# Data Mining based Approach to Classify Location of a Protein

Dikshitha V
6th Sem, Department of CSE
AMC Engineering college,
Bengaluru, India

Chaitra J
6th Sem, Department of CSE
AMC Engineering college,
Bengaluru, India

Divya V. R
6th Sem, Department of CSE
AMC Engineering college,
Bengaluru, India.

***Abstract* — Computational method for predicting protein subcellular localization was developed using Associative Classification technique of data mining. Protein sequences were modeled as document sets. Approach used was to divide 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 were then mined from the protein sequence examples that range from the N-terminus to the C-terminus. Then, a boosting algorithm was applied to those rules to build up a final classifier. Performance analysis of the system shows that the efficiency of the proposed system is on par with the state-of-art predication techniques.**

***Keywords— Associative Classification, Bioinformatics, Boosting, Data Mining, Subcellular Localization.***

## I. INTRODUCTION

Subcellular localization is a key functional characteristic of proteins. An automatic, reliable and efficient prediction system for protein subcellular localization is needed for large-scale genome analysis. Subcellular location is a key function characteristic of potential gene expressing the protein because the protein functions in the specific location in the intact cells to maintain the cell survival.

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.

The proposed approach in this paper 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 this 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.

## II. LITERATURE SURVEY

To determine the subcellular 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.

### *A. Target signals*

The first class uses targeting signals that reside at a specific part of primary sequence. Nakai 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, TargetP, and iPSORT predictors used sorting signals for prediction.

### *B. Amino acid sequences*

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. PLOC uses the compositions of amino acid, amino acid pairs, and gapped amino acid pairs. Eskin and Agichtein, Hawkins et al. used Support Vector Machine (SVM) classifiers with k-mer subsequence features.

### *C. External knowledge base*

The third class exploits the information from external knowledge bases. LOCkey collects keywords annotated to the protein entries of the Swiss-Port database. Proteome Analyst has a predictor which uses text annotations for the homologues of a target protein.

MultiLOC uses Protein sequence motifs from the NLSdb and PROSITE databases. SherLOC 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 through either combining

sorting signals and amino acid composition Information or combining all three types of features.

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.

This paper proposes an approach that uses only sequence feature, while keeping a better classification performance than the previous methods.

## III. SYSTEM DESIGN AND MODEL

This overall procedure is very similar to a document classification which uses both frequent-pattern mining and boosting.

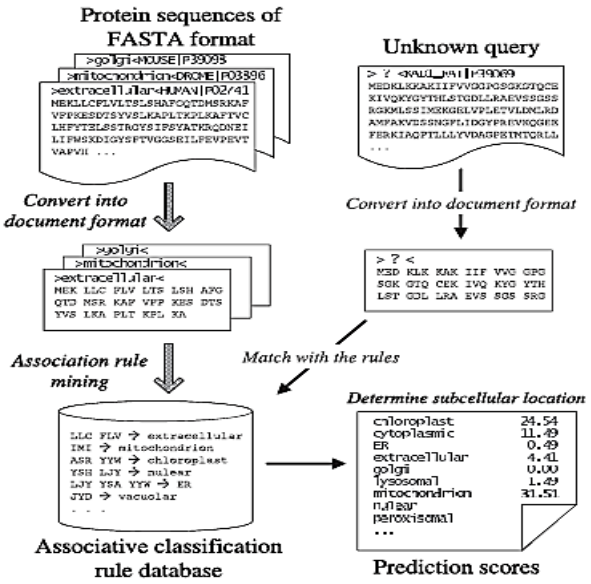


Figure 1 Flow of localization prediction using association rules.

Extraction of association rules from training example

- 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.
- 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.

Prediction of localization of test protein

- 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.

### *A. Mathematical Model*

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 L/3. Let the set of the converted protein sequences be $X=X_1*X_2*\ldots*X_d$, 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:

$$D=\{(x_i\, y_j) \mid x_i \in X\, y_j \in Y\} \qquad (1)$$

where Y is the set of subcellular location labels.

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. A mined association rule would look like this:

(KKH, LRL)→mitochondrion(20,0.3), 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.

This classification method 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:

$$S_j=\Sigma\, s_{ij} \qquad (2)$$

Then, the final prediction label c for x is determined such that:

$$c=arg\ max(\, S_j\,) \qquad (3)$$

## IV. MODEL BOOSTING PREDICTION ACCURACY

Boosting is a novel and powerful machine learning method for classification and regression. It can effectively convert a base or "weak" algorithm with accuracy just slightly better than random guessing into a strong classifier which can achieve high 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. The prediction ability of weak hypotheses is at least better than random guessing.

The frequent pattern mining method used in this system 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 labels).

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: $\theta$
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

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

Figure 2 Algorithm to induce the classifier using boosting.

Algorithm, Boosting Class Association Rules (BCAR), is shown in Fig. 2. 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 at round t is denoted as $w_i^t$ .

In the inner For loop, $r_t$ is applied to all training examples (line 1). If classifies an example correctly, then 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 , the example is deleted from the database (line 2c). This changes the distribution of the training examples. controls the rule selection process globally. The optimal value of is determined empirically using the simulation data which were used for determining the parameters (Section 3).

Line 1-2a of the algorithm corresponds to the weak learner of the original boosting algorithm. The change in the distribution of the training examples determines whether the next rule 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.

After finishing the iteration for all rules a small set of final rules is induced . The final hypothesis outputs the label of the largest score as the estimate of.

## ACKNOWLEDGMENT

We would like to thank Dr. G G Sivashankari and Srividhya V R, Department of CSE, AMCEC for constant encouragement and also we would like to express out deepest thanks to Santosh Pattar for his insightful comments and supporting us throughout.

## REFERENCES

[1] Subcellular Localization Prediction through Boosting Association Rules, Yongwook Yoon and Gary Geunbae Lee, IEEE/ACM transactions on computational biology and bioinformatics, Vol. 9, No. 2, March/April 2012.
[2] Achuthsankar S. Nair, Computational Biology & Bioinformatics: A Gentle Overview, Communications of the Computer Society of India, January 2007.
[3] A Novel Method for Protein Subcellular Localization Based on Boosting and Probabilistic Neural Network, Jian Guo, Yuanlie Lin, Zhirong Sun, Tsinghua University, 10084, China.
[4] Subcellular Localization Prediction through Boosting Association Rules, Yongwook Yoon and Gary Geunbae Lee, IEEE/ACM transactions on computational biology and bioinformatics, Vol. 9, No. 2, March/April 2012.
[5] Jiawei Han and Micheline Kamber, Data Mining, 2nd edition, Elsevier publication, 2006.

.