Journal of Experimental & Theoretical Artificial Intelligence
Vol. ??, No. ?, Month?? 2010, 1–17

Incorporating cellular sorting structure for better prediction of protein subcellular locations

W.Y. Yang[a], B.L. Lu[ab]* and James T. Kwok[c]

[a]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, No. 800, Dongchuan Road, Shanghai, China;* [b]*MOE-Microsoft Laboratory for Intelligent Computing and Intelligent Systems of Shanghai Jiao Tong University, No. 800, Dongchuan Road, Shanghai, China;* [c]*Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong*

(*Received* [redacted]; *final version received* [redacted])

This article explores the interdependences between subcellular locations and incorporates them with support vector machines for prediction of protein subcellular localisation. Traditional prediction systems utilise a ‘flat’ structure of classifiers, such as the one-versus-all and one-versus-one schemes, with amino acid compositions to perform the prediction. Apart from those existing studies that ignore the interdependences between subcellular locations, we take advantage of a hierarchical structure to organise the subcellular locations and model their relationships. Here, we propose to use four kinds of hierarchical prediction methods and make comparative studies on three datasets. Experimental results show that three of the hierarchical models outperform the traditional ‘flat’ model in terms of tree loss values. In particular, one hierarchical model outperforms the traditional ‘flat’ model for all evaluation measures. Moreover, we gained some valuable insights into the sorting process by using hierarchical structures.

Keywords: protein subcellular localisation; support vector machines; hierarchical ontology; structured prediction; optimisation

1. Introduction

The number of protein sequences with unknown functions has increased dramatically in the past decade. As a result, determining the functions of a new protein is a major issue in proteomics and bioinformatics. One of the key steps towards this long-term initiative is the prediction of subcellular locations. However, the protein sorting process is very complex and still not clearly understood. This gives rise to many opportunities for in-depth exploration of protein subcellular localisation.

Nishikawa, Kubota, and Ooi (1983) have conducted a pioneering investigation into predicting protein subcellular locations. They performed the prediction based on amino acid compositions, and found that considerable discriminative ability is contained in the simple representation. Inspired by their seminal work, many studies have been performed to incorporate sequence information for predicting subcellular locations. For example, Reinhardt and Hubbard (1998) used neural networks and constructed a benchmark

*Corresponding author. Email: blu@cs.sjtu.edu.cn

ISSN 0952–813X print/ISSN 1362–3079 online

DOI: 10.1080/0952813X.2010.506303
http://www.informaworld.com

dataset, which has been extensively studied. Chou et al. proposed a battery of methods, such as a covariant discriminant algorithm (Chou and Elrod 1999), pseudo amino acid composition (Chou 2001) and a hybrid approach using gene ontology (GO) (Chou and Cai 2005). Hua and Sun (2001) first applied support vector machines (SVMs), one of the most successful algorithms in machine learning and pattern recognition, to this prediction task. Park and Kanehisa (2003) constructed another SVM-based prediction system, whose objective is to predict a wide coverage scope of subcellular locations (12 locations) with accurate performance. Recently, due to the development of statistical learning theory and the availability of SVM solvers, various methods use SVMs as base tools to construct new prediction systems (Yang and Lu 2005; Höglund, Donnes, Blum, Adoplh, and Kohlbacher 2006; Pierleoni, Martelli, Fariselli, and Casadio 2006; Yang, Lu, and Yang 2006).

A potential drawback of all the above methods is that they use a 'flat' structure to perform predictions. The relationships between subcellular locations, which are in fact highly related to the protein sorting process, are not taken into consideration. In view of this problem, Nair and Rost (2005) have constructed a prediction system called LOCtree for protein subcellular localisation. In short, they constructed a tree structure by hand to mimic the cellular sorting process. Thus, a 'winner-take-all' scheme is used to make a hard decision on each node, down from the root to leaves. They showed that this simple decision tree scheme performs better than conventional methods. In addition, this hierarchical structure can provide insights into the sorting process, such as the accurate distinction between secretory pathway proteins and others. Pierleoni et al. (2006) also proposed a similar method called BaCelLo to incorporate the relationships between subcellular locations. They used another tree structure that is slightly different from the LOCtree and placed more emphasis on balanced performance among all the locations. They achieved the balanced prediction system by adjusting the separating hyperplane to maximise a balanced criterion at each node of the tree structure. Both of the two methods referenced above use SVMs as their node classifiers, and they use a radial basis function (RBF) kernel with tuned parameters, γ and C.

On the basis of these empirical results, there is reason to believe that the relationships between subcellular locations offer valuable information that prediction methods are able to capitalise on. However, LOCtree and BaCelLo only employ a decision tree scheme to incorporate structure information. There are two clear drawbacks for that simple scheme. First, errors may accumulate from the roots to the leaves. In other words, errors made by upper level nodes cannot be recovered by lower level nodes. Second, by using a hard decision in each node, the confidence value of each SVM decision is lost. Considering the concept of structured learning has recently emerged in the machine learning community, in this article we conduct an in-depth exploration on how to better use relationships between subcellular locations to improve prediction performance.

2. Methods

2.1. *SVMs for subcellular localisation*

The SVMs boasts a strong theoretical underpinning, coupled to remarkable empirical results across a growing spectrum of applications. In machine learning, it has become one of state-of-the-art supervised learning algorithms. However, directly using SVMs for predicting protein subcellular locations has one obvious obstacle, as suggested in the literature, which is that conventional SVMs apparently only handle binary classification problems. For multi-class problems, such as protein subcellular localisations, one needs to generalise the original binary SVMs to perform the task. In the following, we introduce several generalisations that we use to predict protein subcellular locations.

2.1.1. *Flat classification*

One of the simplest ways to generalise SVMs from the binary case to multi-class case is the so-called one-versus-all (OVA) scheme (Hsu and Lin 2002). First, we build *N* SVM classifiers with real-valued outputs, each of which is trained by using one of the classes as the positive class and the remaining classes as the negative class. Then, given a new sample, all *N* SVMs are run and output *N* decision values. We select the classifier with the largest output value (the most positive) and take its corresponding category as the resulting prediction. In this article, we take the OVA scheme as our flat classification method.

Additionally, a number of more sophisticated multi-class SVM extensions have also existed, including (OVO) scheme (Hsu and Lin 2002), part-versus-part (PVP) scheme (Lu, Wang, Utiyama, and Isahara 2004) and error correcting output code (ECOC; Dietterich and Bakiri 1995).

Previous researchers have utilised these ‘flat’ schemes numerous times to predict subcellular locations. For example, Hua and Sun (2001) used the OVA scheme to combine three and four SVMs for the prediction of prokaryotic and eukaryotic proteins, respectively. Park and Kanehisa (Park and Kanehisa 2003) used the OVO scheme to combine 66 SVMs for the prediction of 12 subcellular locations. Yang, Chen, Lu, and Kwok (Chen, Lu, and Kwok 2006; Yang and Lu 2006; Yang and Lu 2007) used the PVP scheme to combine a number of small SVMs for more balanced and accurate predictions.

2.1.2. *Hierarchical classification*

As a natural approach to comprehensively organising the data, hierarchical structure appears to be ubiquitous in information categorisation, browsing, searching and visualisation. Accordingly, researchers have proposed a number of methods to perform automated hierarchical classification of texts (Koller and Sahami 1997), web contents (Dumais and Chen 2000) and patents (Cai and Hofmann 2004). These works showed that the incorporation of hierarchical relationships between classes can obtain better results than ‘flat’ classification methods.

Hierarchical classification models present an alternative for multi-class problems. Conceptually, hierarchical models provide a more comprehensive method for protein subcellular localisation and simulate the protein sorting process via a top-down hierarchical structure, which has been shown to improve predictions (Nair and Rost 2005; Pierleoni et al. 2006).

We use two prediction schemes based on a hierarchical structure. The first one is the basic decision tree scheme, which is the same as LOCtree and BaCelLo. We refer to this as the DT-SVM in the sequel. The other scheme utilises the probability information calculated from the SVM outputs (Wu, Lin, and Weng 2004). Instead of making a hard decision at each node, we calculate the probability for each branch. Thus, we can calculate the probability of each leaf by combining the probabilities of all its ancestors in a multiplicative way. Finally, we take the leaf with the largest probability as the resulting prediction and we refer to this scheme as the PM-SVM in the sequel.

2.1.3. *Structured classification*

Maximum margin structured prediction (Taskar, Guestrin, and Koller 2003; Tsochantaridis, Joachims, Hofmann, and Altun 2005) is a more general SVM learning algorithm. It is proposed to handle predictions with interdependent outputs, such as those from a hierarchical structure, label sequences and sequence alignment. Existing structured predictions hold a consensus on defining a *discriminative function* $F: \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, from which a prediction can be derived by maximising F over $\mathbf{y} \in \mathcal{Y}$. Hence, the predicting function is

$$f(\mathbf{x}; \mathbf{w}) = \arg_{\mathbf{y}\in\mathcal{Y}} \max F(\mathbf{x}, \mathbf{y}; \mathbf{w}). \tag{1}$$

As a reasonable assumption, F is usually represented as the following linear form, $F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y})\rangle$, where $\Psi(\mathbf{x}, \mathbf{y})$ is the so-called *joint feature mapping*.

The training formulation follows from the maximum margin principle, which is shown as the following optimisation problem,

$$\begin{aligned} &\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ &\text{s.t. } \forall i,\ \forall \mathbf{y} \in \mathcal{Y}\backslash\mathbf{y}_i:\ \langle \mathbf{w}, \delta\Psi_i(\mathbf{y})\rangle \geqslant 1 - \xi_i, \\ &\quad\ \forall i:\ \xi_i \geqslant 0, \end{aligned} \tag{2}$$

where we define $\delta\Psi_i(\mathbf{y}) = \Psi(\mathbf{x}_i, \mathbf{y}_i) - \Psi(\mathbf{x}_i, \mathbf{y})$, so that the constraint can be written more concisely.

A reasonable assumption is that we want the cost of mistakes to be as low as possible. To be more specific, when a mistake is made, it is better to assign the pattern to a class that is 'near' the correct class. Concerning this intuition, we involve loss function $\Delta(\mathbf{y}_i, \mathbf{y})$, which relies on the relationship between $\mathbf{y}_i$ and $\mathbf{y}$. Existing works have proposed two optimisation forms to incorporate the loss function (Tsochantaridis et al. 2005). We use the one that rescales slack variables as follows:

$$\begin{aligned} &\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ &\text{s.t. } \forall i,\ \forall \mathbf{y} \in \mathcal{Y}\backslash\mathbf{y}_i:\ \langle \mathbf{w}, \delta\Psi_i(\mathbf{y})\rangle \geqslant 1 - \frac{\xi_i}{\Delta(\mathbf{y}_i, \mathbf{y})}, \\ &\quad \forall i:\ \xi_i \geqslant 0. \end{aligned} \tag{3}$$

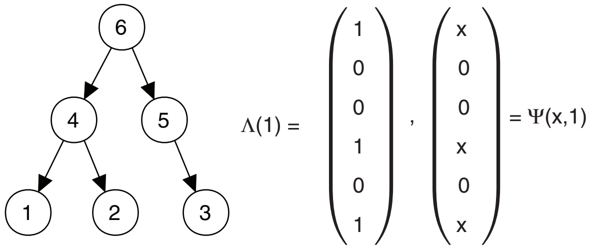
This optimisation problem can be solved by the cutting plane algorithm (Tsochantaridis et al. 2005), and guaranteed to converge in polynomial time.

The joint feature mapping is one of the key tricks for structured prediction. We define the joint feature mapping as a tensor product, $\Psi(\mathbf{x}, \mathbf{y}) = \Lambda(\mathbf{y}) \otimes \mathbf{x}$, where $\Lambda(\mathbf{y})$ is a vector encoding the relationships between a label $\mathbf{y}$ and other nodes in the hierarchy. Formally,

$$\Lambda(\mathbf{y}) = (\lambda_z(\mathbf{y}))_{z\in\mathcal{T}}, \tag{4}$$

where $\mathcal{T}$ denotes the set of all the nodes in the hierarchy, and $\lambda_z(\mathbf{y})$ is defined as

$$\lambda_z(\mathbf{y}) = \begin{cases} 1, & \text{if } z \prec \mathbf{y} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$



$$\Lambda(1) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathrm{x} \\ 0 \\ 0 \\ \mathrm{x} \\ 0 \\ \mathrm{x} \end{pmatrix} = \Psi(\mathrm{x},1)$$

Figure 1. A simple tree hierarchy. This tree hierarchy has three classes and six nodes. The joint feature mapping for an input vector **x** and category 1 is depicted as an example.

Here, the relation $\prec$ denotes that a node z is a predecessor of a label $\mathbf{y}$. Therefore, we can write the joint feature mapping $\Psi(\mathbf{x}, \mathbf{y})$ in the following form,

$$\Psi(\mathbf{x}, \mathbf{y}) = \Lambda(\mathbf{y}) \otimes \mathbf{x} = \begin{pmatrix} \lambda_{z_1}(\mathbf{y}) \cdot \mathbf{x} \\ \lambda_{z_2}(\mathbf{y}) \cdot \mathbf{x} \\ \vdots \\ \lambda_{z_t}(\mathbf{y}) \cdot \mathbf{x} \end{pmatrix}. \tag{6}$$

where z_i for $i = 1$ to t enumerate all the elements in $\mathcal{T}$.

We take an example in Figure 1 to clarify how this joint feature mapping is defined.

2.2. *Interdependent subcellular locations*

2.2.1. *Biological perspective*

Biological protein sorting is a complicated process that has not been clearly elucidated. Therefore, it is hard to model all the relationships between locations consistently into one hierarchical structure. Even in existing works, researchers use different types of hierarchical structures to perform prediction, such as the hierarchical structures used for plants in LOCtree (Nair and Rost 2005) and BaCelLo (Pierleoni et al. 2006). However, there are several general criteria for the hierarchical structure that can be summarised from earlier works. First, proteins destined for the extracellular space, the ER, the Golgi, endosomes and lysosomes are targeted via the same secretory pathway. Hence, proteins from the secretory pathway are regarded as more similar to each other than they are to other intra-cellular proteins. Second, as suggested in earlier works, the intra-cellular proteins can be roughly categorised into three groups: nuclear, cytoplasm and organelles. This categorisation was adopted in both LOCtree (Nair and Rost 2005) and BaCelLo (Pierleoni et al. 2006).

Three hierarchical structures are manually drawn to perform the prediction of subcellular localisations in our experiments and are shown in Figures 4(a), 4(b), 5(a) and 5(b).

2.2.2. *Machine learning perspective*

Note that existing hierarchical structures of subcellular locations are all manually constructed for only a few locations, such as LOCtree (Nair and Rost 2005) for fewer than six locations and BaCelLo (Pierleoni et al. 2006) for fewer than five locations. However, when presented with more subcellular locations, drawing the hierarchy by hand may be inaccurate or inconsistent with all the relationships between locations. As a result, we are interested in how to use an automatic method to analyse the distribution of data, thus assisting human experts into constructing the hierarchical structure for a large number of subcellular locations. We are also interested in whether the resulting structure is biologically interpretable.

Here, we propose a clustering-based method for constructing a binary tree. For the clustering algorithm, we use a toolkit called CLUTO.[1] The clustering-based construction method can be summarised as follows:

(1) According to the used features, cluster all the proteins into *N* clusters by using the bisection method repeatedly.
(2) Represent each subcellular location by an *N*-length vector, which holds the distribution of the corresponding proteins in all *N* clusters. Then compute the similarity between every two subcellular locations using the correlation coefficient.
(3) Use a graph partition-based clustering algorithm to perform clustering based on the similarities between every two locations. As a result, the tree structure can be drawn in a top-down fashion corresponding to the graph partitioning steps.

We use this learning method to construct a hierarchical structure for a dataset, which has 12 subcellular locations. Note that we only use the learning method as a tool to assist structure building. In practice, the resulting hierarchical structure should be selected with biological interpretation by a biological expert from a candidate set.

2.3. *Multiple root structure*

Hierarchical prediction as presented in the previous section is only used for tree structures with one single root. These tree structures can only encode the relationships among proteins from one species. Hence, they cannot capture the relationships among proteins from different species. In fact, these two types of relationships should be encoded by the same hierarchical structure. In consideration of this notion, we combine the subcellular hierarchies of different species together to get a more interpretable structure that has more than one root. These two types of interdependences are depicted in Figure 2.

The multiple root structure can be constructed in a straightforward manner. In particular, the leaf nodes represent subcellular locations for different species. We merge together those leaf nodes belonging to the same subcellular location, but different species. The edges connected to them are also moved to the newly merged leaf node. In this way, we get a multiple root structure that encodes two types of relationships.

According to this variant of the tree structure, we need to change the formulation of structured prediction. First, we must introduce some notations. We denote by $\mathcal{Y}_r$ the set of all leaf nodes that are reachable from the *r*th root, and by $\Psi_r(\mathbf{x}, \mathbf{y})$ the joint feature mapping with respect to the tree rooted at the *r*th root. Since each protein belongs to one species, i.e., one root, we assume that each sample $\mathbf{x}$ belongs to one root, although the leaf node it belongs to may be reachable by more than one root. Let the root that sample $\mathbf{x}$ belongs to be denoted by $r(\mathbf{x})$.

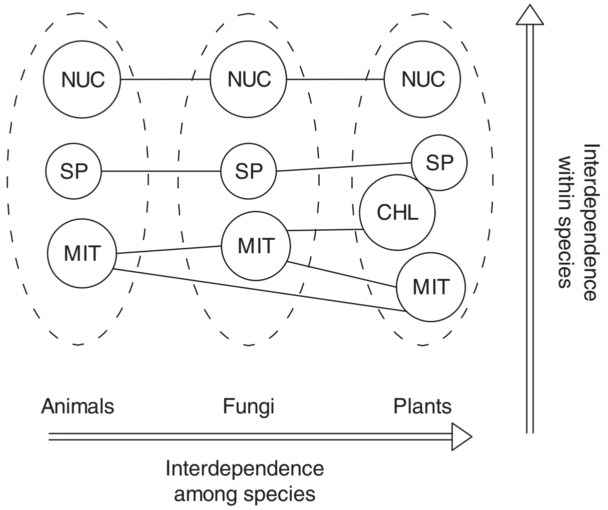


Figure 2. Prosaical illustration of two types of interdependences: CHL, chloroplast; MIT, mitochondrial; NUC, nuclear; SP, secretory pathway.

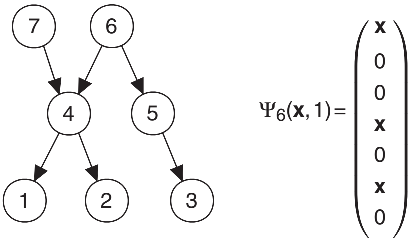

7
6
4
5
1
2
3
$\Psi_6(\mathbf{x}, 1) = (\mathbf{x}, 0, 0, \mathbf{x}, 0, \mathbf{x}, 0)^T$

Figure 3. A multiple root structure. This multiple root structure has three classes and seven nodes. The joint feature mapping for an input vector $\mathbf{x}$ and category 1 with respect to root 6 is depicted as an example. Note that the seventh element of the $\Psi_6(\mathbf{x}, 1)$ is 0, though root 7 is also a predecessor of category 1.

A variant of the original structured prediction for multiple root structure is as follows:

$$\begin{aligned} &\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{s.t. } \forall i,\ \forall \mathbf{y} \in \mathcal{Y}_{r(\mathbf{x})}\backslash \mathbf{y}_i: \end{aligned} \tag{7}$$

220

$$\begin{aligned} &\langle \mathbf{w}, \delta\Psi_{ri}(\mathbf{y})\rangle \geqslant 1 - \frac{\xi_i}{\Delta(\mathbf{y}_i, \mathbf{y})}, \\ &\forall i\colon xi_i \geqslant 0, \end{aligned} \tag{8}$$

where $\delta\Psi_{ri}(\mathbf{y}) = \Psi_r(\mathbf{x}_i, \mathbf{y}_i) - \Psi_r(\mathbf{x}_i, \mathbf{y})$. An illustrating example of $\Psi_r(\mathbf{x}, \mathbf{y})$ is depicted in Figure 3. Similar with the optimisation problem defined in Equation (3), this new optimisation problem can also be solved in polynomial time by the cutting plane algorithm (Tsochantaridis et al. 2005).

3. Experiments

3.1. *Experimental setup*

The SVM algorithm and structured prediction algorithm used in our experiment are modified from the implementation of LibSVM version 2.82[2] and SVM-struct version 3.0[3].

In all the experiments, a linear kernel is used, since it is computationally efficient and achieves competitive performance for high-dimensional inputs. We use the occurrences of k-mers as features for each protein sequence, since this type of feature could provide robust and competitive prediction with other complicated feature extraction methods. This has been empirically proven in Yang et al. (2006). Moreover, as a pre-processing step, the feature vector is normalised to be of unit length ($\|\mathbf{x}\|_2 = 1$). For the training parameters, we follow the heuristic used in SVMlight (Joachims 1998) that sets C to 1 in all SVM runs (since the input vectors are normalised to unit length). Experiments on text classification show that this is a good choice (Cai and Hofmann 2004). The tolerance parameter ϵ is set to 0.01 in all runs for SVM-struct. This value is also used by default in previous works (Cai and Hofmann 2004).

According to the constant C used in our experiment, it is necessary to normalise the joint feature mapping as $\langle \Phi(\mathbf{x}, \mathbf{y}), \Phi(\mathbf{x}, \mathbf{y}) \rangle = 1$ so that the heuristic in SVMlight is applicable. Therefore, instead of using $\lambda_z(\mathbf{y}) = 1$ in Equation (5), we set $\lambda_z(\mathbf{y}) = \sqrt{\frac{1}{d}}$, if $z \prec \mathbf{y}$, where d is the depth of the tree structure.

3.2. *Datasets and evaluations*

In the experiment, we use three datasets to measure the effectiveness of our hierarchical prediction methods. These datasets are constructed by Reinhardt and Hubbard (1998), Park and Kanehisa (2003) and Pierleoni et al. (2006), respectively. According to the author names and their prediction systems, we refer to these three datasets as RH, PLOC and BaCelLo datasets, respectively. The detailed information about these three datasets is listed in Table 1. The specific locational distributions are shown in Table 4–6. For the BaCelLo dataset, we use the same split of training and test sets as in the work by

Table 1. Detailed information of the three used datasets, RH (Reinhardt and Hubbard 1998), PLOC (Park and Kanehisa 2003) and BaCelLo (Pierleoni et al. 2006).

Dataset	No. of locations	No. of sequences	SWISSPROT version	Year
RH (prokaryotic)	3	997	33.0	1998
RH (eukaryotic)	4	2427	33.0	1998
PLOC (prokaryotic and eukaryotic)	12	7579	39.0	2003
BaCelLo (plants)	5	491	48.0	2006
BaCelLo (animals)	4	2597	48.0	2006
BaCelLo (fungi)	4	1198	48.0	2006

Pierleoni et al. (2006). For the RH and PLOC datasets, we use the same fivefolds as in the earlier works (Hua and Sun 2001; Park and Kanehisa 2003) in order to be consistent.

We use standard recall (R), total accuracy (TA), locational accuracy (LA) (Park and Kanehisa 2003) and tree loss (Δ-loss) to measure prediction performance. The standard recall R is used to measure the prediction of each location. The three measures, TA, LA and Δ-loss, are used to measure the overall prediction accuracy across all locations. The tree loss function used in the experiment is

$$\Delta(\mathbf{y}, \hat{\mathbf{y}}) = \left(\sum_{z \in \mathcal{T} : z \prec \mathbf{y},\, z \not\prec \hat{\mathbf{y}}} \frac{1}{2} \right) + \left(\sum_{z \in \mathcal{T} : z \not\prec \mathbf{y},\, z \prec \hat{\mathbf{y}}} \frac{1}{2} \right). \tag{9}$$

This loss function can also be interpreted as the distance between $\mathbf{y}$ and $\hat{\mathbf{y}}$ in the tree structure. We use this same loss function in both training and evaluation.

3.3. *Hierarchical structures for subcellular locations*

Following the descriptions in Section 2.2, we developed the hierarchical structures for the three datasets used in our experiments. For the RH and BaCelLo datasets, the structures are smaller than those of the PLOC dataset. Hence, we directly draw the hierarchical structures by using the aforementioned biological criteria. However, for the PLOC dataset, there are 12 different subcellular locations and 7579 protein sequences. Due to the large numbers of sequences and subcellular locations, we apply the clustering-based method to produce a hierarchical structure for this dataset. We test about 10 different values for N, the number of clusters, and report the structure that is more biologically plausible and has better prediction performance. The hierarchical structure for the PLOC dataset shown in Figure 6 is produced by $N = 50$.

It is worth mentioning that the hierarchical structure automatically generated for the PLOC dataset is quite biologically plausible. The left branch from the root roughly represents the intra-cellular components of cells. Correspondingly, the right branch stands for the secretory pathway and extracellular components. Overall, this automatically generated structure coincides with well-known biological knowledge.

We also constructed for the BaCelLo dataset a more complicated structure called a multiple root tree, which can be used for structured prediction to incorporate the relationships between locations, as well as between species. The structure in Figure 4(c) is constructed and used for training following the steps in Section 2.3. It is produced by merging the leaf nodes of the three tree structures for BaCelLo animals, fungi and plants shown in Figure 4(a) and (b). The red, green and blue lines represent the original tree structures of BaCelLo animals, fungi and plants, respectively. The black lines represent common edges of the three structures.

In Figures 4–6, the double, solid and dashed circled nodes represent that the corresponding SVMs perform relatively well, modestly and poorly, respectively. The abbreviations used are as follows: CHL, chloroplast; CYK, cytoskeleton; CYT, cytoplasmic; ER, endoplasmic reticulum; EXT, extracellular; GOL, Golgi apparatus; LYS, lysosomal; MEM, membrane; MIT, mitochondrial; NUC, nuclear; PER, peroxisomal; RIP, periplasmic; SP, secretory pathway and VAC, vacuolar.

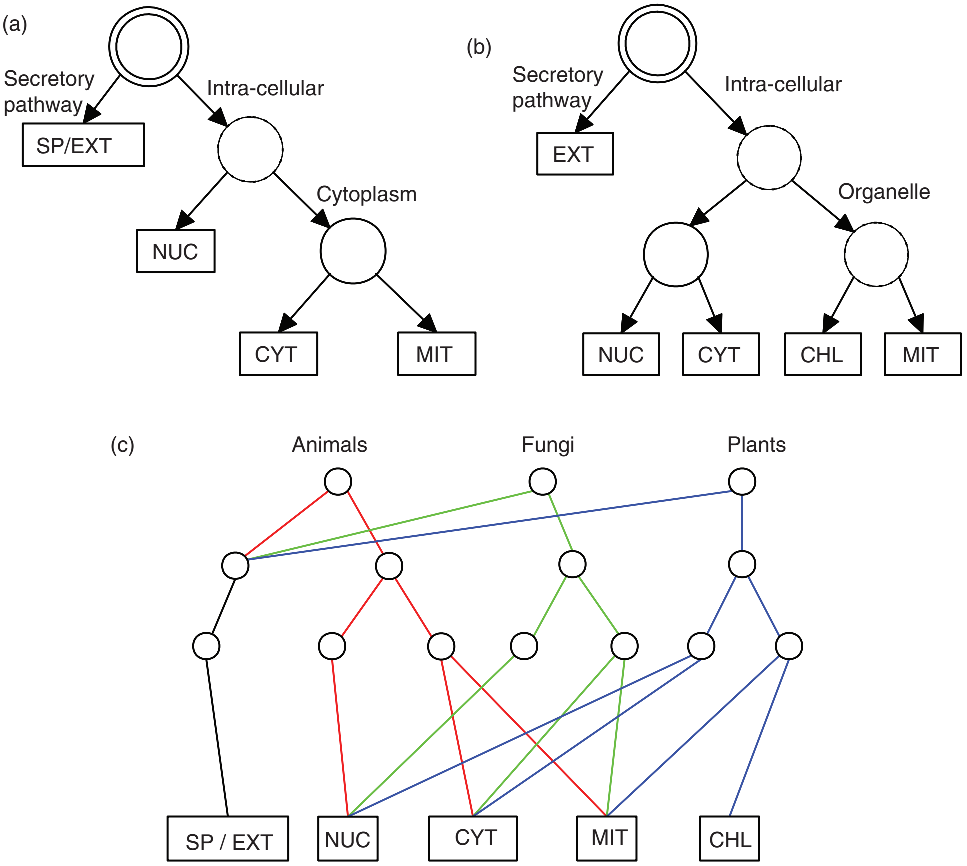


Figure 4. Hierarchical structures of BaCelLo dataset: (a) animals and fungi, (b) plants and (c) multiple root structure for three species.

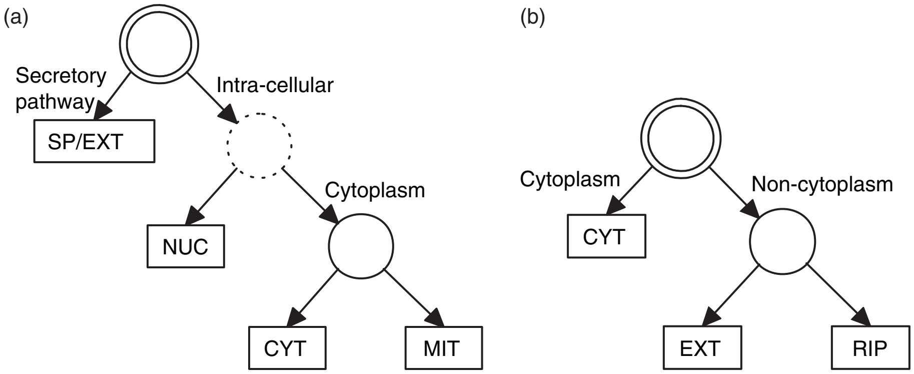


Figure 5. Hierarchical structures of RH dataset: (a) eukaryotic and (b) prokaryotic.

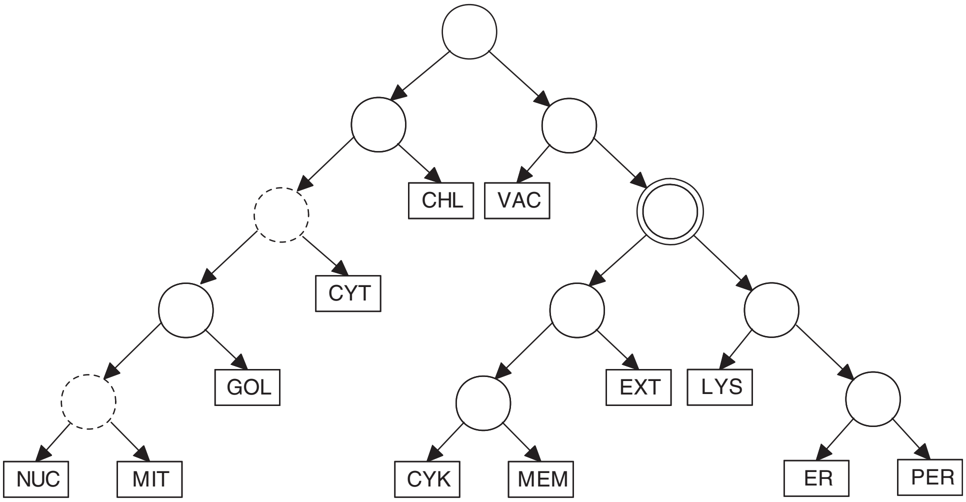


Figure 6. Hierarchical structure for PLOC dataset.

3.4. *Comparison with traditional methods*

We made comparisons between the traditional 'flat' model and four types of 'hierarchical' methods. The flat model used in our experiments is the OVA scheme, which is previously used and empirically proven to be competitive with other SVM-based methods for multiclass problems (Rifkin and Klautau 2004). The other four 'hierarchical' methods are the aforementioned DT-SVM, PM-SVM, SVM-struct and multiple root SVM-struct. Note that DT-SVM is essentially the method used in LOCtree (Nair and Rost 2005) and BaCelLo (Pierleoni et al. 2006).

3.4.1. *Comparison with flat SVM and decision tree SVM*

We carry out the experiments on four kinds of features: amino acid compositions (1-mers), amino acid pair compositions (2-mers), 3-mers and 4-mers. These features are extracted in a similar manner as the *n*-gram features commonly used in text processing. In our experiment, these features indeed give robust prediction performance. We depict in Figure 7 the curves of loss values for the flat model and three types of hierarchical models with respect to feature sets used. We defer the results of the multiple root SVM-struct to Table 3 for the BaCelLo dataset since it is not run on all datasets. We observe that PM-SVM performs better than the flat SVM in nearly all cases, except for two points for the BaCelLo fungi dataset. SVM-struct also performs better than the flat SVM except for four out of a total of 24 points. However, the DT-SVM, which is essentially a decision tree scheme used in LOCtree and BaCelLo, apparently does not perform better than the flat SVM in our experiments.

3.4.2. *Performance of multiple root SVM*

We list the best results obtained for the three datasets in Tables 2 and 3. Note that 3-mer or 4-mer features give the best performance for all three datasets. Therefore, we show the results on the RH and BaCelLo datasets with 3-mer features and the PLOC dataset with

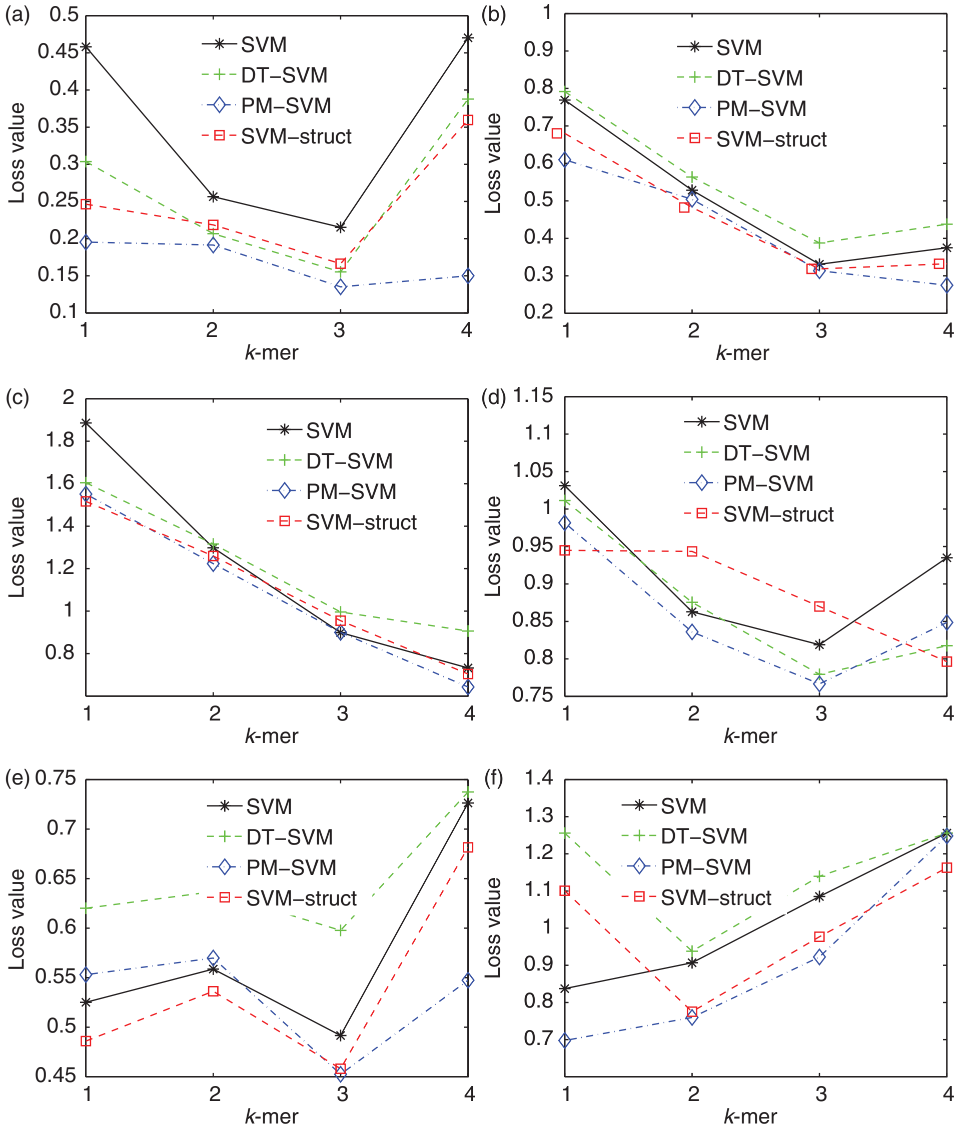


Figure 7. Comparison between flat SVM and three hierarchical prediction methods. Flat SVM refers to traditional OVA SVM. Comparisons are on three datasets (RH, PLOC and BaCelLo) with varying lengths of the *k*-mers. The performance is measured by using tree loss value: (a) RH prokaryotic, (b) RH prokaryotic, (c) PLOC, (d) BaCelLo plants, (e) BaCelLo fungi and (f) BaCelLo plants.

4-mer features. From the comparison, we can observe that PM-SVM and multiple root SVM-struct perform competitively, and both perform better than the other three methods. Note that multiple root SVM-struct is trained by using a mixed tree structure and protein sequences from animals, fungi and plants. In the experiment, it clearly performs better than the SVM-struct trained independently for each species. This strongly suggests that the incorporation of two types of interdependences is reasonable and effective in practice.

Table 2. Comparison between different methods for the RH set on 3-mer features and the PLOC set using 4-mer features. The measures are TA (%), LA (%) and Δ-loss.

	RH prokaryotic			RH eukaryotic			PLOC		
Methods	TA	LA	Δ-Loss	TA	LA	Δ-Loss	TA	LA	Δ-Loss
Flat SVM	88.6	77.3	0.22	82.5	75.4	0.33	79.5	59.8	0.73
DT-SVM	90.9	79.9	0.16	79.7	70.7	0.39	73.5	43.7	0.91
PM-SVM	**91.7**	**84.6**	**0.14**	**83.3**	**78.7**	**0.31**	**82.1**	**67.0**	**0.64**
SVM-struct	90.2	79.1	0.16	82.6	74.5	**0.31**	79.0	58.8	0.70

Table 3. Comparison between different methods for the BaCelLo sets using 3-mer features. The measures are TA (%), LA (%) and Δ-loss.

	Animals			Fungi			Plants		
Methods	TA	LA	Δ-Loss	TA	LA	Δ-Loss	TA	LA	Δ-Loss
Flat SVM	66.5	48.9	0.82	76.0	51.6	0.49	44.2	23.3	1.09
DT-SVM	65.1	50.5	0.78	71.0	46.3	0.60	41.1	22.3	1.14
PM-SVM	**66.8**	**54.0**	**0.77**	76.0	61.7	0.45	**53.5**	**39.0**	0.92
SVM-struct	63.2	47.5	0.87	76.5	57.2	0.46	49.6	25.7	0.98
Multiple root SVM-struct	64.0	51.5	0.86	**77.7**	**63.5**	**0.44**	**53.5**	36.0	**0.89**

Table 4. Comparison of location recalls on the RH set.

Species	Locations (#seq)	Flat	PM
Prokaryotic	Cytoplasm (688)	**1.00**	0.98
	Extracellular (107)	0.74	**0.77**
	Periplasmic (202)	0.58	**0.79**
Eukaryotic	Cytoplasm (684)	**0.81**	**0.81**
	Extracellular (325)	0.80	**0.83**
	Mitochondria (321)	0.46	**0.59**
	Nuclear (1097)	**0.95**	0.92

3.4.3. *Location-by-location comparison*

To clearly show how the PM-SVM and multiple root SVM-struct outperform flat SVM, we list in Tables 4–6 location-by-location comparisons between them on the three datasets used. From comparison on the PLOC dataset, we can observe that PM-SVM and the multiple root SVM-struct perform much better than OVA SVM in most subcellular locations, especially the locations with fewer protein sequences. In contrast, OVA SVM favours those subcellular locations with more protein sequences, e.g., the cytoplasmic, nuclear and membrane, and it performs slightly better in those locations. By sacrificing slightly in the large class and improving greatly in the small class, those two hierarchical

Table 5. Comparison of location recalls on the PLOC set.

Locations (#seq)	Flat	PM
Chloroplast (671)	0.73	**0.77**
Cytoplasmic (1241)	**0.77**	**0.77**
Cytoskeleton (40)	**0.66**	0.36
ER (114)	0.60	**0.72**
Extracellular (861)	0.76	**0.86**
Golgi apparatus (47)	0.06	**0.47**
Lysosomal (93)	0.53	**0.69**
Mitochondrial (727)	0.43	**0.64**
Nuclear (1932)	**0.93**	0.89
Peroxisomal (125)	0.39	**0.40**
Membrane (1674)	**0.96**	0.95
Vacuolar (54)	0.35	**0.56**

Table 6. Comparison of location recalls on the BaCelLo set.

Species	Locations (#seq)	Flat	PM	MR
Animal	Cytoplasm (439)	0.12	0.14	**0.19**
	Mitochondria (188)	0.14	0.29	**0.34**
	Nucleus (1166)	**0.82**	0.78	0.79
	SP (804)	**0.87**	0.86	0.73
Fungi	Cytoplasm (211)	0.07	0.17	**0.30**
	Mitochondria (188)	0.27	**0.45**	**0.45**
	Nucleus (711)	**0.98**	0.91	0.91
	SP (88)	0.75	**0.94**	0.88
Plant	Chloroplast (204)	**0.93**	0.87	0.87
	Cytoplasm (58)	0.00	0.00	**0.17**
	Mitochondria (67)	0.00	0.00	0.00
	Nucleus (121)	0.23	0.41	**0.43**
	SP (41)	0.00	**0.67**	0.33

methods achieve better performance than the flat SVM. This balanced performance also holds for the other two datasets, RH and BaCelLo. The Flat, PM and MR denote OVA SVM, PM-SVM and multiple root SVM-struct in Tables 4–6. The #seq denotes the number of protein sequences in the corresponding subcellular location.

3.5. *Diverse performance of node SVMs*

One of the advantages of hierarchical prediction is that it can provide valuable insights into the sorting process. By comparing the performances of the node SVMs in the hierarchical structure, we pick out those SVMs that perform relatively well or poorly and show them in Figures 4–6 by double or dashed circles, respectively. Actually, these good and bad SVMs strongly suggest some relationships between subcellular locations. In other words, the high performing SVM suggests a clear separation between the protein sequences in its two branches, and vice versa. We note that the work done by Huh et al. (2003) reflects the fact that some proteins may occur in multiple subcellular locations, which is, in fact, closely connected with our work. Roughly, a blurry separation suggests the transfers of proteins between locations.

We can make two main observations based on the SVM performance in Figures 4–6. First, it is hard to discriminate nuclear proteins from those proteins targeted to the mitochondria and cytoplasm. This observation coincides with the work done by Huh et al. (2003) and Chou and Cai (2005), suggesting the existence of ‘multi-location’ proteins between them. Second, it is relatively easy to separate proteins in the extracellular, membrane, and secretory pathway from proteins in the other subcellular locations. This observation coincides with the well-known facts about the secretory pathway and membrane proteins, since proteins destined for these locations are distinct from the other proteins, e.g., the membrane proteins have hydrophobic regions.

4. Conclusion

Since traditional prediction methods ignore the inter-location relationships, we have proposed in this research work to improve the prediction of subcellular locations by incorporation of hierarchical structures. A total of six structures are constructed by either hand or automatic methods. Four tree structures for a small number of locations are drawn by hand, which is in a similar manner with previous methods. One tree structure for a relatively large number of locations is selected from a set of candidate trees, which are all automatically generated by a clustering-based method. The last multiple root structure is put forward to encode another type of relationship between proteins belonging to the same subcellular location, but different species. In the experiments, two of the proposed methods outperform traditional methods on nearly all datasets. The best method appears to be PM-SVM, which picks the leaf node with the highest multiplicative probability. Another good choice is the SVM-struct, which performs better than the flat SVM in terms of tree loss values. A variant of the SVM-struct on multiple root structure is proposed and proven to be better than traditional SVM-struct which is trained on independent tree structures. Moreover, from the experimental results, we gained some valuable insights into the protein sorting process, which supports a close connection with earlier research work. As such, it suggests that using structure in prediction can provide valuable clues for the exploration of the protein sorting process.

Acknowledgements

The authors thank Yang Yang for her valuable discussions and suggestions at the beginning of this research. This research is partially supported by the National Natural Science Foundation of China via the grant NSFC 60773090.

Notes

1. http://glaros.dtc.umn.edu/gkhome/views/cluto.
2. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
3. http://svmlight.joachims.org/svm_struct.html.

References

Cai, L., and Hofmann, T. (2004), ‘Hierarchical Document Categorization with Support Vector Machines’, in *Proceedings of ACM International Conference on Information and Knowledge Management*, pp. 78–87.

Chen, K., Lu, B.L., and Kwok, J.T. (2006), ‘Efficient Classification of Multi-label and Imbalanced Data using Min-max Modular Classifiers’, in *Proceedings of IEEE International Joint Conference on Neural Networks*, pp. 1770–1775.

Chou, K.C. (2001), ‘Prediction of Protein Cellular Attributes using Pseudo-amino Acid Composition’, *Proteins: Structure, Function, and Genetics*, 43, 246–255.

Chou, K.C., and Cai, Y.D. (2005), ‘Predicting Protein Localization in Budding Yeast’, *Bioinformatics*, 21, 944–950.

Chou, K.C., and Elrod, D.W. (1999), ‘Protein Subcellular Location Prediction’, *Protein Engineering*, 12, 107–118.

Dietterich, T.G., and Bakiri, G. (1995), ‘Solving Multiclass Learning Problems via Error-correcting Output Codes’, *Journal of Artificial Intelligence Research*, 2, 263–286.

Dumais, S., and Chen, H. (2000), ‘Hierarchical Classification of Web Content’, in *Proceedings of ACM SIGIR Special Interest Group on Information Retrieval*, pp. 256–263.

Höglund, A., Donnes, P., Blum, T., Adoplh, H., and Kohlbacher, O. (2006), ‘MultiLoc: Prediction of Protein Subcellular Localization using N-terminal Targeting Sequences, Sequence Motifs and Amino Acid Composition’, *Bioinformatics*, 22, 1158–1165.

Hsu, C.W., and Lin, C.J. (2002), ‘A Comparison of Methods for Multiclass Support Vector Machines’, *IEEE Transactions on Neural Networks*, 13, 415–425.

Hua, S., and Sun, Z. (2001), ‘Support Vector Machine Approach for Protein Subcellular Localization Prediction’, *Bioinformatics*, 17, 721–728.

Huh, W.K., Falvo, J.V., Gerke, L.C., Carroll, A.S., Howson, R.W., Weissman, J.S., and O’Shaea, E.K. (2003), ‘Global Analysis of Protein Localization in Budding Yeast’, *Nature*, 425, 686–691.

Joachims, T. (1998), ‘Text Categorization with Support Vector Machines: Learning with Many Relevant Features’, in *Proceedings of European Conference on Machine Learning*, pp. 137–142.

Koller, D., and Sahami, M. (1997), ‘Hierarchically Classifying Documents using very Few Words’, in *Proceedings of International Conference on Machine Learning*, pp. 170–178.

Lu, B.L., Wang, K.A., Utiyama, M., and Isahara, H. (2004), ‘A Part-versus-part Method for Massively Parallel Training of Support Vector Machines’, in *Proceedings of IEEE International Joint Conference on Neural Networks*, pp. 735–740.

Nair, R., and Rost, B. (2005), ‘Mimicking Cellular Sorting Improves Prediction of Subcellular Localization’, *Journal of Molecular Biology*, 348, 85–100.

Nishikawa, K., Kubota, Y., and Ooi, T. (1983), ‘Classification of Proteins into Groups Based on Amino Acid Compostion and Other Characters’, *Journal of Biochemistry*, 94, 997–1007.

Park, K.J., and Kanehisa, M. (2003), ‘Prediction of Protein Subcellular Locations by Support Vector Machines using Compositions of Amino Acids and Amino Acid Pairs’, *Bioinformatics*, 19, 1656–1663.

Pierleoni, A., Martelli, P.L., Fariselli, P., and Casadio, R. (2006), ‘BaCelLo: A Balanced Subcellular Localization Prediction’, *Bioinformatics*, 22, e408–e416.

Reinhardt, A., and Hubbard, T. (1998), ‘Using Neural Networks for Prediction of Subcellular Location of Proteins’, *Nucleic Acids Research*, 26, 2230–2236.

Rifkin, R.M., and Klautau, A. (2004), ‘In Defense of One-vs-all Classification’, *Journal of Machine Learning Research*, 5, 101–141.

2 Taskar, B., Guestrin, C., and Koller, D. (2003), ‘Max-margin Markov Networks’, in *Advances in Neural Information Processing Systems*, ed. ■. ■■, ■: MIT Press, pp. ■■–■■.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005), 'Large Margin Methods for Structured and Interdependent Output Variables', *Journal of Machine Learning Research*, 6, 1453–1484.

Wu, T.F., Lin, C.J., and Weng, R.C. (2004), 'Probability Estimates for Multi-class Classification by Pairwise Coupling', *Journal of Machine Learning Research*, 5, 975–1005.

Yang, W.Y., Lu, B.L., and Yang, Y. (2006), 'A Comparative Study on Feature Extraction from Protein Sequences for Subcellular Localization Prediction', in *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pp. 201–208.

Yang, Y., and Lu, B.L. (2005), 'Extracting Features from Protein Sequences using Chinese Segmentation Techniques for Subcellular Localization', in *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pp. 288–295.

Yang, Y., and Lu, B.L. (2006), 'Prediction of Protein Subcellular Multi-locations with a Min-max Modular Support Vector Machine', in *Proceedings of International Symposium on Neural Networks*, pp. 667–673.

Yang, Y., and Lu, B.L. (2007), 'Incorporating Domain Knowledge into a Min-max Modular Support Vector Machine for Protein Subcellular Localization', in *Proceedings of International Conference on Neural Information Processing*, pp. ■■–■■. 3