Subcellular Localization Prediction through Boosting Association Rules

Yongwook Yoon and Gary Geunbae Lee

Abstract—Computational methods for predicting protein subcellular localization have used various types of features, including N-terminal sorting signals, amino acid compositions, and text annotations from protein databases. Our approach does not use biological knowledge such as the sorting signals or homologues, but use just protein sequence information. The method divides a protein sequence into short k-mer sequence fragments which can be mapped to word features in document classification. A large number of class association rules are mined from the protein sequence examples that range from the N-terminus to the C-terminus. Then, a boosting algorithm is applied to those rules to build up a final classifier. Experimental results using benchmark data sets show that our method is excellent in terms of both the classification performance and the test coverage. The result also implies that the k-mer sequence features which determine subcellular locations do not necessarily exist in specific positions of a protein sequence. Online prediction service implementing our method is available at http://isoft.postech.ac.kr/research/BCAR/subcell.

Index Terms—Clustering classification and association rules, bioinformatics (genome or protein) databases, pattern recognition.

1 Introduction

The genes in DNA direct the synthesis of proteins which are necessary to maintain an organism's biological functions [1]. The synthesized proteins are transported to different subcellular compartments by a complex sorting mechanism [2], so prediction of subcellular locations of proteins has become more important in the design of new drugs. To determine the locations by experimental methods requires considerable time and effort. Therefore, many computational prediction methods have been proposed recently. These methods can be divided into several classes according to what type of features they use.

The first class uses targeting signals that reside at a specific part of primary sequence. Nakai [3] reviewed a diverse range of sorting signals in bacteria, plant, and animal proteins which had been verified through biological experiments. Most of these signals are concentrated near specific positions of the primary protein sequence (N-terminus or C-terminus). PSORT [4], TargetP [5], and iPSORT [6] predictors used sorting signals for prediction.

The second class exploits information which can be extracted from the entire range of amino acid sequences. Several predictors use the amino acid compositions as features in training sequences [7], [8], [9]. PLOC [7] uses the compositions of amino acid, amino acid pairs, and gapped amino acid pairs. Eskin and Agichtein [10] and Hawkins et al. [11] used Support Vector Machine (SVM) classifiers with k-mer subsequence features.

• *The authors are with the Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), San 31, Hyoja-Dong, Pohang 790-784, South Korea. E-mail: {ywyoon, gblee}@postech.ac.kr.*

Manuscript received 17 Sept. 2009; revised 5 June 2011; accepted 8 Aug. 2011; published online 27 Sept. 2011.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2009-09-0160. Digital Object Identifier no. 10.1109/TCBB.2011.131.

The third class exploits the information from external knowledge bases. LOCkey [12] collects keywords annotated to the protein entries of the Swiss-Prot database [13]. Proteome Analyst [14] has a predictor which uses text annotations for the homologues of a target protein. MultiLOC [15] uses protein sequence motifs from the NLSdb [16] and PROSITE [17] databases. SherLOC [18] has a SVM classifier whose feature set consists of texts from PubMed titles and abstracts on proteins of Swiss-Prot. These predictors that were assisted with text features were highly accurate.

Each type of features has strengths and weaknesses. Using only sorting signals at the N-terminus may decrease the coverage for unseen sequence patterns of test proteins. Although global information such as amino acid composition is helpful to improve classification performance, it is not as accurate as sequence information. Thus, recent studies have enhanced prediction accuracy (Acc) through either combining sorting signals and amino acid composition information [15] or combining all three types of features [18]. Chou and Shen [19], [20] constructed an ensemble classifier using the features of the gene ontology database and pseudo amino acid composition.

The predictors which use only sorting signals or sequence motifs as features are apt to suffer from low coverage for test proteins. So are the predictors which use text annotation features, because newly synthesized proteins might not have text information in the protein knowledge bases. We propose an approach that uses only sequence feature, while keeping a better classification performance than the previous methods. A primary protein sequence is divided into short sequence fragments, from which associative classification rules are generated using a frequent pattern mining algorithm [21]. Our classifier improves the coverage for test sequences by generating a huge number of class association rules whose condition part consists of short sequence fragments. However, this may cause a negative influence on the computational

efficiency and the prediction accuracy. To overcome such deficiency, our method improves the classification performance by applying a boosting algorithm to the mined association rules.

Some previous studies have applied association rule mining and boosting to the subcellular localization prediction. She et al. [22] constructed a classifier with associative classification rules to target outer membrane proteins of Gram-negative bacteria. Jin et al. [23] used AdaBoost to predict subcellular locations based on amino acid compositions. Our method is the first to unite associative classification and boosting in a biological application. Unlike other systems which combine several classifiers that use different types of features [15], [18], [20], our approach is very convenient both for the implementation and the application. Experimental results show the performance of our classifier to be equivalent or superior to the previous works.

Section 2 outlines our system for subcellular localization, and explains in detail the general techniques for associative classification. Section 3 describes the procedure of extracting appropriate features for subcellular localization prediction. Section 4 describes the classifier construction procedure using our own boosting algorithm. Section 5 shows the experimental results and analyzes the behavior of our system. Finally, Section 6 includes our conclusions and suggestions for future work.

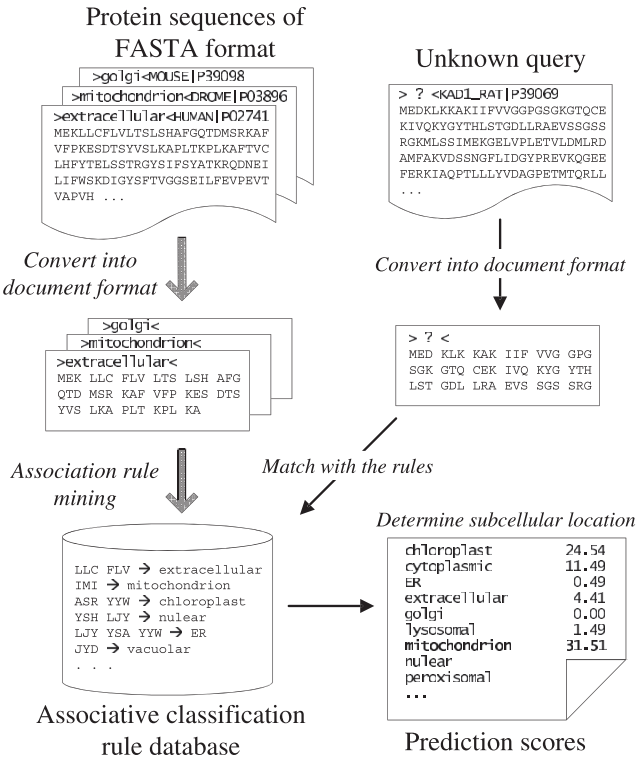


Fig. 1. Flow of localization prediction using association rules.

2 System and Methods

2.1 Overall Flow

This overall procedure is very similar to a document classification which uses both frequent-pattern mining and boosting [24]. First, training examples are collected from protein databases such as Swiss-Prot. An example consists of a primary protein sequence and its subcellular location label (Fig. 1). Protein sequences are divided into fixed-length sequence fragments with subcellular location labels annotated, which can be mapped to words in a document. Frequently occurring sequence fragments and class labels are extracted from the training set, and arranged into associative classification rules. Then, a boosting process is applied to select a smaller number of rules that will constitute the final classifier.

A test protein with unknown subcellular localization is converted into a document format like the case of training examples. Then, the classification rules are applied to the test protein, yielding the prediction scores of all the class labels. After the scores are compared to one another, the localization of the test protein is determined.

2.2 Data Sets

We used three data sets for subcellular localization experiments. The TargetP data set [5] contains 3,678 redundancy-reduced protein sequences from the Swiss-Prot database. We also used its redundant version which contains 11,794 sequences. Plant proteins are annotated with four location labels: chloroplast (chl), mitochondrion (mit), secretory pathway (SP), and other. Nonplant proteins are annotated with three locations: mitochondrion, secretory pathway, and *other*. The *other* category consists of cytoplasmic and nuclear locations.

The MultiLOC data set [15] contains animal, fungal, and plant protein sequences from the Swiss-Prot release 42. The plant proteins have 10 subcellular locations: chloroplast, cytoplasm (cyt), endoplasmic reticulum (er), extracellular space (ext), Golgi apparatus (gol), mitochondrion, nucleus (nuc), peroxisome (per), plasma membrane (pm), and vacuole (vac). Animal proteins lack chloroplasts, and have lysosomes (lys) instead of vacuoles.

We introduced an up-to-date version of UniProtKB[1] protein knowledgebase, the January-2011 Swiss-prot protein data set. We wanted to make up an independent data set which is different from the one used in the previous researches. The protein sequences which already exist in the MultiLOC data set were removed. Plant and animal protein sequences with the CC field of "SUBCELLULAR LOCATION:" were extracted from the knowledgebase. Then, they were assigned subcellular localization labels, according to the rules by which the authors of MultiLOC assigned the sequences [15]. The knowledgebase data were transformed into the FASTA format. As the resulting data contained many homologous sequences, they were redundancy reduced using the ClustalW alignment tool [25]. A total of 13,591 homology-reduced sequences with a pairwise similarity not greater than 80 percent were obtained (Table 1).

In the training phase, primary protein sequences in the data sets are divided into k-character pseudowords or k-mer substrings. In case of $k = 3$, a separated protein sequence would look like this

mitochondrion MLR QII GQA KKH PSL IPL FVF
IGT GAT GAT LYL LRL ALF NPD VCW DRN NPE
PWN KLG PND QYK FYS VNV DYS KLK KER PDF

1. http://www.uniprot.org.

TABLE 1
The Number of Protein Sequences Extracted from Jan-2011 ver of UniProtKB Database

Location	Homology-conserved	Homology-reduced
chloroplast	1654	981
cytoplasm	4009	3133
endoplasmic reticulum	801	442
extracellular space	5200	3243
Golgi apparatus	1043	563
lysosome	282	147
mitochondria	2824	1728
nucleus	907	820
peroxisome	430	262
plasma membrane	3329	2218
vacuole	109	54
Total	20588	13591

To extract more features, a sliding window is applied to the original sequences to obtain two additional sets of 3-mers:

mitochondrion M LRQ IIG QAK KHP SLI PLF VFI GTG ATG ATL YLL RLA LFN PDV CWD RNN PEP WNK LGP NDQ YKF YSV NVD YSK LKK ERP DF

mitochondrion ML RQI IGQ AKK HPS LIP LFV FIG TGA TGA TLY LLR LAL FNP DVC WDR NNP EPW NKL GPN DQY KFY SVN VDY SKL KKE RPD F

2.3 Association Rule Mining

Association rules were devised to represent a degree of association between the items in a transactional log database [26]. A protein sequence corresponds to a transactional log record, and its sequence fragments to the items of the transaction. Let the number of amino acids be 20, the length of a sequence fragment 3, and the average length of protein sequences L. Then, the dimensionality d of the space of a protein sequence example is $\lceil L/3 \rceil$. Let the set of the converted protein sequences be $X = X_1 \times X_2 \times \cdots \times X_d$, where $X_j = \{AAA, AAB, AAC, \ldots\}$ is a set of 3-mer sequence fragments. The set of training examples D can be written as

$$D = \{(x_j, y_j) \mid \ x_j \in X, \ y_j \in Y\},$$

where Y is the set of subcellular location labels.

An class association rule is an association rule whose consequence (the right-hand side of "$\rightarrow$ ") is a class label

$$\{A_1, A_2, \ldots, A_i, \ldots, A_k\} \rightarrow Y, \tag{1}$$

where $A_i \in X_j$. k is called the order of the rule. The support of a rule is the number of training examples in which the pattern and the class label of the rule occur. An association rule r_i is frequent if the support of the rule exceeds some given threshold value min_sup. The confidence of the rule is defined as

$$conf(r_i) = sup(r_i)/sup(\text{the condition of } r_i).$$

A rule is accurate if the confidence of the rule exceeds a given threshold min_conf.

Frequent-item set mining algorithms generate association rules which pass both min_sup and min_conf thresholds.

TABLE 2
Classification Rules for the Prediction Example

id	rule	sup	conf
r_1	$\{a\} \rightarrow \alpha$	20	0.3
r_2	$\{a, b\} \rightarrow \alpha$	5	0.49
r_3	$\{c\} \rightarrow \beta$	44	0.5
r_4	$\{b, f\} \rightarrow \gamma$	6	0.7
r_5	$\{c, d, e, f\} \rightarrow \gamma$	3	1.0
r_6	$\{e\} \rightarrow \delta$	23	0.27
r_7	$\{a, e\} \rightarrow \delta$	11	0.75
r_8	$\{c, d, e\} \rightarrow \delta$	4	0.6

sup and conf denote the support and the confidence of r_i.

We modified the rule mining algorithm of [27] to generate a large number of association rules more efficiently. A mined association rule would look like this

$$KKH, \ \ LRL \rightarrow mitochondrion \ (20, 0.3), \tag{2}$$

where 20 and 0.3 denote the support and the confidence of the rule, respectively. This rule means that the protein sequences with subsequences "KKH" and "LRL" and location label mitochondrion occur 20 times in the training set, and 30 percent of the sequences with "KKH" and "LRL" have mitochondrion as their location label. The confidence of an association rule denotes the probability of the location label given the amino acid subsequences.

2.4 Associative Classification

Numerous powerful classification methods exploiting association rules have been proposed [21], [28], [29]. Our method is different from those methods: it generates a huge number of association rules and constructs a final classifier through boosting the association rules (Section 4). Let $R = \{r_1, r_2, \ldots, r_{|R|}\}$ be the final classification rule set. When we have a test example x, we apply the rules to x. Let s_{ij} be the confidence of rule r_i that the class label of x is c_j. Then, S_j, the total score for c_j after all rules are applied, can be written as

$$S_j = \sum_{r_i \in R \text{ and } c_j \in Y} s_{ij}. \tag{3}$$

Then, the final prediction label $\hat{c}$ for x is determined such that

$$\hat{c} = \arg\max_{c_j \in Y} S_j.$$

As a simple example for this prediction procedure, let a set of items $I = \{a, b, c, d, e, f\}$ and a set of class labels $Y = \{\alpha, \beta, \gamma, \delta\}$. min_sup and min_conf are set to 3 and 0.25, respectively. After classifier construction, final classification rules are produced (Table 2).

Assume a test instance x to include items $\{a, b, e, f\}$. When the rules in Table 2 are applied to x, the pattern of a rule is examined to determine whether it is a subset of the pattern of x, $\{a, b, e, f\}$. In this way, r_1, r_2, r_4, r_6, and r_7 will be matched to x. Then, the scores of class labels $\alpha, \ldots, \delta$ are calculated by summing the scores of the rules. (The confidence of a matching rule is used as the prediction score.) For example, the score for label α is 0.79 (the sum of the confidences of rules r_1 and r_2). Finally, we obtain a prediction score vector for Y

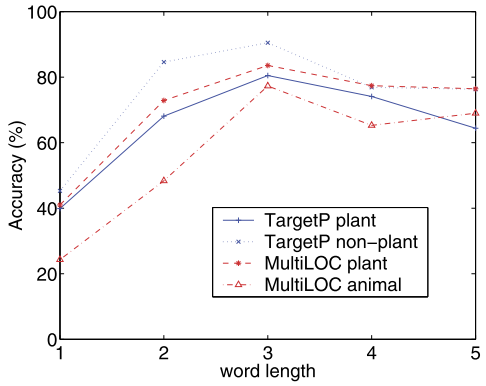


Fig. 2. Prediction accuracy versus subsequence length k.

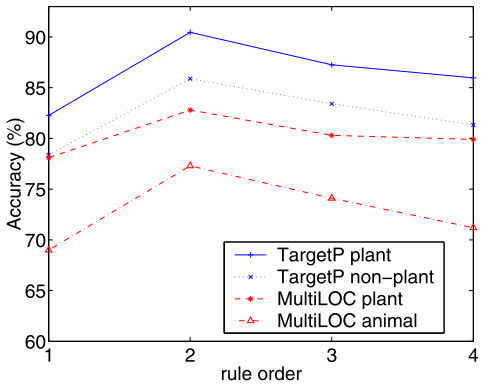


Fig. 3. Prediction accuracy versus maximum rule order.

$$(0.79 \quad 0.0 \quad 0.7 \quad 1.02),$$

where each dimension denotes a class of Y. δ has the highest score (1.02), so it is predicted to be the class label of x.

2.5 Comparison with Other Methods

Previous methods have also used fixed-length subsequences in predicting subcellular localizations. Hawkins et al. [11] used different lengths of k-mers which were arguments to the spectrum kernel function of SVM for predicting nuclear localizations. Eskin and Agichtein [10] used fixed-length k-mers in the sparse kernel of SVM. However, with k-mer features which the kernel functions uses as a whole, it is difficult to give prediction weights to specific subsequences. On the other hand, our method can attach weights to subsequence features for more accurate classification. To complement the subsequence features, some SVM-based methods combined other features such as sequence motifs [11], [15] and text annotations [10], [18] from knowledge bases, while our method uses only pure sequence information.

For multiclass classification, the SVM-based methods should apply a series of SVM classifiers which are trained with individual training sets. On the other hand, our method can perform multiclass classification through training and predicting only once. Even with simpler features and prediction procedures, our approach can obtain an equivalent or better classification performance than other methods. Proteins may exist at more than one subcellular location simultaneously, or shuttle between two locations. Chou and Shen developed a classifier which can handle proteins with multiple sites [19], [30], and Hawkins et al. a prediction model to identify dual localized proteins [11]. Although our method currently conducts the single-label classification (one subcellular location per protein), it can easily be extended to the multilabel classification [24].

3 Determining the Parameters

Selecting the length of subsequences has a trade-off between the specificity (Spec) and the coverage of the induced predictor. As the fragment length increases, the specificity grows but the coverage drops. To find an optimal length, we conducted simulations in which length k was varied, using the TargetP and MultiLOC data set (Fig. 2). We selected randomly one tenth of sequences from each data set to use as simulation data. The y-axis of the figure denotes averaged prediction accuracy over the entire subcellular locations (for other experimental settings, refer to Section 5). Subsequences with three amino acid residues had the best prediction performance in this simulation. Combinations of different values of k-mer substring sets were also tried: for example, the combined set of 2-mer and 3-mer substring sets. But they failed to show better prediction accuracy than the 3-mer set (results not shown).

Another factor which affects prediction accuracy is how many subsequences should be included in the condition part of association rules (the order of rules). As the order of rules increases, the specificity of the classifier should be raised. However, the number of rules generated during the rule mining step increases exponentially with the order of the rules, which makes the computation intractable. In addition, high-order rules suffer from poor coverage for test proteins. We conducted simulations to determine an optimal maximum order of classifying rules (Fig. 3). The case in which the rule order is less than or equal to two had the best performance. No matter whether the type of data is biological sequences or text words [24], the maximum rule order in our approach is chosen as not exceeding four.

The length of primary sequence input L is related to both the prediction accuracy and the computation efficiency. If the input sequence is long, the classification performance increases accordingly. However, if the length exceeds a certain point, the classification accuracy starts to saturate. The related simulation results are shown in Fig. 6. In our method, the maximum input sequence length is set to no longer than 500.

4 Boosting Prediction Accuracy

Boosting is a meta learning algorithm, which improves or boosts the prediction ability of individually weak predictors, by assigning them appropriate weights and combining them into a final strong predictor [31], [32]. The prediction ability of weak hypotheses is at least better than random guessing. Our frequent pattern mining method intentionally generates a large number of classification rules so that it can include modest quality rules as well as very accurate rules, by adjusting min_sup to 3 (a very small quantity) and min_conf to $1/m$ (m is the number of subcellular location

Algorithm BCAR
Input Training database: $D_0 = \{(x_i, y_i)\}_{i=1}^N$,
Class Association Rules: $R = \{r_1, r_2, ..., r_T\}$,
Example weight threshold: θ
Initialize weight vector: $w_i^0 \leftarrow 1$ for $i = 1, ..., N$,
final rule set: $H \leftarrow \{\}$,
sort the rules R to the confidence values
For $t = 1, 2, ..., T$
For $i = 1, 2, ..., |D_{t-1}|$
1) Apply $r_t : x_t \rightarrow y_t$ to (x_i, y_i).
2) If r_t classifies correctly, then
a) Select r_t as a member of the final rule set H: $H \leftarrow H \cup \{r_t\}$
b) Update the weight of the example: $w_i^t \leftarrow w_i^{t-1} \exp[-\text{conf}(r_t)]$
c) If $w_i^t < \theta$ then delete (x_i, y_i) from the training set D_{t-1}: $D_t \leftarrow D_{t-1} - \{(x_i, y_i)\}$
Output the final hypothesis

$$h_f(x) = \arg\max_{y \in Y} \sum_{\phi:\, r_\phi \in H,\, r_\phi(x) = y} \text{conf}(r_\phi)$$

Fig. 4. Algorithm to induce the classifier using boosting.

labels). Simulation results verified that these selections produce the best classification accuracy [24].

Our boosting algorithm, Boosting Class Association Rules (BCAR), is shown in Fig. 4. Before the main boosting iterations, input association rules are sorted in the order of descending confidence. If the confidence values are equal, then the rule with higher support value has a higher rank. The weights of training examples are stored in the weight vector w. The weight of example (x_i, y_i) at round t is denoted as w_i^t.

In the inner For loop, r_t is applied to all training examples $(x_i, y_i) \in D_{t-1}$ (line 1). If r_t classifies an example correctly, then r_t is included in the set of the final classifier H (line 2a) and the weight of the example is decreased (line 2b). If the weight is less than the weight threshold θ, the example is deleted from the database (line 2c). This changes the distribution of the training examples. θ controls the rule selection process globally. The optimal value of θ is determined empirically using the simulation data which were used for determining the parameters (Section 3). In the implementation of BCAR, $\theta' = -\ln\theta$ is used instead of θ. For example, θ' was set to 300 for the TargetP plant data. (Table 3).

Line 1-2a of the algorithm corresponds to the weak learner of the original boosting algorithm [31]. The change in the distribution of the training examples determines whether the next rule r_{t+1} be selected or not (line 2b-c). This weak learning process is very simple and efficient, because it does not regenerate association rules but just selects a rule using the reduced set of training examples. The completeness of the training algorithm of BCAR is explained in the following theorem.[2]

TABLE 3
Simulation to Determine Optimal Values for θ'

Data	10	50	100	300	500	700
TargetP plant	81.8	82.2	84.3	85.3	81.7	81.0
non-plant	80.6	86.5	90.2	91.4	91.8	91.5
MultiLOC plant	81.2	82.4	83.1	83.0	82.8	81.9
animal	68.4	76.6	77.5	76.6	74.7	74.5

TABLE 4
The Number of Classification Rules before and after Boosting

Data		Before Boosting		After Boosting	
		#rules	Acc	#rules	Acc
TargetP	plant	6,105,192	78.7	356,623	85.4
	non-plant	2,057,639	90.1	527,972	91.5
MultiLOC	plant	5,047,055	72.0	305,566	83.0
	animal	20,004,345	64.2	212,158	77.5

Theorem 1. *Suppose BCAR chooses the class association rule whose error $\varepsilon_t \leq 1/2 - \gamma$ for some $\gamma > 0$ at each round t $(t = 1, \ldots, T)$. Then, the error ε of the final hypothesis h_f output by BCAR is bounded above by*

$$\varepsilon = \frac{|\{i : h_f(x_i) \neq y_i\}|}{N} \leq \exp(-T\gamma^2/2). \quad (4)$$

The proof and further analysis of the algorithm can be found in [24].

After finishing the iteration for all rules $\{r_t\}$, a small set of final rules H is induced ($|H| \ll |R|$). The final hypothesis h_f outputs the label of the largest score as the estimate of y. Boosting algorithms reduce the generalization error as well as the training error [31], [32], and never fall in overfitting but keep reducing generalization error if it can be provided with weak classifiers steadily [33]. Table 4 shows prediction results using one subset of 5-fold cross-validation data as testing instances to show the effect of boosting. The numbers of class association rules and the prediction accuracies before and after executing the BCAR algorithm are listed. While boosting reduces the number of classification rules, it improves generalization performance greatly.

5 Results and Discussion

5.1 Experimental Setup

The jackknife test is deemed to be the most objective way of examining the accuracy of a statistical prediction method [20], [34], [35], and has been widely adopted to test various prediction methods [36]. Instead, to reduce the computational time, we adopted the 5-fold cross-validation as the previous works [5], [15], [18] did in their experiments using Neural Network or SVM as a prediction engine. After removing the one tenth of the original data that are reserved for simulation to determine parameters, the remaining data are divided into five subsets, four subsets for training and one for testing.

Before being input to the rule mining process, protein sequences in FASTA format undergo some preprocessing steps. The sequences containing B, Z, or X amino acid residues were excluded to avoid noisy information. Previous methods [4], [5], [15] extracted input subsequences

2. This theorem is for a binary-class version of BCAR.

TABLE 5
Prediction Results of Different Predictors Using MultiLOC Data

Subsets	Location	MultiLOC			SherLOC			BCAR		
		Spec	Sens	MCC	Spec	Sens	MCC	Spec	Sens	MCC
Plant	chl	0.85	0.88	0.85	0.91	0.94	0.92	0.82	0.87	0.76
	cyt	0.85	0.68	0.70	0.91	0.81	0.82	0.88	0.70	0.76
	er	0.54	0.72	0.61	0.63	0.82	0.71	0.86	0.86	0.85
	ext	0.81	0.68	0.70	0.90	0.84	0.84	0.67	0.33	0.46
	gol	0.41	0.75	0.54	0.61	0.84	0.70	1.00	0.67	0.81
	mit	0.79	0.85	0.80	0.88	0.90	0.88	0.67	0.67	0.65
	nuc	0.75	0.82	0.75	0.85	0.89	0.85	0.87	0.93	0.84
	per	0.34	0.71	0.47	0.59	0.85	0.70	0.67	0.67	0.66
	pm	0.89	0.74	0.77	0.96	0.84	0.87	0.60	0.75	0.67
	vac	0.20	0.70	0.36	0.29	0.83	0.48	0.86	0.67	0.75
	(avg)	0.64			0.75			0.79		
	Acc[%]	74.6 (±0.8)			85.1 (±1.1)			83.1 (±1.9)		
Animal	cyt	0.85	0.67	0.68	0.91	0.83	0.82	0.79	0.70	0.67
	er	0.56	0.68	0.60	0.67	0.82	0.73	0.75	0.62	0.68
	ext	0.83	0.79	0.77	0.90	0.86	0.86	0.69	0.87	0.71
	gol	0.36	0.69	0.48	0.55	0.86	0.68	0.99	0.68	0.82
	lys	0.43	0.71	0.53	0.65	0.87	0.74	0.86	0.60	0.71
	mit	0.82	0.88	0.83	0.91	0.93	0.91	0.57	0.72	0.61
	nuc	0.73	0.82	0.73	0.83	0.89	0.84	0.86	0.48	0.62
	per	0.31	0.71	0.44	0.68	0.89	0.77	0.67	0.71	0.68
	pm	0.90	0.73	0.76	0.95	0.85	0.87	0.87	0.90	0.83
	(avg)	0.64			0.78			0.78		
	Acc[%]	74.6 (±1.0)			86.2 (±0.9)			77.5 (±1.1)		

from the N-terminal position. On the other hand, we extracted L-length amino acid residues starting from N-terminal. As quite long sequences may lower the computational efficiency, it is necessary to put a limit to the input length L. However, L is chosen so large that most part of the original protein sequences cannot truncated by this confinement. The input sequence of length L is divided into k-mer subsequences and indexed using a document-indexing library (We used the BOW toolkit [37]).

TABLE 6
Prediction Results of BCAR Using the Jan-2011 Swiss-Prot Data

Subsets	Location	Spec	Sens	MCC
Plant	chl	0.67	0.91	0.59
	cyt	0.82	0.42	0.55
	er	0.45	0.45	0.44
	ext	0.78	0.83	0.79
	gol	1.00	0.59	0.76
	mit	0.77	0.49	0.56
	nuc	0.33	0.33	0.32
	per	0.71	0.45	0.56
	pm	0.98	0.91	0.94
	vac	0.89	0.73	0.80
	(avg)	0.74		
	Acc[%]	73.3 (±0.9)		
Animal	cyt	0.65	0.60	0.51
	er	0.83	0.64	0.72
	ext	0.60	0.90	0.61
	gol	0.93	0.69	0.79
	lys	0.65	0.67	0.65
	mit	0.89	0.70	0.76
	nuc	0.26	0.21	0.18
	per	0.61	0.73	0.66
	pm	0.88	0.56	0.66
	(avg)	0.70		
	Acc[%]	67.1 (±1.1)		

The prediction result of each subcellular location was evaluated with three measures: specificity, sensitivity (Sens), and Matthews correlation coefficient (MCC)

$$Spec = \frac{TP}{TP+FP}, \quad Sens = \frac{TP}{TP+FN},$$
$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FN)(TP+FP)(TN+FN)(TN+FP)}}.$$

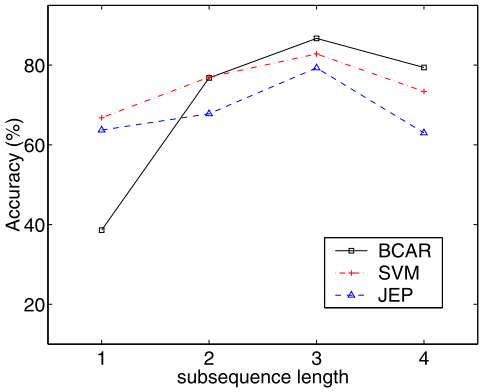
Here, TP, TN, FP, and FN mean the numbers of true positive, true negative, false positive and false negatives, respectively. The overall performance on a whole data set was evaluated using accuracy, which is defined as

$$Acc = \frac{\sum_{j=1}^{j=M} TP_j}{\sum_{j=1}^{j=M} (TP_j + FN_j)},$$

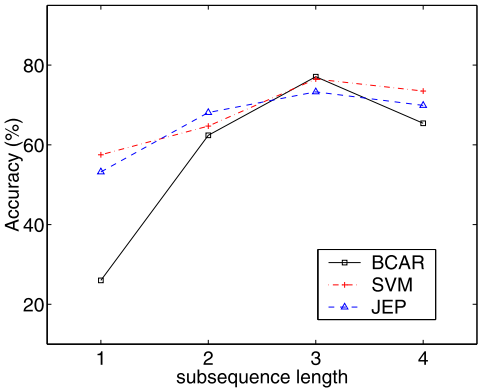
where j is location index, and M is the number of total locations. Another overall measure we used was the averaged specificity which was defined by $\sum_{j=1}^{j=M} Spec_j$. The codes for our prediction system were written with C++ and Perl. The program was run on a Linux machine with 4 GB memory and 2.8 GHz CPU speed.

TABLE 7
Comparison with MultiLOC Using the Jan-2011 Swiss-Prot Data, Measured in the Accuracy (Percent)

Subsets	MultiLOC	BCAR
Plant	37.1	41.2
Animal	46.0	47.1



(a) MultiLOC plant data



(b) MultiLOC animal data

Fig. 5. Comparison of the performances with SVM and JEP classifier.

5.2 Prediction Results

5.2.1 MultiLOC Data

The prediction results for the MultiLOC plant and nonplant data set are shown in Table 5. The first 500 amino acid residues from the N-terminus were used as features. We compared the prediction result of BCAR with those of the MultiLOC [15] and SherLOC [18] predictors. Both MultiLOC and SherLOC are integrated systems of classifiers. They comprise several subclassifiers, each of which uses different kind of features. In addition to all the subclassifiers of MultiLOC, SherLOC has another subclassifier which uses text annotations of the PubMed articles.

In the plant subset, SherLOC has a strong point in the sensitivities for the chl, cyt, ext, mit, per, pm, and vac locations. BCAR had distinctly better performance than MultiLOC and SherLOC predictors for the er, nuc, and per locations, which do not have sorting signals in specific positions of a primary sequence. In the animal subset, SherLOC had strength in sensitivities, as it uses the text features of the homologues when newly discovered proteins have no text annotations. With the abundant frequent subsequence features and the boosting technique, BCAR showed relative strength in the specificities. The computation time during the learning phase of the plant classifier is about 72 minutes, most of which is consumed by the rule generation step.

BCAR showed a more promising result in the overall prediction performance. Based upon statistical significance tests (t-test with 95 percent significance level), BCAR outperformed the MultiLOC predictor in terms of both the average specificity and the accuracy. In comparison with SherLOC, BCAR showed better specificity in the plant subset and equal specificity in the animal subset. In the overall accuracy of the plant subset, BCAR (83.1 percent) approached SherLOC (85.1 percent). In the animal subset, the accuracy of BCAR (77.5 percent) did not surpass the performance of SherLOC. However, the comparison with the SherLOC can be regarded as unfair. BCAR uses only sequence information, while SherLOC uses text annotaions as well as sequence features. Even MultiLOC includes the motif information from the knowledge bases into prediction features.

5.2.2 Jan-2011 Swiss-Prot Data

We performed another prediction experiment with a new protein data, which were extracted from the January 2011 version of the Uniprot knowledge base. As described in Section 2.2, these protein sequences do not occur in the MultiLOC data set. The number of the sequences of the Jan-2011 set is more than twice of the MultiLOC set. We conducted a 5-fold cross-validation prediction of the subcellular localizations (Table 6). The new data set can be freely downloaded at our BCAR online website.

Using this independent data set, we compared the performance of BCAR with the MultiLOC predictor (Table 7). Both predictors were trained using the MultiLOC data set. For the test set, we selected randomly 80 sequences from the Jan-2011 data set for each plant and animal test set. Five sequences are chosen from er, gol, per, vac, lys locations, 10 sequences from the rest locations. The performance of both predictors was measured in the accuracy (percent). The prediction results of MultiLOC were acquired through their website.[3] These overall performances are shown much lower than the other 5-fold cross-validation results, because the MultiLOC training data are diverged far away in timeline from the Jan-2011 data set. The accuracy of BCAR slightly leads that of the MultiLOC predictor.

5.2.3 SVM and Jumping Emerging Patterns (JEP) Classifier

We supplemented our experiments with applying other types of classifiers to k-mer ($k = 1, 2, 3, 4$) subsequence features. In addition to Support Vector Machine, we considered Jumping Emerging Patterns classifiers [38]. While JEP classifiers also extract a large number of patterns which are associated with classes, they differ in the mining methods from the associative classifiers. We used the JEP classifier which was implemented by Gambin and Walczak [39]. Their system can handle only binary classes; thus, we executed a series of JEP classifiers and combined the results for the subcellular localization predictions. We conducted a feature selection using information gain, and chose the highest 100 k-mer subsequences as the input features of JEP. For the SVM classifiers, we used the implementation of the

3. http://abi.inf.uni-tuebingen.de/Services/MultiLoc.

TABLE 8
Greatly Contributing k-mers to the Prediction of Selected Subcellular Localizations

Localization	k-mers ($k <= 3$)
nucleus	WWQ, AWW, CWE, FE, WWP, QWA, GP, ACC, CLW, MHT, SL, MHM, LF
endoplasmic reticulum	QWF, QCC, EL, WIV, DWD, GWE
peroxisome	FHW, KTW, YRH, MHR, SM, WSQ, RSW, PTC, PEW, TM
plasma membrane	TWP, HHN, WPE, QRH, WA

BOW toolkit. MultiLoc plant and animal data of Section 5.2.1 were used for this experiment (Fig. 5).

On the whole, BCAR showed a better performance compared with SVM and JEP. Besides comparing the results of BCAR with those of SVM and JEP classifiers, the purpose of this experiment is to investigate the change of prediction performance when varying the subsequence length k. BCAR is more sensitive to the length of subsequence features. When using only the composition information of single amino acids ($k = 1$), its classification accuracy is very poor. The all three classifiers were at the peak of their performance when k is 3. Thus, 3-mer amino acid residues are considered the most powerful features in the subcellular localization classification.

5.3 Discussion

The boosting algorithm is based on the statistical learning theory. We redesigned the algorithm so that it can be efficiently applied to a real-world classification problem such as subcellular localization, without losing its excellent predicting performance. Although other integrative methods used several types of features, our method showed better performance using only k-mer sequences. All of these k-mers do not have as good prediction ability as the biological features like the sequence motifs or the sorting signals. However, when a huge number of such weak rules are combined using our boosting algorithm, our method can produce a prominent classification performance. Among k-mers which our boosting algorithm produces, there are some strong ones that might be thought of biological motifs. In Table 8, we list a small number of k-mers that contribute greatly to the prediction when using the MultiLOC plant data.

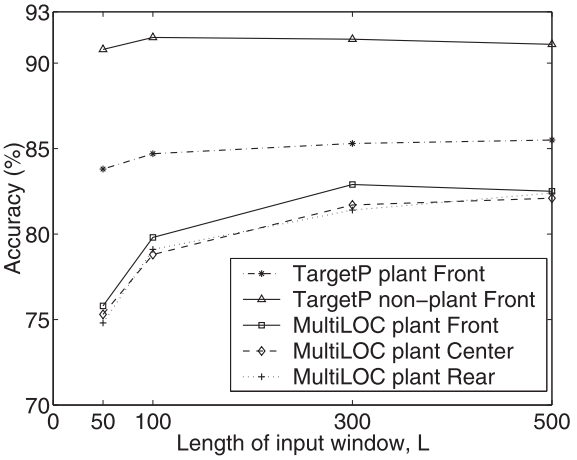


Fig. 6. Change in the classification accuracy for different sequence lengths and positions in a primary sequence.

We listed the scores of Specificity, Sensitivity, MCC, and Accuracy before and after boosting in Table 9. The specificity before boosting, averaged over the entire locations, is about the same as that of after boosting (the last line of Table 9). However, the sensitivity and the MCC after boosting are far better than before boosting. Before boosting, there is unbalance between the performances of individual localization predictors. Thus, a few values of the specificities can be higher than those after boosting. Through boosting, these uneven performances become

TABLE 9
Comparison of the Performances before and after Boosting, Performed with 5-fold Cross-Validations

Subsets	Location	Specificity		Sensitivity		MCC		Acc(%)	
		bef-boost	aft-boost	bef-boost	aft-boost	bef-boost	aft-boost	bef-boost	aft-boost
Plant	chl	0.59	0.81	0.89	0.89	0.60	0.77		
	cyt	0.91	0.82	0.53	0.67	0.67	0.70		
	er	0.88	0.91	0.47	0.76	0.64	0.83		
	ext	0.33	0.53	0.05	0.21	0.07	0.31		
	gol	0.92	1.00	0.50	0.73	0.67	0.84		
	mit	0.92	0.60	0.23	0.61	0.47	0.57		
	nuc	0.86	0.90	0.78	0.95	0.77	0.88		
	per	0.92	0.93	0.53	0.50	0.69	0.67		
	pm	0.92	0.87	0.37	0.63	0.58	0.73		
	vac	0.83	0.82	0.36	0.77	0.54	0.78	74.3	83.1
	(avg)	0.81	0.82	0.46	0.67	0.57	0.71		
Animal	cyt	0.57	0.76	0.72	0.73	0.54	0.66		
	er	0.85	0.76	0.30	0.64	0.50	0.69		
	ext	0.87	0.71	0.28	0.82	0.47	0.70		
	gol	0.87	0.99	0.20	0.61	0.43	0.76		
	lys	0.85	0.90	0.26	0.63	0.48	0.75		
	mit	0.87	0.60	0.55	0.71	0.67	0.63		
	nuc	0.79	0.87	0.44	0.62	0.56	0.71		
	per	0.82	0.63	0.16	0.49	0.38	0.55		
	pm	0.53	0.84	0.83	0.88	0.55	0.80	64.7	77.1
	(avg)	0.78	0.78	0.42	0.68	0.51	0.69		

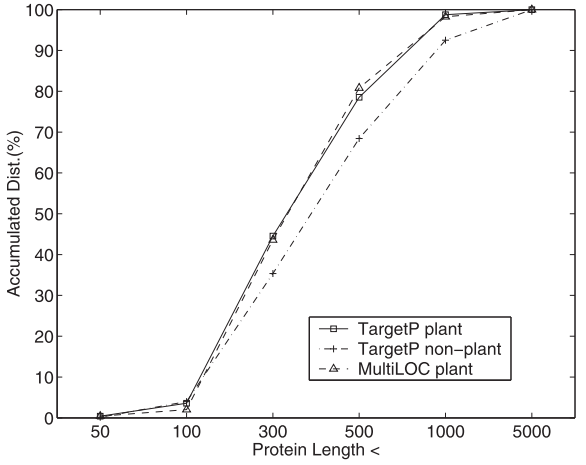


Fig. 7. The accumulated distribution of the length of the training proteins.

normalized, and at the same time, the overall prediction performance (the accuracy) is improved remarkably.

Previous researches concentrate on a specific part of the primary sequence (e.g., the N-terminal sorting signals), or on a small number of sequence motifs which were verified through biological experiments. On the other hand, we collected association rules from a broader range of positions in a primary sequence. Fig. 6 shows average prediction accuracy versus the maximum length of input subsequences. In the case of TargetP data, the first L residues starting from the N-terminus were used. The accuracy increases with the first 100 residues. As the sequence length becomes larger than 100, the accuracy either stops increasing or increases very slowly. This indicates that most discriminating features in TargetP data are concentrated in the first N-terminal section.

The accuracy changes in MultiLOC plant data were plotted for three different subsets of input subsequences. These input sequences have different starting positions in the primary sequence: front, center, and rear. (For example, the 100 point of the x-axis in the MultiLOC plant Rear line in Fig. 6 represents a 100-residue subsequence which starts from the C-terminus of the primary sequence.) Up to 100 residues, the accuracy increases rapidly. For lengths larger than 100, the accuracy improves constantly although the rate is reduced a little. This indicates that the N-terminal targeting peptides are not the only sequence features which help to raise the accuracy. More interestingly, the accuracies of the subsequences from the center or rear region are little lower than that of the subsequences from the front region (the N-terminus).

The accumulated distribution of the length of the training proteins also supports the above argument. Fig. 7 shows the distribution for MultoLOC plant, TargetP plant, and non-plant training set, which displays shares of different protein lengths in the training set. Eighty percent of the proteins have sequence lengths shorter than 500. In the case of the training set with 500 protein length (Fig. 6), most 3-mers extracted from the C-terminus were used in the prediction of localization. This implies that 3-mers selected from the C-terminus hold equal predicting power with 3-mers from the N-terminus.

6 Conclusion

BCAR uses only the amino acid sequence information of proteins for localization prediction. It does not need other types of features from the motif database or the text knowledge bases. After mining association rules from protein sequences, BCAR boosts those rules so as to induce a more general and accurate localization predictor. Many experiments on benchmark data sets used in previous studies show that BCAR has an equivalent or better prediction performance than the state-of-the-art results. BCAR is very simple for practical use, compared with other prediction methods that consist of a set of many SVM's or neural networks.

Our method could be well applied to newly synthesized proteins which have no known information other than amino acid sequence. We plan to raise the prediction accuracy through combining other types of classifiers which use text annotations or sorting signals as features. The target for localization prediction will be extended to proteins with multiple locations. Our method could also be applied to other biological prediction tasks such as the classification of protein structures and protein-protein interactions.

Acknowledgments

The authors thank Torsten Blum for providing with the redundant data set of MultiLOC and Tomasz Gambin for the source code of the JEP classifier. This research was supported by The Ministry of Knowledge Economy (MKE), Korea, under the Information Technology Research Center (ITRC) support program supervised by the National IT Industry Promotion Agency (NIPA) (NIPA-2011-C1090-1131-0009).

References

[1] B. Eisenhaber and P. Bork, "Wanted: Subcellular Localization of Proteins Based on Sequence," *Trends in Cell Biology*, vol. 9, pp. 169-170, 1998.

[2] G. Schatz and B. Dobberstein, "Common Principles of Protein Translocation across Membranes," *Science*, vol. 271, no. 5255, pp. 1519-1526, 1996.

[3] K. Nakai, "Protein Sorting Signals and Prediction of Subcellular Localization," *Advances in Protein Chemistry*, vol. 54, pp. 277-344, 2000.

[4] K. Nakai and M. Kanehisa, "A Knowledge Base for Predicting Protein Localization Sites in Eukaryotic Cells," *Genomics*, vol. 14, pp. 897-911, 1992.

[5] O. Emanuelsson, H. Nielsen, S. Brunak, and G. von Heijne, "Predicting Subcellular Localization of Proteins Based on Their n-Terminal Amino Acid Sequence," *J. Molecular Biology*, vol. 300, pp. 1005-1016, 2000.

[6] H. Bannai, Y. Tamada, O. Maruyama, K. Nakai, and S. Miyano, "Extensive Feature Detection of N-Terminal Protein Sorting Signals," *Bioinformatics*, vol. 18, no. 2, pp. 298-305, 2002.

[7] K.-J. Park and M. Kanehisa, "Prediction of Protein Subcellular Locations by Support Vector Machines Using Compositions of Amino Acids and Amino Acid Pairs," *Bioinformatics*, vol. 19, no. 13, pp. 1656-1663, 2003.

[8] Y.-D. Cai and K.-C. Chou, "Predicting Subcellular Localization of Proteins in a Hybridization Space," *Bioinformatics*, vol. 20, no. 7, pp. 1151-1156, 2004.

[9] K.C. Chou and D.W. Elrod, "Protein Subcellular Location Prediction," *Protein Eng.*, vol. 12, pp. 107-118, 1999.

[10] E. Eskin and E. Agichtein, "Combining Text Mining and Sequence Analysis to Discover Protein Functional Regions," *Proc. Pacific Symp. Biocomputing*, pp. 288-299, 2004.

[11] J. Hawkins, L. Davis, and M. Boden, "Predicting Nuclear Localization," *J. Proteome Research*, vol. 6, pp. 1402-1409, 2007.

[12] R. Nair and B. Rost, "Inferring Sub-Cellular Localization through Automated Lexical Analysis," *Bioinformatics,* vol. 28, pp. S78-S86, 2002.

[13] A. Bairoch and R. Apweiler, "The Swiss-Prot Protein Sequence Database and Its Supplement Trembl in 2000," *Nucleic Acids Research,* vol. 28, no. 1, pp. 45-48, 2000.

[14] Z. Lu, D. Szafron, R. Greiner, P. Lu, D.S. Wishart, B. Poulin, J. Anvik, C. Macdonell, and R. Eisner, "Predicting Subcellular Localization of Proteins Using Machine-Learned Classifiers," *Bioinformatics,* vol. 20, no. 4, pp. 547-556, 2004.

[15] A. Höglund, P. Dönnes, T. Blum, H.-W. Adolph, and O. Kohlbacher, "Multiloc: Prediction of Protein Subcellular Localization Using N-Terminal Targeting Sequences, Sequence Motifs and Amino Acid Composition," *Bioinformatics,* vol. 22, no. 10, pp. 1158-1165, 2006.

[16] R. Nair, P. Carter, and B. Rost, "NLSdb: Database of Nuclear Localization Signals," *Nucleic Acids Research,* vol. 31, pp. 397-399, 2003.

[17] A. Bairoch and P. Bucher, "Prosite: Recent Developments," *Nucleic Acids Research,* vol. 22, pp. 3583-3589, 1994.

[18] H. Shatkay, A. Höglund, S. Brady, T. Blum, P. Dönnes, and O. Kohlbacher, "Sherloc: High-Accuracy Prediction of Protein Localization by Integrating Text and Protein Sequence Data," *Bioinformatics,* vol. 23, no. 11, pp. 1410-1417, 2007.

[19] K.-C. Chou and H.-B. Shen, "Hum-PLoc: A Novel Ensemble Classifier for Predicting Human Protein Subcellular Localization," *Biochemical and Biophisical Research Comm.,* vol. 347, pp. 150-157, 2006.

[20] K.-C. Chou and H.-B. Shen, "Review: Recent Progresses in Protein Subcellular Location Prediction," *Analytical Biochemistry,* vol. 370, pp. 1-16, 2007.

[21] B. Liu, W. Hsu, and Y. Ma, "Integrating Classification and Association Rule Mining," *Proc. Int'l Conf. Knowledge Discovery and Data Mining,* pp. 80-86, 1998.

[22] R. She, F. Chen, K. Wang, and M. Ester, "Frequent-Subsequence-Based Prediction of Outer Membrane Proteins," *Proc. Int'l Conf. Data Mining and Knowledge Discovery,* pp. 436-445, 2003.

[23] Y. Jin, B. Niu, K. Feng, W. Lu, Y. Cai, and G. Li, "Predicting Subcellular Localization with Adaboost Learner," *Protein and Peptide Letters,* vol. 15, no. 1, pp. 286-289, 2008.

[24] Y. Yoon and G.G. Lee, "Text Categorization Based on Boosting Association Rules," *Proc. Second IEEE Int'l Conf. Semantic Computing (ICSC 2008),* pp. 136-143, 2008.

[25] M.A. Larkin, G. Blackshields, N.P. Brown, R. Chenna, P.A. Mcgettigan, H. Mcwilliam, F. Valentin, I.M. Wallace, A. Wilm, R. Lopez, J.D. Thompson, T.J. Gibson, and D.G. Higgins, "Clustal w and Clustal x Version 2.0," *Bioinformatics/Computer Applications in the Biosciences,* vol. 23, pp. 2947-2948, 2007.

[26] R. Agrawal, T. Imielinski, and A.N. Swami, "Mining Association Rules between Sets of Items in Large Databases," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* pp. 207-216, 1993.

[27] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* W. Chen, J. Naughton, and P. A. Bernstein, eds., pp. 1-12, 2000.

[28] W. Li, J. Han, and J. Pei, "CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules," *Proc. IEEE Int'l Conf. Data Mining (ICDM),* pp. 369-376, 2001.

[29] J. Wang and G. Karypis, "Harmony: Efficiently Mining the Best Rules for Classification," *Proc. SIAM Int'l Conf. Data Mining (SDM),* 2005.

[30] K.C. Chou and H.B. Shen, "Cell-PLoc: A Package of Web Servers for Predicting Subcellular Localization of Proteins in Various Organisms," *Nature Protocols,* vol. 3, no. 2, pp. 153-162, http://dx.doi.org/10.1038/nprot.2007.494, 2008.

[31] R.E. Schapire, "The Strength of Weak Learnability," *Machine Learning,* vol. 5, pp. 197-227, citeseer.ist.psu.edu/schapire90strength.html, 1990.

[32] Y. Freund and R.E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Computer and System Sciences,* vol. 55, no. 1, pp. 119-139, 1997.

[33] R.E. Schapire, Y. Freund, P. Barlett, and W.S. Lee, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *Proc. Int'l Conf. Machine Learning (ICML),* pp. 322-330, 1997.

[34] C.C.L.Z.C. Zhou, X.B., and X.Y. Zou, "Using Chou's Amphiphilic Pseudo-Amino Acid Composition and Support Vector Machine for Prediction of Enzyme Subfamily Classes," *J. Theoretical Biology,* vol. 248, pp. 546-551, 2007.

[35] K.C. Chou and H.B. Shen, "Foldrate: A Web-Server for Predicting Protein Folding Rates from Primary Sequence," *The Open Bioinformatics J.,* vol. 3, pp. 31-50, 2009.

[36] K.C. Chou and C.T. Zhang, "Review: Prediction of Protein Structural Classes," *Critical Rev. in Biochemistry and Molecular Biology,* vol. 30, pp. 275-349, 1995.

[37] A.K. McCallum, "Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering," http://www.cs.cmu.edu/mccallum/bow, 1996.

[38] J. Li, G. Dong, and K. Ramamohanarao, "Making Use of the Most Expressive Jumping Emerging Patterns for Classification," *Knowledge and Information Systems,* vol. 3, no. 2, pp. 131-145, 2001.

[39] T. Gambin and K. Walczak, "A New Classification Method Using Array Comparative Genome Hybridization Data, Based on the Concept of Limited Jumping Emerging Patterns," *BMC Bioinformatics,* vol. 10(Suppl 1), article S64, 2009.

Yongwook Yoon received the BS degree in mechanical engineering at Yonsei University, Korea in 1990, the MS degree in computer science and the PhD degree in computer science and engineering from the Pohang University of Science and Technology (POSTECH) in 2004 and 2010, respectively. As a senior researcher of the Central R&D Laboratory of Korea Telecom (KT), he has been assigned the projects of data mining and knowledge processing. His research area also includes the natural language processing, machine learning algorithms and its applications to Bioinformatics.

Gary Geunbae Lee received the BS/MS degree in computer engineering from Seoul National University, Korea, and the PhD degree in computer science from UCLA. He has been a professor at CSE department POSTECH in Korea since 1991. His research focuses on human language technology researches including natural language processing, speech recognition/synthesis, speech translation, and web/text mining. He has authored more than 150 papers in international journals, and has served as a technical committee member for several international conferences such as ACL, COLING, IJCAI, SIGIR, IUI, Interspeech, ASRU, and ICASSP. He is a member of the IEEE.