Weave Amino Acid Sequences for Protein Secondary Structure Prediction

Xiaochun Yang
Department of Computer Science,
Brigham Young University,
Provo, Utah, 84602 U.S.A.
xiaochuny@cs.byu.edu

Bin Wang
Institute of Software,
Northeastern University,
Shenyang, 110004 China.P.R.
wang_b2001@hotmail.com

ABSTRACT

Given a known protein sequence, predicting its secondary structure can help understand its three-dimensional (tertiary) structure, i.e., the folding. In this paper, we present an approach for predicting protein secondary structures. Different from the existing prediction methods, our approach proposes an encoding schema that weaves physio-chemical information in encoded vectors and a prediction framework that combines the context information with secondary structure segments. We employed Support Vector Machine (SVM) for training the CB513 and RS126 data sets, which are collections of protein secondary structure sequences, through sevenfold cross validation to uncover the structural differences of protein secondary structures. Hereafter, we apply the sliding window technique to test a set of protein sequences based on the group classification learned from the training set. Our approach achieves 77.8% segment overlap accuracy (SOV) and 75.2% three-state overall per-residue accuracy (Q_3), which outperform other prediction methods.

General Terms

Design, Experimentation

Keywords

Protein structure prediction, protein secondary structure, SVM, encoding schema

1. INTRODUCTION

Understanding the function and the physiological role of proteins is a fundamental requirement for the discovery of novel medicines and protein-based products with medical and industrial applications. To enhance their understanding, many researchers focus on the functional analysis of genome these days. This functional analysis is a difficult problem, and the determination of protein three-dimensional structures is essential in the analysis of the genome functions. A protein three-dimensional structure can help understand its function. Unfortunately, it happens that three-dimensional protein structures cannot be accurately predicted from protein (amino acid) sequences. An intermediate, but useful and fundamental, alternative step is to predict protein secondary structures since a protein three-dimensional structure results partially from its secondary structure.

The protein secondary structures can be categorized into several states: a-helix (H), 3_{10} helix (G), pi helix (I), isolated β-bridge (B), extended β-strand (E), hydrogen bonded turn (T), bend (S) and random coil (-) [1]. In the previous work on predicting protein secondary structures, researchers focused on classifying these states into three consolidated classes: helix, strand and coil classes, among which the helix and strand classes are the major repetitive secondary structures [11]. These days the number of known protein sequences has far exceeded the number of known protein secondary structures, which have been obtained through chemical and biological methods such as the crystallography and NMR methods [14], and the rapid pace of the discovery of genomic sequences has further widened the gap. Thus, computational tools for predicting protein secondary structures from protein sequences are needed and they are essential in narrowing the widening gap [14]. After four decades of research work, the prediction of the secondary structures has not yet attained the satisfying accuracy rate. (See Table 1.)

The secondary structure prediction methods developed in the past [8; 17; 19] use local information of a single sequence, but they exhibit major drawbacks. For example, the accuracy measures (Q3) on these methods are relatively low (57% for the 1st generation and 63% for the 2nd generation). A usual criticism on using the neural network prediction approach [17] is that the methods are "black boxes." The neural network approach may be an effective classifier; however, it is hard to explain why a given pattern is classified as a helix rather than a strand, and thus it is difficult to derive a conclusion on the merit of the approach. Recently, a more promising approach on solving the predicting problem appears in [7]. Hua et al. introduce a prediction approach based on the SVM [18], which separated the input data, i.e., the protein secondary structure segments, into two classes to uncover the protein secondary structural classes.

However, most of the existing works did not consider the roles of physio-chemical properties of each amino acid residue, and to our best knowledge, none of them consider the roles of contexts of helices, strands, or coils in protein sequences which results in the lower accuracy prediction. Our prediction approach builds up an *encoding schema* that weaves the properties of different chemical groups into secondary structure segments and a *two dimensional prediction framework* that considers not only interested segments themselves (e.g. helices and strands) but also the contexts of those interested

Table 1: Prediction of protein secondary structures during the three generations

	1st Generation	2nd Generation	3rd Generation
Time Period	1960s - 1970s	1980s	1990s
Standard Method	GOR1 [8]	GOR3 [13; 19]	PHD [17]
Approach	Statistical Information	Local Interactions Among Amino Acids	Homologous Protein Sequences
Prediction Accuracy (Q3)	57%	63%	72%

segments. Encoding schema includes: (i) n-gram for determining the length of training segments, (ii) cardinal vectors for catching similar physio-chemical properties of amino acid residues, and (iii) probability of different residues for capture the tendency of a residue location has a conformation of helical (strand or coil) structure, while two dimensional prediction framework uses sliding window technique to exhaustively exam all the possible protein sequence segments and synthesizes classifiers derived from the helical (strand or coil) segments and context segments to the final prediction. We use two data sets CB513 [3] and RS126 [16], which contain 513 and 126 sequences of protein secondary structures, respectively, that have been discovered by using different chemical and biological experiments and employ SVM to train samples (segments) in the data sets. Hereafter, we use the two dimensional prediction framework to compare each predicted segments with known protein classes. Our prediction approach gets encouraging experimental results by adopting various accuracy measures, and it has some advantages over other prediction methods, which are listed as follows.

- It weaves part of biological information, such as chemical groups of each residue, and frequency of residues into the encoding schema which combines the classical encoding schema, matrix calculation, and statistical method, where we use classical encoding schema to represent the order of residues, matrix calculation to combine biological information and order of segments together, and statistical method to capture the frequency of each residue in the training set.
- Our two dimensional prediction is correlated, i.e. each prediction is not made in isolation, taking account of the predictions for neighboring residues. It allows protein classification based on not only the local regions of similarity but also the context regions of similarity in contrast to the detection of global similarities used in the traditional search techniques. Each prediction is based on multiple decision functions that get rid of the inherent bias/noise.
- The approach is theoretically sound since the SVM method has been adopted by researchers in mathematic and computer science since its discovery in 1997 in solving the classification problems, and it has been proved to be efficient in the determination of the decision function in separating two different (in our work, the protein) classes, which is exactly what we need in solving the prediction of protein secondary structures.

The rest of the paper is organized as follows. In Section 2, we simply introduce SVM, in Section 3, we propose an encoding schema that transforms amino acid segments into vectors for further training, and in Section 4, we present a two dimensional prediction framework, In Section 5, we discuss our experimental results, and finally we give our conclusions.

2. REVIEW OF SUPPORT VECTOR MACHINE

SVM [18] has been successfully applied in solving a wide range of pattern recognition problems, including face detection [12], text classification [4], and etc. It is popular due to effective avoidance of overfitting, the ability to handle large feature spaces, information condensing of data sets, and promising empirical performance.

The basic idea of SVM is that for a training set which contains l samples, where each sample is represented by an n-dimensional input vector $\vec{\mathbf{x}}_i$ $(1 \leq i \leq l)$ and the training set is regarded as an n-dimensional input space $\Re^n$, SVM tries to estimate a *decision function* f: $\Re^n \rightarrow \{\pm 1\}$, i.e. for each n-dimensional input vectors $\vec{\mathbf{x}}_i$ $(i = 1, 2, \ldots, l) \in \Re^n$, there is a classification identifier $y_i \in \{+1, -1\}$ that indicates to which class, $+1$ or -1, $\vec{\mathbf{x}}_i$ belongs. The decision function is shown as follows,

$$f(\vec{\mathbf{x}}) = sign(\sum_{i=1}^{l} y_i \alpha_i \cdot \vec{\mathbf{x}} \cdot \vec{\mathbf{x}}_i + b) \tag{1}$$

where l is the number of vectors $\vec{\mathbf{x}}_i$ $(1 \leq i \leq l)$ in the training set, vector $\vec{\mathbf{x}}$ is a variant that represents a testing vector in a test set, y_i $(\in \{+1, -1\})$ is a classification identifier, and coefficients α_i are obtained by solving the following convex Quadratic Programming (QP) problem:
Maximize

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{i=1}^{l} \alpha_i \alpha_j \cdot y_i y_j \cdot (\vec{\mathbf{x}}_i \cdot \vec{\mathbf{x}}_j) \tag{2}$$

with constraints

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \ 0 \leq \alpha_i, \ i = 1, 2, \ldots, l. \tag{3}$$

Vector $\vec{\mathbf{x}}$ can be classified by the decision function in the testing phase by calculating distances (dot products) between $\vec{\mathbf{x}}$ and $\vec{\mathbf{x}}_i$. (Note that the decision function depends only on the dot products of vectors in $\Re^n$.)

In the case where a linear boundary is inappropriate for classifying samples in the training set, a straightforward method can be used to map vectors in $\Re^n$ into a high dimensional *feature space* $\mathcal{H}$ via a nonlinear map Φ: $\Re^n \mapsto \mathcal{H}$. Therefore, SVM maps the input vector, $\vec{\mathbf{x}}_i \in \Re^n$, into a *feature vector*, $\Phi(\vec{\mathbf{x}}_i) \in \mathcal{H}$. Then the training problem only depends on the dot products of vectors, $\Phi(\vec{\mathbf{x}}) \cdot \Phi(\vec{\mathbf{x}}_i)$, in $\mathcal{H}$. However, if $\mathcal{H}$ is a high-dimensional space, $\Phi(\vec{\mathbf{x}}) \cdot \Phi(\vec{\mathbf{x}}_i)$ will be very expensive to compute [12]. Fortunately, the dot product in $\mathcal{H}$ has an equivalent expression $K(\vec{\mathbf{x}}, \vec{\mathbf{x}}_i)$ in $\Re^n$ such that $K(\vec{\mathbf{x}}, \vec{\mathbf{x}}_i) = \Phi(\vec{\mathbf{x}}) \cdot \Phi(\vec{\mathbf{x}}_i)$, where $K(\vec{\mathbf{x}}, \vec{\mathbf{x}}_i)$ is a kernel function. Therefore, the kernel function is used to perform $\Phi(\vec{\mathbf{x}}) \cdot \Phi(\vec{\mathbf{x}}_i)$ in the input space $\Re^n$ rather than the potentially high dimensional feature space $\mathcal{H}$, i.e. using kernel function in $\Re^n$ without need to explicitly know what Φ is. If the kernel function contains a bias term[1], b can be accommodated within the kernel function, and hence the decision function in Equation 1 can be rewritten into Equation 4.

$$f(\vec{\mathbf{x}}) = sign(\sum_{i=1}^{l} y_i \alpha_i \cdot K(\vec{\mathbf{x}}, \vec{\mathbf{x}}_i)) \tag{4}$$

where coefficients α_i satisfy the Equation 5 with the constraints in Equation 6.
Maximize

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{i=1}^{l} \alpha_i \alpha_j \cdot y_i y_j \cdot K(\vec{\mathbf{x}}_i, \vec{\mathbf{x}}_j) \tag{5}$$

with constraints

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \; 0 \le \alpha_i \le C, \; i = 1, 2, \ldots, l. \tag{6}$$

where, C is a unper bound value. A large C assigns a high penalty to the classification errors[12].
The most commonly accepted employed kernel functions are shown in Table 2. For a given training set, only the kernel function and the parameter C are needed to be selected to specify one SVM.

3. ENCODING SCHEMA

Each known protein sequence is composed of a number of structured segments. We first extract n-gram segments from known protein sequences with consideration of structured segments and the contexts of them. Then we represent each extracted segments as an input vector for the further training and prediction.

3.1 N-gram Extraction

It has been well accepted that the order in an amino acid sequence plays the most important role in predicting its secondary structure. Having looked at a lot of sequences, we know which segment tends to construct a secondary structure and which does not. For this task, we cannot possibly consider each residue separately. Therefore, we use n-gram model (any segments with n amino acid residues, where $n > 0$) to extract segments from amino acid sequences and transform them into input vectors. The total number of possible segments of an amino acid sequence using n-gram extraction is $m-n+1$, where m (>0) is the number of residues in the sequence. The choice of n is determined by the average length of different structure segments (helix or strand). Based on our study, for most helices the amino acid length is between five and 17, and for most strands the length is between five and 11. Thus, we will choose varied n between five and 17 and begin with a small size and gradually increase the size to determine the "best" size for future helix/non-helix (or strand/non-strand) prediction. (It is anticipated that the best sizes, n, of different classifications should achieve the best prediction result for their classification.)

3.1.1 *Subsequences without context*

In the case that extracting segments from a training set without considering context, we only consider structure segments (i.e., helix, strand, or coil) in known protein sequences, which includes identifying structure segments in protein sequences from the training set and extracting n-gram subsequences $m-n+1$ times from each structure segment sequentially, where m is the number of residues in the segment.

Example 1. Given a protein sequence "`MNIFEMLRID`" with a helix segment from position 2 to 7 (underlined), the four 3-gram subsequences are "`NIF`," "`IFE`," "`FEM`," and "`EML`."

3.1.2 *Subsequences with context*

Locally encoding structure segments has its inherent disadvantage, for example, the contexts of structure segments has been excluded. Consider the protein sequence "`MNIFEML RID`" again. Residue '`M`' and subsequence "`RID`" are contexts of helix segment "`NIFEML`" which would lead us to infer that the nearest neighbors (one or more residues) of "`NIFEML`" must have significant different properties from "`NIFEML`," otherwise these neighbors should be included in "`NIFEML`" as a partial helical structure.

Definition 1. N-gram context subsequences of structure segments. Given a protein sequence s and a secondary structure segment s_s of size m that begins at position i in s, the *left n-gram context* of s_s is a subsequence of size n that begins at position i - $\lfloor \frac{n}{2} \rfloor$ in s, and the *right n-gram context* of s_s is a subsequence of size n that begins at position $i + m$ - $\lceil \frac{n}{2} \rceil$ in s.

Example 2. Given a protein sequence "`DQAMRYKT`," the underlined residues are helical structures. The left 5-gram context subsequence of "`QAM`" is "`∅DQAM`," where "`∅D`" are the left two neighbors, where $\emptyset$ indicates that the followed "`D`" is the head of the sequence. The right 5-gram context string of it is "`QAMRY`," where "`RY`" are the right two neighbors.

3.2 Encoding Schema of Amino Acid Residues

The transformation from n-gram subsequences into input vectors is an *encoding* process. The classical encoding schema was adopted along with using neural network in predicting protein secondary structures [13], where in a segment, each residue is encoded as an orthogonal binary vector $[1, 0, \ldots, 0]$, $\ldots$, or $[0, 0, \ldots, 1]$, in which '1' indicates the relative position of the corresponding residue. This type of vectors is 21-dimensional, and each one of the first twenty coefficients of these vectors denotes a distinct amino acid in the segment and the last unit is used for specifying whether the corresponding residue is the "head" or the "tail" of the sequence [13]. However, even if chemical and biological methods, such as crystallography and NMR, are used, the "head" and the "tail" of a segment can not be accurately determined [14]. Therefore, we use the first 20 units of each 21-dimensional vector to represent a residue, which we call 20-dimensional *residue vector*. In this paper, we adopt the following ordered list of residues in [11] to encode a residue at the relative position in its corresponding sequence.

[1]Many employed Kernels have a bias term and any finite kernel function can be made to have one[12].

Table 2: The commonly accepted kernel functions

Typical Kernel function	Explanation
$K(\vec{\mathbf{x}}_i, \vec{\mathbf{x}}_j) = ((\vec{\mathbf{x}}_i \cdot \vec{\mathbf{x}}_j) + \theta)^d$	Polynomial function
$K(\vec{\mathbf{x}}_i, \vec{\mathbf{x}}_j) = \exp(-\gamma\vert\vec{\mathbf{x}}_i - \vec{\mathbf{x}}_j\vert^2)$	Gaussian radial basis function
$K(\vec{\mathbf{x}}_i, \vec{\mathbf{x}}_j) = \tanh(\vec{\mathbf{x}}_i \cdot \vec{\mathbf{x}}_j - \theta)$	Multi layer perceptron function

Table 3: Five groups according to chemical properties of amino acid residues.

R-group classification	5D cardinal vectors	Amino acid residues
Nonpolar, aliphatic (c_1)	[1, 0, 0, 0, 0]	`A,V,L,I,M`
Aromatic (c_2)	[0, 1, 0, 0, 0]	`F,Y,W`
Polar, uncharged (c_3)	[0, 0, 1, 0, 0]	`G,S,P,T,C,N,Q`
Positively charged (c_4)	[0, 0, 0, 1, 0]	`K,H,R`
Negatively charged (c_5)	[0, 0, 0, 0, 1]	`D,E`

A, V, L, I, M, F, Y, W, G, S, P, T, C, N, Q, K, H, R, D, E

Example 3. Given a training set of protein sequences, the 20-dimensional residue vector of 'V" is $[0,1,0,\ldots,0]_{1\times 20}$, where the second coefficient of the vector is 1, since 'V' is the 2nd residue in the ordered list given above.

In our prediction approach, instead of using 20-dimensional residue vector to represent a residue, we incorporate statistical information and the physio-chemical features of each residue into our encoding schema to enhance the classical approach by considering the probability of each residue and different biological groups [11] to which a residue belongs. In Table 3, twenty different amino acid residues are classified into five different *R-groups*, which combines Hydropathy index, molecular weights M_r, and pI value together [11]. Besides this R-group classification, amino acid residues of secondary structure sequences can be classified into many different groups according to their biological meanings, such as exchange group[5], hydrophobicity group[11], and etc (see appendix). [11] indicates that if two amino acids have similar biological properties, they should form similar structures. Therefore, we define *cardinal vector* to incorporate the different biological groups information of amino acid residues into a secondary structure sequence.

Definition 2. Cardinal vector. Given an m-class biological group of amino acid residues, its m-dimensional (mD) cardinal vector is a binary vector $\hat{vc}_m = [a_{c_1}, a_{c_2}, \ldots, a_{c_m}]$, where a_{c_i} $(1 \le i \le m) \in [0,1]$ and $\sum_{i=1}^{m} a_{c_i}^2 = 1$.

Definition 3. Probability of an m-class group with helix (strand or coil) structure in a training set. Given a training set s and an m-class group C=$\{c_1, \ldots, c_m\}$ of amino acid residues, the probability of a residue i in c_j (1$\le i \le m$) has helical structure is:

$$p^H_{i,c_j} = \frac{1}{N_H} \sum_{i \in c_j} N_{H_i} \tag{7}$$

where p^H_{i,c_j} $(1 \le i \le 20)$ is the probability that residue i belongs to group c_j and is a helix in s, N_H is the total number of residues that have the conformations of helical structures in the training set, and N_{H_i} is the number of residues in which residue i belongs to c_j and residue i is a helix in s. (p^E_{i,c_j} for strand and p^C_{i,c_j} for coil structures are computed accordingly.)

We use Kronecker product[20] of residue vectors, cardinal vectors, and frequency of residues in the training set to encode a residue, and we use concatenation of vectors of n consecutive residues to encode an amino acid subsequence with n residues. Residue vectors distinguish a residue from others, cardinal vectors catch the similar physio-chemical properties of residues, the frequency of a residue captures the tendency of the residue that locates in which structure, and concatenation of vectors exhibits the order of residues in a protein subsequence.

Definition 4. Encoding schema of protein segments. Given a training set s and an m-class biological group C=$\{c_1, \ldots, c_m\}$, the encoded vector of a residue i w.r.t. a c_j (1$\le i \le m$) is

$$\begin{aligned} v_{i,c_j} &= p^H_{i,c_j} \times v_{r_i} \otimes \hat{vc}_{m_{c_j}} \\ &= p^H_{i,c_j} \times [a_1 \times \hat{vc}_{m_{c_j}}, \ldots,\ a_{20} \times \hat{vc}_{m_{c_j}}] \\ &= [p^H_{i,c_j} \times a_1 \times a_{c_1}, \ldots, p^H_{i,c_j} \times a_{20} \times a_{c_1}, \ldots, \\ &\quad\ p^H_{i,c_j} \times a_1 \times a_{c_m}, \ldots, p^H_{i,c_j} \times a_{20} \times a_{c_m}] \end{aligned} \tag{8}$$

where v_{r_i} (= $[a_1, a_2, \ldots, a_{20}]$) is the residue vector of residue i, $\hat{vc}_{m_{c_j}}$ (= $[a_{c_1}, a_{c_2}, \ldots, a_{c_m}]$) is the cardinal vector of the m-class group to which i belongs, and p^H_{i,c_j} is the probability that residue i is a helix in s and is belonged to group c_j. For each subsequence s_s in s, the encoded schema of s_s is the concatenation of vectors of the consecutive residues in s_s. Hence, the size of an encoded vector of a residue is 20 $\times$ m, and the size of an encoded vector of subsequence with n residues is 20 $\times$ m $\times$ n.

Example 4. Consider a training set that contains the following two known protein sequences and their DSSP structures,[2]

$$s_1 : \underbrace{\texttt{LWQFNGMIKCK}}_{H}\ \underbrace{\texttt{PS}}_{T(Coil)}\ \underbrace{\texttt{PLLD}}_{H}\ \underbrace{\texttt{FNN}}_{S(Coil)}\ \underbrace{\texttt{GCY}}_{T(Coil)}$$

$$s_2 : \underbrace{\texttt{NLLLSDW}}_{E}\ \underbrace{\texttt{W}}_{\_(Coil)}\ \underbrace{\texttt{HQ}}_{S(Coil)}\ \underbrace{\texttt{S}}_{\_(Coil)}\ \underbrace{\texttt{IHKQEVGLS}}_{H}$$

where H is α-helix, E is extended β-strand, T is hydrogen bonded turn, S is bend, and _ is random coil. Hence,

Helix (H)	: `LWQFNGMIKCK`, `PLLD` and `IHKQEVGLS`
Strand (E)	: `NLLLSDW`
Coil (S, T, _)	: `PS`, `FNNGCY`, and `WHQS`

[2]DSSP (Dictionary of Secondary Structure Protein) is available at http://www.sander.ebi.ac.uk/dssp.

Let's consider the first helix segment, i.e. "LWQFNGMIKCK," extracted from s_1 and the first coil segment, i.e., "PS," also extracted from s_1. If we use 3-gram and R-group, i.e., 5-group (see Table 3) to train (determine) a helix/non-helix classifier, "LWQ" is represented as a 300-dimensional (20 $\times$ 5 $\times$ 3) vector V_1, in which the first 100 units of V_1 (see below) encodes the amino acid residue 'L.' Since 'L' belongs to the *Nonpolar, aliphatic R group* (c_1), the probability that 'L' is a helix and 'L' belongs to c_1 (in the training set) is 33% since $p^H_{L,c_1} = \frac{8}{24} = 0.33$, where 24 is the number of helix, and 8 is the sum of all the residues that belong to c_1 that have the conformation of helical structure in the training set.

$$V_1 = [\ \underbrace{\underbrace{0,\ldots,}_{10}\underbrace{0.33}_{p^H_{L,c_1}},\underbrace{0,\ldots,}_{89}}_{L}\underbrace{\underbrace{0,\ldots,}_{35}\underbrace{0.08}_{p^H_{W,c_2}},\underbrace{0,\ldots,}_{62}}_{W}\underbrace{\underbrace{0,\ldots,}_{70}\underbrace{0.33}_{p^H_{Q,c_3}},\underbrace{0,\ldots}_{29}}_{Q}\ ]$$

Both strand and coil structures can be regarded as non-helix in the helix/non-helix classification. Thus, "PS_" is a non-helix subsequence with size 3, and '_' in "PS_" denotes a space needed for making up the required length. The encoding vector of "PS_" is V_2, which is shown below.

$$V_2 = [\ \underbrace{\underbrace{0,\ldots,}_{52}\underbrace{0.33}_{p^H_{P,c_3}},\underbrace{0,\ldots,}_{47}}_{P}\underbrace{\underbrace{0,\ldots,}_{45}\underbrace{0.33}_{p^H_{S,c_3}},\underbrace{0,\ldots}_{54}}_{S}\ \underbrace{\underbrace{0,\ldots}_{100}}_{\_}\ ]$$

4. PREDICTION FRAMEWORK

4.1 Training of Encoded Vectors Using SVM

Protein secondary structure prediction is a typical three-class classification problem. A simple strategy to handle the three-classification problem is to cast it to a set of binary categorizations. For an n-class classification, $n-1$ classes are used to construct training sets. The ith class will be trained with all of the samples in the ith class with "y = +1" labels and all other example data with "y = −1" labels. If input vectors are transformed from helix (strand or coil) segments, then their learned decision function is called *vertical decision function*, denoted f_v. Otherwise, if input vectors are transformed from context subsequences, then their learned decision function is called *horizontal decision function*, which can be further classified into *left horizontal decision function* derived from the left contexts subsequences, denoted f_{hl}, and *right horizontal decision function* derived from the right contexts subsequences, denoted f_{hr}. The final prediction synthesizes f_v and f_{hl} (f_{hr}) that will be discussed in detail in Section 4.3.

4.2 Sliding Window

The sliding window method is adopted to move the window over the test protein sequence. A window, which is a one-dimensional frame of slots, can be adopted for overlaiding on and moving over a test protein sequence to examine different segments of its amino acids. It extracts an protein segment of length L ($L \geq 1$) to match a potential structure class using the derived decision function generated after the training phase. Note that the size of the sliding window should be the same size of the n-gram so that the test vectors can be fed into the same input space with the training vectors.

4.3 Synthesis of Deduced Decision Functions

We develop a heuristic algorithm to synthesize the deduced decision function for the final prediction.

Algorithm 1. Prediction of a protein sequence s with a sliding window of size n, and return a labeled sequence.

```
string predict (string s, int n, SVMmodel model,
          SVMmodel l_model, SVMmodel r_model)
{
  bool mark = true;
  int pre_p = -1; //prediction of the previous segment
  string s_p = new(strlen(s)); // predict sequence
  int begin = 0; //begin position of sliding window
  while (begin < strlen(s)) {
     int current = svm_predict(segment(s,i,n), model);
     int left = svm_l_predict(l_seg(s, i, n), l_model);
     int right = svm_r_predict(r_seg(s, i, n), r_model);
     if (mark)
        begin += l_synthesis(s_p, begin, current,
                             left, n, &pre_p, &mark);
     else
        begin += r_synthesis(s_p, begin, current,
                             right, n, &pre_p, &mark);
  }
  return s_p;
}
int l_predict(int begin, int current, int left,
          int size, bool *pre_p, bool *mark)
{
  int len = size;
  if((current==-1) && (left==-1))
     { mark_seq(s_p, begin, size, -1); *pre_p = -1; }
  else if ((current==1) && (left==-1)) {
     if (begin_position == 0)
        { mark_seq(s_p, begin, len, 1); *pre_p = 1; }
     else {  mark_seq(s_p, begin, len, *pre_p);
             if (*mark)  *mark = false;
             else *mark = true;
     }
  }
  else if ((current==-1) && (left==1)) {
     len = size/2; mark_seq(s_p, begin, len, -1);
     *mark = false; *pre_p = -1;
  }
  else if ((current==1) && (left==1)) {
     len = 1; mark_seq(s_p, begin, len, 1);
     *pre_p = 1;
  }
  return len;
}
int r_predict(int begin, int current, int right,
            int size, bool *pre_p, bool *mark)
{
  int len = size;
  if((current==-1) && (right==-1))
     { mark_seq(s_p, begin, len, -1); *pre_p = -1; }
  else if ((current==-1) && (right==1)) {
     len = 1; mark_seq(s_p, begin, len, 1);
     *pre_p = 1;
  }
  else if ((current==1) && (right==-1)) {
     mark_seq(s_p, begin, len, *pre_p);
     if ((*pre_p)==1) *mark = true;
     else *mark = false;
  }
  else if ((current==1) && (right==1)) {
     len = size/2; mark_seq(s_p, begin, len, 1);
     *pre_p = 1;
  }
  return len;
}
```

For a helix/non-helix (strand/non-strand, coil/non-coil) classifier, we select the best accuracy value (will be discussed in Section 5.1), and the corresponding decision function $f(\vec{\mathbf{x}})$ (including the kernel function and the upper bound C), and n-gram for the prediction phase. Thus, given an unknown sequence s, we first use sliding window with the window size n and different biological groups (cardinal vectors) to transform subsequences of s into different sets of input vectors. Hereafter, for each set of vectors, we apply the derived decision functions to classify each subsequence into either the helix or non-helix (strand or non-strand, coil or non-coil) class. If different biological groups cause conflict predictions, a vote strategy will be used to decide the final prediction.

5. EXPERIMENTAL DATA

The experiments are carried out on the CB513 set and RS126 set with 513 and 126 non-redundant protein amino acid sequences, respectively, whose secondary structures are known. In these two data sets, the assignment of secondary structure is usually performed by DSSP[9], STRIDE[6], or DEFINE[15].

Different assignment methods have different explanations on partial of residues in protein sequences, therefore they will influence the prediction accuracy as discussed by the constructor of CB513. In this paper, we select DSSP assignment in order to compare with other prediction approaches. DSSP assignment classify eight states secondary structure, `H`, `G`, `I`, `B`, `E`, `T`, `S`, and `-` [1], into three classes *helix*(H), *strand*(E), and *coil*(C). Here, we use methods in [7; 17] that call `H`, `G`, and `I` as *helix*(H), `E` as *strand*(E), and other states as *coil*(C). This assignment is slightly different from other assignments reported in different literatures. For example, in the CASP competition[10], H contains DSSP classes `H` and `G`, while E contains DSSP classes `E` and `B`.

We use multiple cross-validation trials to minimize variation in the results caused by a particular choice of training or test sets. An sevenfold cross validation is used on CB513 (RS126), where CB513 (RS126) is divided into seven subsets randomly, and each subset has similar size. Among these seven subsets, data in six subsets are selected as training data and data in the remaining subset is test data.

5.1 Evaluation Metrics

For a given sequence, we evaluate our prediction approach and compare it with some prediction methods that have been well accepted using evaluation metrics. First, we evaluate the effectiveness of SVM in our prediction application domain. Hereafter, we select the best parameters (e.g. kernel function and upper bound C) in SVM according to the highest accuracy value in Equation 9, and use the selected paraments for further prediction and comparison. For the evaluation of the effectiveness of our prediction approach, we give the following metrics.

The prediction accuracy using SVM is calculated as:

$$AC_{SVM} = \frac{N_{c_i}}{N_i} \times 100\% \qquad (9)$$

where i is a test set, N_{c_i} is the number of residues correctly predicted in i, and N_i is the number of residues in i.

We also employed several standard performance measures to assess prediction accuracy. Three-state overall per-residue accuracy (Q_3) and segment overlap calculation *SOV* [17] were calculated to evaluate accuracy.

Q_3, The overall per residue three-state accuracy method [17], is the most widely-used method for assessing the quality of a predicting approach, which is defined as

$$Q_3 = \frac{N_H + N_S + N_C}{N} \times 100\% \qquad (10)$$

where N_H, N_S, and N_C are the numbers of residues correctly predicted in the helix, strand, and coil, respectively, and N is the number of residues in the predicted sequences. Q_3, however, does not consider the problem of over-predictions, e.g. non-helix residues are treated as helix, or under-predictions, e.g. helix residues are treated as non-helix.

Another popular accuracy measurement, the *Segment overlap calculation* (SOV) has been proposed to assess the quality of a prediction in a more realistic manner. This is done by taking into account the type and position of secondary structure segment, the natural variation of segment boundaries among families of homologous proteins and the ambiguity at the end of each segment. SOV is calculated as

$$Sov = \frac{1}{N}\sum_{s} \frac{minov(s_{obs}; s_{pred}) + \delta}{maxov(s_{obs}; s_{pred})} \times len(s_{obs}) \qquad (11)$$

where N is the number of residues in the predicted sequences, s is a segment in the predicted sequences, s_{obs} and s_{pred} are observed and predicted segments, respectively, $minov$ is the actual overlap between s_{obs} and s_{pred}, $maxov$ is the extent of the either segment, and $len(s_{obs})$ is the length of the observed segment. δ is an integer chosen to be smaller than $minov$ and smaller than one-half the length of s_{obs}, δ = 1, 2, or 3 for short, intermediate, and long segments.

The per-residue accuracy for each type of secondary structure ($Q_H, Q_E, Q_C, Q_H^{pre}, Q_E^{pre}, Q_C^{pre}$) was also calculated. We distinguish Q_I and Q_I^{pre} (here, I=H, E, and C) as:

$$Q_I = \frac{N_{I_{cp}}}{N_{I_o}} \times 100\%, \qquad Q_I^{pre} = \frac{N_{I_{cp}}}{N_{I_p}} \times 100\% \qquad (12)$$

where, $N_{I_{cp}}$ is number of residues correctly predicted in state I, N_{I_o} is number of residues observed in state I, and N_{I_p} is number of residues predicted in state I.

In the following sub-sections, we use LIBSVM [2], an SVM software, to evaluate our prediction approach from different point of views, which include (i) determination of kernel function and parameter C, (ii) selection of window size, and (iii) comparison with other prediction methods.

5.2 Determination of Kernel Functions and Parameters in SVM

Given a data set, a proper kernel function and its parameters must be chosen to construct the SVM classifier. Different kernel functions and paraments will construct different SVM classifier. We hope to select an optimal kernel function and its parameters which can achieve high AC_{SVM}. However, there are no successful theoretical methods for determining the optimal kernel function and its parameters. In [7], the author has proved that the gaussian radial basis function $\exp(-\gamma|\vec{\mathbf{x}}_i - \vec{\mathbf{x}}_j|^2)$ can provide superior performance in generalization ability and converge speed. Therefore, we select different parameters and upper bound values C on $\exp(-\gamma|\vec{\mathbf{x}}_i - \vec{\mathbf{x}}_j|^2)$ to compare the testing accuracies on helix/non-helix (H/¬H) test. Since the average of helix in

CB513 is eight, we use 8-gram to select the kernel function and parameter C for H/¬H.

As shown in Table 4, for each C, we select γ from 0.001 to 50, where 0.001 is the ratio of 1 to the number of dimensions 800 (20 × 5 × 8) in the training set. We found that the best accuracy (84.72%) is achieved by $\exp(-\gamma|\vec{\mathbf{x}}_i - \vec{\mathbf{x}}_j|^2)$ with γ being equal to 0.2 and C being equal to 1.5. It is well known that a large C value imposes a high penalty to the classification errors [12]. However, Table 4 shows that for each γ between 0.001 and 50, when C increases, the accuracy on the test set first increases to a peak value, and then decreases. This is because too small a value of C gives a poor accuracy while too large a value of C leads to overfitting. A proper value of C balances these problems. We found that C=1.5 is appropriate for not only H/¬H but also other classifiers. In the following tests we set $C = 1.5$ and $\gamma = 0.2$ to construct different SVM classifiers.

5.3 Dependency of Window Size

There is a trade off between the number of residues used and the level of "noise" in deciding the size of the window. If the window size is too large, the prediction process may be misled by the noise, while if the window size is too small, then the number of residues for predicting the protein structure may be insufficient. Therefore, A proper window size could lead to a good performance.

We construct three different binary classifiers. The optimal window size, n_{op}, for each binary classifier was determined on the CB513. The results in Table 5 indicate that the optimal window size is related to the average length of the secondary structure segments. In general, longer n_{op} values mean segments require larger optimal window sizes. (The average length of helix, strand, and coil in CB513 are 8, 5, and 5, respectively.) It is interesting to find that the optimal window size based on our prediction approach is the similar size with the one based on previous NN approaches [13; 17], e.g. the optimal window sizes for H/¬H are both 13.

5.4 Effectiveness of Our Prediction Approach

We now evaluate the effectiveness of our prediction approach. We first compare our prediction approach with PHD [17] and an SVM-based method [7]. PHD is one of the most accurate and reliable secondary structure prediction methods based on NNs, and the SVM-based method is a new proposed method that can outperform PHD. Our comparison is based on the same data sets (CB513 and RS126), the same secondary structure definitions, and the same accuracy assessments. For easy to illustrate, we call the SVM-based method SVM_{01} and our prediction method SVM_{SSP}. The comparison results among PHD, SVM_{01}, and SVM_{SSP} are shown in Table 6, where '-' means that the results cannot be obtained from the papers.

On the CB513 set, the Q_3 is 75.2%, which is 1.7% higher than SVM_{01}, and the SOV with SVM_{SSP} achieves 77.8%, which is 1.6% higher than SVM_{01}. Q_3 and SOV scores for the RS126 set are improved by 3.0% and 2.9%, respectively, compared with PHD results.

The comparison results between PHD and the two SVM-based methods indicate that the methods based on SVM outperformed PHD, because SVM-based method can successfully avoid many problems which other machine learning approaches often encounter. For example, structures of NNs (especially the size of the hidden layer) are hard to be determined, gradient based training algorithms only guarantee finding local minima, too many model parameters have to be optimized, overfitting problems are hard to be avoided, etc. In addition, the comparison results between SVM_{01} and SVM_{SSP} prove that the encoding schema of SVM_{SSP} and two dimensional prediction framework play significant roles on predicting protein secondary structures.

Table 6: Comparison with results of other systems.

	CB513		RS126		
	SVM_{01}	SVM_{SSP}	PHD	SVM_{01}	SVM_{SSP}
Q_3	73.5%	75.2%	70.8%	71.2%	73.8%
Q_H	75.0%	77.5%	72.0%	73.0%	75.1%
Q_E	60.0%	64.8%	66.0%	58.0%	61.2%
Q_C	79.0%	79.9%	72.0%	75.0%	81.2%
Q_H^{pre}	79.0%	82.1%	73.0%	77.0%	80.9%
Q_E^{pre}	67.0%	70.5%	60.0%	66.0%	69.7%
Q_C^{pre}	70.0%	70.6%	-	69.0%	70.1%
Sov	76.2%	77.8%	73.5%	74.6%	76.4%

6. CONCLUSIONS

In this paper, we propose an approach in predicting protein secondary structures using SVM. Our prediction approach takes a segment extracted from a given sequence using a sliding window and matches it against the structural classification results computed by SVM. We incorporate the biological classifications of amino acid residues and statistical information into an encoding schema, consider not only structured segments but also their contexts, and choose the most ideal kernel function, parameters, and size of the window for the prediction of different protein structure classes. The experimental results show that our two dimensional prediction framework outperforms the existing prediction methods.

7. ACKNOWLEDGEMENTS

The authors would like to thank Jianyin Shao of the Chemistry and Biochemistry department at Brigham Young University for his helpful discussions and suggestions. Special thanks are also given to Chih-Chung Chang and Chih-Jen Lin for providing LIBSVM.

8. REFERENCES

[1] P. D. Bank. http://www.rcsb.org/pdb/, 2002.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: a Library for Support Vector Machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.

[3] J. A. Cuff and G. J. Barton. Evaluation and Improvement of Multiple Sequence Methods for Protein Secondary Structure Prediction. *Proteins: Struct. Funct. Genet.*, 34:508–519, 1999.

[4] H. Drucker, D. Wu, and V. Vapnik. Support Vector Machines for Span Categorization. *IEEE Trans. on Neural Networks*, 10:1048–1054, 1999.

[5] M. O. D. (ed). Atlas of Protein Sequence and Structure. *National Biomedical Research Foundation (Washington, D. C.)*, 5, 1972.

Table 4: Accuracies AC_{SVM} of different parameters γ and upper bounds C for H/¬H under 8-gram on CB513 set, where * marks the best accuracy for each C.

C	γ							
	0.001	0.01	0.2	0.3	0.5	1	5	50
0.01	50.00%	50.00%	51.39%	65.83%	80.83%	*83.89%	50.00%	50.00%
0.1	50.00%	50.00%	83.89%	83.89%	83.61%	*84.44%	83.61%	51.11%
1.5	51.39%	82.50%	*84.74%	81.94%	82.50%	80.28%	53.89%	53.89%
5	79.72%	79.72%	80.83%	*81.94%	81.39%	81.67%	53.89%	53.89%
50	78.61%	81.28%	*81.67%	81.11%	81.67%	82.78%	78.06%	53.89%

Table 5: Accuracies AC_{SVM} of different window sizes for each binary classifier on CB513.

Binary classifier	Window size (n-gram)							Best window size
	5	7	9	11	13	15	17	n_{op}
H/¬H	79.38%	83.91%	84.82%	85.99%	86.72%	86.13%	84.54%	13
E/¬E	79.26%	81.65%	83.89%	84.06%	83.38%	82.45%	81.61%	11
C/¬C	75.12%	78.18%	79.80%	81.06%	80.58%	80.47%	79.01%	11

[6] D. Frishman and P. Argos. Knowledge-Based Protein Secondary Structure Assignment. *Proteins*, 23:566–579, 1995.

[7] S. Hua and Z. Sun. A Novel Method of Protein Secondary Structure Prediction with High Segment Overlap Measure: Support Vector Machine Approach. *Bioinformatics*, 308:397–407, 2001.

[8] J.Garnier, D.J.Osguthorpe, and B.Robson. Analysis of the Accuracy and Implications of Simple Methods for Predicting the Secondary Structure of Globular Proteins. *J. Mol Biol*, 120:97–120, 1978.

[9] W. Kabsch and C. Sander. A Dictionary of Protein Secondary Structure. *Biopolymers*, 22:2577–2637, 1983.

[10] J. Moult and et al. Critical Assessment of Methods of Protein Structure Prediction (CASP): Round II. *Proteins. supplement 1.*, 29(S1):2–6, 1997.

[11] D. Nelson and M. Cox. *Lehninger Principles of Biochemistry Amino*. Worth Publishers, 2000.

[12] E. E. Osuna, R. Freund, and F. Girosi. Support Vector Machines: Training and Applications (A. I. Memo1602). MIT A.I.Lab, 1997.

[13] N. Qian and T. J. Sejnowski. Predicting the Secondary Structure of Globular Proteins Using Neural Network Models. *J. Mol. Biol*, 202:865–884, 1988.

[14] H. H. Rashidi and K. L. Buehler. *Bioinformatics Basics Applications in Biological Science and Medicine*. CRC Press, 2000.

[15] F. M. Richards and C. E. Kundrot. Identification of Structural Motifs from Protein Coordinate Data: Secondary Structure and First-Level Supersecondary Structure. *Proteins*, 3:71–84, 1988.

[16] B. Rost and C. Sander. Prediction of Protein Secondary Structure at Better Than 70% Accuracy. *J. Mol Biol*, 232:584–599, 1993.

[17] B. Rost, C. Sander, and R. Schneider. Redefining the Goals of Protein Secondary Structure Prediction. *J. Mol Biol*, 235:13–26, 1994.

[18] V. Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag, New York, 1995.

[19] M. J. Zvelebil, G. J. Barton, W. R. Taylor, and et al. Prediction of Protein Secondary Structure and Active Sites Using the Alignment of Homologous Sequences. *J. Mol Biol*, 195:957–961, 1987.

[20] D. Zwillinger, S. G. Krantz, and K. H. Rosen, editors. *Standard Mathematical Tables and Formulae (30th edition)*. CRC Press, 1996.

APPENDIX

A. CARDINAL VECTORS WITH PHYSIOCHEMICAL PROPERTIES

The exchange group and hydrophobicity are shown in Table 7 and 8, respectively.

Table 7: Exchange groups of amino acid residues.

Exchange groups	6D cardinal vectors	Amino acid residues
(A)	[1, 0, 0, 0, 0, 0]	C
(C)	[0, 1, 0, 0, 0, 0]	A,G,P,S,T
(D)	[0, 0, 1, 0, 0, 0]	D,E,N,Q
(E)	[0, 0, 0, 1, 0, 0]	H,K,R
(F)	[0, 0, 0, 0, 1, 0]	I,L,M,V
(G)	[0, 0, 0, 0, 0, 1]	F,W,Y

Table 8: Hydrophobicity groups of amino acid residues.

Hydrophobicity groups	2D cardinal vectors	Amino acid residues
hydrophobic(O)	[1, 0]	C,D,E,G,H,K,N,Q,R,S,T,Y
hydrophilic(I)	[0, 1]	A,F,I,L,M,P,V,W