97.32 | — | 86.84 | — | 79.43 | — | 42.97 | — |

performance of 93.01%, better than HYBIRD of 91.6% and PSORTb of 82.6%.

### *Experiment 6: PSLDoc under different prediction thresholds versus PSORTb v.2.0 on the PS1444 data set*

***Prediction confidence.*** The probability estimated by LIBSVM is used for determining the confidence levels of classifications. The class with the largest probability is chosen as the final predicted class. The confidence of the final predicted class, *prediction confidence,*[32] could be defined as the value of the largest probability minus the second largest probability. Figure 6 shows the relationship between accuracy and prediction confidence. For proteins with prediction confidence in the range 0.9–1, the prediction accuracy is near 100% (99.12%).

***Prediction threshold.*** Gardy *et al.* suggested that when a prediction system is unable to generate a confident prediction, the program outputs a result of "Unknown" because biologists usually prefer correct predictions (high precision) over prediction coverage (recall).[12] To provide prediction results with higher precision, we determine a *prediction threshold* to filter out prediction results with low confidence. That is, the SVM classifier predicts results only when the prediction confidence is above the threshold, otherwise the SVM classifier will output Unknown.[12,13] The recall and precision for each prediction threshold are shown in Figure 7.

Table VI shows the performance of PSLDoc under different prediction thresholds. Setting the prediction threshold to 0.7, PSLDoc achieves slightly better recall than PSORTb v.2.0 (83.66% vs. 82.6%), whereas the precision of PSLDoc is better than PSORTb v.2.0 (97.89% vs. 95.8%). In addition, when the prediction threshold is set to 0.3, PSLDoc achieves comparable precision to PSORTb v.2.0 (95.77% vs. 95.8%), and PSLDoc's recall is much better than that of PSORTb v.2.0 (89.27% vs. 82.6%).

## DISCUSSION

In PLSA, we associate proteins and gapped-dipeptides with topics. Through analyzing the trained PLSA model with $P(w|z)$ and $P(z|d)$ for gapped-dipeptide $w$, topic $z$ and protein $d$, gapped-dipeptide signatures in proteins with different localization sites are discovered for the PS1444 data set. Some of these signatures have been reported in the literature as motifs critical for stability or localization. We also discuss the problem of polysemy and solve it through the PLSA model.

### Gapped-dipeptide signatures for Gram-negative bacteria localization sites

In Figure 8, we show the distribution of topic versus protein as visualized by $P(z|d)$ for topic $z \in Z$ and pro-

**Table V**
*Comparison of PSLDoc, HYBRID, and PSORTb v.2.0 on the PS1444 Data Sets*

| | PSLDoc | | HYBRID | | PSORTb v.2.0 | |
|---|---|---|---|---|---|---|
| Loc. sites | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* | *Acc.* (%) | *MCC* |
| CP | 94.96 (94.24) | 0.91 (0.91) | 95.00 | 0.89 | 70.10 | 0.77 |
| IM | 93.20 (93.53) | 0.94 (0.94) | 90.60 | 0.92 | 92.60 | 0.92 |
| PP | 89.13 (89.13) | 0.87 (0.85) | 88.80 | 0.84 | 69.20 | 0.78 |
| OM | 95.65 (95.14) | 0.95 (0.94) | 95.10 | 0.93 | 94.90 | 0.95 |
| EC | 90.00 (87.37) | 0.87 (0.86) | 85.30 | 0.87 | 78.90 | 0.86 |
| Overall | 93.01 (92.45) | — | 91.60 | — | 82.60 | — |

The PSLDoc performance of incorporating a three-way data split procedure is indicated in the parentheses.

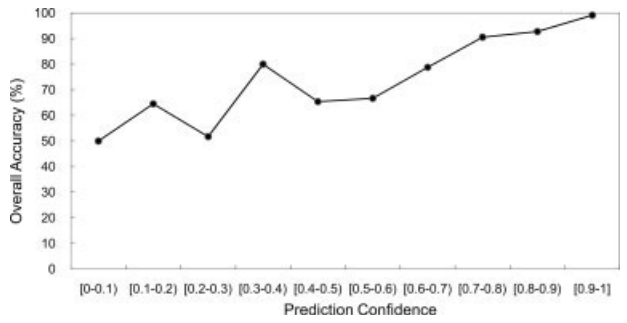


**Figure 6**

*Overall accuracy of PSLDoc with respect to prediction confidence. [x,y) represents the prediction confidence is more than x but under y.*

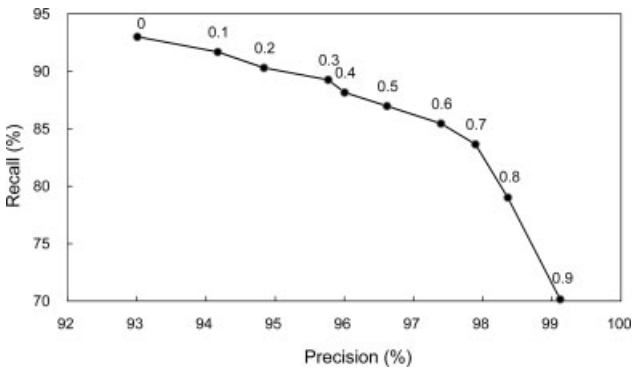


**Figure 7**

*Overall accuracy of PSLDoc with respect to prediction confidence. The value above the point denotes the corresponding prediction threshold.*

tein $d \in D$. In the figure, the size of topic ($|Z|$) is set to 80 according to the conclusion from Experiment 2.

To find site-topic preference, we then cluster proteins according to their localization sites for examining preferred topics for each localization site. The *site-topic preference* of the topic $z$ for a localization site $l$ is calculated by averaging $P(z|d)$, where $d$ (a protein) belongs to $l$ class (i.e., $d$ has localization site $l$.) The site-topic preference over topics per localization site is shown in Figure 9. We can observe from the figure that topics can be divided into five groups such that each group "prefers" a specific localization site.

We say a topic $z$ *prefers* a localization size $l$, if the corresponding site-topic preference is the largest of all localization sites. For some topics preferring PP and EC classes, the difference of the site-topic preference between their own preferring site and other sites are not obvious in Figure 9. This also reflects the relative poor performance of PSLDoc in PP and EC classes.

The distribution of topic versus gapped-dipeptide is visualized by $P(w|z)$ for gapped-dipeptide $w \in W$ and topic $z \in Z$ as shown in Figure 10. In the figure, the size of gapped-dipeptides ($|W|$) is set to 5600 ($=14 \times 20 \times 20$) following the conclusion of Experiment 2.

To list gapped-dipeptides of interest, we select 10 preferred topics for each localization site according to *site-preference confidence*, which is defined as the largest site-topic preference minus the second largest site-topic preference. For each topic, five most frequent gapped-dipeptides are selected. We list the gapped-dipeptide signatures of 10 preferred topics corresponding to each of the localization sites in Table VII.

## Gapped-dipeptide signatures reflecting motifs relevant to protein localization sites

Interestingly, some of the signatures in Table VII found by PSLDoc have been reported in the literature as motifs critical for stability or localization. One example is observed in the integral membrane (IM) proteins, in which helix–helix interactions are stabilized by aromatic residues.[46] Specifically, the aromatic motif (WXXW or W2W) is involved in the dimerization of transmembrane (TM) domains by π–π interactions.[46] Remarkably, one preferred topic predicted for the IM class includes this motif (W2W) among other signatures of aromatic residues. Another example is found in the outer membrane (OM) class, where the C-terminal signature sequence is recognized by the assembly factor, OMP85, regulating the insertion and integration of OM proteins in the outer membrane of gram-negative bacteria.[47] The C-terminal signature sequence contains a Phe (F) at the C-terminal

**Table VI**

*Comparison of PSLDoc Under the Prediction Threshold 0.7, PSLDoc Under the Prediction Threshold 0.3 and PSORTb v.2.0*

| | PSLDoc_PreThr = 0.7 | | | | | PSLDoc_PreThr = 0.3 | | | | | PSORTb v.2.0 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Loc. sites | TP | FP | FN | Pre. | Rec. | TP | FP | FN | Pre. | Rec. | TP | FP | FN | Pre. | Rec. |
| CP | 216 | 6 | 62 | 97.30 | 77.70 | 243 | 13 | 35 | 94.92 | 87.41 | 195 | 15 | 83 | 92.86 | 70.14 |
| IM | 273 | 3 | 36 | 98.91 | 88.35 | 285 | 6 | 24 | 97.94 | 92.23 | 286 | 14 | 23 | 95.33 | 92.56 |
| PP | 202 | 8 | 74 | 96.19 | 73.19 | 226 | 17 | 50 | 93.00 | 81.88 | 191 | 9 | 85 | 95.50 | 69.20 |
| OM | 366 | 2 | 25 | 99.46 | 93.61 | 372 | 6 | 19 | 98.41 | 95.14 | 371 | 10 | 20 | 97.38 | 94.88 |
| EC | 151 | 7 | 39 | 95.57 | 79.47 | 163 | 15 | 27 | 91.57 | 85.79 | 150 | 4 | 40 | 97.40 | 78.95 |
| Total | 1208 | 26 | 236 | 97.89 | 83.66 | 1289 | 57 | 155 | 95.77 | 89.27 | 1193 | 52 | 251 | 95.82 | 82.62 |

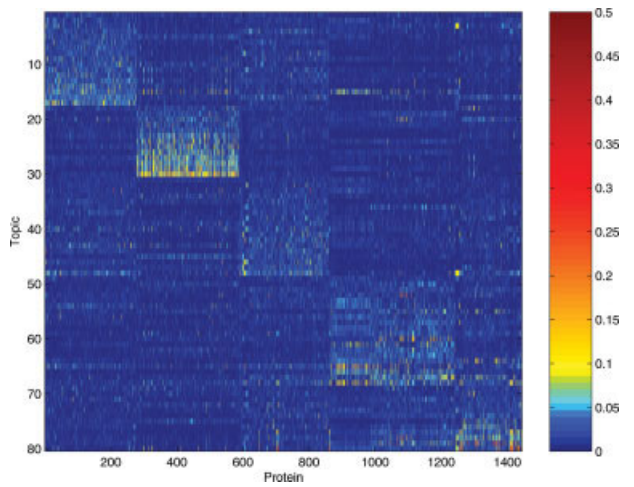


**Figure 8**

*Distribution of topic versus protein plotted as an image with its colormap,[d] where the topics are sorted such that topics "preferring" (to be explained in the third paragraph) the same localization site are grouped together. Each element P(z|d) corresponds to a rectangular area in the image and its color is decided by the value.*

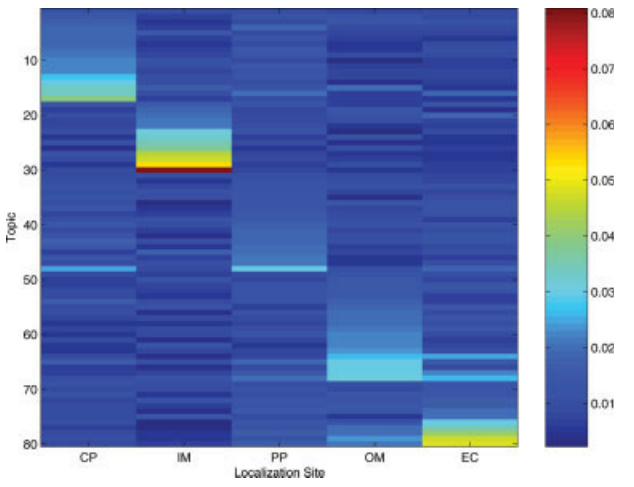


**Figure 9**

*Distribution of site-topic preference versus localization site. The 80 topics are divided into five groups of 17, 13, 18, 20, and 12 topics that prefer CP, IM, PP, OM, and EC, respectively.*

position, preceded by a strong preference for a basic amino acid (K, R).[47] One of the preferred topics indeed contains this motif (R0F.).

The above findings demonstrate the sensitivity of PSLDoc for capturing gapped-dipeptide signatures relevant to localization sites. Thus, the predicted signa-

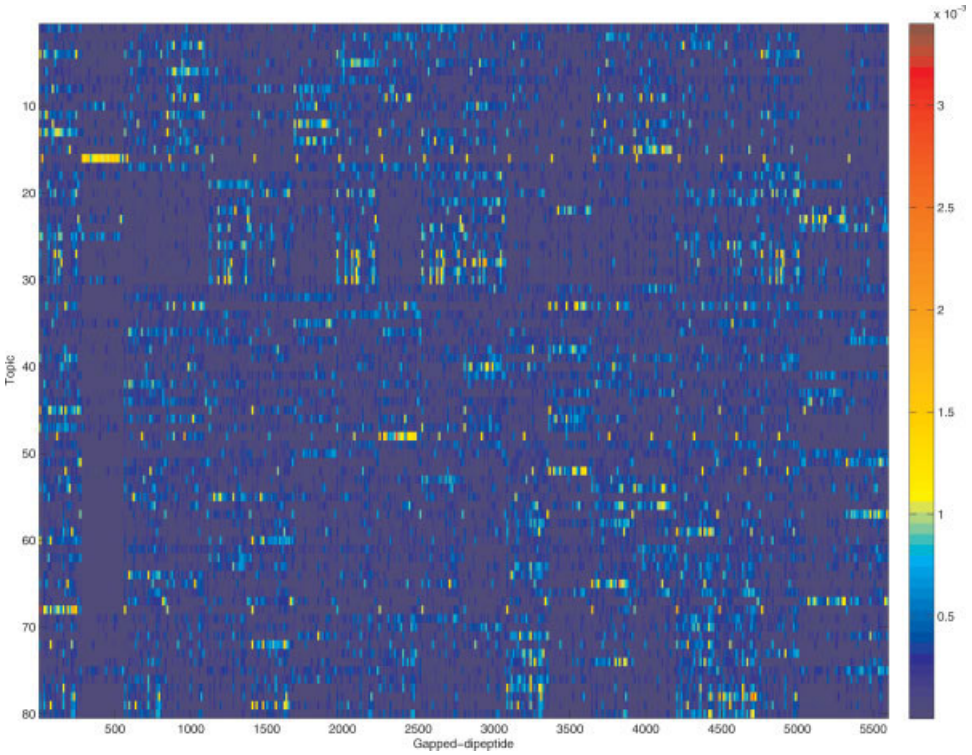


**Figure 10**

*Distribution of topic versus gapped-dipeptide.*

**Table VII**
*Gapped-Dipeptide Signatures for Each Gram-Negative Bacteria Localization Site*

| Site | Gapped-dipeptide signatures | | |
|---|---|---|---|
| CP | E0E, K1I, K5V, K1V, D0E; | L1H, L5H, L3H, H4L, H0L; | A12C, A9C, A13C, A5C, A7C; |
| | R3R, R6R, R2R, R0R, R9R; | A6A, A13A, A7A, A10A, A11A; | I0E, R6I, I3R, I3K, R6V; |
| | H3H, H1H, H7H, H13H, H10H; | H1M, H2M, H11M, M0H, H0M; | A4E, E1E, A2E, V4E, A9E; |
| | E4E, K6E, E6E, E3E, E0E | | |
| IM | I2I, I3I, I0I, L0I, I0F; | L7L, L4L, L10L, L3L, L6L; | M3M, M2M, M0M, M8M, M6M; |
| | V2I, V2V, V3I, V3V, I0V; | T2F, T6F, F3F, T4F, T8F; | A1A, A7L, A4A, A1C, A11L; |
| | W3W, W0W, W2W, W6W, W4W; | Y12L, Y1L, Y11L, L0Y, L1L; | M2T, M3T, M10T, M4T, M0L; |
| | F10P, F8P, F12P, F3P, F13P | | |
| PP | A1A, A2A, A0A, A3A, M4A; | M0H, W1Q, W1H, W1K, W5Q; | P1E, P0E, E0P, P0K, E1P; |
| | D0D, Q0D, D3D, D3Q, D11D; | W0E, E4W, W11E, E0W, W13E; | K3K, K0K, K2K, K1K, K7K; |
| | A3A, A7A, A1P, A6R, A10R; | P3N, N4P, N3P, N5P, N0P; | H6G, G3M, H7D, G11H, H11G; |
| | A10A, A11A, A6A, A12A, A3A | | |
| OM | T1R, R3T, R1T, T5R, P0P; | R0F, R4F, Y13R, R6F, R2F; | N4N, N0N, N10N, N7N, F1N; |
| | Q6Q, Q1Q, Q3Q, Q13Q, Q4Q; | S0F, A3F, F0S, R9F, F7F; | G0G, A0G, A1G, G1A, G3A; |
| | N1Q, N1N, Q1Q, N12N, Q11V; | W2N, N2W, N0W, D2W, N13W; | Q5R, R1Q, Q1R, Q3R, R2Q; |
| | Y1Y, Y0Y, Y5Y, Y4Y, Y12Y | | |
| EC | S6S, S2S, T11T, S13S, T6S; | G8G, G0G, G7G, G9G, G6G; | T1T, T3T, T5T, T9T, T10T; |
| | N10N, N9N, N13N, N11N, N12N; | N1N, N3N, N4N, N11N, N1T; | I5Y, Y12S, Y3S, Y9S, Y6I; |
| | Q2N, N1Q, Q1Q, N3Q, Q7Q; | K1S, S6S, S5S, S11M, S0S; | S3G, G3G, G4S, G3S, G2G; |
| | N0N, N12V, N4V, V12N, N9V | | |

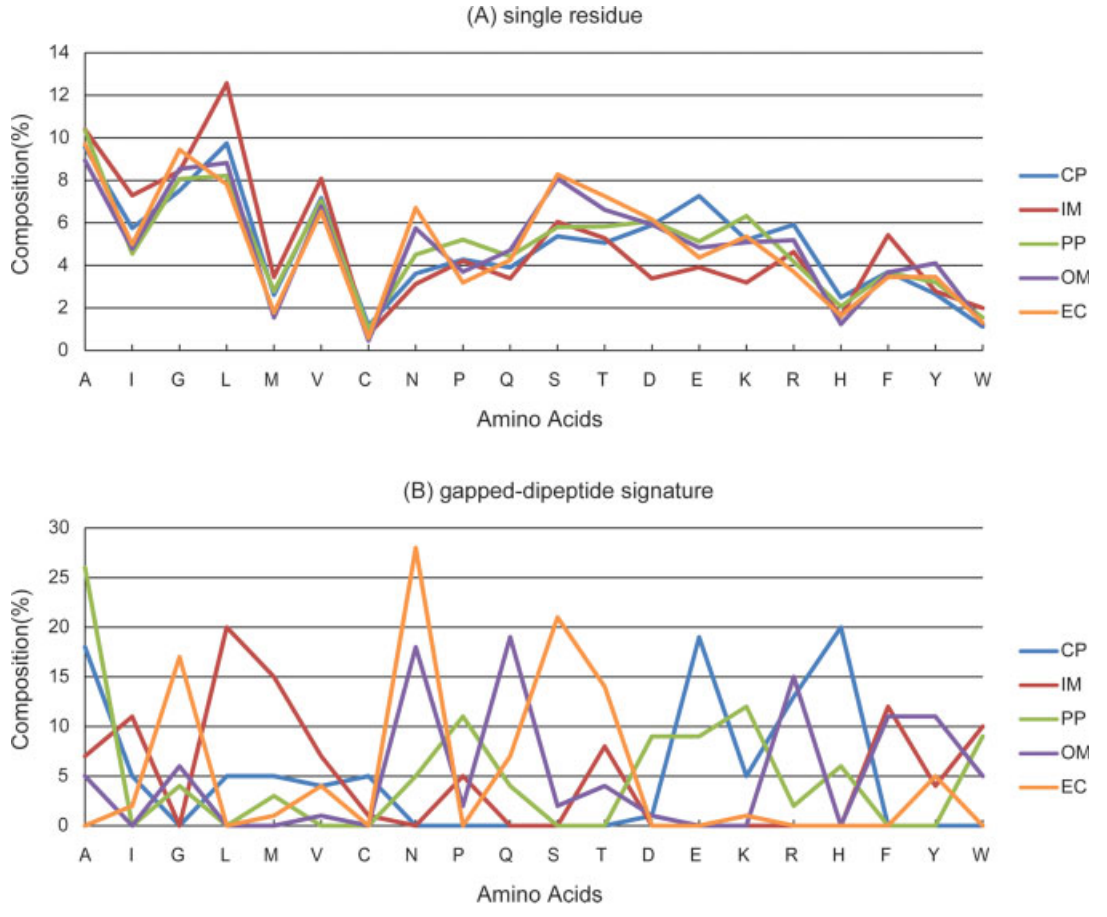


**Figure 11**
*The amino acid compositions of single residues (**A**) and selected gapped-dipeptide signatures (**B**) in different localization sites.*

tures can provide important clues for further studies of uncharacterized sequence motifs related to protein localization.

### Comparison of gapped-dipeptide signature encoding and amino acid composition

Figure 11 shows the amino acid compositions of single residues and gapped-dipeptide signatures for each localization site, respectively. It is observed that the distributions of 20 amino acids calculated from single residues and gapped-dipeptide signatures are quite different. The distribution from single residues [Fig. 11(A)] has no clear separation for some amino acids but the distribution from gapped-dipeptide signatures [Fig. 11(B)] has a clear separation among five classes.

From Figure 11(A,B), it is observed that for some amino acids, general amino acid composition bias have an effect on the gapped-dipeptide signatures (e.g., CP: E; IM: I, L; PP: P, K; OM: Y; EC: G, N). That is, amino acids having high composition in a localization site tend to also have high composition in gapped-dipeptide signatures of the localization site. For example, there are relatively high proportions for Ile and Leu in both single residue and gapped-dipeptide signature compositions in IM proteins. However, many amino acids have high compositions in at least two localization sites. Therefore, it is difficult to predict localization site based on single residue compositions. From the amino acid composition of gapped-dipeptide signatures, we observe a clear separation among different localizations for several amino acids, which are indistinguishable at the single residue level (i.e., A, M, V, Q, S, H, W). Specifically, Met, Val, and Trp have similar proportions across all five localizations in single residue composition. The small differences in single amino acid composition for these residues are amplified by examining the gapped-dipeptide signature compositions and thus, they can be used for predicting localization site in a discriminative manner. We further analyze the correlation between single amino acid and gapped-dipeptide signature compositions by the Pearson correlation coefficient whose definition for a series of $n$ measurements of variables $X$ and $Y$ is as follows:

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \quad (17)$$

The Pearson correlation coefficient (r) between the two compositions (single residues vs. gapped dipeptide signatures) for CP, IM, PP, OM, EC, and all localization sites are 0.29, 0.50, 0.41, 0.07, 0.50, and 0.36, respectively. The correlation for all localization sites is medium (in range 0.30–0.49).[48]

In summary, the gapped-dipeptide signatures predicted by PSLDoc can (1) successfully capture the compositional bias inherent at the single residue level, and (2) better resolve ambiguity in discriminating amino acid compositions for each localization site.

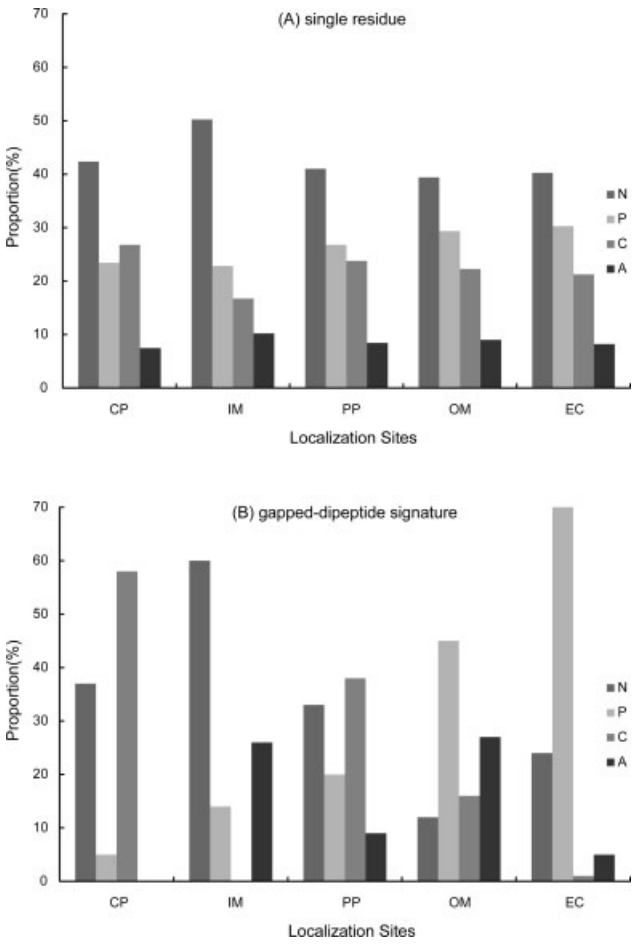


**Figure 12**

*The amino acid compositions of single residues (**A**) and predicted gapped-dipeptide signatures (**B**) for each protein class distinguished by the localization site. Localization sites: CP, IM, PP, OM, and EC. Amino acid groups: N (nonpolar: AIGLMV); P (polar: CNPQST); C (charged: DEHKR); and A (aromatic: FYW).*

### The physicochemical preference of gapped-dipeptide signatures

To further analyze the physicochemical preference of gapped-dipeptide signatures, each amino acid is classified into one of the four groups: nonpolar (AIGLMV), polar (CNPQST), charged (DEHKR), and aromatic (FYW). Figure 12 shows the grouped amino acid compositions of single residues and gapped-dipeptide signatures for each localization class. The grouped amino acid composition of single residues for each localization site has very similar preferences, but different preferences are observed for gapped-dipeptide signature composition. For example, in Figure 12(A), IM, PP, OM, and EC have similar distribu-

tion, but in Figure 12(B), each localization has distinct distribution of grouped amino acid composition. This also lends support to the second point in the previous section, that gapped-dipeptide signature can better resolve ambiguity in discriminating amino acid compositions for each localization. Furthermore, our analysis shows that the amino acid compositions of the predicted gapped-dipeptide signatures exhibit some over-represented patterns for a particular compartment.

Gapped-dipeptide signatures predicted for CP, IM, and EC classes have distinct preferences for different groups of amino acids, possibly reflecting the physicochemical constraints imposed by the environment of a subcellular compartment. In particular, the signatures predicted for IM has a high percentage of nonpolar amino acids (60%) and no charged (0%) amino acids. This can be explained in terms of the physicochemical properties of the lipid bilayer, in which nonpolar amino acids are favored in the transmembrane domains of IM proteins.[49] In contrast, charged amino acids are disfavored due to the penalty incurred in energy terms during the assembly of IM proteins.[50] CP and EC classes are found to contain a high percentage of charged and polar amino acids, respectively. The role of charged amino acids in the cytoplasm is probably related to pH homeostasis in which they act as buffers, whereas secreted proteins in the EC classes may require more polar amino acids for promoting interactions in the solvent environment.[51]

Although gapped-dipeptide signatures are found, PSLDoc performs training and testing procedures solely based on the topics of the PLSA model. In addition, Hofmann[41] also noted that PLSA can capture the semantic meaning of words, in our case, the gapped-dipeptides. This part will be discussed in the following section.

### PSLDoc's capability to solve the polysemy of gapped-dipeptides

In document classification, a word with two different meanings is called polyseme [e.g., "bank" means (i) an organization that provides various financial services or (ii) the side of a river]. Hofmann mentioned PLSA could deal with polysemy and gave an example about the word "segment" [(i) an image region or (ii) a phonetic segment].[41,52] Such a word $w$ would have a high probability in two different topics. The hidden topic variable, $P(w|z)$, associated with each word occurrence in a particular document is used to determine which particular topic $w$ is assigned to, depending on the context of the document. Sivic *et al.*[53] applied PLSA to images and discussed the polysemy on images. We discuss the polysemy effect on gapped-dipeptides.

A gapped-dipeptide may prefer two localization sites, e.g., "A6A" prefer CP and PP in Table VII. It is sometimes difficult to determine the localization site of a protein based on the weight of a polysemous gapped-dipeptide. PLSA can be used to remedy the polysemy effect of a gapped-dipeptide by associating the gapped-dipeptide with different topics. For example, "A6A" is among the top five frequent dipeptides of Topic 73 in CP and Topic 6 in PP that their probabilities $P(w|z)$ are sorted in a decreasing order as shown in Table VIII.

**Table VIII**
*"A6A" is Among the Top Five Frequent Gapped-Dipeptides of Topic 73 and Topic 6, Where Gapped-Dipeptides Are Arranged to the Decreasing Order of P(w|z)*

| Topic 73 | Topic 6 |
|---|---|
| A6A | A10A |
| A13A | A11A |
| A7A | A6A |
| A10A | A12A |
| A11A | A3A |

For example, two proteins from PS1444 data set, chemotaxis protein cheZ and Endoglucanase B,** contain subsequences of the polysemous gapped-dipeptide "A6A." They are in different classes, CP class and PP class, respectively, and some of their relevant information is listed in Table IX. Using the original vector space, the two proteins have $P\{w =$ "A6A," $d_{44}\} = 0.7001$ and $P\{w =$ "A6A," $d_{680}\} = 0.651$, which differ slightly, and thus it is difficult to distinguish them. However, using the posterior probabilities of Topic 73 and Topic 6, given the different occurrences of "A6A" based on the PLSA reduced vector space can distinguish the two proteins and determine their classes. That is, since $(P\{z_{73}|w =$ "A6A," $d_{44}\}$, $P\{z_6|w =$ "A6A," $d_{44}\}) = (0.0794, 0.0)$ and $(P\{z_{73}|w =$ "A6A," $d_{680}\}$, $P\{z_6|w =$ "A6A," $d_{680}\}) = (0.0, 0.0596)$, and Topics 73 and 6 are associated with different classes, the proteins $d_{44}$ and $d_{680}$ can be distinguished to be in CP and PP classes. This example demonstrates PLSA's capability to remedy the polysemy effect of gapped-dipeptides.

## CONCLUSION

We present a new PSL prediction method, PSLDoc, based on gapped-dipeptides and PLSA, and demonstrate that it is suitable for proteins of a wide range of sequence homologies. PSLDoc extracts features from gapped-dipeptides of various distances, where evolutionary information from the PSSM is utilized to determine the weighting of each gapped-dipeptide such that its performance is comparable to the homology search method in the high-homology data set. These features are further reduced by PLSA and incorporated as input vectors for SVM classifiers. PSLDoc performs very well in low-homology data set with overall accuracy of 86.84%. It

**Chemotaxis protein cheZ and Endoglucanase B are 44th and 680th proteins in PS1444, respectively. We use $d_{44}$ and $d_{680}$ to denote them for ease.

**Table IX**
*Examples of Two Proteins in PS1444 Having the Gapped-Dipeptide "A6A"*

| Protein name | $P\{w_j = \text{"A6A"}, d_i\}$ | $(P\{z_{73}\|d_i, w_j = \text{"A6A"}\}, P\{z_6\|d_i, w_j = \text{"A6A"}\})$ | "A6A" | Localization site |
|---|---|---|---|---|
| 116294 | 0.7001 | (0.0794, 0.0) | AIAEAAEA(40[a])<br>ASQPHQDA(75) | CP |
| 121816 | 0.651 | (0.0, 0.0596) | APGDPGSA(362)<br>AQWGVSNA(409)<br>AQYGGFLA(420) | PP |

[a]The number in brackets denotes the starting position of the gapped-dipeptide "A6A."

can also achieve very high precision by using a flexible prediction threshold. Experiments show PSLDoc performs better than some of the current methods in overall accuracy by 1.51%. Because of the generality of this method, it can be extended to other species or multiple localization sites in the future. Through analyzing the amino acid composition of gapped-dipeptide signatures, there is a relationship between the amino acid group and localization sites. For future work, we will incorporate the amino acid groups with gapped-dipeptides to design a new representation of terms for predicting protein subcellular localization.

## ACKNOWLEDGMENTS

We wish to express our gratitude to the two anonymous reviewers for their valuable and enlightening comments in improving the readability of this manuscript. We thank Ching-Tai Chen and Han-Kuen Liang for helpful discussion.

## REFERENCES

1. Nakai K, Kanehisa M. Expert system for predicting protein localization sites in gram-negative bacteria. Proteins 1991;11:95–110.
2. Nakai K, Horton P. PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization. Trends Biochem Sci 1999;24:34–35.
3. Nair R, Rost B. Better prediction of sub-cellular localization by combining evolutionary and structural information. Proteins: Struct Funct Genet 2003;53:917–930.
4. Reinhardt A, Hubbard T. Using neural networks for prediction of the subcellular location of proteins. Nucleic Acids Res 1998;26: 2230–2236.
5. Yu CS, Chen YC, Lu CH, Hwang JK. Prediction of protein subcellular localization. Proteins 2006;64:643–651.
6. Bhasin M, Garg A, Raghava GPS. PSLpred: prediction of subcellular localization of bacterial proteins. Bioinformatics 2005;21:2522–2524.
7. Hua SJ, Sun ZR. Support vector machine approach for protein subcellular localization prediction. Bioinformatics 2001;17:721–728.
8. Nair R, Rost B. Mimicking cellular sorting improves prediction of subcellular localization. J Mol Biol 2005;348:85–100.
9. Park KJ, Kanehisa M. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. Bioinformatics 2003;19:1656–1663.
10. Wang J, Sung WK, Krishnan A, Li KB. Protein subcellular localization prediction for Gram-negative bacteria using amino acid subalphabets and a combination of multiple support vector machines. BMC Bioinformatics 2005;6:174.
11. Yu CS, Lin CJ, Hwang JK. Predicting subcellular localization of proteins for Gram-negative bacteria by support vector machines based on n-peptide compositions. Protein Sci 2004;13:1402–1406.
12. Gardy JL, Laird MR, Chen F, Rey S, Walsh CJ, Ester M, Brinkman FSL. PSORTb v.2.0: expanded prediction of bacterial protein subcellular localization and insights gained from comparative proteome analysis. Bioinformatics 2005;21:617–623.
13. Gardy JL, Spencer C, Wang K, Ester M, Tusnady GE, Simon I, Hua S, deFays K, Lambert C, Nakai K, Brinkman FSL. PSORT-B: improving protein subcellular localization prediction for Gram-negative bacteria. Nucleic Acids Res 2003;31:3613–3617.
14. Lu Z, Szafron D, Greiner R, Lu P, Wishart DS, Poulin B, Anvik J, Macdonell C, Eisner R. Predicting subcellular localization of proteins using machine-learned classifiers. Bioinformatics 2004;20:547–556.
15. Rey S, Acab M, Gardy JL, Laird MR, DeFays K, Lambert C, Brinkman FSL. PSORTdb: a protein subcellular localization database for bacteria. Nucleic Acids Res 2005;33:D164–D168.
16. Costa EP, Lorena AC, Carvalho AeCPLF, Freitas AA, Holden N. Comparing several approaches for hierarchical classification of proteins with decision trees. Brazilian Symposium on Bioinformatics. Brazil, 2007. pp 126–137.
17. Cheng BYM, Carbonell JG, Klein-Seetharaman J. Protein classification based on text document classification techniques. Proteins 2005;58:955–970.
18. King BR, Guda C. ngLOC: an n-gram-based Bayesian method for estimating the subcellular proteomes of eukaryotes. Genome Biol 2007;8:R68.
19. Valdes-Perez RE, Pereira F, Pericliev V. Concise, intelligible, and approximate profiling of multiple classes. Int J Hum Comput Stud 2000;53:411–436.
20. Namburu SM, Tu H, Luo J, Pattipati KR. Experiments on supervised learning algorithms for text categorization. IEEE Aerospace Conference, Big Sky, Montana; 2005. pp 1–8.
21. Manning CD, Schütze H. Foundations of statistical natural language processing. Cambridge, MA: MIT Press; 1999. xxxvii, 680 p.
22. Salton G, Wong A, Yang CS. Vector-space model for automatic indexing. Commun ACM 1975;18:613–620.
23. Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Inf Process Manage 1988;24:513–523.
24. Liang HK, Huang CM, Ko MT, Hwang JK. Amino acid coupling patterns in thermophilic proteins. Proteins 2005;59:58–63.
25. Cedano J, Aloy P, PerezPons JA, Querol E. Relation between amino acid composition and cellular location of proteins. J Mol Biol 1997;266:594–600.
26. Chou KC, Elrod DW. Protein subcellular location prediction. Protein Eng 1999;12:107–118.
27. Garg A, Bhasin M, Raghava GPS. Support vector machine-based method for subcellular localization of human proteins using amino acid compositions, their order, and similarity search. J Biol Chem 2005;280:14427–14432.

28. Nair R, Rost B. Sequence conserved for subcellular localization. Protein Sci 2002;11:2836–2847.
29. Myers EW, Miller W. Optimal alignments in linear-space. Comput Appl Biosci 1988;4:11–17.
30. Cuff JA, Barton GJ. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins: Struct Funct Genet 1999;34:508–519.
31. Hua SJ, Sun ZR. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. J Mol Biol 2001;308:397–407.
32. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 1999;292:195–202.
33. Lin HN, Chang JM, Wu KP, Sung TY, Hsu WL. HYPROSP II—a knowledge-based hybrid method for protein secondary structure prediction based on local prediction confidence. Bioinformatics 2005; 21:3227–3233.
34. Rost B, Sander C. Prediction of protein secondary structure at better than 70-percent accuracy. J Mol Biol 1993;232:584–599.
35. Wu KP, Lin HN, Chang JM, Sung TY, Hsu WL. HYPROSP: a hybrid protein secondary structure prediction algorithm—a knowledge-based approach. Nucleic Acids Res 2004;32:5059–5065.
36. Altschul SF, Madden TL, Schaffer AA, Zhang JH, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 1997;25: 3389–3402.
37. McGuffin LJ, Bryson K, Jones DT. The PSIPRED protein structure prediction server. Bioinformatics 2000;16:404–405.
38. Wheeler DL, Chappey C, Lash AE, Leipe DD, Madden TL, Schuler GD, Tatusova TA, Rapp BA. Database resources of the National Center for Biotechnology Information. Nucleic Acids Res 2000; 28:10–14.
39. Kumar CA, Gupta A, Batool M, Trehan S. Latent semantic indexing-based intelligent information retrieval system for digital libraries. J Comput Inf Technol 2006;14:191–196.
40. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. J Am Soc Inf Sci 1990;41:391–407.
41. Hofmann T. Unsupervised learning by probabilistic latent semantic analysis. Machine Learn 2001;42:177–196.
42. Chang C-C, Lin C-J. LIBSVM: a library for support vector machines. 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
43. Wu TF, Lin CJ, Weng RC. Probability estimates for multi-class classification by pairwise coupling. J Machine Learn Res 2004;5:975–1005.
44. Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim Biophys Acta 1975; 405:442–451.
45. Ritchie MD, White BC, Parker JS, Hahn LW, Moore JH. Optimization of neural network architecture using genetic programming improves detection and modeling of gene–gene interactions in studies of human diseases. BMC Bioinformatics 2003;4:28.
46. Sal-Man N, Gerber D, Bloch I, Shai Y. Specificity in transmembrane helix-helix interactions mediated by aromatic residues. J Biol Chem 2007;282:19753–19761.
47. Robert V, Volokhina EB, Senf F, Bos MP, Van Gelder P, Tommassen J. Assembly factor Omp85 recognizes its outer membrane protein substrates by a species-specific C-terminal motif. PLos Biol 2006;4: 1984–1995.
48. Cohen J. Statistical power analysis for the behavioral sciences. Hillsdale, NJ: L. Erlbaum Associates; 1988. xxi, 567 p.
49. Ulmschneider MB, Sansom MSP, Di Nola A. Properties of integral membrane protein structures: derivation of an implicit membrane potential. Proteins 2005;59:252–265.
50. Hessa T, Kim H, Bihlmaier K, Lundin C, Boekel J, Andersson H, Nilsson I, White SH, von Heijne G. Recognition of transmembrane helices by the endoplasmic reticulum translocon. Nature 2005;433:377–381.
51. Booth IR. Regulation of cytoplasmic pH in bacteria. Microbiol Rev 1985;49:359–378.
52. Hofmann T. Probabilistic latent semantic analysis. In: Laskey KB, Prade H, editors. Proceedings of the fifteenth conference on uncertainty in artificial intelligence; July 30–August 1, 1999; Stockholm, Sweden. pp 289–296.
53. Sivic J, Russell BC, Efros AA, Zisserman A, Freeman WT. Discovering object categories in image collections. Proceedings of the IEEE international conference on computer vision (ICCV). Beijing, China; 2005. pp 1331–1338.