Protein Secondary Structure Prediction With Classifier Fusion

by

İsa Kemal Pakatcı

Submitted to the Graduate School of Sabancı University
in partial fulfillment of the requirements for the degree of
Master of Science

Sabancı University

August, 2008

Protein Secondary Structure Prediction With Classifier Fusion

İsa Kemal Pakatcı

EE, Master's Thesis, 2008

Thesis Supervisor: Hakan Erdoğan

Keywords: Protein, Structure, Secondary Structure Prediction

Abstract

The number of known protein sequences is increasing very rapidly. However, experimentally determining protein structure is costly and slow, so the number of proteins with known sequence but unknown structure is increasing. Thus, computational methods for prediction of structure of a protein from its amino acid sequence are very useful. In this thesis, we focus on the problem of a special type of protein structure prediction called secondary structure prediction. The problem of structure prediction can be analyzed in categories. Some sequences can be enriched by forming multiple alignment profiles, whereas some are single sequences where one cannot form profiles. We look into different aspects of both cases in this thesis.

The first case we focus in this thesis is when multiple sequence alignment information exists. We introduce a novel feature extraction technique that extracts unigram, bigram and positional features from profiles using dimension reduction and feature selection techniques. We use both these novel features and regular raw features for classification. We experimented

with the following types of first level classifiers: Linear Discriminant Classifier (LDCs), Support Vector Machines (SVMs) and Hidden Markov Models (HMMs). A novel method that combines these classifiers is introduced.

Secondly, we focus on protein secondary structure prediction of single sequences. We explored different methods of training set reduction in order to increase the prediction accuracy of the IPSSP (Iterative Protein Secondary Structure Prediction) algorithm that was introduced before [34]. Results show that composition-based training set reduction is useful in prediction of secondary structures of orphan proteins.

Sınıflandırıcı Birleştirimi İle Protein İkincil Yapısı Kestirimi

İsa Kemal Pakatcı

EE, Master Tezi, 2008

Thesis Supervisor: Hakan Erdoğan

Anahtar Kelimeler: Protein, Yapı, İkincil Yapı Kestirimi

Özet

Bilinen protein dizileri sayısı çok hızlı artmaktadır, fakat proteinlerin yapısını deneysel metotlarla belirlemek maliyetli ve yavaş olduğu için yapısı bilinen proteinlerin sayısı ile dizisi bilinen proteinlerin sayısı arasındaki fark gittikçe artmaktadır. Bu yüzden amino asit zinciri bilinen bir proteinin yapısının hesaplamalı yollarla bulunması bu farkı kapatmak açısından önemlidir. Bu tezde ikincil yapı adı verilen protein yapısının kestirimi üzerine yoğunlaşılmıştır. İkincil yapı kestirimi kategoriler halinde incelenebilir. Bazı diziler çoklu dizi profilleri ile zenginleştirilebilirken bazı diziler için profil çıkartılamaz. Bu iki durum da bu çalışmada incelenmiştir.

Yoğunlaştığımız ilk durum çoklu dizi hizalama bilgisinin olmadığı durumdur. Boyut düşürme ve öznitelik seçimi yöntemleri kullanılarak tekli, çiftli ve pozisyon özniteliklerini profil bilgisinden çıkaran yeni bir öznitelik çıkarma yöntemi geliştirdik. Çıkarılan bu öznitelikler ile ham öznitelikleri sınıflandırma için kullandık. Kullandığımız ilk seviye sınıflandırıcılar saklı Markov modeli, destek vektör makinesi, doğrusal ayırtaç sınıflandırıcısıdır. Bu ilk seviye sınıflandırıcıları birleştiren yeni bir yöntem sunulmuştur.

İkinci olarak, tek dizi protein ikincil yapısı kestirimi problemine yoğunlaştık. Bu problem için daha önceden önerilmiş olan IPSSP algorithmasının performansını arttırmak için değişik eğitim kümesi indirgeme yöntemleri incelenmiştir. Deney sonuçları eğitim kümesi indirgemenin, yetim proteinlerin ikincil yapısının kestirimi için işe yaradığını göstermektedir.

Acknowledgements

I wish to express my utmost gratitude to my supervisor, Hakan Erdoğan for his guidance, encouragement and most of all his patience. He had put so much effort in supporting me throughout this study.

Many thanks to my thesis jury members, Canan Atılgan, Hikmet Budak, Uğur Sezerman and Hüsnü Yenigün for having kindly accepted to read and review this thesis.

I would like to thank Zafer Aydın from Georgia Institute of Technology for his contributions and ideas that was very helpful in development of this work.

I would like to thank my very best friend Ahmet Tuysuzoglu, for motivating me and making me believe to my work; I also thank my firends Umut and İsmail Fatih for their support.

I would like to thank TÜBITAK for the generous financial support and Istanbul Technical University National High Performance Computing Center for providing parallel computing resource that accelerated the simulations in our work.

Finally, I would like to thank my family for their support of all means.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Proteins are the building blocks of life and understanding their function is essential for human health. However determining a function of a protein is a hard, time consuming and costly process. It has long been known that protein function is closely related to its 3D structure, therefore understanding the structure of a protein is crucial in function prediction. There are experimental methods for protein structure determination such as X-ray crystallography and NMR spectroscopy both of which require significant amount of time and investment. Alternative methods are computational structure prediction methods which are very cheap and efficient. Although these methods are less accurate than experimental methods, protein sequence-structure gap is increasing after large-scale genome sequencing projects began and we need fast and accurate ways to predict structural information. Computational prediction of structure of proteins have been studied in the literature. Prediction of 3D structure of proteins is a hard problem and biologists have defined local 1-D structures such as secondary structure and solvent accesibility which are easier to predict. In this work, we develop new computational methods for secondary structure prediction of proteins which we hope will be competitive with existing approaches.

1.2 Problem Definition and Literature Review

Definition of protein secondary structure problem in simple terms is the following: Given an aminoacid sequence of a protein, assign each aminoacid to

one of three secondary structure states: α-helix, β-sheet, or loop. Because of the importance of this problem, many computational methods have been proposed and now we review some of them.

We can divide the history of development of prediction methods into three generations. First generation methods [12, 28, 17] use single amino acid statistics derived from small sequence databases. Basically these methods used the probability of each aminoacid to be in a particular secondary structure state. Second generation methods [25] extended this concept and took neighborhood information of aminoacids into account. Many pattern recognition algorithms are applied to chemical properties that is extracted from adjacent aminoacids. The accuracy of first and second generation methods was below 70%.

First algorithm that surpassed 70% boundary was PHD [30] algorithm which can be considered as the first method in third generation of secondary structure prediction algorithms. It used neural networks of multiple levels which was a new idea and many successor methods make use of this idea. The Q_3 accuracy of PHD method was 71.7% and segment overlap measure (SOV) of the method was 68%. In 1999, David Jones proposed PSIPRED algorithm [20] which introduced the idea of using position spesific scoring matices (PSSM) produced by the PSI-BLAST alignment tool. This method has a special strategy to avoid using unrelated proteins and polluting the profile generated. Similar to PHD, PSIPRED also uses neural networks which achieve a Q_3 score of 76.5 and SOV score 73.5%. This method is further developed and according to the assesment results in EVA [1], which evaluates protein secondary structure servers in real time, PSIPRED reaches Q_3

accuracy of 77.9% and SOV score of 75.3%. Another comparable algorithm proposed is the Jpred2 algorithm [15] which achieves 76.4% Q_3 accuracy and 74.2% SOV score. This algorithm uses 3 layers of neural networks similar to PHD method but it uses different types of features such as position specific scoring matrices, PSIBLAST frequency profile, HMM and multiple sequence alignment profiles. There are also support vector machine (SVM) based methods [19, 22] among which a notable one is SVMpsi algorithm which combines binary SVM classifier in directed acyclic graph form, claims finally achieving a Q_3 score of 78.5% and SOV score of 77.2%.

Best of state of the art protein secondary structure prediction methods is PORTER [27] which achieves Q_3 accuracy of 79.1% and SOV score of 75%. The idea of this method is to overcome the shortcoming of classic feed-forward neural networks by using bidirectional recurrent neural networks which can take the whole protein chain as input. Furthermore five two-layer BRNN models which have different architecture, size and initial weights are avereged in PORTER method.

1.3 Contributions

Contributions of this thesis can be listed as follows:

1. Three different classifier types, namely hidden Markov model (HMM), linear discriminant classifier (LDC), and support vector machines (SVM) have been implemented and their performances are compared on a standard benchmark dataset for the secondary structure prediction problem.

2. A new algorithm that combines outputs of linear discriminant classifiers, support vector machines and hidden Markov models is proposed.

3. A new feature extraction technique based on unigram, bigram and positional statistics is introduced and compared with standard features used in the literature.

4. Ratio of single sequence proteins to all proteins in five different organisms is calculated for different values of similarity thresholds.

5. Effect of using different similarity measures in training set reduction phase to prediction accuracy for the single sequence problem is analyzed.

1.4 Outline

In chapter 2, we give basic information about proteins and multiple sequence alignments which are heavily used in prediction of protein secondary structure. Overview of the problem and our work on determining single sequence protein percentages in some organisms is also presented in this chapter. In chapter 3, we give details of feature extraction methods used in our work. Details of proposed method is presented is chapter 4. In chapter 5, we present different training set reduction methods for improving the accuracy of single sequence prediction algorithm. Finally in chapter 6, conclusions are made and possible extensions of our work is discussed.

2 Secondary Structure Prediction: Background and Overview

In this chapter, some introductory information about proteins and their structure is given. We introduce sequence alignment methods, which are essential tools for protein secondary structure prediction. General overview of the problem is given and performance measures for assessing proposed algorithms are described.

2.1 Proteins

Proteins are large organic molecules that consist of a chain of amino acids which are joined by peptide bonds. Proteins are essential in organisms and they play a key role in almost every process within cells. For example almost all enzymes, which are molecules that catalyze biochemical reactions, are proteins. Because of their importance, proteins are most actively studied molecules after their discovery by Jöns Jakob Berzelius in 1838 [3].

Amino acid is a molecule that consist of a amino group and a carboxyl group. Hundreds of types of amino acids have been are found in nature but only 20 of them can be found in proteins [31]. There are also two other non-standard amino acids (Selenocysteine and Pyrrolysine) that are known to occur in proteins but since these are very rare only standard 20 types of amino acids will be considered. The term 'residue' can be used as an alternative to the term amino acid since residue means a unit element of a biological sequence.

Proteins fold into a stable structure in 3D which are uniquely determined

by the composition of its amino acids under nearly same environmental conditions such as pH, pressure, temperature. Structures of proteins have been investigated in 4 groups (Figure 1):

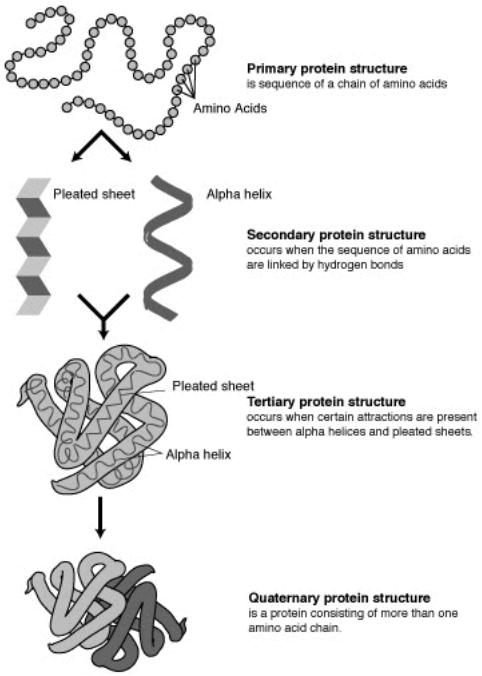

Figure 1: Types of protein structures
(From http://upload.wikimedia.org/wikipedia/commons/a/a6/Protein-structure.png)

1. Primary structure: Amino acid sequence.

2. Secondary structure: Repeating local patterns. Although some alternative definitions exist, there are mainly 3 types of secondary structures:

   - α-helix, spring-like structure which we denote as H,
   - β-sheet, generally twisted pleated sheet-like structure which we denote as E,

- loop, non regular regions between α-helix and β-sheet which we denote as L.

3. Tertiary structure: Overall shape in 3D, spatial relationship between atoms. The 'fold' can be used as an alternative to tertiary structure.

4. Quaternary structure: Structure of the protein complex. Some proteins consist of more than one protein subunits whose interaction form a protein complex.

Common methods for experimental structural determination of proteins are X-ray crystallography and NMR spectroscopy, both of which can produce information at atomic resolution.

2.2 Sequence Alignment

Sequence alignment refers to alignment of sequences by possibly introducing gaps, where the goal is to have highest similarity between aligned residues. The aim of this method is to evaluate the evolutionary origin of each residue in a protein since a residue can be changed over time. There may be insertions or deletions, so lengths of the sequences are not necessarily the same. Sequence alignment score is a measure to assess the similarity of aligned sequences. When there are two sequences that are aligned, this process is called pairwise alignment. Sample pairwise alignment is shown in Figure 2. In this figure, red residues indicate matching, blue residues are residues that are similar, dashes denote gaps where this means either there was a deletion in gapped sequence or there was an insertion in the other sequence.

Figure 2: Sample pairwise sequence alignment
(From http://www.ncbi.nlm.nih.gov/Web/Newsltr/Summer00/images/aminot.gif)

In order to calculate the score of a given alignment, a substitution matrix where each entry in the matrix indicates substitution score for each pair of aligned amino acids, should be used. For each pair of aligned residues, score is looked up from this matrix and scores for each residues are summed to calculate the final alignment score. All gaps may be given a fixed score but general strategy for scoring gaps is to penalize first gap in gapped region by a gap opening penalty and penalize remaining gaps by a gap extension penalty. There are two types of alignment: global and local. In global alignment all residues of the both sequences are aligned, but in local alignment highly similar subsequences are aligned. In protein secondary structure prediction problem local alignments are preferred because they capture more information about distantly related sequences.

Finding optimum local alignment for a given pair of proteins can be achieved by dynamic programming. Smith Waterman algorithm [32] calculates highest local alignment score given query and subject sequences. In general one wants to search a sequence database for significantly similar sequences to the given query sequence. It may seem that we may use raw alignment score for selecting significantly similar sequences, but since lengths of the sequences are not the same, this measure is highly variable with length and is inappropriate. A more appropriate criteria for evaluating significance

of sequence similarity is the e-value criteria which is defined as

$$E = K \cdot m \cdot n \cdot e^{-\lambda S},$$

where S is the alignment score, m and n are the lengths of query sequence and sequence in the alignment respectively. K and λ parameters control the weighting of length the of the sequences and the similarity score. E-value is the expected number of pairs of randomly chosen segments whose alignment score is at least S, therefore lower e-value means there is a significant similarity between sequences.

2.3 Multiple Sequence Alignment

Multiple sequence alignment is a generalization of pairwise alignment which is used to incorporate more than two sequences. Multiple alignment methods align all of the sequences in a set which are assumed to be evolutionary related. Since biological sequences behave similarly in a family, multiple alignment is more suitable for extracting evolutionary information. Generally, first stage of multiple alignment is that an e-value threshold is set and those alignments whose e-value are less than this threshold are searched in the database. Once the proteins above a certain threshold are extracted, a distance matrix of all $N(N-1)/2$ pairs including the query protein is constructed by pairwise dynamic programming alignments. Then multiple alignments are calculated using statistical properties of these clusters. More information about multiple sequence alignment can be found in [8] (AMPS) and [33] (CLUSTALW).

Sequence Frequency Profile

Sequence frequency profile is a $20 \times N$ matrix where N is the number of residues in the query protein. It is obtained from multiple sequence alignment by counting the number of occurrences of each type of residue in the alignment. These counts are divided to the number of non-gap symbols for each position in order to get frequency of each type of residue in each position. Frequency information of multiple sequence alignment is also used in the secondary structure prediction problem which is one of our methods in this work.

Position Specific Scoring Matrix (PSSM)

PSSM is also a $20 \times N$ matrix generated by PSI-BLAST program which iteratively searches for local alignments in a database. Multiple alignment is calculated through the search and position-specific scores for each position in the alignment are calculated. Highly conserved positions receive high position specific scores and weakly conserved positions receive scores near zero. Most important difference between PSSM and frequency profile is that PSSM is calculated by weighting alignments according to their alignment score whereas frequency profile does not distinguish between alignments. PSSMs are also heavily used in secondary structure prediction problem and more information about them can be found in [6].

Figure 3: Sample secondary structure prediction of a protein. Secondary structures are α-helix (H), β-sheet (E) and loop (L). Stars indicate correctly predicted structures.

2.4 Overview of the problem

To restate the problem we can say that our aim is to predict secondary structure sequence given the amino acid sequence of a protein. Sample secondary structure prediction is shown in Figure 3.

As mentioned in the introduction chapter, there is a huge gap between the number of protein sequences we know and the number of proteins whose structure are known. For example currently there are more than 6 million chains in the NR database which includes almost all known protein chains organized by organism name. On the other hand Protein Data Bank [4] which includes all publicly available solved structures, contains 52103 structures where 7279 of them were solved in 2007. This phenomenon is called the sequence-structure gap.

When a biologist obtains the sequence of a protein whose structure he/she tries to predict, there may be four different cases depending on the sequence:

1. Structure of the sequence is already experimentally determined and considered known. The structure is simply looked up from a database of structures such as PDB.

2. There is another protein whose structure is known and is similar (ho-

mologous) to the input protein, then secondary structure can be predicted with high accuracy since sequential similarity is highly related with structural similarity. This problem is known as homology modeling and accuracies can be as high as 85%-95% depending on the level of sequence similarity[7]. This case is considered to be trivial in the literature and machine learning algorithms are deemed unnecessary in this case. In this work we do not deal with this problem.

3. There exist sequences with significant similarity to the input protein but their structures are not known. Similar sequences can be used to generate a multiple-alignment sequence profile which contains evolutionary information. This information is very useful in prediction of secondary structure. In chapter 4, we explore methods aiming to solve this case of the problem.

4. There is no sequence that is significantly similar to the input protein. In this case, this protein is called an orphan protein or the protein sequence is called a single sequence. We refer to the problem as protein secondary structure prediction in single sequence condition. An alternative definition of single sequence condition is that there may be at most 1 sequence similar to the input sequence so that one cannot reliably form a sequence profile. In section 5 we explore different training set reduction methods for predicting structure in the single sequence condition.

In this work, we explore the probability of a person to face each case except for case 1 since we assume that this person is given a new protein with un-

known structure. In other words, we calculate the percentages of proteins that fall into one of the categories 2, 3 and 4 above. To do this, we used the NR database. We extracted proteins belonging to five organisms that are very different in organism complexities. These organisms are Homo sapiens (Human), Sulfolobus solfataricus, Mycoplasma genitalium, Methanococcus jannaschii, Bacillus subtilis. We aligned each protein of each selected organisms to all other proteins in the NR database using PSI-BLAST with one iteration. Three types of statistics are calculated from the results for each e-value:

1. No-hit percentage: Percentage of proteins that has no significantly similar protein in the NR database. This is an estimate of the probability of observing a new protein that falls into case 4.

2. At most one hit: Percentage of proteins that has at most one significantly similar protein in the NR database. This is an estimate of the probability of observing a new protein that falls into case 4 of alternative definition of single sequence condition.

3. No hit with known structure: Percentage of proteins that has no significantly similar protein whose structure is known (in PDB). This is the estimation of probability of one observes a protein that falls into case 3 or 4.

Table 1 shows calculated statistics for human proteins and for e-values between $10^{-5}$and 1. A typical e-value may be 10^{-3} and for this e-value table 1 shows that approximately 6% of human proteins are orphan, thus, we can say that, the probability of a new human protein sequence to be in case 4 is

E-value	No hits (%)	At most 1 hit (%)	No hit with known structure
10^{-5}	7.0	10.0	56.8
10^{-4}	6.6	9.6	56.1
10^{-3}	6.3	9.2	55.4
10^{-2}	5.8	8.7	54.5
10^{-1}	5.3	8.0	53.6
10^{0}	4.5	6.8	52.5

Table 1: Percentage of proteins in human which do not have significantly similar proteins in the NR database for a given e-value

0.06. For the same e-value, approximately 55% of human proteins fall into category 3 or 4. If we separate orphan proteins, we can say that 49% of the human proteins fall into category 3. Remaining percentage of human proteins (45%) fall into category 2. Figures showing calculated statistics for a broader range of e-values and tables showing calculated statistics for other selected organisms are provided in the Appendix.

2.5 Performance Measures

There are different performance measures to assess protein secondary structure prediction accuracy. The most commonly used one is Q_3 which is the overall percentage of correctly predicted residues. Formally:

$$Q_3 = \frac{\sum_{k\in\{H,E,L\}} \#of\ correctly\ predicted\ residues\ for\ class\ k}{\sum_{k\in\{H,E,L\}} \#of\ \ residues\ for\ class\ k}.$$

The per residue accuracy is a measure of accuracy for each state which is defined as

$$Q_k = \frac{\#of\ correctly\ predicted\ residues\ for\ class\ k}{\#of\ \ residues\ for\ class\ k} \quad k \in \{H, E, L\}\,.$$

Segment overlap measure (SOV) is introduced in order to evaluate methods by secondary structure segments rather than individual residues:

$$SOV = \frac{1}{N} \sum_{k \in \{H,E,L\}} \sum_{S(i)} \left[\frac{minOV(s_1, s_2) + \delta(s_1, s_2)}{maxOV(s_1, s_2)} \times length(s_1) \right],$$

where s_1 and s_2 are observed and predicted secondary structure segments for each state k, $S(i)$ is set of all pairs (s_1, s_2) of segments where $s_1$and s_2 have at least 1 residue in common, $length(s_1)$ is number of residues in s_1, $minOV(s_1, s_2)$ is number of residues in overlapping region of s_1 and s_2, $maxOV(s_1, s_2)$ is total extent where any of s_1 and s_2 has residue in state k, N is the total number of residues in the database. There are 2 definitions for $\delta(s_1, s_2)$ which are given in 1994 [29], and 1999 [35], but we will define a recent version:

$$\delta(s_1, s_2) = min \left\{ \begin{array}{c} maxOV(s_1, s_2) - minOV(s_1, s_2) \\ minOV(s_1, s_2) \\ int(0.5 \times length(s_1)) \\ int(0.5 \times length(s_2)) \end{array} \right\}.$$

SOV score may provide better scoring in cases where Q_3 scores are high but predicted and correct segment lengths are significantly different.

Another measure is correlation coefficient measure for each class which is introduced by Matthews [24] which is defined as

$$C_i = \frac{tp_i \cdot tn_i - fp_i \cdot fn_i}{\sqrt{(tp_i + fn_i)(tp_i + fp_i)(tn_i - fn_i)(tn_i - fp_i)}} \quad i \in \{H, E, L\},$$

where tp_i is the number of residues that are correctly identified as class i (true positive), tn_i is the number of residues that are correctly rejected (true negative), fp_i is the number of residues incorrectly predicted to be in class i (false positive), fn_i is the number of residues incorrectly rejected to be in class i (false negative).

3 Feature Extraction

In this chapter, two different types of raw features used in this work are introduced. We also explain the details of our feature extraction methodology applied to both of these raw features.

3.1 Features Used

Two types of raw features used in this work are frequencies in multiple sequence alignment and PSI-BLAST generated position specific scoring matrix which is mapped to 0-1 range as in PSIPRED method by the following transformation:

$$\frac{1}{1-e^{-x}},$$

where x is entry in the position specific scoring matrix. We will call these features raw frequency features (FREQ) and raw pssm features (PSSM) respectively. For each residue in the input protein whose secondary structure is to be determined, there are 20 features each of which correspond to one of 20 amino acid types. When we select a window of size w we get $21 \times w$ matrix of features for each residue in the input protein where the 21^{st} row indicates whether each position in the window is in or out of the protein. For positions that fall outside the protein, all other entries except the 21^{st} are set to zero. We will denote this matrix for a specific residue as T where $T_{i,j}$ denotes freq or pssm feature corresponding to amino acid type i and residue whose position is j before or after the residue in consideration. For example $T_{3,-1}$ denotes the 3^{rd} raw feature (PSSM or FREQ) of the residue just before the residue whose secondary structure is to be predicted.

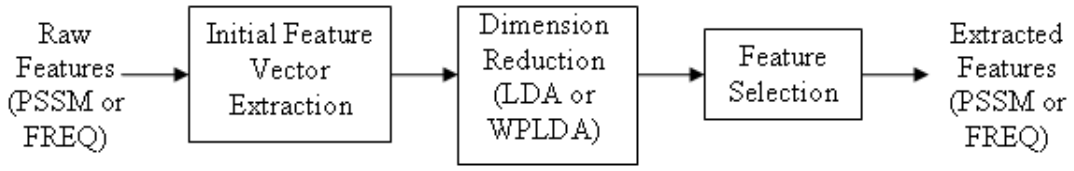

Figure 4: Processing stages for feature extraction

Given the raw feature matrix T for a residue, we process the raw data in 3 stages:

1. Initial feature vector extraction,
2. Dimension reduction,
3. Feature selection.

3.2 Initial feature vector extraction

We applied three different methods for initial feature vector extraction. These methods correspond to choosing a subvector of raw features and processing each of the subvectors separately.

1. Unigram vectors $\mathbf{u}_i = [T_{i,-l}, T_{i,-l+1}, T_{i,-l+2}, \ldots, T_{i,-1}, T_{i,0}, T_{i,1}, \ldots T_{i,l-1}, T_{i,l}]$ where $l = (w-1)/2$ is half window size. Since matrix T has 21 rows, there are 21 w-dimensional vectors of this type.
2. Position vectors $\mathbf{p}_i = [T_{1,i}, T_{2,i}, T_{3,i}, \ldots, T_{21,i}]$ are raw feature vectors corresponding to i position before/after residue in consideration. If window size is w then there are w vectors of this type.

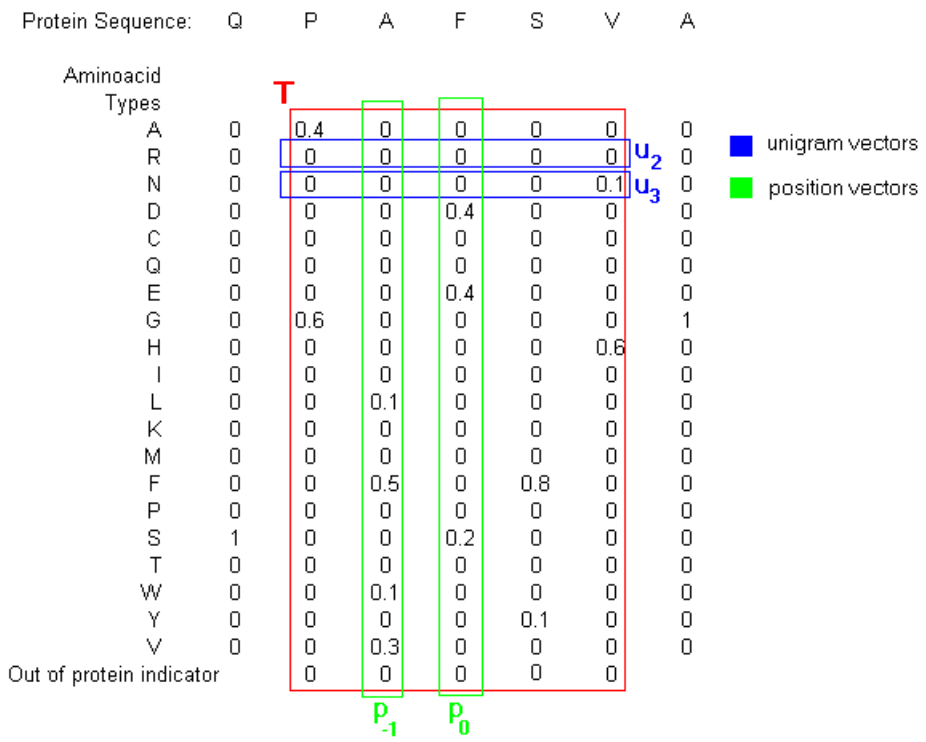

Figure 5: Raw frequency features for protein sequence QPAFSVA and initial vector types: unigram and position. T holds the raw features within a window for predicting secondary structure of residue F.

3. Bigram vectors $\mathbf{b}_{i,j} = [u_{i,1}u_{j,2}, u_{i,2}u_{j,3}, \ldots, u_{i,w-1}u_{j,w}]$ where $u_{k,m}$ denotes the m^{th} dimension of unigram vector u_k. Since bigram vectors are constructed for each pair of unigram vectors there are $21 \times 21 = 441$ vectors of this type.

Figure 5 shows raw frequency features, matrix T, unigram and position vectors.

3.3 Dimension Reduction

After initial feature vectors are extracted, we applied two dimension reduction techniques both of which reduce vectors of any dimension to $C-1$ dimensions where C is the number of classes. Since we have 3 classes, we reduced every vector described in the previous section to 2 dimensions. We now explain the dimension reduction methods used.

3.3.1 Linear Discriminant Analysis (LDA)

Linear discriminant analysis is a feature dimension reduction technique that aims to find direction(s) that maximizes the separation between classes. For example in a situation like Figure 6, it can be seen that projecting data onto vector w_2 does not help separating the classes but projecting data onto vector w_1 separates each class into different clusters.

Formal Definition

We are given a labeled set

$$\mathcal{D} = \{(\mathbf{x}_i, c_i) | \mathbf{x}_i \in \mathbb{R}^p, 1 \leq c_i \leq C\}_{i=1}^n,$$

where $\mathbf{x}_i$ is a data point in p-dimensional feature space and c_i is the corresponding class label. Between class covariance matrix B and within class covariance matrix W are defined as

$$B = \frac{1}{C-1} \sum_{j=1}^{C} n_j (\mu_\mathbf{j} - \mu)(\mu_\mathbf{j} - \mu)^T,$$

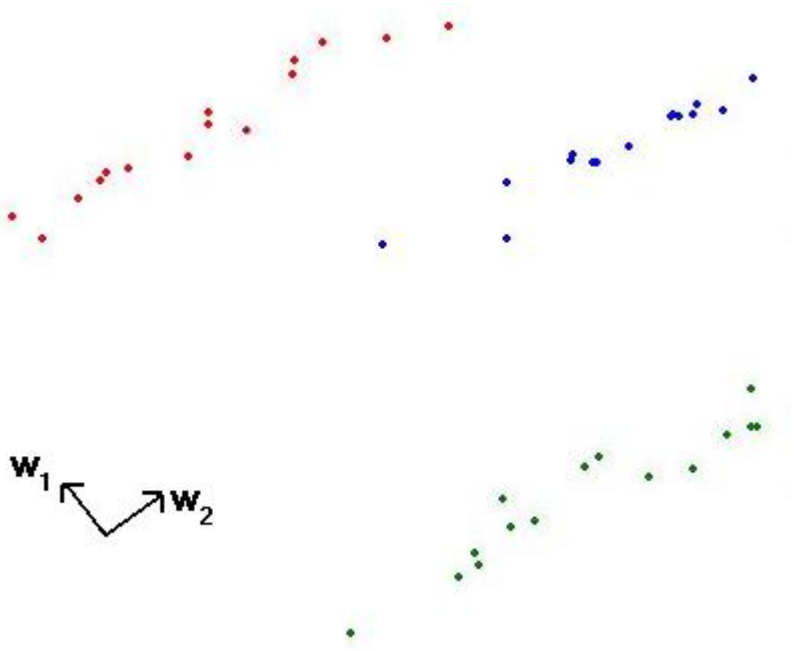

Figure 6: Data points and candidate vectors for projection

$$W = \frac{1}{n-C} \sum_{j=1}^{C} \sum_{i \in N_j} (x_i - \mu_j)(x_i - \mu_j)^T,$$

where n_j denotes number of points belonging to class j, μ denotes the overall mean, μ_j denotes mean of points belonging to class j, N_j is the set of indices of points that belong to class j. LDA finds a vector $\mathbf{w}$ that maximizes

$$\frac{\mathbf{w}^T B \mathbf{w}}{\mathbf{w}^T W \mathbf{w}},$$

since $\mathbf{w}^T B \mathbf{w}$ and $\mathbf{w}^T W \mathbf{w}$ is the between class and within class variance when data is projected onto $\mathbf{w}$ respectively and we want between class separation high and within class separation low. Solution of this problem is the generalized eigenproblem $(B - \lambda W)\mathbf{w} = 0$. If W is non-singular it can be trans-

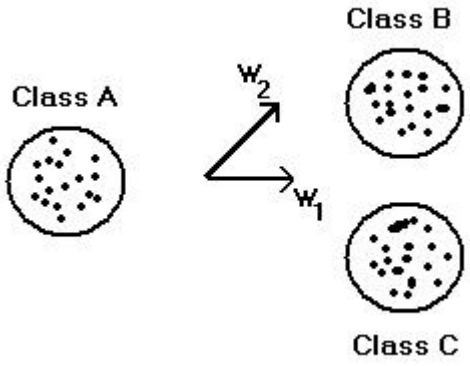

Figure 7: Sample dataset which results in different vectors for LDA and Weighted Pairwise LDA

formed into a standard eigen value problem $(W^{-1}B - \lambda I)\mathbf{w} = 0$. Therefore solution is eigenvectors of $W^{-1}B$. Rank of $W^{-1}B$ is $C-1$ hence there is at most $C-1$ eigenvectors corresponding to nonzero eigenvalues. Eigenvalues of $W^{-1}B$ are called separations, which give a measure of separation of classes over the space obtained by projecting data onto corresponding eigenvector.

3.3.2 Weighted Pairwise LDA

Weighted pairwise LDA (WPLDA) is introduced by Marco Loog [23] in order to overcome the shortcoming of standard LDA that overvalues the separation of a single class in multiclass case. For example, for the dataset shown in Figure 7, standard LDA finds vector $w_1$which does not separate classes B and C, because it favors the discrimination of class A from others. A better projection vector is w_2 since all classes are separated when data is projected onto w_2. WPLDA also produces at most $C-1$ eigenvectors.

3.4 Feature Selection

After reducing each initial feature vectors to 2 dimensions by applying one of the methods described above, we get what we call reduced feature vectors. Then we select some of these features in the following way:

For each type of features, separation values (eigenvalues of $W^{-1}B$) for each dimension of reduced features are sorted and features are selected whose separation values are highest and sum of them is equal to some fraction, p, of the sum of all separation values. For example, there are 21 unigram reduced feature vectors each of which is 2-dimensional, therefore we have 42 reduced features and separation values corresponding to each of them. We select some of these features such that some fraction, p, of total separation values is conserved. Formally, let $sp_1, sp_2,.., sp_n$ be separations in ascending order. Then, the feature corresponding to the separation sp_i is selected if $\sum_{k=1}^{i} sp_k \leq p \cdot \sum_{k=1}^{n} sp_k$. This feature selection mechanism is applied within each type of features: unigram, bi gram and position. As a result of this process we get features called extracted features in this work.

To sum up, there are 4 main features in our system: raw pssm and frequency features (raw PSSM and raw FREQ), extracted pssm and frequency features (extracted PSSM and extracted FREQ). There are 3 subtypes of extracted features: unigram, position and bigram extracted features. We used these features in our experiments which will be explained in the following chapter.

4 Proposed Methods, Experiments and Results

In this chapter, framework of our proposed method on protein secondary structure prediction problem is explained in detail and results of our experiments are given. First we give information about the dataset that we used. Afterwards, three types of general purpose classification algorithms used in this work are explained. Architecture of our system and its optimization procedure is given together with accuracy of the components of the system. We conclude this chapter by giving final results of our experiments and discussing on them.

4.1 Database and Assessing Accuracy

In our experiments we used CB513 dataset [13] which is a standard benchmarking dataset for comparison of protein secondary structure prediction methods. This dataset which contains 513 proteins, is constructed such that there are no sequence similarity between any pair of proteins in order to enable algorithm developers to emulate case 3 given in section 2.4. If there were any pair of proteins which are homologs of each other and one of them is in the training data and the other in the testing data, one can get artificially high accuracy in secondary structure prediction by using homology modeling. Thus, to emulate case 3 (when there is no homolog with known structure), one needs a database which has no pair that have sequence similarity higher than a certain value. There are 8 secondary structure states which is known as DSSP definitions in this dataset. 8 to 3 state reduction is applied as in jpred method in [14]. In order to obtain position specific scor-

ing matrices, the non-redundant (NR) database that contains more than 6 million chains is filtered by pfilt program as in [21] and PSI-BLAST search is done using this database with three iterations. Multiple alignment frequency profile is obtained by calculating frequencies of alignment data provided by distribution material [2] of JPred method which uses CLUSTALW program to obtain multiple sequence alignment. We used 5-fold cross validation in order to assess the accuracy of our method.

4.2 Classification Algorithms

4.2.1 Hidden Markov Models

Hidden Markov Models (HMMs) are statistical models that are suitable for sequential data. They are heavily used in speech recognition applications but also used in many bioinformatics problems as well. A Hidden Markov Model assumes that each observation is generated by one of finitely many states and probability of being at any state given the previous state is fixed (Markov assumption). States are not directly observable which is why it is called "hidden". In our case, states correspond to secondary structures or subsections of secondary structures and observations correspond to features which are extracted from the amino acid sequence.

Formal Definition

We will denote the state at time t as q_t. An HMM is characterized by the following set of parameters:

1. $S = \{S_1, S_2, ..., S_n\}$: Set of states.

2. T: Set of observations that can be generated by each state. This set can be discrete or continuous but in our case this set will be the set of real vectors.

3. $A = \{a_{ij}\}$: state transition probability distribution where

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i) \quad 1 \leq i, j \leq n,$$

which means the transition probability from state S_i to state S_j. Since this is a probability distribution for each source state we have the following constraint: $\sum_{i=1}^{n} a_{ij} = 1 \quad 1 \leq j \leq n$.

4. $B = \{b_i(o)\}$: Observation probability distribution for state i where

$$b_i(o) = P(o|S_i) = P_{\theta_i}(o)$$

for any $o \in T$. Here θ_i denotes the parameters of distribution for state i. For example if we assume that observations are sampled from multivariate Gaussian distribution, θ_i will be union of mean vector and covariance matrix parameters.

5. $\pi = \{\pi_i\}$: Initial state distribution where

$$\pi_i = P(q_1 = S_i).$$

Given state space, observation space and allowed transitions (topology), parameters of a model that must be estimated would be $\lambda = \{A, \Theta, \pi\}$ where $\theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$. In general one wants to find the best matching state

sequence given the observation sequence, O, and the model parameters λ which is known as the decoding problem. Another problem which is known as the training problem, is to find parameters of the most likely model that generates the given observation sequence.

4.2.2 Linear Discriminant Classifier

Linear Discriminant Classifier (LDC) is a classifier that assumes data is generated by a multivariate Gaussian distribution with equal covariance matrices among classes. It uses a discriminant function which is derived from this assumption and assigns a data label to the class that maximizes this function.

Formal Definition

We are given a labeled set

$$\mathcal{D} = \{(\mathbf{x}_i, c_i) | \mathbf{x}_i \in \mathbb{R}^p, 1 \leq c_i \leq C\}_{i=1}^n.$$

Common covariance matrix W is calculated same way as within covariance matrix is calculated in LDA given in section 3.3.1. Discriminant function for class j is:

$$g_j(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mu_j)^T W^{-1}(\mathbf{x} - \mu_j) + lnP(j),$$

where μ_j is mean of class j and $P(j)$ is prior probability of class j which is calculated as the fraction of training examples belonging to class j. LDC assigns a given data point $\mathbf{x}$ to class c_j such that

$$\arg\max_j g_j(x).$$

4.2.3 Support Vector Machines

Support vector machine (SVM) is a machine learning tool that has been popular in recent years. It simultaneously minimizes the training error and maximizes the margin which is defined as the minimum of distances of the training points to the decision boundary. For example in Figure 8, H_1 separates training data (black and white dots) perfectly but its margin is small whereas H_2 has large margin and also separates training data perfectly. On the other hand training error of H_3 is very high. SVM is initially proposed as a linear classifier for two classes but it is extended to non-linear case using the kernel trick. For multiclass case, one-versus-one or one-versus-rest binary classifiers can be constructed for each class or pair of classes respectively and output of these binary classifiers can be combined. For a detailed information on combination of binary SVMs see [9].

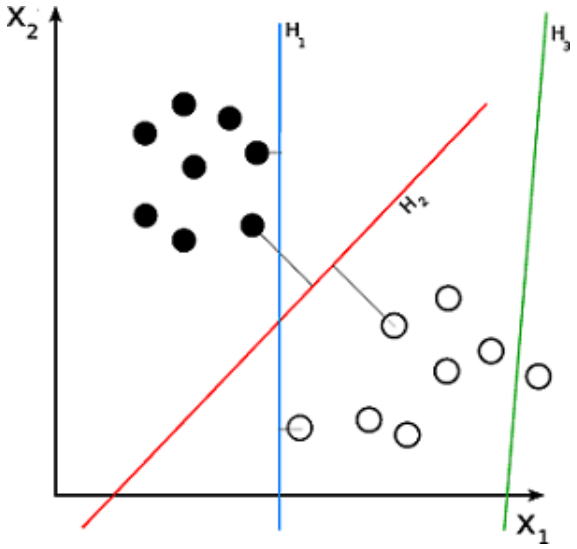

Figure 8: Hyperplanes that separate feature space
(From http://upload.wikimedia.org/wikipedia/commons/2/20/Svm_separating_hyperplanes.png)

Formal Definition

We are given training data

$$\mathcal{D} = \{(\mathbf{x}_i, c_i)|\mathbf{x}_i \in \mathbb{R}^p, c_i \in \{-1, 1\}\}_{i=1}^n,$$

where x_i is a point in p-dimensional space and c_i denotes the class label which is either 1 or -1. We want to find a hyperplane which separates the points belonging to class 1 from points belonging to class -1. Any hyperplane in p-dimensional space can be defined by a vector $\mathbf{w} \in \mathbb{R}^p$ and a scalar b . We want to choose w and b such that

$$\mathbf{w} \cdot \mathbf{x}_i - b \geq 1 \quad \forall(\mathbf{x}_i, 1) \in D,$$

$$\mathbf{w} \cdot \mathbf{x}_i - b \leq -1 \quad \forall(\mathbf{x}_i, -1) \in D,$$

which can be combined as

$$c_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 \quad \forall(\mathbf{x}_i, c_i) \in D.$$

Margin is $\frac{2}{\|\mathbf{w}\|}$ so in order to maximize margin $\|\mathbf{w}\|$ should be minimized. Hence, the problem reduces to finding $\mathbf{w}$ and b that minimizes $\|\mathbf{w}\|$ subject to constraints in given the equation above. If the data is not linearly separable then, we can insert non-zero slack variables ξ_i in constraints and minimize sum of these variables. In this case problem is formulated as

$$\min \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i \quad \text{such that} \quad c_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1 - \xi_i \quad 1 \leq i \leq n.$$

The parameter C controls the trade-off between large margin and small error. This problem can be solved by standard quadratic optimization techniques.

In order to get a non-linear classifier kernel functions can be used instead of dot product. A common kernel function is the Gaussian kernel which is defined as

$$K(x_i, x_j) = e^{-||x_i - x_j||^2/2\sigma^2}.$$

4.3 Proposed System Architecture

There are 2 layers of classifiers in our system. In the first layer there are 9 different classifiers which differ in type and features they use.

1. Linear discriminant classifier which uses features in position specific scoring matrix in a sliding window of specified size,

2. Support vector machine which uses same features in 1,

3. Linear discriminant classifier which uses reduced features obtained by applying one of the dimension reduction methods mentioned above to position specific scoring matrix,

4. Support vector machine which uses same features in 3,

5. Linear discriminant classifier which uses features in frequencies of multiple sequence alignment in a sliding window of specified size,

6. Support vector machine which uses same features in 5,

7. Linear discriminant classifier which uses reduced features applied to frequency features,

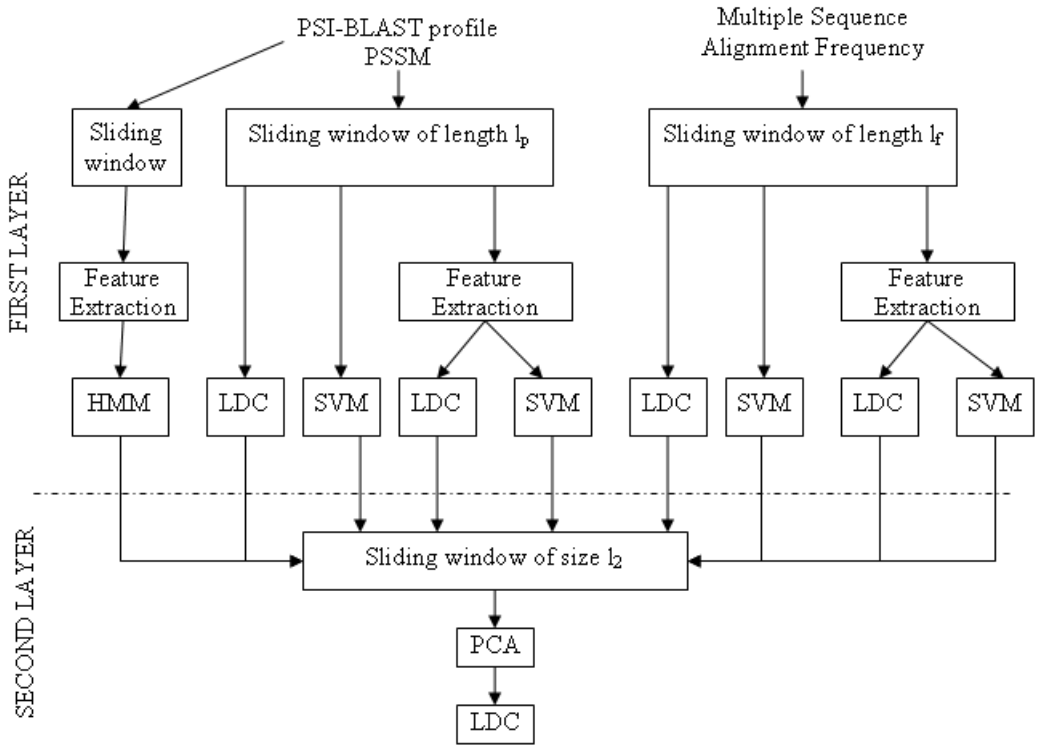

Figure 9: Architecture of our system

8. Support vector machine which uses same features in 7,

9. Hidden Markov model which performs best among the models described in section 4.2.1.

Output of each of these classifiers is a 3 dimensional vector, v, depending on the output encoding. We used 2 types of output encoding: In posterior encoding scheme, each element in this vector represents posterior probability of corresponding residue to be in each of the secondary structure classes. This scheme is not used in hidden Markov model. The other type of encoding is binary encoding, where each element in the output vector denotes whether or not corresponding residue is assigned to each secondary structure by the classifier. In other words one of the elements of v is 1 and the others are 0.

In order to combine outputs of the first level classifiers, we concatenate the output vectors of all first layer classifiers, therefore we have $9 \times 3 = 27$ dimensional vector for each residue. After that we extract features by applying a sliding window of size l_2. Since an additional dimension is added for in-protein indicator,the dimension of the resulting space is $l_2 \times 28$. We applied principal component analysis (PCA) which linearly maps data to lower dimension such that at most a fraction, F, of the total variance in the data is preserved. We choose this fraction to be 80% which is selected experimentally. For more details of PCA see [26]. After the dimension of the outputs of the first level classifiers are reduced, resulting features are fed to second level classifier which we choose to be linear discriminant classifier. The architecture of our system is illustrated in Figure 9.

4.4 Parameter Optimization and Results

4.4.1 First layer sliding window size parameter

To determine the window size in the first layer, we applied sliding window with a window size ranging from 9 to 19. We used the linear discriminant classifier on raw frequency and pssm features separately. The window size parameter which maximizes the performance of this classifier is selected.

The results of sliding window size experiments are shown in Table 2. According to these results, using pssm features performs %5-6 better than using frequency features which is a significant difference. The reason behind this difference may be that position specific scoring matrices can capture more evolutionary information and sequence similarity between distantly related proteins.

Another observation is that accuracy increases as window size increases except for frequency feature and sliding window size 19. In both of the features there is at least 1% increase while changing window size from 9 to 19. By increasing window size more information is fed to classifier which enables capturing relationships between distant residues, but more information does

Window size	PSSM features Q_3 (%)	FREQ features Q_3(%)
9	72.0	67.0
11	72.6	67.4
13	73.0	67.8
15	73.1	68.1
17	73.2	68.2
19	73.2	68.0

Table 2: Q_3 accuracies for different sliding window sizes and raw feature types where LDC is used as a classifier

not necessarily mean more accuracy in practice since high dimension needs more training data which is also known as curse of dimensionality. Therefore the drop of accuracy as we go from window size 17 to window size 19 by using frequency features can be explained by lack of training data.

Considering results in Table 2 we selected 17 as l_f parameter since that value maximizes prediction accuracy. For l_p parameter we also selected 17 because there is no difference between window size of 17 and 19 and the lower window size is preferred for simplicity.

4.4.2 Dimension reduction method parameter

Two dimension reduction methods, LDA and WPLDA, are applied to bigram features. Reason for selecting bigram features is that bigram features are nonlinear mapping of original space unlike unigram and position features and may capture nonlinear interactions of pairs of residues. For both of dimension reduction methods, features are selected as to preserve %80 and %90 of total separations. We fixed window size parameter to 17 for reasons discussed earlier. The result of experiments are shown in Table 3.

The results in Table 3 show that pssm features perform significantly better

	PSSM features		FREQ features	
	#of features	Q_3(%)	#of features	Q_3(%)
LDA, p=90%	320	72.2	254	67.7
LDA, p=80%	265	71.4	167	66.6
WPLDA, p=90%	284	70.5	251	64.9
WPLDA, p=80%	203	69.6	179	64.1

Table 3: Accuracies for different types of dimension reduction methods and fraction of separations conserved (p)

than frequency features which is consistent with observation in optimizing window size parameter. Conserving %90 of separations performs 1% better results than conserving %80 of separations which shows that using more data results in better prediction accuracy in this case. LDA dimension reduction method is superior to WPLDA method about %2 in almost all cases which may be because of that there is no secondary structure that is far away from others in this space. Therefore, WPLDA causes reduction in accuracy while considering pairwise distances between classes.

In the light of these results we selected LDA dimension reduction method with p=90%. After applying this method to pssm features, we observed that the three bigrams with highest separations are (Alanine-Leucine),(Valine-Valine) and (Leucine-Alanine). Sum of separations of these bigrams consists %6.8 of all separation values. For each bigram, position of bigram and absolute value of corresponding coefficient in LDA dimension reduction vector, is shown in Figure 10. Coefficient determines the importance of bigram at corresponding position. For bigrams (Alanine-Leucide) and (Leucine-Alanine) most important position is 10^{th} position which is two position right of center residue (since window size is 17, center position is 8). Bigram (Valine-Valine) is most important when it is found in one position to the right of the center residue. Amino acids, Valine, Alanine and Leucine are hydrophobic amino acids which means they are repelled from mass of water. These bigrams may be a clue in determining the factors leading to secondary structure formation.

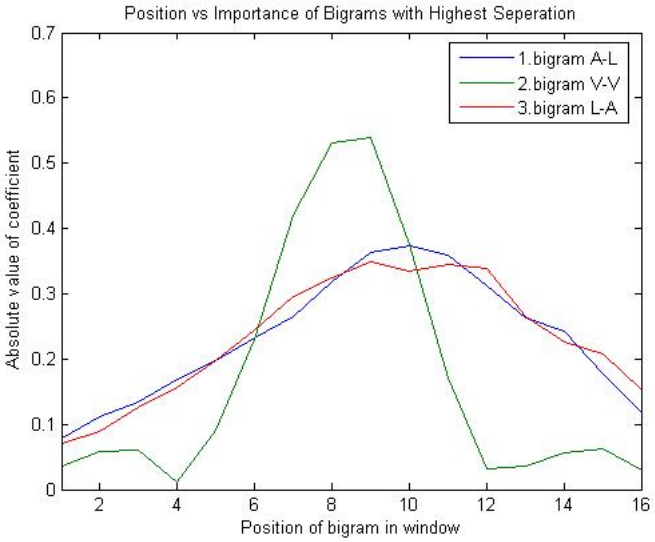

Figure 10: Position of bigram vs importance of bigram

4.4.3 Features used

As mentioned in chapter 3, there are three types of features extracted. Bigram features are used in selecting dimension reduction method. Now we will consider whether adding other 2 types of features, unigrams and position features, increases accuracy. Table 4 shows results of adding unigram and position features. As can be seen from the table adding unigram features increases accuracy about 0.3% and, position features increases accuracy about 0.1%. Low increase may be because of the fact that there are much more features in bigrams than unigrams or position features. Although increase rates are low, we select all three types of features for our system because these new information may be helpful in second level classification.

	PSSM features		FREQ features	
	#of features	$Q_3(\%)$	#of features	$Q_3(\%)$
Bigram	320	72.2	254	67.6
Bigram + unigram	335	72.5	271	68.0
Bigram + unigram + position	351	72.7	291	68.1

Table 4: Accuracies for different types of extracted features

4.4.4 Support Vector Machine Parameters

As mentioned in Section 4.2.3 there are 2 parameters in SVMs with Gaussian kernels; C parameter controlling the trade-off between misclassification tolerance and large margin, and a γ parameter which is the variance of the Gaussian kernel. These parameters of support vector machines are optimized by grid search procedure proposed in [10]. C parameters are searched within set $\{2^{-2}, 2^{-1}, 2^0, ..., 2^6\}$ and γ parameter are searched within set $\{2^{-6}, 2^{-5}, 2^{-4}, ..., 2^0\}$. Since our data is large, we selected 10000 of the residues as training data and 2500 of the residues as testing data. SVM with each pair of parameters, (C, γ), is trained on training set and parameters that gives maximum accuracy on testing set are selected.

	PSSM			FREQ		
	C	γ	$Q_3(\%)$	C	γ	$Q_3(\%)$
Extracted features	0.5	2^{-3}	74.6	4	1	70.2
Raw features	1	2^{-5}	75.8	2	2^{-4}	71.9

Table 5: Accuracy of SVM different type of features with optimized C and γ parameters

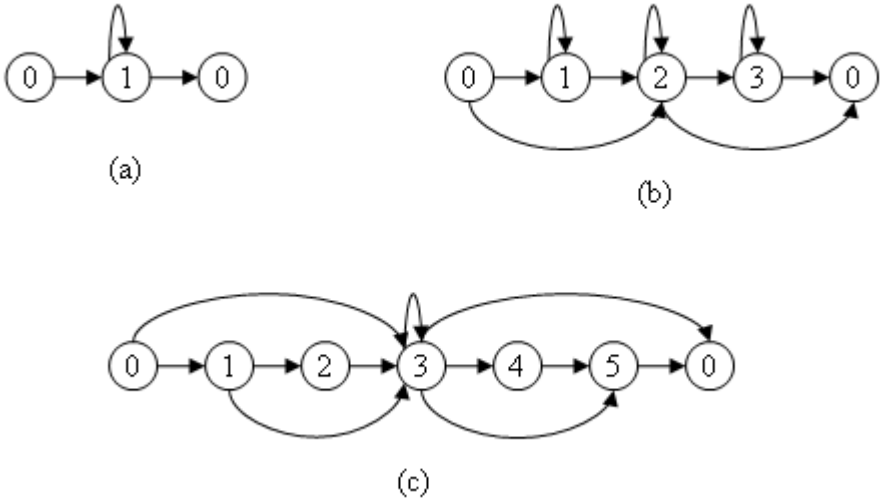

Figure 11: 1,3 and 5 emitting state HMM models used in this work

4.5 Used HMM Topologies and Results

4.5.1 Used Topologies

We used 1, 3 and 5 state topologies for each of secondary structure states, $\alpha-$helix, $\beta-$sheet and loop, which are shown in Figure 11.

We used multivariate Gaussian distribution as observation probability distribution with 2 options: full covariance matrix, diagonal covariance matrix. For full covariance matrix models we tried tying covariance matrices and mean vectors of secondary structures in 3 ways:

1. Tying covariance matrices within each secondary structure model (TSC),
2. Tying all covariance matrices (TAC),
3. Tying all covariance matrices and mean vectors within each secondary structure model (TAC+TM).

Models	1-state (%)	3-state (%)	5-state (%)
Diagonal Cov	63.6	65.0	64.3
Full Cov (TSC)	62.2	64.0	65.3
Full Cov (TAC)	66.6	65.1	68.1
Full Cov (TAC+TM)	66.6	66.0	66.5

Table 6: Q_3 Accuracies of used models for each covariance matrix formation

We used unigram pssm features which is reduced by LDA with parameter p=90% and sliding window size is 11 since this window size was optimized using same optimization procedure in sections 4.4.2 and 4.4.1in unigram feature space.

4.5.2 Results and Discussion

Results in table 6 shows that 5-state model performs better when full covariance matrix is used whereas 3-state model performs better when diagonal covariance is used. This shows that 1-state model is not sufficient to model secondary structures which means that beginning, internal and ending parts of segments of secondary structures does not behave same. This result is consistent with the finding of IPSSP algorithm discussed in chapter 5 which says that modeling segment boundaries differently than segment internal regions increases prediction accuracy. Another observation is that using full covariance matrix generally performs better than using diagonal covariance matrix. Reason for this may be that there are relationships between states of same secondary structure, i.e. a helix in the boundary of segment is correlated with a helix in the middle of a segment. Since this correlation is not used in diagonal covariance matrix case accuracies may drop. This result is not obvious before experimentation because full covariance matrix may have

performed worse, since there are much more parameters in full covariance matrix and estimating these parameters needs a lot of training data. For instance in 5-state model tying all covariances performs better than tying covariances within same state because tying all covariances reduces parameter size which reduces amount of training data needed.

4.6 Combining Classifiers

To combine classifiers, outputs of all classifiers are fed into another LDC classifier after applying principal component analysis (PCA) with variance preservation parameter 80%. Generally methods in the literature that use multiple kind of features in the first level of classification, do not use different type of features for second level classification but there are more than one second level classifiers. These second level classifiers are then combined by third level classifier. Theoretically second and third level classification can be combined to a single classifier which is the approach taken in this work. This approach enables second level classifier to use relationship between different type of features around the neighbor of the center residue. Sample second layer classification is shown in Figure 12.

Results of combination for different second layer window sizes are shown in table 7. Results show that using posterior encoding is roughly 1% better than using binary encoding for window sizes 9 and 11 whereas for window size 7 both encodings give comparable results. Reason for this result may be that posterior encoding includes more information than binary encoding. Simple majority voting combination of first level classifiers gives 70.1% accuracy, therefore using LDC as second level classifier is better than majority voting.

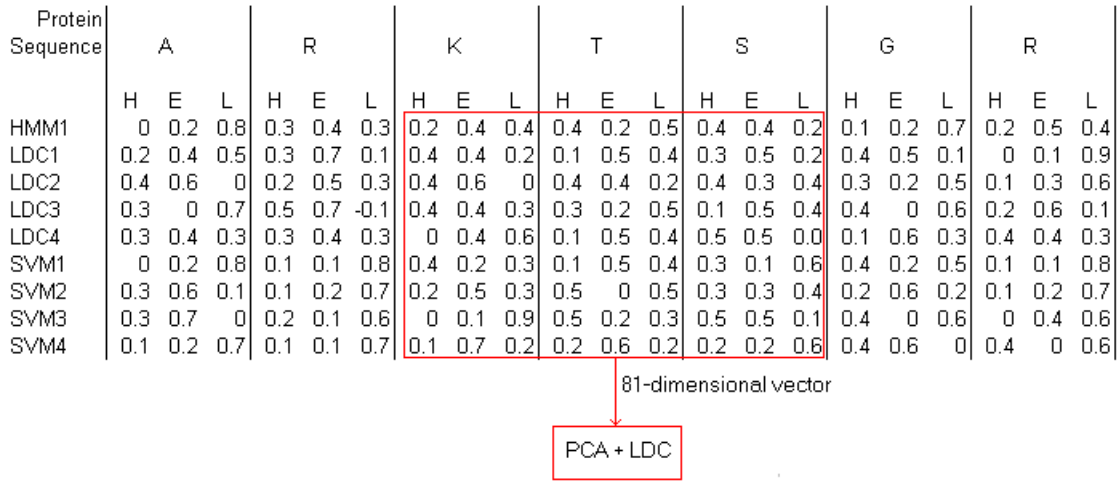

Figure 12: An example of second layer classification with sliding window (l_2) of size 3 with posterior encoding. Structure of central residue in the window (T) is to be determined. For each residue in the window, posterior probabilities for each secondary structure state is shown for each classifier.

This is because majority voting is simple rule that does not take into account training data which are outputs of first layer classifiers, whereas LDC can model the relationships between the outputs of the first layer classifiers.

Figure 13 shows the distribution of accuracies using posterior encoding and window size $l_2 = 11$ where bin size of accuracies is 5%. Numbers in the x-axis of the figure is the upperbound of the bin (i.e. bar corresponding to x value 70 is the frequency of accuracies between 65% and 70%). From the

Encoding	$l_2 = 7$ $Q_3(\%)$	$l_2 = 9$ $Q_3(\%)$	$l_2 = 11$ $Q_3(\%)$
Binary	72.6	71.6	71.5
Posteriror	72.4	72.6	72.7

Table 7: Results of the final secondary structure prediction for different second layer window sizes (l_2) and different encoding schemes used in combining classifiers

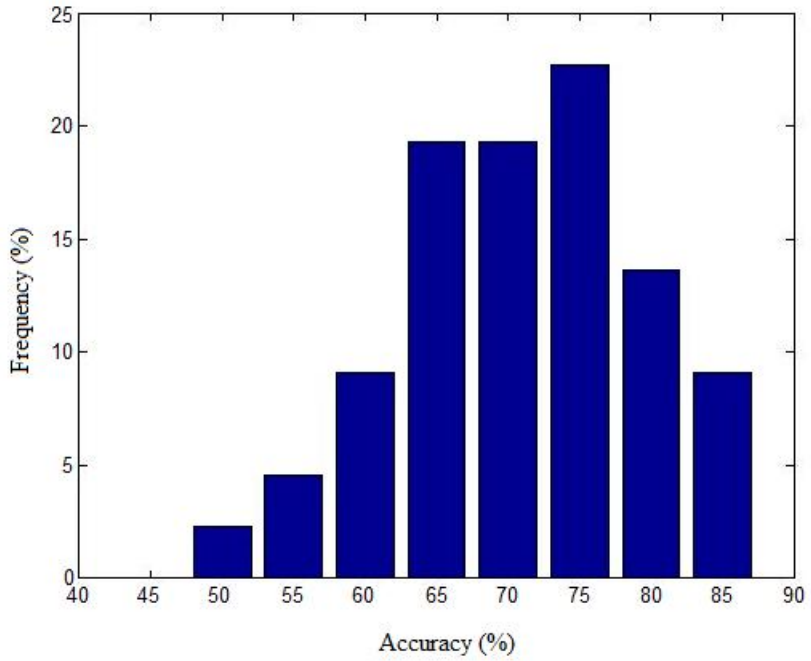

Figure 13: Accuracy distribution of combined classifier using posterior encoding and window size 11

figure it can be seen that distrubution makes a peak at range 70%-75%. This means that given a protein, most likely range that accuracy of the secondary structure prediction of this protein falls into is 70%-75%. Minimum and maximum accuracies are found to be 47.9% and 87.2% respectively.

5 Training set reduction for single sequence prediction

In earlier chapters, we have focused on predicting the secondary structure of sequences with more than 2 homolog proteins. As we have shown in section 2.4 6% percent of proteins in human do not have any homologs with e-value 0.001 which means single sequence condition. In this chapter we are going to consider improving IPSSP algorithm [34] which is designed to predict protein secondary structure in single sequence condition. Different methodologies that are applied in training reduction phase of this algorithm is explained and results are given.

5.1 Iterative Protein Secondary Structure Parse Algorithm

Amino acid and DNA sequences have been successfully analyzed using hidden Markov models (HMM) where the character strings generated in "left-to-right" direction. In a hidden semi-Markov model (HSMM), a transition from a hidden state into itself cannot occur, and a hidden state can emit a whole string of symbols rather than a single symbol. The hidden states of the model used in protein secondary structure prediction are the structural states {H, E, L} designating α-helix, β-strand and loop segments, respectively. Transitions between the states are characterized by a probability distribution. At each hidden state, an amino acid segment with uniform structure is generated according to a given length distribution, and the segment likelihood distribution. The IPSSP algorithm utilizes three HSMMs and an iterative training

procedure to refine the model parameters. The steps of the algorithm can be summarized as follows:

IPSSP Algorithm

1. For each HSMM, compute the posterior probability distribution that defines the probability of an amino acid to be in a particular secondary structure state. This is achieved by using the posterior decoding algorithm (also known as the forward-backward algorithm).

2. For each HSMM, compute a secondary structure prediction by selecting the secondary structure states that maximize the posterior probability distribution.

3. For each HSMM, reduce the original training set using a distance measure that compares the training set proteins to the predictions computed in step 2. Then, train each HSMM using the reduced dataset and compute secondary structure predictions as described in steps 1 and 2.

4. Repeat step 3 until convergence. At each iteration, start from the original dataset and perform reduction.

5. Take the average of the three posterior probability distributions and compute the final prediction as in step 2. It has been observed that performing the dataset reduction step only once (i.e., one iteration) generated satisfactory results [34].

5.2 Training Set Reduction Methods

In this section, we describe three dataset reduction methods that are used to refine the parameters of an HSMM: composition based reduction, alignment based reduction and reduction using Chou-Fasman parameters. In each method, the dataset reduction is based on a similarity (or a distance) measure. We considered two types of decision boundaries to classify proteins as similar or dissimilar. The first approach selects the first 80% of the proteins in the original dataset that are similar to the input protein. The second approach applies a threshold and selects proteins accordingly.

A. Composition Based Reduction

In this method, the distance between the predicted secondary structure and the secondary structure segmentation of a training set protein is computed as follows:

$$D = max(|H_p - H_t|, |E_p - E_t|, |L_p - L_t|),$$

where H_p, E_p and $H_p$denote the composition of α-helices, β-strands and loops in the predicted secondary structure, respectively. Similarly H_t, E_t and H_t represent the composition of α-helices, β-strands and loops in the training set protein. Here, the composition is defined as the ratio of the number of secondary structure symbols in a given category to the length of the protein. For instance, H_p is equal to the number of α-helix predictions divided by the total number of amino acids in the input protein. After sorting the proteins in the training set, we considered two possible approaches to construct the

reduced set: (1) selection of the first 80% of the proteins with the lowest D values; (2) selection of the proteins that satisfy $D < 0.35$.

B. Alignment Based Reduction

In this method, first, pairwise alignments of the given protein to training set proteins are computed. Then proteins with low alignment scores are excluded from the training set. As in the composition based method, two approaches are considered to obtain the reduced dataset: (1) selection of the first 80% of the proteins with the highest alignment scores; (2) selection of the proteins with alignment scores above a threshold. Here, the threshold is computed by finding the alignment score that corresponds to the threshold used in the composition based reduction method. In the following sections, we will give more details on pairwise alignment settings.

1) Alignment Scenarios: We considered the following cases:

- Alignment of secondary structures (SS),
- Alignment of amino acid sequences (AA),
- Joint alignment of amino acid sequences and secondary structures (AA+SS).

In the first case, the aligned symbols are the secondary structure states, which take one of the three values: H, E, or L. In the second case, the symbols are the amino acids and finally, in the third case, the aligned symbols are the pairs of amino acid and secondary structure type.

2) Score Function: The score of an alignment is computed by summing the scores of the aligned symbols (matches and mismatches) as well as the gapped regions. This is formulated as follows:

M_{ss}	H	E	L
H	2	-15	-4
E	-15	4	-4
L	-4	-4	2

Table 8: Secondary Structure Similarity Matrix

$$S = \sum_{k=1}^{r} (\alpha M_{aa}(a_k, b_k) + \beta M_{ss}(c_k, d_k)) + G,$$

where S is the alignment score, r is the total number of match/mismatch pairs, G is the total score of the gapped regions, a_k, b_k represent the k^{th} amino acid pair of the aligned proteins (the input and the training set protein, respectively), c_k, d_k denote the k^{th} secondary structure pair of the aligned proteins, $M_{aa}(.)$ is the amino acid similarity matrix, $M_{ss}(.)$ is the secondary structure similarity matrix, and finally, the parameters α, and β determine the weighted importance of the amino acid and secondary structure similarity scores, respectively. To compute possible alignment variations described in the previous section, α and β take the following values: (1)$\alpha = 0$; $\beta = 1$ to align secondary structures; (2) $\alpha = 1$; $\beta = 0$ to align amino acid sequences; (3) $\alpha = 1$; $\beta = 1$to align amino acid and secondary structures in a joint manner.

3) Similarity Matrices: We used the BLOSUM30 table [18] as the amino acid similarity matrix and the Secondary Structure Similarity Matrix (SSSM) [5] shown in Table 8.

4) Gap Scoring: When a symbol in one sequence does not have any counterpart (or match) in the other sequence, then that symbol is aligned to a gap symbol '-'. Allowing gap regions in an alignment enables us to

better represent the similarity between the aligned sequences in a biologically meaningful manner. In the state-of-the-art gap scoring, opening a gap is penalized more than extending it. For example, in the "affine gap scoring", which is one of the most widely used gap scoring techniques, starting a gap is scored by the parameter go, and extending a gap region is scored by ge. In that case, the total gap score in (2) is computed as

$$G = N_0 g_0 + N_e g_e,$$

where N_0 is the total number of gap openings, and N_e is the total number of gap extensions. In this work, we set the parameters g_0, and g_e to -12, and -2, respectively.

5) Optimum Alignment: Given a scoring function, the computation of the optimum (best scoring) alignment can be found using a dynamic programming approach. In this work, we used the Smith-Waterman algorithm to compute the local alignment between a pair of proteins. Further details on the alignment algorithms and dynamic programming can be found in Durbin et al [16].

6) Score Normalization: After computing the raw score of an alignment, it is useful to normalize it to a statistically meaningful range. In this work, we normalized the alignment score by the average length of the aligned proteins. In that case, the normalized score is computed as $2\frac{rawscore}{l_1+l_2}$, where l_1, and l_2 are the lengths of the aligned proteins.

C. Reduction using Chou-Fasman parameters

In this method, the training set reduction is based on the Chou-Fasman distance measure, which is defined as

$$D_{cf} = \sum_{k \in \{H,E,L\}} \left[\frac{1}{l_p} \sum_{j=1}^{l_p} f_k(q(j)) - \frac{1}{l_t} \sum_{j=1}^{l_t} f_k(h(j)) \right].$$

Here, l_p is the length of the input protein, l_t is the length of the training set protein, $q(j)$ is the j^{th} amino acid of the input protein, $h(j)$ is the j^{th} amino acid of the training set protein, and $f_k(z)$ is the Chou-Fasman coefficient that reflects the propensity of the amino acid of type z to be in the secondary structure state k. These coefficients can be computed as described in [11]. In this formulation, the secondary structure information of the proteins is not used and each amino acid is allowed to take three possible secondary structure states. In a slightly modified version of this method, we defined the Chou-Fasman distance using the secondary structure information as follows:

$$D_{cf,2} = \frac{1}{l_p} \sum_{j=1}^{l_p} f_{k(q(j))}(q(j)) - \frac{1}{l_t} \sum_{j=1}^{l_t} f_{k(h(j))}(h(j)),$$

where $k(q(j))$ is the predicted secondary structure state for the j^{th} amino acid of the input protein, and $k(h(j))$ is the secondary structure state for the j^{th} amino acid of the training set protein. In Chou-Fasman based reduction, we computed the reduced dataset by selecting the first 80% of the proteins with the lowest Chou-Fasman distances.

5.3 Results and Discussion

In our simulations, we used the EVA set of "sequence unique" proteins [1] derived from the PDB database [4]. We removed sequences shorter than 30 amino acids and arrived to a set of 2720 proteins. To reduce eight secondary structure states used in the DSSP notation to three, we used the following conversion rule: H, G to H; E, B to E; I, S, T, ' ' to L. We used the PDB SELECT dataset to compute the Chou-Fasman coefficients as in [11]. Here, the coefficients reflect the propensity of an amino acid to be either in H, E, or L state, which are defined using the above conversion rule. We evaluated the performances of the methods by a leave-one-out cross validation experiment (jacknife procedure). At each step, a protein is chosen as the test example and is taken out from the dataset. The remaining proteins form the training set and are used to estimate the parameters of the hidden semi-Markov model (i.e., transition, length and emission distributions). Since the true secondary structures were available, we used the maximum-likelihood estimation procedure, in which the observed frequencies for the desired quantities are divided by a proper normalization factor to compute the probability values. After estimating the model parameters, we predicted the secondary structure sequence of the test protein and repeated the leave-one-out procedure until all the proteins in the test set are evaluated. To save computation time, we restricted our test data to the first 600 proteins in the dataset, which gave a good approximation to the true result. Then, we computed the performance measures by taking the true secondary structures of the proteins as reference. To evaluate the performance, we chose the three state- per-residue accuracy (Q_3) as the overall sensitivity measure, which is computed as the total num-

Method	$Q_3(\%)$
Composition based	67.01
Alignment Based (SS)	67.00
Alignment Based (AA+SS)	66.92
Alignment Based (AA)	66.69
Chou-Fasman Based (D_{cf})	66.65
No Re-training	66.59
Chou-Fasman Based ($D_{cf;2}$)	66.50

Table 9: Sensitivity Measures of the Training Set Reduction Methods. The top 80% of the proteins are classified as similar to the input protein.

ber of correctly predicted amino acids in all dataset proteins divided by the total number of amino acids in the dataset.

From the results shown in Tables 9 and 10, the composition based reduction method performs better than the other reduction methods. This is mainly because of the fact that composition based reduction does not impose strong constraints, which serves to compensate for the errors made in the initial secondary structure prediction. In addition, threshold based reduction is slightly better than the reduction that selects the first 80% of the most similar proteins. Among the methods being compared, the composition based reduction method with thresholding gave the most accurate result, where the secondary structure prediction accuracy is improved by 0.6% compared to the condition with no retraining. Another advantage of the composition based method is its low computational complexity.

Comparing the alignment based reduction methods, the best result is obtained by the method that aligns secondary structures. Joint alignments of amino acid sequences and secondary structures did not perform better than secondary structure alignments. This is not surprising because in single

Method	$Q_3(\%)$
Composition based	67.17
Alignment Based (SS)	67.12
Alignment Based (AA+SS)	67.06

Table 10: Sensitivity Measures of the Training Set Reduction Methods. The dataset proteins are classified as similar to the input protein by applying a threshold.

sequence condition the input protein is not statistically similar to dataset proteins at the amino acid level. Therefore, the discriminative power of the amino acid similarity matrix is weaker than the secondary structure similarity matrix.

6 Conclusion and Future Work

In this study, we applied hidden Markov model, support vector machine, linear discriminant classifiers that uses features extracted from position spesific scoring matrices and multiple sequence alignment profiles by a novel method proposed. Although using these features did not result in better predictions than using standart features, results are comparable. Considering the fact that extracting different features by dimension reduction is a new idea applied to this problem, this approach is open to development. An example development would be better mapping of PSSM matrix to 0-1 range to reflect substitution probabilites of amino acids since we observed that pssms contain very important information that can be used in determination of secondary structure.

Classifier combination method which uses outputs of several classifiers as well as prediction of adjacent residues, performed worse than SVM using raw pssm matrix. The reason for this result may be that second level classifier is trained on intrinsically same data as first level classifiers and since classifiers generally perform very well in training data, there may be less things to learn for second level classifier.

For prediction of single sequence proteins, we showed that the training set reduction followed by the re-estimation of the model parameters improves the secondary structure prediction accuracy. Among the methods being compared, the composition based reduction technique with thresholding generates the most accurate results. This is mainly because of the fact that composition based reduction does not impose strong constraints, which serves to compensate for the errors made in the initial secondary structure prediction.

6.1 Future Work

Prediction accuracy can be improved by training different classifiers for four different structural classes, all-α, all-β,α/β and $\alpha+\beta$. One can also define a reliability measure so that the prediction of the classifier which gives maximum reliability is chosen for a given test sequence. This approach is expected to result in better accuracy since aminoacid composition of different structural classes are very different; therefore different classifiers which are focused on each class can perform better. Furthermore, predictions whose reliability is less than a certain threshold can be handled by different classifiers.

In order to improve classifier combination, one can train first level classifiers on random samples (bootstrapping) from training data and then train second level classifiers on the whole training data. Another solution would be partitioning data into three parts: one part for training the first layer, one part for training the second layer using the output of the first layer, and one part for testing the overall combined classifier. These approaches would enable second layer combiner classifier to learn the behavior of first layer classifiers on unseen data, which in turn should improve overall performance.

For prediction of single sequence proteins, the threshold parameter used to construct the reduced dataset can be optimized. In addition, the methods analyzed can be applied to the second class of prediction algorithms, which utilize evolutionary information in the form of alignment profiles or multiple alignments. In that case, we expect the alignment based method to perform significantly better than the other reduction methods, because the accuracy of the initial secondary structure prediction will be comparably higher than that obtained in the single-sequence condition.

References

[1] Eva: Evaluation of automatic structure prediction servers. http://cubic.bioc.columbia.edu/eva/.

[2] Jpred - training data. http://www.compbio.dundee.ac.uk/jpred.

[3] Protein - wikipedia. http://en.wikipedia.org/wiki/Protein.

[4] Rcsb protein data bank. http://www.rcsb.org/pdb/home/home.do.

[5] L. R. Murphy A. Fadel A. Wallqvist, Y. Fukunushi and R. M. Levy. Iterative sequence/secondary structure search for protein homologs: comparison with amino acid sequence alignments and application to fold recognition in genome databases. *Bioinformatics*, 16(11):988–1002, 2000.

[6] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–3402, September 1997.

[7] D. Baker and A. Sali. Protein structure prediction and structural genomics. *Science*, 294(5540):93–96, October 2001.

[8] G. J. Barton. Protein multiple sequence alignment and flexible pattern matching. *Methods in enzymology*, 183:403–428, 1990.

[9] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *COLT '92: Pro-*

ceedings of the fifth annual workshop on Computational learning theory, pages 144–152, New York, NY, USA, 1992. ACM Press.

[10] Chih-Chung Chang Chih-Wei Hsu and Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.

[11] P. Chou and G. Fasman. Empirical predictions of protein conformation. *Annu. Rev. Biochem*, 47:251–276, 1978.

[12] P. Y. Chou and G. D. Fasman. Prediction of protein conformation. *Biochemistry*, 13(2):222–245, January 1974.

[13] J. A. Cuff and G. J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins*, 34(4):508–519, March 1999.

[14] J. A. Cuff, M. E. Clamp, A. S. Siddiqui, M. Finlay, and G. J. Barton. Jpred: a consensus secondary structure prediction server. *Bioinformatics*, 14(10):892–893, 1998.

[15] James A. Cuff and Geoffrey J. Barton. Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins: Structure, Function, and Genetics*, 40(3):502–511, 2000.

[16] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, July 1999.

[17] 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(1):97–120, March 1978.

[18] S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *P.N.A.S. USA*, 89:10915–10919, 1992.

[19] Sujun Hua and Zhirong Sun. A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach. *J. Mol. Biol*, 308:397–407, 2001.

[20] D. T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol*, 292(2):195–202, September 1999.

[21] S. Jones, M. Stewart, A. Michie, M. B. Swindells, C. Orengo, and J. M. Thornton. Domain assignment for protein structures using a consensus approach: Characterization and analysis. *Protein Sci*, 7(2):233–242, February 1998.

[22] H. Kim and H. Park. Protein secondary structure prediction based on an improved support vector machines approach. *Protein engineering*, 16(8):553–560, August 2003.

[23] Marco Loog, Robert P. W. Duin, and Haeb R. Umbach. Multiclass linear dimension reduction by weighted pairwise fisher criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(7):762–766, 2001.

[24] B. W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochim Biophys Acta*, 405(2):442–451, October 1975.

[25] K. Nishikawa and T. Ooi. Correlation of the amino acid composition of a protein to its structural and biological characters. *Journal of biochemistry*, 91(5):1821–1824, May 1982.

[26] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 6:559–572, 1901.

[27] Gianluca Pollastri and Aoife Mclysaght. Porter: a new, accurate server for protein secondary structure prediction. *Bioinformatics*, 21(8):1719–1720, April 2005.

[28] B. Robson and E. Suzuki. Conformational properties of amino acid residues in globular proteins. *Journal of molecular biology*, 107(3):327–356, November 1976.

[29] B. Rost, C. Sander, and R. Schneider. Redefining the goals of protein secondary structure prediction. *J Mol Biol*, 235(1):13–26, January 1994.

[30] Burkhard Rost and Chris Sander. Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology*, 232(2):584–599, July 1993.

[31] W. Sakami and H. Harrington. Amino acid metabolism. *Annual Review of Biochemistry*, 32(1):355–398, 1963.

[32] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[33] J. D. Thompson, D. G. Higgins, and T. J. Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–4680, November 1994.

[34] Y. Altunbasak Z. Aydin and M. Borodovsky. Protein secondary structure prediction for a single sequence using hidden semi-markov models. *BMC Bioinformatics*, 7(178), 2006.

[35] Adam Zemla, Ceslovas Venclovas, Krzysztof Fidelis, and Burkhard Rost. A modified definition of sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins: Structure, Function, and Genetics*, 34(2):220–223, 1999.

A Appendix

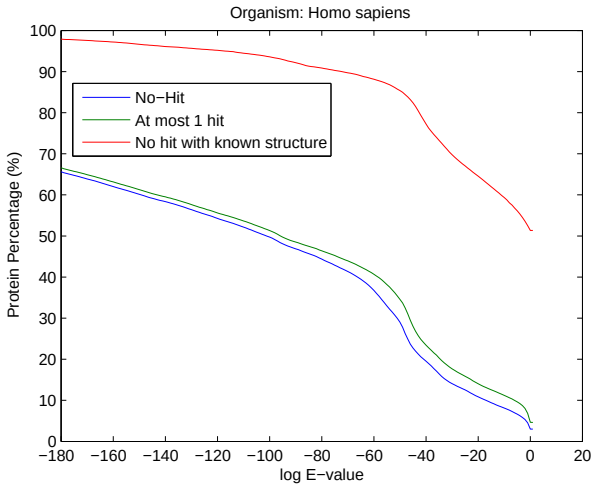

Figure 14: Percentage of proteins in human which does not have significantly similar proteins in NR database for a given e-value

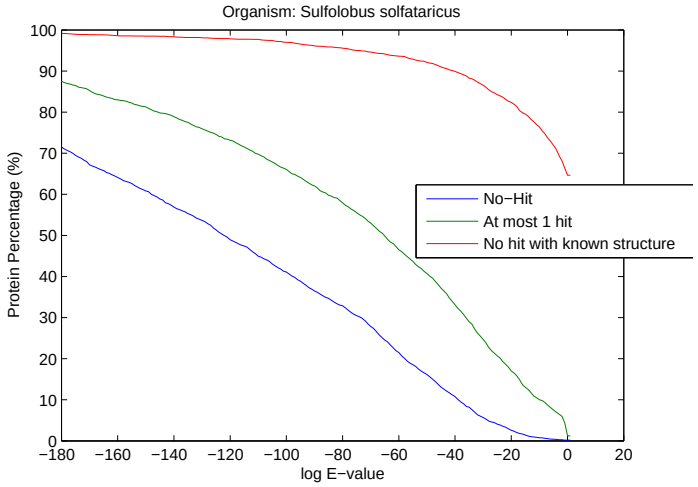

Figure 15: Percentage of proteins in Sulfolobus solfataricus which does not have significantly similar proteins in NR database for a given e-value

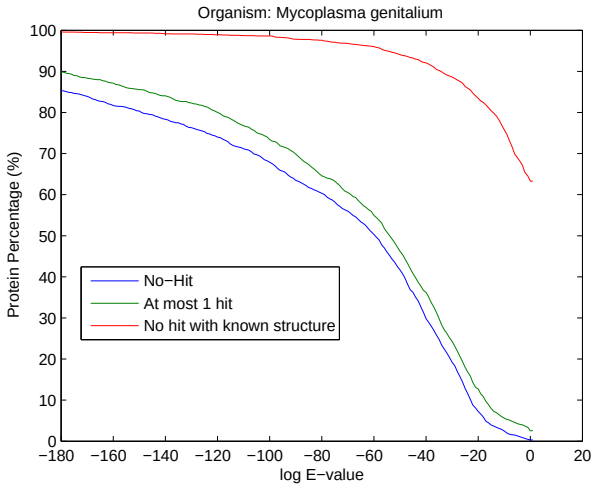

Figure 16: Percentage of proteins in Mycoplasma genitalium which does not have significantly similar proteins in NR database for a given e-value

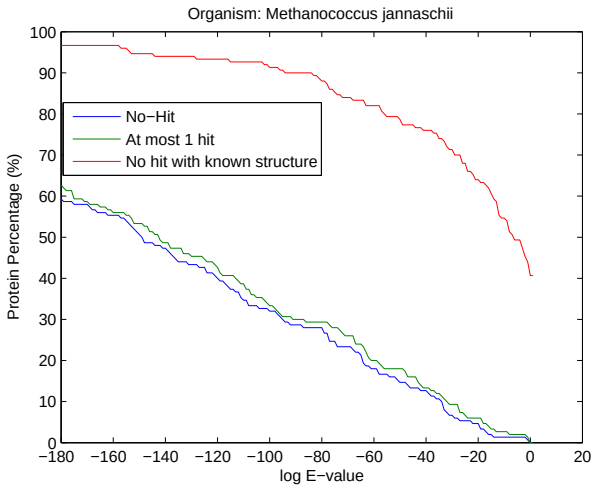

Figure 17: Percentage of proteins in Methanococcus jannaschii which does not have significantly similar proteins in NR database for a given e-value

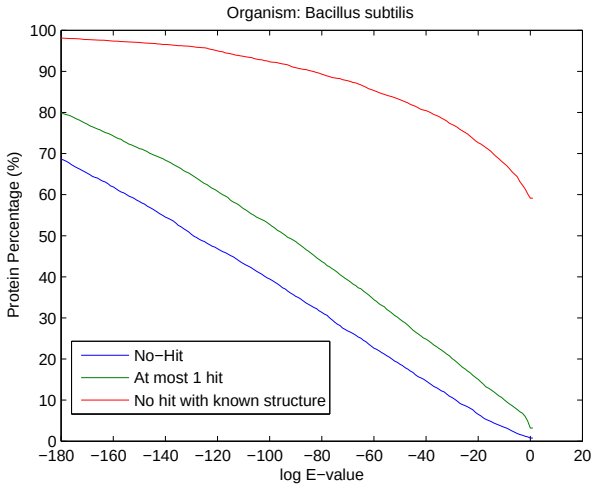

Figure 18: Percentage of proteins in Bacillus subtilis which does not have significantly similar proteins in NR database for a given e-value

e-value	No hits (%)	At most 1 hit (%)	No hit with known structure (%)
10^{-5}	0.5	8.1	72.9
10^{-4}	0.4	7.5	72.0
10^{-3}	0.3	7.0	71.0
10^{-2}	0.3	6.5	69.4
10^{-1}	0.3	6.0	68.1
10^{0}	0.2	4.4	66.4

Table 11: Percentage of proteins in Sulfolobus solfataricus which does not have significantly similar proteins in NR database for a given e-value

e-value	No hits (%)	At most 1 hit (%)	No hit with known structure (%)
10^{-5}	1.5	4.6	69.9
10^{-4}	1.3	4.2	69.0
10^{-3}	1.1	4.1	68.2
10^{-2}	0.9	4.0	67.1
10^{-1}	0.6	3.7	65.3
10^{0}	0.4	3.4	64.6

Table 12: Percentage of proteins in Mycoplasma genitalium which does not have significantly similar proteins in NR database for a given e-value

E-value	No hits (%)	At most 1 hit (%)	No hit with known structure(%)
10^{-5}	1.3	2	49.3
10^{-4}	1.3	2	49.3
10^{-3}	1.3	2	49.3
10^{-2}	1.3	2	47.3
10^{-1}	1.3	2	45.3
10^{0}	0.7	1.3	44

Table 13: Percentage of proteins in Methanococcus jannaschii which does not have significantly similar proteins in NR database for a given e-value

E-value	No hits (%)	At most 1 hit (%)	No hit with known structure(%)
10^{-5}	2.2	8.2	65.0
10^{-4}	1.9	7.7	64.3
10^{-3}	1.6	7.2	63.2
10^{-2}	1.4	6.8	62.3
10^{-1}	1.2	6.1	61.4
10^{0}	1.1	4.9	60.2

Table 14: Percentage of proteins in Bacillus subtilis which does not have significantly similar proteins in NR database for a given e-value