Rule Extraction from SVM for Protein Structure Prediction

Jieyue He[1], Hae-jin Hu[2], Bernard Chen[2], Phang C Tai[3], Rob Harrison[2], and Yi Pan[2]

[1] School of Computer Science and Engineering, Southeast University, NJ 210096, China
[2] Department of Computer Science, Georgia State University, Atlanta, GA 30303 USA
[3] Department of Biology, Georgia State University, Atlanta, GA 30303-4110, USA

Summary. In recent years, many researches have focused on improving the accuracy of protein structure prediction, and many significant results have been achieved. However, the existing methods lack the ability to explain the process of how a learning result is reached and why a prediction decision is made. The explanation of a decision is important for the acceptance of machine learning technology in bioinformatics applications such as protein structure prediction. The support vector machines (SVMs) have shown better performance than most traditional machine learning approaches in a variety of application areas. However, the SVMs are still black box models. They do not produce comprehensible models that account for the predictions they make. To overcome this limitation, in this chapter, we present two new approaches of rule generation for understanding protein structure prediction. Based on the strong generalization ability of the SVM and the interpretation of the decision tree, one approach combines SVMs with decision trees into a new algorithm called SVM_DT. Another method combines SVMs with association rule (AR) based scheme called SVM_PCPAR. We also provide the method of rule aggregation for a large number of rules to produce the super rules by using conceptual clustering. The results of the experiments for protein structure prediction show that not only the comprehensibility of SVM_DT and SVM_PCPAR are much better than that of SVMs, but also that the test accuracy of these rules is comparable. We believe that SVM_DT and SVM_PCPAR can be used for protein structure prediction, and understanding the prediction as well. The prediction and its interpretation can be used for guiding biological experiments.

1 Introduction

For the past few decades, many studies have focused on the accuracy of protein structure prediction using machine learning technologies such as neural networks or support vector machine and have achieved good results [3,4,10,12,24]. In spite of this, these methods do not explain the process of how a learning

© Springer-Verlag Berlin Heidelberg 2008

result is reached and why a decision is made. It is important to be able to explain how a decision is made for the acceptance of the machine learning technology, especially for applications such as bioinformatics since the reasons for a decision is a useful guide for the "wet experiments". The extracted rules can also be used later as a basis for advanced approaches to deduce biological features.

In most of these cases, the performance of support vector machines (SVMs) is either similar to or better than that of traditional machine learning approaches, including neural networks. It is especially important for the field of computational biology because it is used for pattern recognition problems including protein remote homology detection, microarray gene expression analysis, recognition of translation start sites, protein structure prediction, functional classification of promoter regions, prediction of protein-protein interactions, and peptide identification from mass spectrometry data [17]. Nevertheless, like the neural networks, the SVMs are black box models. They do not have the ability to produce comprehensible models that account for their predictions.

Recent researches try to extract the embedded knowledge in trained neural networks in the form of symbolic rules in order to improve comprehensibility in the field of neural networks (NNs) [26–28]. These rule extraction methods serve for several purposes: to provide NNs with explanatory power, to acquire knowledge for symbolic AI systems, to explore data, to develop hybrid architectures and to improve adequacy for data mining applications [18].

With SVM, some researchers have started to address the issue of improving the comprehensibility. Rule-extraction from SVM [18] and learning-based rule-extraction from SVMs technique [1] are two examples. Some of the limitations of these two approaches are discussed in [25].

Although some researchers have started to apply SVMs and decision trees in bioinformatics areas, all of these have not integrated the merits of both SVMs and decision trees. For example, Krishnan et al. [29] have done a comparative study of SVMs and decision tree to predict the effects of single nucleotide polymorphisms on protein function. In his paper [14], Lin classified genes by names using decision trees and SVMs. The result showed that, although the prediction errors of both methods were acceptably low for production purpose, SVM outperforms decision trees. There is also some research using the decision tree to produce rules for bioinformatics, such as automatic rule generation for protein annotation with the C5.0 data mining algorithm [20] applied on SWISS-PROT [19].

In this chapter, two novel approaches of rule-extraction for understanding protein structure prediction are presented. One approach combines SVM with decision tree into a new algorithm called SVM_DT, which proceeds in four steps. This algorithm first trains SVMs. Next, a new training set is generated by careful selection from the result of SVMs. Third, this new training set is used to train a decision tree learning system and extract the corresponding rule sets. Finally, it decodes the rules into logical rules with biological meaning according to encoding schemes. Another method combines SVM with a new association rule based classifier, pattern based classification with predictive association rules (PCPAR), into an algorithm called SVM_PCPAR with the similar process applied to SVM_DT.

Since a large number of rules are difficult for researchers to interpret and analyze, we use conceptual clustering to cluster huge number of rules based on similarity, and then aggregate the rules in each cluster to generate new super-rules. These super-rules represent the consensus rule pattern and the essential underlying relationship of classification. Because the super-rules come from each clusters, the researchers can not only understand the general trend and ignore the noise, but also interactively focus on the key aspects of the domain by using super-rules and selectively view the original rules in the corresponding cluster.

Based on protein secondary structure prediction with the RS126 data sets and transmembrane segments prediction with the 165 low-resolution data sets [5], the results show that they have similar accuracy while SVM_DT and SVM_PCPAR are more comprehensible. Hence, SVM_DT and SVM_PCPAR can be used both for prediction and guiding biological experiments.

This chapter is organized as follows. Section 2 describes the method of SVM_DT and presents the experiments of protein secondary structure prediction on RS126 data sets and transmembrane segments prediction on 165 low-resolution data set. Section 3 presents the method of extracting rules from SVM based on association rule based method. Section 4 is about the rule clustering and super rules generation. Finally, Section 5 summarizes the main contribution of this chapter and discusses some issues of the methods that should be further investigated.

2 Rule Generation by Combing SVM and DT

SVM have shown strong generalization ability in many application areas, including protein structure prediction. However, it is a black box model. On the other hand, a decision tree has good comprehensibility. It motivates us to integrate merits of both support vector machine and decision tree to generate rules for understanding protein structure prediction. This approach combines SVM with decision tree into a new algorithm called SVM_DT.

2.1 SVM_DT

SVM represents novel learning techniques that have been introduced in the framework of structural risk minimization (SRM) inductive principle and in the theory of Vapnik Chervonenkis (VC) [22] bounds. SVM has a number of interesting properties, including effective avoidance of over fitting, the ability to handle large feature spaces, and information condensing of the given data set, etc.

The basic idea of applying SVM for solving classification problems can be stated briefly in two steps. First, SVM transforms the input space to a higher dimension feature space through a non-linear mapping function. Second, it constructs the separating hyperplane with maximum distance from the closest points of the training set [2].

Decision tree learning [15] is a means for approximating discrete-valued target functions, in which the learned function is represented by a decision tree. It is one of the most popular classification methods and has been used in many research areas, such as personalized recommender system based on web [6] and effective technology commercialization [7]. Learned trees can also be re-represented as sets of if–then rules to improve human readability. Let us suppose, in a set of records, each record has the same structure, consisting of a number of attribute/value pairs. One of these attributes represents the category of the record. The problem is to determine a decision tree that, on the basis of answers to question about the non-category attributes, predicts correctly the value of the category attribute. In the decision tree, each node corresponds to a non-categorical attribute and each arc to a possible value of that attribute. A leaf of the tree specifies the expected value of the categorical attribute for the records described by the path from the root to that leaf. There are many decision tree algorithms. The results of the experiment [13] show the C4.5 [20] tree-induction algorithm provides good classification accuracy.

C5.0 is a new version of C4.5. They use the gain ratio criterion, which is based on information theory and produces suboptimal trees heuristically [20]. At first, a decision tree is built using the training set. In a second step, the decision tree is pruned by replacing a whole subtree by a leaf node. If a decision rule establishes that the expected error rate in the subtree is greater than in the single leaf, the replacement takes place.

Decision trees can sometimes be quite difficult to understand. Thus, the rule sets that consist of simple if–then rules are derived from a decision tree: write a rule for each path in the decision tree from the root to a leaf. In that rule, the leaf-hand side is easily built from the label of the nodes and the labels of the arcs. Rules are ordered by class and sub-ordered by confidence, and a default rule is created for dealing with instances that are not covered by any of the generated rules. The default rule has no antecedent and its consequence is the class that contains the most training instances not covered by any rule. Each of the rule sets produced is then evaluated using the original training data and the test data.

SVM claims to guarantee generalization, i.e. its decision model reflects the regularities of the training data rather than the incapability of the learning machine. SVM reveals the classification by looking at the critical cases. On the other hand, the advantage of the Decision Trees algorithm is easily comprehensible; it describes what attributes are important for classification [14]. Thus, the motivation of combining SVM and decision tree to classify is the desire of combining the strong generalization ability of SVM and the strong comprehensibility of rule induction. Specifically, our new algorithm SVM_DT employs SVM as a pre-process of decision tree.

Suppose we are given a training data set $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, where x_i is the feature vector and y_i is the expected class label or target of the *i*th training instance. At first, SVMs are trained using N-fold cross validation. That is, for data set S, we divided it into N subsets with similar sizes (k) and similar distribution of classes. We perform the tests for the N runs, each with a different subset as the test set ($\mathrm{Te\_svm}^i, i = 1 \ldots N$) and with the union of the other N-1 subsets as the training set ($\mathrm{Tr\_svm}^i, i = 1 \ldots N$). Then, from each test set ($\mathrm{Te\_svm}^i i, i = 1 \ldots N$), based on the result of prediction P^i_svm, we select cases that are correctly predicted by SVMs into new data set ($S^i\_svm, i = 1 \ldots N$). Finally, we use the original test data $\mathrm{Te\_svm}^i, i = 1 \ldots N$ as test data set ($\mathrm{Te\_dt}^i, i = 1 \ldots N$) and the union of the other N-1 subsets S^i_svm as the training set ($\mathrm{Tr\_dt}^i, i = 1 \ldots N$) to train decision trees and induce the rule sets. In summary, the pseudo-code of SVM_DT algorithm is shown in [49]. Since support vector machine usually has strong generalization ability and we select the new data set from the correct result of SVMs as our inputs to DT, we believe that some bad ingredients of S, such as the noise, may be reduced by the process of SVMs, and some weak cases may be filtered by SVMs. It is indicated that new data set S^i_svm data is better than the original training data set S for rule induction based on our experiment results that will be shown later. This is the reason why we use support vector machine as a pre-process of decision tree.

2.2 Protein Second Structure Prediction with SVM_DT

We apply the method of SVM_DT to the prediction of protein secondary structure. On one hand, the method is used to generate the rule sets for explaining how a secondary structure can be classified, and on the other hand, it is applied to evaluate the performance of the algorithm. We use RS126 [21] as a data set which was proposed by Rost and Sander. Based on their definition, it is a non-homologous set. This set was used in many researches on protein secondary structure prediction such as the experiments by Hua [10] and Kim [12]. The protein secondary structure prediction can be analyzed as a typical classification problem where the class (secondary structure) of a given instance is predicted based on its sequence features. The goal of secondary structure prediction is to classify a pattern of adjacent residues as helix (H), sheet (E) or coil (C, the remaining part) based on the idea that the segments of consecutive residues prefer certain secondary structure.

In this study, firstly, we combined orthogonal matrix and BLOSUM62 matrix [8] as encoding schemes [9]. The orthogonal encoding scheme is the simplest profile which assigns a unique binary vector to each residue, such as (1, 0, 0. . .), (0, 1, 0. . .), (0, 0, 1. . .) and so on. The BLOSUM62 matrix is a measure of difference between two distantly related proteins. Namely, the values in the BLOSUM62 matrix mean "log-odds" scores for the possibility that a given amino acid pair will interchange with each other and it contains the general evolutionary information among the protein families. This BLOSUM62 matrix was applied as an encoding scheme by converting its data range to [0,1]. In the encoding schemes, the information about the local interactions among neighboring residues can be embedded as a feature value, because the feature values of each amino acid residue in a window mean the weight of each residue in a pattern. Therefore, the optimal window length 13 was adopted by testing different window lengths from 5 to 19. We construct three one vs. one binary classifiers (H/∼H, E/∼E, and C/∼C).

Secondly, to train the SVM, we selected the kernel function $K(x, y) = e^{-\lambda\|x-y\|^2}$ based on the previous studies [10, 12], and the parameter of the kernel function λ and the regularization parameter C were optimized based on tests [9]. With the data set, we ran sevenfold cross validation in the experiments. That is, we divided the data set into seven subsets with similar sizes and similar distribution of classes. Then, we performed the tests for the seven runs, each with a different subset as the test set and with the union of the other six subsets as the training set. In this experiment, we used SVM$^{\text{light}}$ [11] software. In each run, we fed the training data into SVM$^{\text{light}}$ to get the model and used test data as validation.

Thirdly, in order to compare the prediction result from SVM on test data to the original data set, and to see if they were consistent, we selected the instance into a new data set which was used later for building rules. We repeat the process until seven sets of new data have been finished. Then, combining six of them as a training data and original test data as test data to train decision tree of C4.5 and C4.5 rules, we get seven group rule sets. For comparison, we also applied the original train data and test data directly into C4.5 and C4.5 rules. All the average accuracy of binary classifier by three methods is shown as Fig. 1, respectively. From Fig. 1 we can see that the accuracy of binary classifier by SVM (SVM$^{\text{light}}$) is better than that of binary classifier by the decision tree (C4.5), but the accuracy of binary classifier by SVM_DT is better than that of binary classifier by the decision tree. We believe that this is a benefit from the generalization ability of SVM.

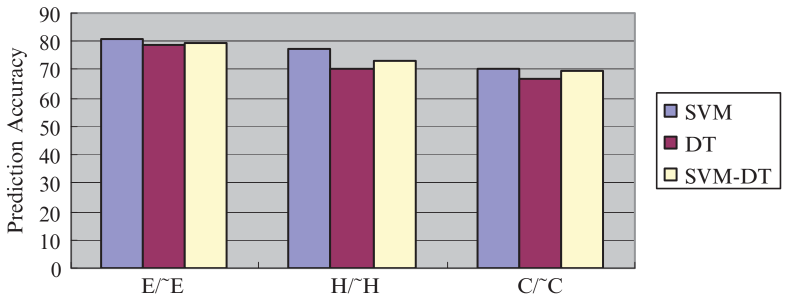

Fig. 1. Comparison of accuracy of E/∼E, H/∼H and C/∼C with three methods

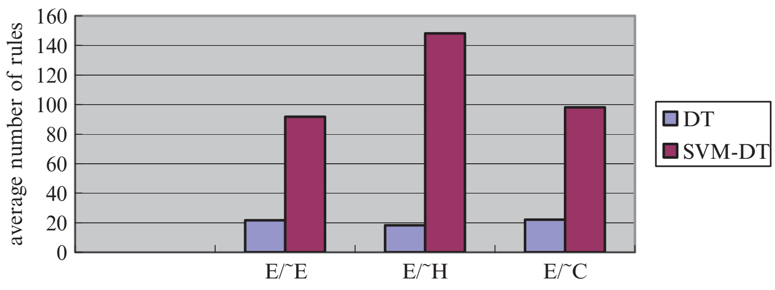

Fig. 2. Comparison of number of rules with DT method and SVM_DT method

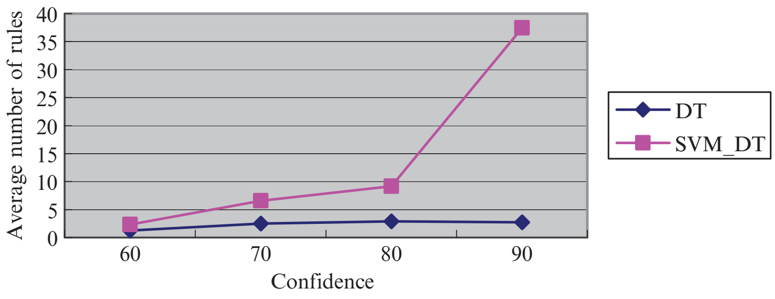

Fig. 3. Comparison average of number of rules with different confidence values of two methods

The average number of rules produced by DT and SVM_DT are shown in Fig. 2. From Fig. 2, we can see SVM_DT generated more rules than DT. In addition, Fig. 3 also shows that the average number of rules produced by SVM_DT is much more than that produced by DT under the same confidence values. This means that SVM_DT not only generates more rules but also generates rules with better quality for prediction. This observation indicates that the training data set processed by SVM is better than the original training data set for rule induction. The reason is that SVM reveals the classification by looking at the critical cases and by selecting the correct output results from SVM; SVM_DT can get the data set that has less noise.

Finally, based on the encoding schemes, we decoded the rules. We obtained a group of logical rules which have biological meaning and then we checked them in the original sequence data according to the logical rules, to verify the accuracy of them. Some of the results are shown in Table 1 with five columns. In the second column there are rules which produced by SVM_DT, and their corresponding rules with biological meaning by decoding based on the encoding schemes are shown in the third column. In the fifth column there are validation examples which are selected from the original sequence according to the logical rule in the third column and the explanation in the fourth column.

Table 1. Two example of protein secondary structure produced by SVM_DT

Rule num.	Rule produced by SVM_DT	Rule with biological meaning	Explanation	Examples
Rule 469	IF A222 $<=$ 0.035 and A260 $>$ 0 and A300 $>$ 0 THEN 'E' [90.6%]	IF Sq[2] in {'C', 'I', 'F', 'W', 'V'} and Sq[3] = 'V' and Sq[4] = 'V' THEN St[3] = 'E' [90.6%]	If the target is 'V', and one amino acid before the target is one of {C,I,F,W,V}, and the one next to the target is 'V', the second structure of the target is 'E' with 81.2% accuracy.	>2FOX: Sequence Length: 138 rule 469: the position is: 108 CVV CEE rule 469: the position is: 109 VVV EEC
Rule 471	IF A481 $>$ 0 THEN '$\sim$E' [96.8%]	IF Sq[7] = 'A' THEN St[1] = '$\sim$E ' [96.8%]	If the sixth amino acids after the target is 'A', the second structure of the target is not 'E' with 82.4% accuracy.	>1TGS: Sequence Length: 56 rule 471: the position is: 0 TSPQREA CCCCCCC >1UBQ: Sequence Length: 76 rule 471: the position is: 21 TIENVKA CHHHHHH

Although the accuracy of the binary classifier by SVM_DT is not better than that of the binary classifier by the SVM, we have gotten the rule sets. We also found that the rules generated have strong biological meaning. For example:

```
IF
    Sq[2]='V' and Sq[4] in {'C', 'I', 'L', 'V'}
    and Sq[5]='G'
THEN
    St[1]='E' [87.1%]
```

This rule can also be explained biologically. The amino acid in position two is the hydrophobic amino acid valine (V) and position four is one of the hydrophobic amino acids ('C', 'I', 'L', or 'V') followed by a glycine (G) in position five. If this forms a sheet (E), then the two hydrophobic amino acids point in the same direction (possibly into the core of the protein), thus stabilizing a sheet.

2.3 Transmembrane Segments Prediction and Understanding Using SVM_DT

Transmembrane (TM) proteins are the integral membrane proteins that can completely cross from the external to the internal surface of a biological membrane. TM proteins are critical targets for drug design. However, because of their hydrophobic properties, the conventional experimental approaches, such as X-ray crystallography or nuclear magnetic resonance (NMR) cannot be easily applied to determine their 3D structures. Therefore, computational or theoretical approaches have become important tools for identifying the structures and functions of TM proteins. Many significant results have been achieved in the prediction of transmembrane segments [24,29]. In spite of these results, the existing methods do not explain the process of the prediction.

In this study, the data set given by Rost et al. is tested and this is labeled as data set of 165 low-resolution. According to Rost et al. [29], the 165 proteins are expert-curated set from the SWISS-PROT database which was originally collected by Möller et al. [30]. The test method with these data sets is a sevenfold cross validation test.

In this research, we use two encoding schemes. One is the combined orthogonal and Blosum62 matrix (OB), the other is position-specific scoring matrix (PSSM) generated by PSI-BLAST. These PSSM values are position-specific scores for each position in the alignment. In this matrix, highly conserved positions have high scores and weakly conserved positions have low scores close to zero. This scheme is originally used to perform the prediction of protein secondary structure by Jones [31]. The author used this PSSM as an encoding profile for his neural network. As another approach, Kim [12] applied this matrix to train the SVM for the prediction of protein secondary structure. According to the author, this PSSM shows better performance than the frequency matrix generated by the multiple sequence alignments. Therefore, in this study, this encoding scheme is applied to test the performance in the prediction of transmembrane segments.

Four methods with different encoding schemes are used in the experiments. Because we focused on the rules extraction for understanding prediction of transmembrane segments, we should get the logical rules which have biological meaning. In the first method, PSSM matrix as encoding schemes are fed into SVM and DT(PSSM_PSSM). In the second method, PSSM matrix as encoding schemes are fed into SVM and the sequences are directly fed into DT(PSSM_SEQ). In the third method, the combined orthogonal and Blosum62 matrix as encoding schemes are fed into SVM and DT(OB_OB). In the fourth method, PSSM matrix as encoding schemes are fed into SVM and the combined orthogonal and Blosum62 matrix as encoding schemes are fed into DT(PSSM_OB).

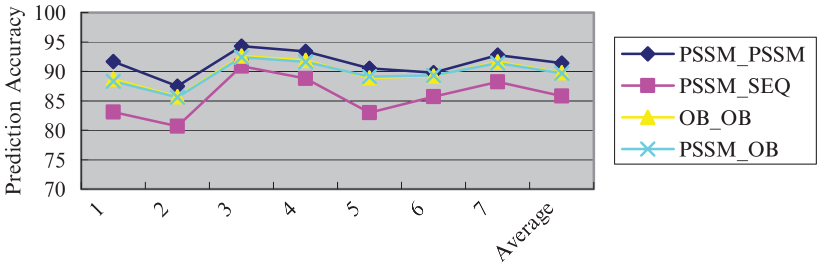

Fig. 4. Comparison of prediction accuracy of seven group rule sets with different encoding schemes

The comparison of prediction accuracy of seven group rule sets with different encoding schemes is shown in Fig. 4. From this figure, we can see the method of (PSSM_PSSM) achieves the highest average prediction accuracy. However, because PSSM is position-specific scoring matrix which is related to the context of amino acid sequence, the rules produced by DT cannot be decoded into logical rules with biological meaning. Although the method of (PSSM_SEQ) has the lower prediction accuracy than the other methods, its rules do not need to be decoded. In Fig. 4, the method of (OB_OB) and (PSSM_OB) show similar accuracy.

In order to analyze the quality of the logical rules, we compare the average rule accuracy, prediction accuracy, and percentage of rule numbers and support of seven group rule sets for the confidence of rules range of 95–100 (OB_OB). We obtain the average rule accuracy is 94.1%, average prediction accuracy is 89.6%, average percentage of rule numbers is 84.8% and average support of seven group rule sets is 83.7%. All of these show that the rules with confidence 95–100 not only have the high rule accuracy, but also have the high percentage of rule numbers and high support. The average percentage of rule numbers and support are all over 80% which means that a majority of rules obtained is of high quality.

We also analyze the rules encoded by PSSM_SEQ. The comparison of average confidence, accuracy, support, and percentage of rule numbers of seven group rule sets with confidence (90–100) is shown in Fig. 5. The results of experiment also indicate that the average prediction accuracy of rules is 93.4 for all of the rules with a confidence greater than 90. At the same time, its support is 78.0 and the percentage of rule numbers is 62.6. This means that these rules not only have high quality, but also are the majority of the rules obtained. From Fig. 5, we could find that the rules with confidence value from 97 to 99 even have a higher support value and percentage of rule numbers. The corresponding accuracies of the rules are also very high. These observations suggest that these rules are more important and valuable.

Empirical results show that the prediction accuracy is usually lower than the confidence of the rules. However, they are usually very consistent and proportional in values. A rule with a high rule confidence often produces high

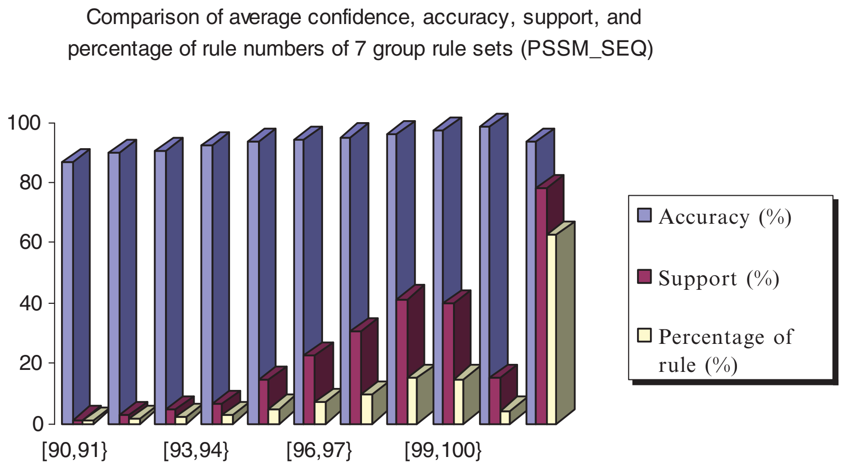

Fig. 5. Comparison of average confidence, accuracy, support, and percentage of rule numbers of seven group rule sets (PSSM_SEQ)

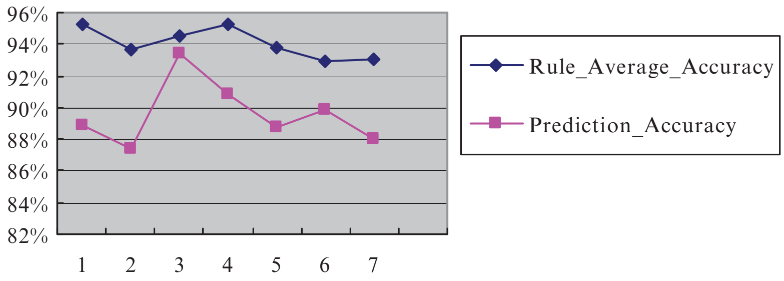

Fig. 6. Comparison of average rule accuracy, prediction accuracy of seven group rule sets for the confidence of rules range of 95–100 (OB_OB)

prediction accuracy, while a rule with a low confidence usually generates low prediction accuracy, just as Fig. 6 shows.

We decoded the rules into logical rules with biological meaning according to encoding schemes. Table 2 shows one example of logical rules and their explanation with SVM_DT (PSSM_SEQ). There are three differences between the logical rules produced by PSSM_SEQ and by OB_OB:

1. Because the rules generated by PSSM_SEQ have biological meaning, they do not need to be decoded. However, the rules generated by OB_OB should be decoded into logical rules by encoding schemes.
2. The number of rules produced by PSSM_SEQ is much greater than that of the rules produced by OB_OB. Usually, the number of rules of one set by PSSM_SEQ is about 2,000, and the number of rules in one set by OB_OB is about 200.

Table 2. One example of logical rule and their explanation with SVM_DT (PSSM_SEQ)

1	Logical rule with biological meaning	Rule 3570:
		A5 = I
		A7 = L
		A11 = F
		−> class +1 [85.6%]

2	Rule explanation	If the second amino acid before the target is 'I', the fouth amino acid following the target is 'F' , and at the same time, the target is 'L', the segment of the target is 'T' (transmembrane) with an confidence 85.60%, prediction accuracy is 83.33% when we do the experiment on the test data.

3. The rules generated by PSSM_SEQ are simpler than that of rules by OB_OB.

The reason of these differences is that the input attributes of DT in PSSM_SEQ are characters, while the input attributes of DT in OB_OB are continuous.

3 Extracting Rule from SVM Based on Association Rule

3.1 Association Rule Based Method

In the previous section, the learning-based rule extraction approaches applied decision tree as a second learning algorithm to extract the rules from SVM [25]. The advantage of DT is that the significance of rules is measured by their contribution to the overall accuracy of the classifier therefore systematic accuracy-based rule pruning is possible [32]. This method searches for rules locally based on a heuristic by adding one capable attribute at a time according to the order of goodness.

An alternative of the DT algorithm is an association rule (AR) based scheme. This method searches for all rules globally based on the cooperative prediction of several attributes and assesses each rule individually without considering the interaction with other rules [32]. The final rule set covers the training data in all possible ways hence the number of rules are usually large compared with DT method. The set with the large number of rules has the potential to find the true classification template from the training data if the over-fitting rules are pruned properly. Therefore, in AR based approach, rule pruning has been a main interest to the researchers.

Recently Hu et al. [33] attempted to apply the AR based method to extract rules from SVM on transmembrane segments prediction. They devised the pattern based classification with predictive association rules (PCPAR) scheme based on classification based on predictive association rules (CPAR) method [34] to handle the dataset with a sliding window scheme. This section begins with basic concepts of AR mining algorithm. Then traditional and recent AR mining algorithms are presented and finally rule generation method based on SVM_PCPAR scheme is described in detail.

3.2 Association Rule Mining

Basic Concepts

A formal definition of association rule mining is as follows [35].

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, or items. Let X be an itemset which is a subset of I. Let $D = \{t_1, t_2, \ldots, t_n\}$ be a set of transactions called a transaction database. Each transaction t has a transaction identifier, *tid* and a transaction itemset such as $t = (t_{id}, t \text{ - } itemset)$. A transaction t contains an itemset X if $X \subseteq T$.

In a transaction database D, each itemset X has a support, $supp(X)$ which is the ratio of transactions in D containing X.

$$supp(X) = |X(t)|/|D|, \tag{1}$$

where $X(t) = \{t \text{ in } D | t \text{ contains } X\}$. A large or frequent itemset is defined as an itemset whose support is equal to, or greater than, the user-specified minimal support threshold.

An association rule is an implication $X \rightarrow Y$, where itemsets X and Y are disjoint, $\mathrm{X} \cap \mathrm{Y} = \phi$. In each association rule, there are two quality measures, support and confidence. The support is the number of occurrences of each pattern and the confidence is the strength of implication. These measures are defined formally as follows:

- The support of a rule $X \rightarrow Y$ is the support of $X \cup Y$
- The confidence of a rule $X \rightarrow Y, conf(X \rightarrow Y)$ is the ratio

$$|(X \cup Y)(t)|/|X(t)|, \text{or } supp(X \cup Y)/supp(X).$$

When a transaction database D is given, mining association rules is generating all association rules which have support and confidence values equal to, or greater than, the user-specified minimal support and confidence threshold respectively.

Association Rule Mining Algorithms

Most of the traditional association rule mining algorithms are based on support-confidence model which is suitable for analyzing the market basket data [36]. For example, a Apriori is a famous and commonly-used algorithm based on this model [37].

However, in other applications such as bioinformatics or system traces, the number of occurrences may not be a good metric to measure the significance of a pattern [38]. In bioinformatics, researchers try to find statistically important sequential patterns from the sequential data. Since the frequency of each symbols in a sequence may not evenly distributed (some symbols occur more often than other symbols), a pattern with common symbols occur more often than that with rare symbols. Therefore, the frequency (support) may not always indicate the importance of a pattern. Researchers should consider both the frequent patterns and the "surprising" patterns [38]. Sometimes a few numbers of "unexpected" rare patterns could provide more information than a large number of "expected" frequent patterns. Wang and Yang adopted the information metric [39] to characterize these surprising patterns. In their research, information is used to measure the degree of "surprise" when a pattern actually occurs. Also, the information gain metric is devised to characterize the accumulated information of a pattern.

Besides the Wang and Yang's approaches, first order inductive learner (FOIL), predictive rule mining (PRM) and classification based on predictive association rules (CPAR) also applied the information metric for the rule generation. FOIL [40] is a greedy algorithm that repeatedly searches for the attribute with the highest information gain. Once this attribute is appended to a rule, all the examples which are not satisfying the rule are removed from both the positive and negative examples. After the rule is added into a rule set, this process is repeated until all positive examples in the data set are covered. For selection of attributes, "FOIL Gain" is defined such as follows to measure the information gained from appending this attribute to the current rule.

$$gain(p) = |P^*| \left(\log \frac{|P^*|}{|P^*| + |N^*|} - \log \frac{|P|}{|P| + |N|} \right), \tag{2}$$

where $|P|$ and $|N|$ are positive and negative examples that satisfy the current rule. After attribute p is added to the rule, there are $|P^*|$ positive and $|N^*|$ negative examples satisfying the new rule's body.

The FOIL algorithm was later further improved by Yin and Han to achieve higher accuracy and efficiency. It is called the predictive rule mining (PRM) algorithm [34]. PRM algorithm is a "weighted" version of FOIL [34]. In PRM, if an example is covered by a rule, without removing it, its weight is reduced by multiplying a decay factor. This algorithm was then further improved by the same authors to produce CPAR [34].

Association Rule Based Classifiers

Associative classification is an approach to integrate association rule (AR) mining and classification [41]. It applies AR mining algorithm to generate the whole set of association rules. Based on this complete rule set, a small subset of significant rules is selected and this set is used for prediction. Two typical AR based classifiers are classification based on associations (CBA) [41] and classification based on multiple association rules (CMAR) [42]. These classifiers are based on support and confidence framework which is not suitable for large dataset.

Classification based on predictive association rules (CPAR) is a more advanced AR based classifier based on information metric [34]. In CPAR, Laplace accuracy is used to measure the accuracy of rules. Given a rule r, it is defined as follows:

$$Laplace\ accuracy\ (r) = \frac{(N_c + 1)}{(N_{total} + m)}, \tag{3}$$

where m is the number of classes, N_{total} is the total number of examples that satisfy the rule's body, among which N_c examples belong to the predicted class, c of the rule. For classification, the best k rules of each class are selected from the rule sets of each class. By comparing the averaged Laplace accuracy of the best k rules of each class, the class with the best accuracy is chosen as the predicted class.

The pattern based classification with predictive association rules (PCPAR) [33] is a modified version of classification based on predictive association rules (CPAR). It is devised to handle the dataset with sliding window scheme. The rule generation part of PCPAR is the same as that of CPAR algorithm except the fact that in PCPAR each attribute window is able to participate in the AR training with different initial weight. The main differences of PCPAR and CPAR are in the post processing and the classification scheme. CPAR algorithm doesn't have any post processing step after rule generation. The PCPAR algorithm incorporates the post processing step to create more general patterns by decoding and merging the rules. For example, the following rules are the same even though the antecedents display different feature values. If we decode these rules, the antecedents of the following rules have the meaning of the amino acid 'EE' occurring position 5 and 6, 6 and 7, and 7 and 8, respectively.

$$\begin{aligned} \{87, 107\} &\rightarrow \{261\}, \\ \{107, 127\} &\rightarrow \{261\}, \\ \{127, 147\} &\rightarrow \{261\}. \end{aligned}$$

As can be observed from the above, the absolute location of each attribute is not important in the sliding window scheme. Rather we should focus on the pattern of the features. With the example above, by decoding and rule merging, we can find a pattern of 'EE' occurring somewhere in a window. This pattern is simpler and also more general than the rules.

The PCPAR classification is based on the patterns created from the post process (decode-merge process) after rule mining. Each test data is checked against all the patterns of each class and the final class is determined based on the following cases. For each test instance, there are four possible situations:

1. It matches with the positive patterns only.
2. It matches with the negative patterns only.
3. It matches with both the positive and negative patterns.
4. It matches with none of them.

In the first and the second case, the final class is positive and negative class respectively. In the third case, by comparing the normalized numbers of patterns matched, the class with bigger number of patterns is selected as a final class. Finally, if no matched pattern is found with a test instance, the class is selected as a negative class by default.

Rule Generation Based on SVM_PCPAR Model

SVM_PCPAR model borrows the idea from the SVM_DT [25] for combining classifiers. This algorithm combines the SVM with a new AR based classifier, pattern based classification with predictive association rules (PCPAR) with the following process (Fig. 7).

First, SVM is trained with the two highly performed encoding profiles including the orthogonal and Blosum62 combined matrix and PSSM. Next, with the output of SVM, correctly predicted set is chosen as a new training set for AR mining. These two steps are the pre-process for the AR mining. The rationale of this pre-process is that since SVM usually has strong generalization ability, some noise or uncertain instances can be filtered out by

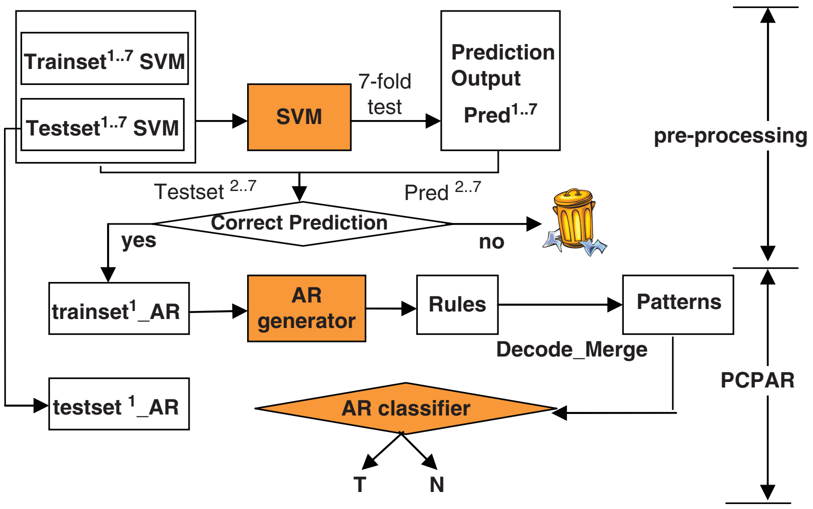

Fig. 7. SVM_PCPAR model

Positive rules					Decoded positive rule body	Positive pattern	
(428)	{31 50 170}	→	{262}	0.97	L 2 I 3 I 9	*LI*****I****	0.92
(429)	{150 231 250}	→	{262}	0.96	I 8 L 12 I 13	*******I***LI	0.92
(430)	{50 70 150}	→	{262}	0.96	I 3 I 4 I 8	**II***I*****	0.94
(431)	{30 50 130}	→	{262}	0.96	I 2 I 3 I 7	***V**I*I****	0.94
(432)	{110 130 210}	→	{262}	0.95	I 6 I 7 I 11	****LFI******	0.91
(433)	{130 150 230}	→	{262}	0.95	I 7 I 8 I 12	***FAI*******	0.91
(434)	{150 170 250}	→	{262}	0.95	I 8 I 9 I 13	****II*F*****	0.92
(435)	{80 130 170}	→	{262}	0.95	V 4 I 7 I 9	*L**LLV******	0.91
(436)	{91 114 130}	→	{262}	0.95	L 5 F 6 I 7	L*L*L*I******	0.97
(437)	{60 110 150}	→	{262}	0.95	V 3 I 6 I 8	**I***L*V****	0.91

Negative rules					Decoded negative rule body	Negative pattern	
(1)	{132}	→	{261}	1.0	K 7	******K******	0.99
(2)	{107 127}	→	{261}	1.0	E 6 E 7	*****EE******	1.00
(3)	{127 147}	→	{261}	1.0	E 7 E 8	****A*R******	0.99
(4)	{87 107}	→	{261}	1.0	E 5 E 6	******K**L***	0.99
(5)	{67 87}	→	{261}	1.0	E 4 E 5	*****SR******	1.00
(6)	{47 67}	→	{261}	1.0	E 3 E 4	******E*****A	1.00
(7)	{147 167}	→	{261}	1.0	E 8 E 9	*****GE******	1.00
(8)	{167 187}	→	{261}	1.0	E 9 E 10	****E**K*****	1.00
(9)	{81 122}	→	{261}	1.0	A 5 R 7	******R**A***	0.99
(10)	{132 191}	→	{261}	1.0	K 7 L 10	******PE*****	0.99

Fig. 8. Example of Decode_Merge process in SVM_PCPAR model

this process [25]. Third, the new training data is normalized to adjust the format for PCPAR training. Fourth, the normalized data is applied to PCPAR to train and generate the rules. Once the rule sets are generated, they are decoded into biologically meaningful rules using a decode table. The example of decoded rule bodies and patterns obtained by merging the same rules are presented in Fig. 8. The same rules are identified by examining the decoded rule body. For example, if a positive rule body is decoded into (V 3 I 6 I 8), it means that amino acids V, I, I occur at position 3, 6 and 8 in a slide window. Since the encoding profile of our AR based classifier is composed of the sliding windows of amino acid sequences, these positions could be any of (1, 4, 6), (2, 5, 7), (4, 7, 9), (5, 8, 10), (6, 9, 11), (7, 10, 12), and (8, 11, 13) with the window size 13. The decoded rule body (V 3 I 6 I 8) can be merged with (V 4 I 7 I 9) since these are the same. Because of this reason, we should rely on the relative positional expression (pattern) rather than the absolute positional information. If we use the previous example again, the positive rule body, (V 3 I 6 I 8) can be expressed as the positive pattern, (V**I*I). It means that only if this pattern comes somewhere within a window, it becomes a positive class. Here, the '*' can be considered as a "don't-care" character. Based on this kind of patterns defining positive and negative classes respectively, the test data can be classified with pattern match. The PCPAR classification algorithm performs this to determine the final class of the amino acid in the middle of a sliding window.

The first column is the positive and negative rules with the Laplace accuracy values, the second is the decoded rule bodies based on the decode table, and the third column is the patterns created from the rule merge process. The Laplace accuracy values in the third column are averaged values from the same rules.

4 Rule Clustering and Super_rule Generation

By combining SVM with decision tree, we extract rules for understanding transmembrane segments prediction. However, rules we have gotten are as many as 20,000. Such a large number of rules are difficult for researcher to interpret and analyze [48]. This can often hamper the knowledge discovery process. Clearly, it will not be satisfactory for researchers to simply use arbitrary small subset of rules because the subset of rules can't cover all the data of the domains.

Many researchers have addressed this problem by proposing a number of approaches to produce a suitable rule set. Usually, rules have to be pruned and grouped at the post mining stage, so that only a reasonable number of rules have to be inspected and analyzed. For example, one method of reduced error pruning is used in the decision tree [20] which takes each of nodes in the tree as pruning candidate. Another successful method for finding high accuracy is rule post-pruning used in C4.5 [20]. Chawla [43] presented pruning of association rules using directed hypergraph which maps a set of association rules into a directed hypergraph and systematically removing circular paths, redundant and backward edges that may obscure the relationship between the target and other frequent item. The clustering is often used for the grouping rules to deal with huge rules. Association rule clustering system (ARCS) is the example [16] which clusters all those two attribute associate rules where the right-hand side of the rules satisfies the segmentation criteria. This approach is based on the geometric properties. In the papers [44–46], the main idea of the approaches is distance based clustering for association rules. The distance is strongly correlated with support, and high support rules will on average tend to have higher distances to everybody else.

In this study, the target of the research can be described as "global" in the sense that we want to compress the rule base into smaller one set with the assurance that very little useful information is lost as possible. Moreover, the rules are not created by frequent items as the primitive association rule mining methods do. Therefore, unlike the methods in [43–46], we present the method of rule clustering and super rules generation based on the conceptual clustering [47]. In conceptual clustering, a group of objects forms a class only if it is describable by a concept. Different from conventional clustering, conceptual clustering consists of two steps: First, discover the appropriate clusters.

Second, form descriptions for each cluster. In the study, descriptions of each cluster are super rules.

Therefore, a novel approach of rule clustering for super-rule (C_SuperRule) generation is presented. We use the clustering to group huge number of rules based on similarity, and then aggregate the rules in each cluster to generate new super-rules. These super-rules represent the consensus rule pattern and the essential underlying relationship of classification. Since the super-rules come from each of clusters, the researchers not only can understand the general trend and ignore the noise but also can interactively focus on the key aspects of the domain by using super-rules and selectively view the detailed rules in the corresponding of cluster.

In this research, the rules are used for protein structure prediction by amino acid sequence. It means that if the length of the sliding windows is L, the corresponding structure of amino acid at the position of [L/2] will be predicted. Therefore, we could take the rules as follows:

For a rule $\Re : \mathrm{B} \rightarrow \mathrm{A}, \mathrm{A} \in [-1, 1]$,
B: IF Sq[1]∩Sq[2] ∩...∩Sq[L], Sq[i]=Q[1]∪Q[2]∪...∪Q[20],Q[k]∈ Λ, or Q[k] = ϕ, $\Lambda[20] = \{$A,R,N,D,C,Q,E,G,H,I,L,K,M,F,P,S,T,W,Y,V$\}$.

B could be express as a matrix M : L × 20,

$$M_{i,j} = \begin{cases} 1 & if \quad sq[i] = \Lambda[j], 1 \leq j \leq 20, 1 \leq i \leq L \\ 0 & \qquad\qquad or \end{cases}$$

For example: IF Sq[1] = {R,S,T}
and Sq[2] = {N,T}
and Sq[3] = {A,G}
and Sq[5] = {D}
THEN
St[7] = N

The precondition part of the rule can be a matrix as Fig. 9.

We use k-means to cluster rules according to the similarity of rules. K-means clustering algorithm is the most widely used method in partition category due to its fast speed and easy understanding. The method uses an object called centroid which is the mean point in a cluster, and tries to minimize the intra-cluster distance between any point in the cluster and the centroid. We applied this method in our classification rules clustering by combining similar rules together to generate more general and error-tolerant rules. First, based on the prediction results of the rules we classify the rule set into positive rule set and negative rule set. Then, the rules in positive rule set and negative rule set are separately clustered by combining similar rules together. Next, the score matrix of each rule cluster is calculated. The score matrix of the cluster means the frequency for the specific amino acid

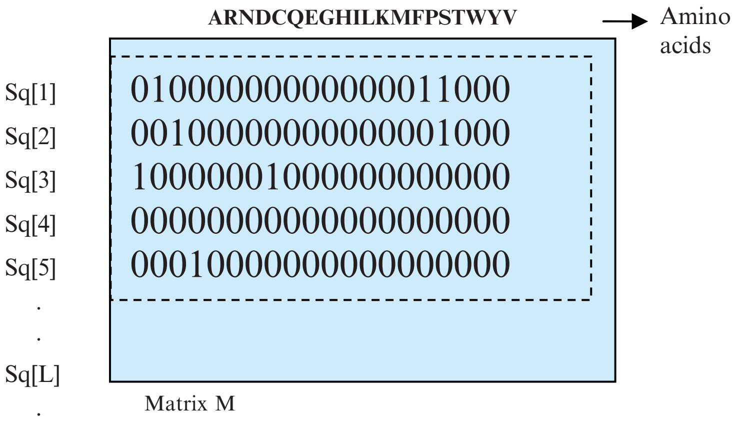

Fig. 9. The example of matrix of the rule

residue in a given window position for a cluster. It is obtained by the following formula:

$$SC_{ij} = \frac{\sum_{k=1}^{n} M_{ij}^{k}}{n} \times 100, \tag{4}$$

where, n is the number of rules in the cluster.

Finally, the super_rules are generated according to a given threshold in the clusters. The pseudo-code of the C_SuperRule algorithm is shown in Fig. 10.

In this study, the experiment of the C_SuperRule is performed based on the rules created by the method in Sect. 2.3. We use random method to generate initial centroids positions; we set K equals to 20 for transmembrane prediction rules, 30 or 35 (depends on the result) for non-transmembrane prediction rules. Comparison of percentage of rule numbers of seven group super-rules for the accuracy range of prediction "T" is shown as Table 3. From Table 3 we can see the percentage of rule number of accuracy over 85 is about 60%. It means the super-rules have high quality. The example of super-rules and explanation is shown in [50]. The super-rules are very useful in guiding biological experiments. In the clustering, we get the cluster score matrix, such as Fig. 11; these indicate the profile of the amino acid in each position of 13 windows for transmembrane prediction. The higher frequency of the amino acid in the position implies that in this position this amino acid is more important for the corresponding structure. We believe that it will be very useful information for biology.

The experiments show that these super-rules not only have high quality but also are different from the rules before clustering because these super-rules are produced by aggregating the detailed rules and indicate the general trend.

```
C_SuperRule
input: Logical Rule set $L\_R_i$, i=1,...,N
output: Super-rule set S_R
Process:
    $S\_R = \Phi$
    FOR i = 1 to N { /* for each $L\_R_i$ set, classify into $R_+$ and $R_-$*/
        IF      $L\_R_i$ is positive rule
                $L\_R_+ = L\_R_+ \cup L\_R_i$
        ELSE
                $L\_R_- = L\_R_- \cup L\_R_i$
        ENDIF
    END IF
    FOR $L\_R_+, L\_R_-$
        / rules clustering by combining similar rules together /
        {rule_number1, $C\_R_+$} = k-means ($L\_R_+$)
        {rule_number2, $C\_R_-$} = k-means ($L\_R_-$)
    ENDFOR
    FOR i = 1 to rule_number1
        Calculate the cluster score matrix of $SC\_R^i_+$
        $S\_R_i$ = create_super_rule($SC\_R^i_+$)
        $S\_R = S\_R \cup S\_R_i$
    ENDFOR
    FOR i = 1 to rule_number2
        Calculate the cluster score matrix of $SC\_R^i_-$
        $S\_R_i$ =create_super_rule($SC\_R^i_-$)
        $S\_R = S\_R \cup S\_R_i$
    ENDFOR
```

Fig. 10. C_SuperRule Algorithm

Table 3. Comparison of percentage of rule numbers of super-rules which covers each accuracy range of prediction "T"

	Accuracy	Percentage of rule numbers
1	95–100	26.9
2	90–95	18.9
3	85–90	15.5
4	80–85	15.2
5	<80	23.5

5 Conclusions

The explanation of a decision is important for the acceptance of machine learning technology in bioinformatics applications such as protein structure prediction. In this chapter, we present two approaches for rule generation from SVM for protein structure prediction, and we also discuss the rule clustering for huge number of rules at the post mining stage. Empirical results on several

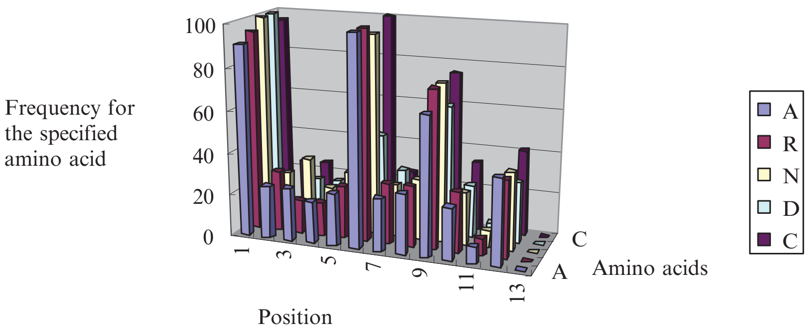

Fig. 11. One of the cluster score matrix of {A,R,N,D,C}

data sets demonstrate the efficacy of our methods. The explanation of the rules is very useful in biology domain. These rules with biological meaning not only indicate what a prediction is, but also how a decision is made. These rules can guide the "wet experiments" since they can help to identify the explicit sequence features causing the prediction and to recognize the specific mutation invalidating the prediction or presumably altering the behavior of the protein. Therefore, we can narrow the experiment scope and focus only on certain changes in the amino acid sequence.

There is still much work need to be done to improve the current approaches. First, we trained the SVM with the encoded training data. When we generated rules produced by SVM_DT, we needed to decode them into the biologically meaningful rules. One of the encoding schemes such as PSSM, we could get higher prediction accuracy than other encoding schemes. However, we were not able to decode them into the logical rules with biological meaning. One of the methods researchers used is to replace kernel functions of SVMs with PSSM. This will solve the decoding problem. But how to use it and which matrix is suitable for use as a kernel function for protein structure prediction is our future work.

Second, Andrews et al. [26] have presented a framework of rule quality evaluation, namely FACC, for evaluating the quality of the rules extracted from neural networks. In detail, the FACC framework comprises four criteria, namely fidelity, accuracy, consistency, and comprehensibility. Zhou [51] points out that the accuracy, consistency, and comprehensibility (ACC) instead of the FACC framework should be used for rule extraction using neural networks, while the fidelity, consistency, and comprehensibility (FCC) instead of the FACC framework should be used for rule extraction for neural networks. In the study, we focused on the accuracy and comprehensibility. How to evaluate the quality of the rules generated from SVM still needs to be studied.

6 Acknowledgements

The authors would like to thank Professor Thorsten Joachims for making SVM$^{\text{light}}$ software available, Professor F. Coenen for making association rule mining software available, and thank RuleQuest Research for a ten-day evaluation licence of C5.0 software available. This research was supported in part by the scholarship under the State Scholarship Fund of China, the Nation of 985 project and Jiangsu Science Fund of BK2007105, and the U.S. National Institutes of Health (NIH) under grants R01 GM34766-17S1, P20 GM065762-01A1, and the U.S. National Science Foundation (NSF) under grants CCF-0514750, and CCF-0646102. This work was also supported by the Georgia Cancer Coalition and computer hardware used was supplied by the Georgia Research Alliance.

References

1. Barakat, N. and Diederich, J.: Learning-based Rule-Extraction from Support Vector Machine. The third Conference on Neuro-Computing and Evolving Intelligence (NCEI'04) (2004).
2. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):121–167 (1998).
3. Casbon, J.: Protein Secondary Structure Prediction with Support Vector Machines (2002).
4. Chandonia, J.M. and Karplus, M.: New Methods for accurate prediction of protein secondary structure. Proteins (1999) 35, 293–306.
5. Chen, C.P., Kernytsky, A. and Rost, B.: Transmembrane helix predictions revisited. Protein Science, vol. 11, (2002), pp. 2774–2791.
6. Cho, Y.H., Kim, J.K. and Kim, S.H. :A personalized recommender system based on web usage mining and decision tree induction. Expert Systems with Applications, Volume 23, Issue 3, 1, (2002), 329–342.
7. Sohn, S. Y. and Moon, T.H.: Decision Tree based on data envelopment analysis for effective technology commercialization. Expert Systems with Applications, Volume 26, Issue 2, (2004), 279–284.
8. Henikoff, S. and Henikoff, J.G.: Amino Acid Substitution Matrices from Protein Blocks. PNAS 89, 10915–10919 (1992).
9. Hu, H., Pan, Y., Harrison, R. and Tai, P. C.: Improved Protein Secondary Structure Prediction Using Support Vector Machine with a New Encoding Scheme and an Advanced Tertiary Classifier. IEEE Transactions on NanoBioscience, Vol. 3, No. 4, Dec. 2004, pp. 265–271.
10. Hua, S. and Sun, Z.: A Novel Method of Protein Secondary Structure Prediction with High Segment Overlap Measure: Support Vector Machine Approach. J. Mol. Biol. (2001) 308: 397–407.
11. Joachims, T.: SVMlight. http://www.cs.cornell.edu/People/tj/svm_light/ (2002).
12. Kim, H. and Park, H.: Protein Secondary Structure Prediction Based on an Improved Sup port Vector Machines Approach (2002).

13. Lim, T.S., Loh, W.Y. and Shih, Y.S.: A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty_Tree Old and New Classification Algorithm. Machine Learning, Vol. 40, no. 3, pp. 203–228, Sept. 2000.
14. Lin, S., Patel, S. and Duncan, A.: Using Decision Trees and Support Vector Machines to Classify Genes by Names. Proceeding of the Europen Workshop on Data Mining and Text Mining for Bioinformatics, 2003.
15. Mitchell, M.T.: Machine Learning. McGraw-Hill, US (1997).
16. Lent, B., Swami, A. N. and Widom, J. Clustering association rules. In ICDE, 1997, pages 20–231.
17. Noble, W.S.: Kernel Methods in Computational Biology. B. Schoelkopf, K. Tsuda and J.-P. Vert, ed. MIT Press (2004) 71–92.
18. Núñez, H., Angulo, C. and Catala, A.: Rule-extraction from Support Vector Machines. The European Symposium on Artifical Neural Networks, Burges, ISBN 2-930307-02-1, 2002, pp. 107–112.
19. Kretschmann, E., Fleischmann, W. and Apweiler, R.: Automatic Rule Generation for protein Annotation with the C4.5 Data Mining Algorithm Applied on SWISS-PROT. Bioinformatics, (2001), 17(10).
20. Quinlan, J.R.: C4.5:Programs for Machine Learning. San Mateo, Calif: Morgan Kaufmann, 1993.
21. Rost, B. and Sander, C.: Prediction of protein Secondary Structure at Better than 70% Accuracy. J. Mol. Biol. (1993) 232, 584–599.
22. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, Inc., New York (1998).
23. Yang, Z.R. and Chou, K.: Bio-support Vector Machines for Computational Proteomics. Bioinformatics 20(5), 2004.
24. Sikder, A.R. and Zomaya, A.Y.: An "overview of protein-folding techniques: issues and perspectives," Int. J. Bioinformatics Research and Applications, Vol. 1, issure 1, pp. 121–143, 2005.
25. He, J., Hu, H., Harrison, R., Tai, P.C. and Y. Pan, "Transmembrane segments prediction and understanding using support vector machine and decision tree," Expert Systems with Applications, Special Issue on Intelligent Bioinformatics Systems, vol. 30, pp. 64–72, 2006.
26. Andrews, R., Diederich, J. and Tickle, A.: A Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks. Knowledge-Based Systems (1995), 8(6), pp. 373–389.
27. Tickle, A., Andrews, R., Mostefa, G. and Diederich, J.: The Truth will come to light: Directions and Challenges in Extracting the Knowledge Embedded within Trained Artificial Neural Networks. IEEE Transactions on Neural Networks, (1998), 9(6), pp. 1057–1068.
28. Zhou., Z.-H. and Jiang, Y.: NeC4.5.: neural ensemble based C4.5. IEEE Transactions on Knowledge and Data Engineering, (2004), 16(6): 770–773.
29. Chen, C.P., Kernytsky, A. and Rost, B.: Transmembrane helix predictions revisited. Protein Science, vol. 11, (2002), pp. 2774–2791.
30. Möller, S., Kriventseva, Apweiler, E.: V. and R.: A collection of well characterized integral membrane proteins. Bioinformatics, vol. 16, (2000), pp. 1159–1160.
31. Jones, D. T.: "Protein Secondary Structure Prediction Based on Position-specific Scoring Matrix," J. Mol. Biol, vol. 292, (1999), pp. 195–202.

32. Wang, K., Zhou, S. and Y. He, “Growing Decision Trees On Support-Less Association Rules,” presented at Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’00), Boston, MA, 2000.
33. Hu, H., Wang, H., Harrison, R., P.C. Tai, and Y. Pan, “Understanding the Prediction of Transmembrane Proteins by Support Vector Machine using Association Rule Mining,” presented at IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB ’07), Honolulu, Hawaii, 2007.
34. Yin, X. and Han, J. “CPAR: Classification based on Predictive Association Rules,” presented at SIAM Int. Conf. on Data Mining (SDM’03), San Fransisco, CA, 2003.
35. Zhang, C. and Zhang, S.: Association Rule Mining: Models and Algorithms: Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 2002.
36. Agrawal, R., Imielinski, T. and A. Swami: “Database mining: A performance perspective,” presented at IEEE Transactions on Knowledge and Data Engineering, 1993a.
37. Agrawal, R. and Srikant, R.: Fast Algorithms for Mining Association Rules, presented at 20th Int’l Conference on Very Large Databases, Santiago, Chile, 1994.
38. Wang, W. and Yang, J.: Mining Sequential Patterns from Large Data Sets: Springer, 2005.
39. Blahut, R.: Principles and Practice of Information Theory: Addison-Wesley Publishing Company, 1987.
40. Quinlan, J. R. and Cameron-Jones, R. M.: FOIL: A Midterm report, presented at European Conference on Machine Learning (ECML-93), Vienna, Austria, 1993.
41. Liu, B., Hsu, W. and Ma, Y.: Integrating classification and association rule mining, presented at The Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)′, New York, 1998.
42. Jayasinghe S, H. K. and White S.H.: Energetics, stability, and prediction of transmembrane helices., J. Mol. Biol., vol. 312, pp. 927–934, 2001.
43. Chawla, S., Davis, J., Pandey, G. On Local Pruning of Association Rules Using Directed Hypergraphs. Proceedings of the 20th International Conference on Data Engineering, ICDE 2004: 832.
44. Gupta, G., Strehl, A. and Ghosh. J. Distance based clustering of association rules. In Intelligent Engineering Systems Through Artificial Neural Networks (Proceedings of ANNIE 1999), ASME Press, November, 1999., volume 9: pages 759–764.
45. Lele, S., Golden, B., Ozga, K. and Wasil, E. Clustering Rules Using Empirical Similarity of Support Sets Lecture Notes In Computer Science; Vol. 2226 archive, Proceedings of the 4th International Conference on Discovery Science table of contents, 2001, Pages: 447–451.
46. Toivonen, H., Klemettinen, M., Ronkainen, P. and Mannila. H. Pruning and grouping discovered association rules. In MLnet Workshop on Statistics, Machine Learning and Discovery in Databases, April, 1995: pages 47–52.
47. Han, J. and Kambr, M.: Data Mining concepts and Techniques, Higher Education Press, Morgan Kaufmann Publishers. 2001.
48. Wang, J. ed.: Encyclopedia of Data Warehousing and Minging, Hershey, PA: IGI, 2005, 190–195.

49. He, J. Hu, H. Harrison, R., Tai, P.C. and Pan, Y.: Rule Generation for Protein Secondary Structure Prediction with Support Vector Machines and Decision Tree, IEEE Transactions on NanoBioscience, Vol. 5, No. 1, March 2006, pp. 46–53.
50. He, J. Hu, H. Harrison, R., Tai, P.C., Dong, Y. and Pan, Y : Rule Clustering and Super_rule Generation for Transmembrane Segments Prediction, Proceedings of IEEE Computational Systems Bioinformatics Conference (CSB 2005), August 8–11, 2005, Califormia, USA, Poster, pp. 224–227.
51. Zhou, Z.-H. Rule extraction:using neural networks or for neural networks? Journal of Computer Science and Technology, 2004, 19(2), 249–253.