Proceedings of the 2007 IEEE Symposium on Computational
Intelligence in Bioinformatics and Computational Biology (CIBCB 2007)

Understanding the Prediction of Transmembrane Proteins by Support Vector Machine using Association Rule Mining

Hae-Jin Hu, Hao Wang, Robert Harrison, Phang C. Tai and Yi Pan*

Abstract—**With the efforts to understand protein structure, many computational approaches have been made recently. Among them, the support vector machine (SVM) methods have been recently applied and showed successful performance compared with other machine learning schemes. However, despite the high performance, the SVM approaches suffer from the problem of understandability since it is a black-box model. To overcome this limitation, this study attempted to combine the SVM with the association rule based classifier which can present the meaningful explanation about the prediction. To perform this task, a new association rule based classifier (PCPAR) was devised based on the existing classifier, CPAR, to handle the sequential data. PCPAR creates the patterns by merging the generated rules and then classifies the sequential data based on the pattern match. The experimental result presents the following: with sequential data, the PCPAR scheme shows better performance with respect to the accuracy and the number of generated patterns than CPAR method whether applied alone or combined with SVM. The combined scheme of SVM_PCPAR generates more compact patterns than the combined scheme of SVM with decision tree, SVM_DT, with similar performance. These patterns are easily understandable and biologically meaningful.**

Index Terms— **support vector machine, association rule based classifier, decision tree, CPAR, PCPAR**

I. INTRODUCTION

Over the past few decades, there were many approaches made to understand the protein structure. Since the initial experimental approaches drew a lot of cost and time, they were replaced with computational prediction methods. The progress of the machine learning technology provided various advanced tools for prediction. Among the many machine learning approaches, the support vector machine methods are the most recently applied in the structure prediction[1]. They show successful performance compared with other machine learning schemes[2-4]. However, despite the highly accurate prediction, the support vector machine approaches suffer from the problem of interpretability since they are a black-box model; the predictions made by SVM cannot be interpreted in a biologically meaningful way.

To overcome this limitation, a few approaches have been made to extract the rules from SVM. Núñez et al [5] introduced the SVM + Prototypes method. The concept of this method is that based on the output of the decision function from SVM, the K-means clustering algorithm is applied to determine prototype vectors (centers of clusters). By combining these prototype vectors with support vectors based on geometric methods, regions such as ellipsoids or hyper-rectangles are defined in the input space. Since each region defines a rule, all the regions can be transferred to if-then rules. However, this approach is known to have a scalability problem [6]. With the large number of patterns, if there is an overlap among different attributes, the explanation capability could suffer.

Barakat and Diederich applied the "learning-based" rule extraction technique to understand the SVM prediction[6]. Since SVM has a lack of interpretability, the authors combined it with a learning algorithm which has this capability by nature. They used the SVM as the first classifier to generate the patterns for the second learning algorithm. Based on these patterns, the second learning algorithm learns what the SVM has learned and generates rules as output. As a second learning algorithm, C5 decision tree algorithm is used to generate both the decision trees and rule sets. However, Barakat and Diederich's decision trees may generate the rules with much lower accuracy than that of SVM [7]. The reason is that some of the rules generated from their decision trees were based on the data set which has the same attributes used for training the SVM but with modified target (class) values by SVM.

He et al introduced a new learning-based rule extraction algorithm called SVM_DT [7]. This algorithm combines the SVM with the decision tree according to the following four steps. The first step is training the SVM. Next, from the output of the SVM, a correctly predicted set is chosen as a new training set for decision tree. Third, this new training data is applied for training a decision tree learning system and extracting the rule sets. Finally, the generated rules are decoded into biologically meaningful rules. This approach is performed well in terms of both the test accuracy of the rules

Manuscript received Oct 31, 2006. This research was supported in part by the NIH, under grants 3 R01 GM34766-17S1 (P.C.T), and 1 P20 GM065762-01A1 (R.W.H), and the NSF under grants CCF-0514750, and CCF-0646102 (Y.P), and the Georgia Cancer Coalition (R.W.H.). Hae-Jin Hu is a fellow of the Molecular Basis of Disease Program. *Asterisk indicates corresponding author.*
Robert Harrison is with the Computer Science Department and the Biology Department, Georgia State University.
Phang C. Tai is with the Biology Department, Georgia State University.
*Yi Pan is with the Computer Science Department, Georgia State University, Atlanta, GA 30303-4110, USA (fax: 404-463-9912, phone: 404-651-0649, e-mail: pan@cs.gsu.edu).

and the comprehensibility when it is applied to the problem of transmembrane segments prediction. However, the decision tree based method searches for rules locally based on a heuristic by adding one capable attribute at a time according to the order of goodness. This kind of attribute selection method may deteriorate the structure in which several attributes cooperatively decide the class.

As an alternative rule mining approach, this paper introduces the association rule (AR) mining method into the SVM prediction. This AR based 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 [8]. 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.

The aim of this paper is to interpret the protein structure prediction of SVM based on the association rule mining method. For this purpose, the transmembrane structure prediction is selected as a working domain, since the prediction consists of simple binary class decision of T (Transmembrane) or N (Not transmembrane). Second, a few association rule (AR) mining algorithms are studied to select a suitable system for this domain. Third, the selected AR mining algorithm is modified to adopt the system of this study and to improve the performance. Fourth, the positive and the negative patterns are obtained respectively from the generated rules by AR mining method. Finally, based on a new classification algorithm of this study, a test set is evaluated.

II. METHODS

A. Prediction of transmembrane proteins

Transmembrane (TM) proteins are the integral membrane proteins that can completely cross from the external to the internal surface of a biological membrane. These TM proteins have important functions in biological systems such as ion channels or receptors. Due to these essential roles in the cellular functions, the research of TM proteins has been appealing to many drug designers. However, because of their hydrophobic properties, the conventional experimental approach such as X-ray crystallography or NMR (Nuclear Magnetic Resonance) 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.

Traditional methods of identifying the TM segments are mostly based on the hydrophobicity scales [9, 10]. Ever since these hydrophobicity based schemes were introduced, there have been many approaches to improve prediction, such as refining the hydrophobicity scale [11, 12], improving the hydrophobicity scales directly [13], analyzing the transmembrane database statistically [14, 15], or applying evolutionary information into neural network [16].

In 2002, Chen et al. evaluated 27 advanced prediction schemes and many simple hydrophobicity based schemes using the high- and low- resolution data sets[17]. According to their analysis, there was no method which consistently showed the best performance. Also, they stated that the simple hydrophobicity based schemes showed less accuracy than the advanced schemes which do not entirely depend on the hydrophobicity scales.

The previous approach of this study was improving the performance of the prediction of transmembrane segments based on the support vector machine[18]. The position-specific scoring matrix (PSSM) was adopted as an optimal encoding scheme by testing different scoring matrices, such as hydrophobicity matrix, the combined orthogonal and Blosum62 matrix, and PSSM. By optimizing the sliding window size and the SVM kernel parameters, this PSSM encoding scheme demonstrated the highest accuracy in terms of Q_2 among the common prediction methods, and produced consistent results on the blind test data.

B. Support Vector Machine

The realistic data have a complicated and nonlinear relationship between class and the parameters that describe the data. The application of the SVM with a linear separation is of relatively little value. However, the SVM can be generalized to complicated spaces by using a non-linear kernel. The kernel is used to map the data in an arbitrary manner so that it can be resolved into separable classes. Clearly, the choice of kernel is critical to the success of the SVM. This study uses a radial basis kernel since it was optimal when used for secondary structure prediction[2, 3, 4].

$$K(x,y) = e^{-\gamma\|x-y\|^2} \tag{1}$$

Where x and y are two input vectors containing different feature values and γ is the radial basis kernel parameter. Based on the above radial basis kernel function, the final non-linear hyper plane decision function has the form

$$f(x) = sign\left(\sum_{i=1}^{SV} \alpha_i y_i K(x, x_i) + b\right) \tag{2}$$

Where x_i are the support vectors, SV is the number of support vectors, $K(x, x_i)$ is the kernel function, α_i are the Lagrange multipliers, b is the bias term. The SVM^{light} software was used to implement the SVM (http://svmlight.joachims.org/). Calculations were carried out on a DELL 4CPU 1.9GHz Xeon using a hyper threading Linux kernel (version 2.4.18smp - Dell installed Red Hat 8.0).

C. Association rule mining

C.1. Basic Concepts

A formal definition of association rule mining is as follows [19]: Let $I = \{i_1, i_2, ..., i_m\}$ be a set of literals, or items. Let X be an itemset which is a subset of I, $X \subseteq I$. Let $D = \{t_1, t_2, ..., t_n\}$ be a set of transactions called a transaction database. Each itemset X is related to a set of transactions $t_X = \{t \in D \mid t \supseteq X\}$ where each transaction t contains the itemset X.

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

$$\text{supp(X)} = |\text{X(t)}| / |\text{D}| \quad (3)$$

Where $X(t) = \{t \text{ in } D \mid t \text{ contains } X\}$. 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, $X \cap Y = \phi$. 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

$$\text{supp}(X \cup Y) / \text{supp}(X).$$

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.

C.2. Association rule mining algorithms

Most of the traditional association rule mining algorithms are based on the support-confidence model such as above[20]. This scheme measures the significance of an association rule based on two factors such as support and confidence [19]. Apriori is a famous, and commonly-used algorithm for mining frequent itemset based on this model[21]. The support-confidence model is suitable for analyzing the market basket data. 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 [22]. 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 [22]. 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 [23] 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.

Wang and Yang's information model is formally defined such as follows [22].

Let $E = \{e_1, e_2, \ldots\}$ be a set of distinct events. The event sequence is a sequence of events in E. A periodic pattern is an ordered list of events occurring repeatedly in the event sequence. The information contained by an event e_i is defined as

$$I(e_i) = -\log_{|E|} \Pr ob(e_i) \quad (4)$$

Where $|E|$ and $Prob(e_i)$ are the number of events in E and the probability that e_i occurs, respectively. As can be noticed from the above equation, the frequent event contains less information than that of a rare event.

A pattern P of length m is a tuple of m events in the form of $(p_1, p_2, \ldots, p_m)$ where $p_i \in E \cup \{*\}$, $(1 \le i \le m)$ and at least one position has to be filled with an event in E. Where the symbol * is used to represent the "don't care" position in a pattern. In a random event sequence without any advanced knowledge of the correlation among the events, a pattern P will occur with a probability

$$\text{Prob(P)} = \text{Prob}(p_1) \text{ x } \text{Prob}(p_2) \text{ x } \ldots \text{ x } \text{Prob}(p_m)$$

The information contained by P is

$$I(P) = -\log_{|E|} Prob(P) = I(p_1) + I(p_2) + \ldots I(p_m)$$

Given a pattern $P = (p_1, p_2, \ldots, p_m)$ and a segment S of m events $s_1, s_2, \ldots, s_m$, we say that S supports P if , for each event p_i in P, $p_i = *$ or $p_i = s_i$. For example, the segment a, g, c, b supports the pattern (a, g, *, *) while the segment b, g, c, d does not. The information gain of P in an event sequence D is defined as

$$G(P) = I(P) \; x \; (Support(P) - 1) \quad (5)$$

Where $Support(P)$ is the number of segments that supports P.

Besides the Wang and Yang's approaches, FOIL(First Order Inductive Learner), PRM(Predictive Rule Mining) and CPAR(Classification based on Predictive Association Rules) also applied the information metric for the rule generation. FOIL [24] 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.

$$gain(p) = |P^*| \left(\log \frac{|P^*|}{|P^*| + |N^*|} - \log \frac{|P|}{|P| + |N|} \right) \quad (6)$$

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 body of the new rule.

The FOIL algorithm was later further improved by Yin and Han to achieve higher accuracy and efficiency[25]. This algorithm was then further improved by the same authors to produce CPAR [25]. In CPAR, not only the attribute with the best gain generates a rule, but also the additional attributes with similar best gain generate new rules. To measure the accuracy of rules, CPAR adopts the Laplace accuracy. It is defined as follows given a rule *r*:

$$Laplace\ accuracy\ (r) = \frac{(N_c + 1)}{(N_{total} + m)} \quad (7)$$

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. Since our AR based classifier is based on this CPAR algorithm, it will be described in detail in the next section.

D. A New Rule Based Classifier: SVM_PCPAR

D.1. PCPAR classifier

The PCPAR(Pattern based Classification with Predictive Association Rules) is a modified version of CPAR (Classification based on Predictive Association Rules) [25]. CPAR is an integrated system of association rule (AR) mining and classification. CPAR adopts an information metric based algorithm which is more effective than support-confidence based counterpart in bioinformatics data mining. This algorithm is suitable for capturing the rules from the general cases in which each itemset is related randomly to one another. When we handle the dataset in which some itemsets are related to each other with the sliding window scheme, we should take another step to embed this knowledge into the generated rule sets. The PCPAR algorithm is devised based on this requirement.

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. It is based on the idea that each window of feature values may not contribute equally to become a positive or negative class. Some window might actively participate in decision making. Others might help a little bit without strong confidence. Since we use the SVM in the previous step, the prediction values obtained from SVM testing can be converted into the normalized weights representing the confidence of decision. With the help of this weight information, each window can participate with different initial weight for training.

The main differences of PCPAR and CPAR are in the post processing and the classification scheme. CPAR algorithm does not have any post processing step after rule generation. The rule generator generates the rules regardless of the fact that some of the itemsets are related with sliding window. 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 (rule body) 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.

$$\{87, 107\} \rightarrow \{261\},$$
$$\{107, 127\} \rightarrow \{261\},$$
$$\{127, 147\} \rightarrow \{261\}$$

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; even if the rules miss 'EE's occurring somewhere else like position 11 and 12 or 12 and 13, the pattern covers all these cases.

The classification scheme of CPAR is that it finds the best k (default = 5) rules for each class based on the subset concept. Once the average accuracy is obtained with k rules, the final class is determined as the class with the higher average accuracy value. However, this classification algorithm is not very effective with highly imbalanced data. In case of TM proteins of this study, 80% of data belong to the negative class (non-transmenbrane) and 20% belong to the positive class (transmembrane). The Laplace accuracies of the negative class which takes major portion of the whole data are generally higher than those of the positive class. If we apply the CPAR classification algorithm, most test data is classified as the negative class.

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 (TABLE I). 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.

D.2. SVM_PCPAR model

SVM_PCPAR model borrows the idea from the SVM_DT [7] for combining classifiers. This algorithm combines the SVM with a new AR based classifier, PCPAR(Pattern based Classification with Predictive Association Rules) with the following process. First, SVM is trained with the two highly performed encoding profiles including the orthogonal and Blosum62 combined matrix and PSSM. Next, based on 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 this process [7]. Third, the new training data is applied to PCPAR to train and generate the rules. Fourth, the generated rules are decoded and merged into the patterns. Finally, based on these patterns, new data is predicted. The pseudo-code of SVM_PCPAR model is presented in Fig. 1. Detailed description of the above process is given as follows.

Assume a training data set $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2),\ldots, (\boldsymbol{x}_m, y_m)\}$ is given, where $\boldsymbol{x}_i$ is the feature vector and y_i is the class label (target value) of the i-th training instance. Initially, SVM performs N-fold cross validation test. Where, the data set S is divided into N subsets with similar sizes. N runs are performed each with a different test set (Te_svm^i, i=1…N) and with the union of the other N-1 training set (Tr_svm^i, i=1…N). Next, by comparing the target values of test set (Te_svm^i, i=1…N) with those of prediction result P^i_svm, correctly predicted instances are selected to form a new data set (S^i_svm, i=1…N). Next, by applying the original test data Te_svm^i, i=1…N as test data set (Te_ar^i,i=1…N) and the union of the other N-1 subsets S^i_svm as the training set (Tr_ar^i,i=1…N), the AR mining system is trained to generate the rule sets. Once the rule sets are generated, they are decoded into biologically meaningful rules based on the decode table which maps each residue and its location into the feature values. The example of decoded rule bodies and the pattern sets generated by merging the same rules is presented in Fig. 2. 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, we can classify the test data using pattern match process. The PCPAR classification algorithm performs this to determine the final class of the amino acid in the middle of a sliding window.

The advantage of SVM_PCPAR is that it adopts more optimal rule generation scheme in which the dependency of multiple attributes are embedded in the rule generation process. Moreover, by applying a more advanced rule decoding and merging process, compact and easily understandable patterns can be obtained. Finally, with a new classification algorithm based on a pattern match process, a sequential test data can be classified effectively.

E. Data sets

This study adopted 165 low-resolution data set given by Rost et al [17]. According to the authors, the 165 protein set is expert-made set from the Swiss-Prot database which was originally collected by Möller et al [26]. This data set is applied to SVM classifier with the 7-fold cross validation test [27, 28]. As a blind test set, 497 transmembrane proteins were obtained from the Swiss-Prot (release 45.5) and TrEMBL (release 28.5) database [29] with the feature keyword 'transmem' and by filtering out the proteins which have 'possible', 'potential', or 'probable' transmembrane segments in their feature description and by removing the same sequences as the 165 data set.

F. Encoding Schemes

For SVM training, four different encoding schemes, including hydrophobicity matrix, orthogonal matrix, the combined orthogonal and Blosum62 matrix, and PSSM were tested with the 165 low-resolution data set by the 7-fold cross validation test [18]. Among the four different encoding schemes, the orthogonal and Blosum62 combined matrix and PSSM are selected for SVM training.

For training the AR based classifier, the sequence encoding is adopted for simplicity of decoding. This encoding consists of the sliding windows of 13 amino acids residues with the target value (class) of the 7th residue which resides in the middle of the window.

TABLE I
FOUR POSSIBLE CASES FROM THE PATTERN MATCH

Number of + patterns matched	Number of - patterns matched	Final Class
n	0	+
0	m	-
n	m	+ if (n/s > m/t), else -
0	0	-

n>0, m>0, s is the number of all patterns in the positive class and t is the number of all patterns in the negative class.

```
Input: training set S
Output: pattern set P

Process:
  FOR each subset i of N
        {Tr_svm^i, Te_svm^i }
            =Create_cross_validation_data(S)
  END FOR

  FOR each subset i of N
    Create prediction data P^i_svm=SVM(Tr_svm^i, Te_svm^i)
    SET new data set S^i_svm= φ
    FOR each case of P^i_svm_j in the prediction data P^i_svm
      IF P^i_svm_j is correct
        Create new data set S^i_svm= S^i_svm ∪ Te_svm^i_j
      END IF
    END FOR
  END FOR

  FOR each subset i of N
     SET new train data for AR mining Tr_ar^i = φ
     Create test data for AR mining
        Te_ar^i = SequenceEncoding(Te_svm^i)
     Create train data for AR mining
        Tr_ar^i = SequenceEncoding( Tr_ar^i ∪ {S^j_svm,
                                    j=1...N, j ≠ i} )
  END FOR

  SET rule set and pattern set R= φ , P= φ
  FOR each subset i of N
     Create rule set R^i = AR (Tr_ar^i, Te_ar^i)
     Create pattern set P^i = Decode_Merge(R^i)
     Calculate the accuracy of test set
        Te_ar^i = AR_Classify(Te_ar^i, P^i)
     SET P=P ∪ P^i
  END FOR
```

Fig. 1. Pseudo-code of SVM_PCPAR algorithm.

G. Prediction Accuracy

To evaluate the performance of the prediction scheme, the most commonly used two-state overall percentage measure, Q_2 is adopted. In this study, Q_2 measure counts the number of correctly predicted transmembrane residues out of all residues. This measure is defined as,

$$Q_2 = \frac{Number\ of\ correctly\ predicted\ residues}{Number\ of\ all\ residues} \times 100 \qquad (8)$$

III. Experimental Results

The average accuracies and the average numbers of rules or patterns for 7 fold test are compared in TABLE II and TABLE III. In both tables, the first two columns are the results from two different AR based classifiers, CPAR and PCPAR without pre-processing. The remaining columns are the results obtained by combining the SVM result with the AR based classifier with different schemes. In OB_SEQ, the orthogonal and Blosum62 combined matrix encoding is applied for SVM and sequence encoding for AR based classifier. In PSSM_SEQ, the PSSM is used for SVM and sequence encoding for AR based classifier. The last two columns are the results obtained by applying different initial weights in each sliding window of sequence encoding. As observed from these tables, the new PCPAR scheme performs better than CPAR method whether it is applied alone, or combined with SVM. In both cases, the accuracy is improved a bit (less than 2%) when combined with SVM. When compared among the combined schemes, OB_SEQ scheme shows better performance than PSSM_SEQ counterpart or weighted window scheme. The weighted window scheme does not improve the result compared with un-weighted one. This implies that it does not generate enough patterns from the training data to capture the relationship among the neighborhood residues.

With respect to the number of rules or patterns, the PCPAR scheme generates about 50 ~ 60 % less patterns than CPAR rules whether applied alone or combined with SVM. In both cases, fewer patterns are generated when combined with SVM. When compared among the combined schemes, OB_SEQ scheme generates fewer patterns than PSSM_SEQ scheme but still shows higher accuracy than PSSM_SEQ.

In terms of understandability, the pattern based PCPAR scheme is easier to understand the biological meaning than the rule based CPAR counterpart. For example, two rules in CPAR such as (A3 = V, A6 = I, A8 = I → positive class) and (A4 = V, A7 = I, A9 = I → positive class) can be expressed as (V**I*I → positive class) in PCPAR.

In Fig. 3, the frequency of amino acids in the rule body is displayed when the rules are generated based on 165 TM proteins. As can be seen, most non-polar amino acids such as A, I, L, M, F and V occur in the positive rule set only and most charged polar amino acids such as D, E, K and R occur in the negative rule set only.

The performance of SVM_PCPAR is compared with SVM_DT in TABLE IV. With respect to the accuracy, SVM_DT shows slightly better performance (0.8 %) than SVM_PCPAR. However, considering the rules or patterns generated, SVM_PCPAR is more effective than SVM_DT with about 60% less patterns. Moreover the patterns generated from

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***
(5) {67 87} → {261} 1.0	E 4 E 5	0.99
(6) {47 67} → {261} 1.0	E 3 E 4	*****SR****** 1.00
(7) {147 167} → {261} 1.0	E 8 E 9	******E*****A 1.00
(8) {167 187} → {261} 1.0	E 9 E 10	*****GE****** 1.00
(9) {81 122} → {261} 1.0	A 5 R 7	****E**K***** 1.00
(10) {132 191} → {261} 1.0	K 7 L 10	******R**A*** 0.99
		******PE***** 0.99

Fig. 2. Example of Decode_Merge process in SVM_PCPAR model. The first column is the positive and negative rules with the Laplace accuracy, the second is the decoded rule bodies, and the third column is the created patterns from the rule merge process. The Laplace accuracy values in the third column are the averaged values from the same rules.

SVM_PCPAR are easier to understand the biological meaning than SVM_DT. For example, a rule (A3 = I, A7 = L, A9 = V → positive class) in SVM_DT is presented with a simpler form such as (I***L*V → positive class) in SVM_PCPAR based on the discovered patterns from the rules.

To measure the quality of generated patterns from SVM_PCPAR, a blind test was performed based on the 497 TM protein set with known structure. In TABLE V, the result is compared with that of SVM_CPAR. As can be observed, SVM_PCPAR scheme showed better performance with respect to the accuracy and to the number of patterns.

IV. CONCLUSION

This study attempted to understand the protein structure prediction of SVM by combining the SVM with the AR mining method. To achieve this goal, transmembrane structure prediction was selected as an initial working domain. The new AR based classifier PCPAR was devised by modifying the existing classifier CPAR to adapt the sequential data encoding with the sliding window scheme. The results show that this new AR based classifier performs well in terms of accuracy and of the number of rules. This scheme can be applied in the problem of predicting protein solvent accessibility, identifying protein-protein interaction sites from primary structure, and predicting protein secondary structure. Since all these problems adopt the encodings based on sequential information with the sliding window method, the performance improvement can be expected by applying this new scheme.

In the future, for tuning up the current AR based classifier, the AR mining parameters will be optimized and different encoding schemes will be tested.

ACKNOWLEDGMENT

The authors would like to thank Professor T. Joachims for making SVM^{light} software available, Professor F. Coenen for making AR mining software available and Professor B. Rost for providing the 165 low-resolution and 36 high-resolution data sets.

TABLE II
ACCURACY COMPARISON OF CPAR AND PCPAR SCHEME (%)

Without preprocess	Encoding (SEQ)	Combined with SVM	Encoding (OB_SEQ)	Encoding (PSSM_SEQ)	Encoding (OB_SEQ) w/ weight
CPAR	81.0	SVM_CPAR	82.9	82.3	80.2
PCPAR	84.1	SVM_PCPAR	85.6	85.0	82.0

TABLE III
COMPARISON OF NUMBER OF RULES OR PATTERNS GENERATED

Without preprocess	Encoding (SEQ)	Combined with SVM	Encoding (OB_SEQ)	Encoding (PSSM_SEQ)	Encoding (OB_SEQ) w/ weight
CPAR	1408	SVM_CPAR	1168	1359	289
PCPAR	830	SVM_PCPAR	659	740	141

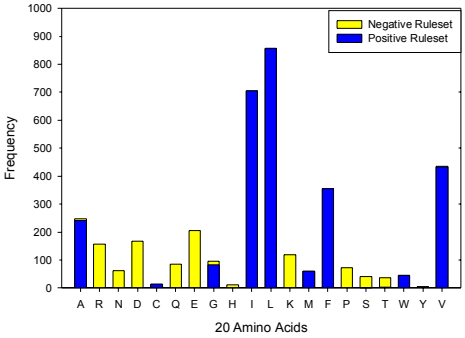

Fig. 3 Frequency of amino acids in the rule body of 165 TM proteins

TABLE IV
COMPARISON OF SVM_PCPAR WITH SVM_DT

Group	SVM_DT (PSSM_SEQ)	SVM_PCPAR (PSSM_SEQ)
1	83.1	85.3
2	80.7	81.8
3	90.9	88.7
4	88.8	87.1
5	83.0	83.5
6	85.7	84.7
7	88.2	84.3
Avg. accuracy (%)	85.8	85.0
Avg. # of rules (patterns)	about 2000	740

TABLE V
RESULT OF THE BLIND TEST BASED ON 497 TM PROTEINS

	Accuracy (%)	Number of Rules or Patterns
SVM_CPAR	80.6	1302
SVM_PCPAR	83.1	737

REFERENCES

[1] V. Vapnik and C. Cortes, "Support vector networks," *Machine Learning*, vol. 20, pp. 273-293, 1995.

[2] S. Hua and Z. Sun, "A Novel Method of Protein Secondary Structure Prediction with High Segment Overlap Measure: Support Vector Machine Approach," *J. Mol. Biol*, vol. 308, pp. 397-407, 2001.

[3] H. Kim and H. Park, "Protein Secondary Structure Prediction Based on an Improved Support Vector Machines Approach," *Protein Eng*, vol. 16, pp. 553-560, 2003.

[4] H. Hu, Y. Pan, R. Harrison, and P. C. Tai, "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, pp. 265-271, 2004.

[5] H. Núñez, C. Angulo, and A. Català, "Rule extraction from Support Vector Machines," presented at The European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 2002.

[6] N. Barakat and J. Diederich, "Learning-based Rule-Extraction from Support Vector Machine," presented at The third Conference on Neuro-Computing and Evolving Intelligence (NCEI'04), 2004.

[7] J. He, H. Hu, R. Harrison, P. C. Tai, 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 b.

[8] K. Wang, S. Zhou, 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.

[9] J. Kyte and R. F. Doolittle, "A Simple Method for Displaying the Hydropathic Character of a Protein," *Journal of Molecular Biology*, vol. 157, pp. 105-132, 1982.

[10] D. M. Engelman, T. A. Steitz, and A. Goldman, "Identifying nonpolar transbilayer helices in amino acid sequences of membrane proteins," *Ann. Rev. Biophys. Biophys. Chem.*, vol. 15, pp. 321-353, 1986.

[11] K. Nakai and M. Kanehisa, "A knowledge base for predicting protein localization sites in eukaryotic cells," *Genomics*, vol. 14, pp. 897-911, 1992.

[12] P. Klein, M. Kanehisa, and C. D. Lisi, "The detection and classification of membrane-spanning proteins," *Biochim Biophys Acta*, vol. 815, pp. 468-476, 1985.

[13] S. Jayasinghe, K. Hristova, and S. H. White, "Energetics, stability, and prediction of transmembrane helices.," *J Mol Biol*, vol. 312, pp. 927-934, 2001.

[14] K. Hofmann and W. Stoffel, "TMbase - A database of membrane spanning proteins segments," *Biol. Chem.*, vol. 374, pp. 166, 1993.

[15] C. Pasquier, V. J. Promponas, G. A. Palaios, J. S. Hamodrakas, and S. J. Hamodrakas, "A novel method for predicting transmembrane segments in proteins based on a statistical analysis of the SwissProt database:the PRED-TMR algorithm," *Protein Engin*, vol. 12, pp. 381-385, 1999.

[16] B. Rost, R. Casadio, P. Fariselli, and C. Sander, "Prediction of helical transmembrane segments at 95 % accuracy," *Protein Sci*, vol. 4, pp. 521-533, 1995.

[17] C. P. Chen, A. Kernytsky, and B. Rost, "Transmembrane helix predictions revisited," *Protein Science*, vol. 11, pp. 2774-2791, 2002.

[18] H. Hu, Y. Pan, R. Harrison, and P. C. Tai, "Transmembrane Segments Prediction with Support Vector Machine Based on High Performance Encoding Schemes," *Proc. of the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, 2004.

[19] C. Zhang and S. Zhang, *Association Rule Mining: Models and Algorithms*: Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 2002.

[20] R. Agrawal, T. Imielinski, and A. Swami, "Database mining: A performance perspective," presented at IEEE Transactions on Knowledge and Data Engineering, 1993a.

[21] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," presented at 20th Int'l Conference on Very Large Databases, Santiago, Chile, 1994.

[22] W. Wang and J. Yang, *Mining Sequential Patterns from Large Data Sets*: Springer, 2005.

[23] R. Blahut, *Principles and Practice of Information Theory*: Addison-Wesley Publishing Company, 1987.

[24] J. R. Quinlan and R. M. Cameron-Jones, "FOIL: A Midterm report," presented at European Conference on Machine Learning (ECML-93), Vienna, Austria, 1993.

[25] X. Yin and J. Han, "CPAR: Classification based on Predictive Association Rules," presented at SIAM Int. Conf. on Data Mining (SDM'03), San Fransisco, CA, 2003.

[26] S. Moller, E. V. Kriventseva, and R. Apweiler, "A collection of well characterized integral membrane proteins," *Bioinformatics*, vol. 16, pp. 1159-1160, 2000.

[27] B. Rost and C. Sander, "Combining evolutionary information and neural networks to predict protein secondary structure," *Proteins*, vol. 19, pp. 55-72, 1994.

[28] S. K. Riis and A. Krogh, "Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments," *J. Comput. Biol.*, vol. 3, pp. 163-183, 1996.

[29] A. Bairoch and R. Apweiler, "The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000," *Nucleic Acids Res.*, vol. 28, pp. 45-48, 2000.