# Protein Structure Prediction using Decision Lists

by

**Volkan KURT**

**A Thesis Submitted to the**
**Graduate School of Sciences and Engineering**
**in Partial Fulfillment of the Requirements for**
**the Degree of**

**Master of Science**

**in**

**Computational Sciences and Engineering**

**Koç University**

**September 2005**

# Acknowledgements

I dedicate this work to my father Feridun, my mother Tülin and my younger brother Altuğ. If my loving family wasn't with me all through my master study, in fact, all through my life, I wouldn't be able to accomplish any of my goals hither.

My advisor Asst. Prof. Deniz Yuret has inspired me with his intelligence, perseverance and scientific insight. Assisted with the invaluable comments of my co-advisor Prof. Burak Erman, I was able to successfully conclude this study. I now have a *Yuret Number* of 1 and an *Erman Number* of 1 (Pólya number?).

Asst. Prof Alkan Kabakçıoğlu, Asst. Prof. Metin Türkay and Asst. Prof. Halil Kavaklı have also contributed to my study and my scientific background significantly, let alone attending my thesis defense as judges.

My friends Hakan Burak Çömden, Necmiye Genç, Selçuk Sümengen, Emre Özbalcı, Erk Subaşı, Burak Öztürk, Ayçıl Çeşmelioğlu, Sabri Bora Erdemli and Faik Can Meral deserve probably the greatest credit for they stack with me throughout my study and they didn't deny me their moral support.

My love, Tutku Rüya Özmen kept me on track. If it wasn't for her love and for her care, I couldn't have gotten this far.

I thank all the people I have known, and of whom I could or couldn't mention; especially to the reader who is about to read details of my work. I hope this work contributes somehow to yours, thus to the wealth of humankind.

# Abstract

Proteins are building blocks of life. Structure of these building blocks plays a vital role in their function, and consequently in the function of living organisms. Although, increasingly effective methods are developed to determine protein structure, it is still easier to determine amino acid sequence of a protein than its folded structure and the gap between number of known structures and known sequences is increasing in an accelerating manner. Structure prediction algorithms may help closing this gap.

In this study, we have investigated various aspects of structure prediction (both secondary and tertiary structure). We have developed an algorithm (Greedy Decision List learner, or GDL) that learns a list of pattern based rules for protein structure prediction. The resulting rule lists are short, human readable and open to interpretation. The performance of our method in secondary structure predictions is verified using seven-fold cross validation on a non-redundant database of 513 protein chains (CB513). The overall three-state accuracy in secondary structure predictions is 62.5% for single sequence prediction and 69.2% using multiple sequence alignment. We used GDL to predict tertiary structure of a protein based on its backbone dihedral angles phi and psi. The effect of angle representation granularity to the performance of tertiary structure predictions has been investigated.

Existing structure prediction approaches build increasingly sophisticated models emphasizing accuracy at the cost of interpretability. We believe that the simplicity of the GDL models provides scientific insight into the relationship between local sequence and structure in proteins.

**Keywords:** protein structure prediction, secondary, tertiary, greedy decision list learner

# Özet

Proteinler hayatın yapıtaşlarıdır. Bu yapıtaşlarının yapıları ise işlevlerinde, dolayısıyla da canlı organizmaların işlevlerinde hayati bir rol oynar. Protein yapısının tespiti için her seferinde öncekilerden daha etkili yöntemler geliştirilse de, hala bir proteinin amino asit dizisini bulmak katlanmış yapısını bulmaktan daha kolaydır ve bilinen protein yapıları ile bilinen dizilerin sayıları arasındaki fark ivmelenerek artmaktadır. Yapı tahmin yöntemleri bu farkın kapanmasında yardımcı olabilir.

Bu çalışmada, yapı tahmininin (hem ikincil hem üçüncül yapı) çeşitli yönlerini inceledik. Örüntü tabanlı protein yapı tahmini kurallarından oluşan bir liste öğrenen bir işlemsel süreç (Açgözlü Karar Listesi öğrenici, veya İngilizce kısaltmasıyla GDL) geliştirdik. Sonuçta oluşan kural listeleri kısa, okunaklı ve yoruma açıktır. Yöntemimizin ikincil yapı tahminlerindeki başarımı, 513 protein zinciri içeren artıksız bir veri kümesi üzerinde (CB513) 7-kat çapraz doğrulama kullanarak tasdiklendi. Yöntemin ikincil yapı tahminindeki genel üç-durumlu doğruluğu, sadece dizi bilgisini kullanarak %62.5 ve çoklu dizi hizalaması kullanarak %69.2. GDL'i bir proteinin üçüncül yapısını omurgasının iki-düzlemli açıları phi ve psi üzerinden tahmin etmek için kullandık. Açıların gösteriminde kullanılan ufalanmanın üçüncül yapı tahminlerinin başarımına etkisi incelendi.

Mevcut yapı tahmini yaklaşımları, doğruluğu yorumlanabilirliğin önünde tutarak gitgide karmaşıklaşan modeller inşa ediyorlar. İnanıyoruz ki, GDL modellerinin sadeliği, proteinlerin yerel dizisi ve yapıları arasındaki ilişkiye bilimsel bir sezgi sağlamaktadır.

**Anahtar kelimeler:** protein yapı tahmini, ikincil, üçüncül, açgözlü karar listesi öğrenici

# Table of Contents

# List of Figures



# List of Tables

# List of Formulas

# 1. Introduction

## *1.1 Proteins*

*Proteins* are complex organic compounds that consist of amino acids joined by peptide bonds. Proteins are essential to the structure and function of all living cells and viruses. Many proteins function as enzymes or form subunits of enzymes. Some proteins play structural or mechanical roles. Some proteins function in immune response and the storage and transport of various ligands. Proteins serve as nutrients as well; they provide the organism with the amino acids that are not synthesized by that organism. Proteins are amongst the most actively studied molecules in biochemistry and they were discovered by the Swedish scientist, Jöns Jakob Berzelius in 1838 [1].

An *amino acid* is any molecule that contains both an amino group and a carboxylic acid group. An *amino acid residue* is the residuals of an amino acid after it forms a *peptide bond* and loses a water molecule. Since we are interested in amino acids that form proteins, it is safe to use the terms residue and amino acid interchangeably. There are 20 different amino acids in nature that form proteins. Two other nonstandard amino acids are known to occur in proteins (Selenocysteine [2] and Pyrrolysine [3]) but these are very rare so only the standard 20 amino acids will be considered throughout this work.

Proteins are amino acid compounds and the composition of amino acids in a protein defines the three dimensional form that the protein folds to. These structures are unique in the sense that a given sequence of amino acids always folds into almost the same structure under the same environmental conditions (pressure, temperature, pH etc. There are exceptions but that is very rare.). Structures of proteins are investigated under four primary groups (Figure 1.1):

- *Primary Structure* is the sequence of amino acids in the protein.
- *Secondary Structure* is the composition of common patterns in the protein. Some patterns are frequently observed in the native states of proteins. This structure class includes regions in the protein of these patterns but it does not include the coordinates of residues.
- *Tertiary Structure* is the native state, or folded form, of a single protein chain. This form is also called the functional form. Tertiary structure of a protein includes the coordinates of its residues in three dimensional space.
- *Quaternary Structure* is the structure of a protein complex. Some proteins form a large assembly to function. This form includes the position of the protein subunits of the assembly with respect to each other.

There are a number of methods with varying resolution to determine the structure of proteins. For example, the primary structure can be determined by means of *mass spectrometry* [4], the secondary structure content (i.e. percentages of the common motifs) can be determined up to some certainty by means of *circular dichroism spectroscopy* [5] and the tertiary structure can be determined by means of

*x-ray crystallography* or *NMR spectroscopy* [6]. These methods require more time and effort as the expected resolution from the method increases.

There are also theory based methods in protein structure determination like *homology modeling*, *threading* or *ab initio* modeling. These methods are referred to as structure prediction methods. Homology modeling can be briefly described as fitting a known sequence to the experimentally determined three dimensional structure of a protein that is similar in sequence [6]. *Threading* is fitting a sequence to a database of known structures using a heuristic scoring method and finding the most likely structure [7]. Ab initio methods are methods that predict structure from scratch, i.e. they do not rely on known structure of the homologous proteins [6].

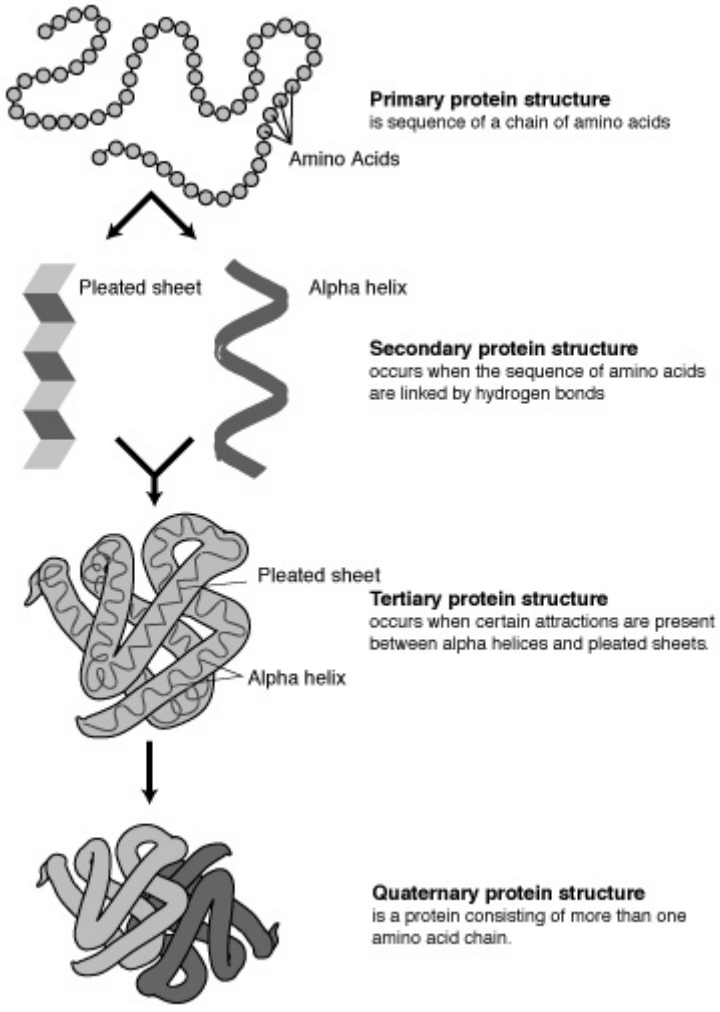


**Figure 1.1:** Different representations of protein structure.

(From ***http://www.genome.gov/Pages/Hyperion/DIR/VIP/Glossary/Illustration/protein.shtml***)

## 1.2 Decision Lists

This work concentrates on various aspects of protein structure prediction. The machine learning algorithm utilized in secondary and tertiary structure predictions is called *decision lists*. A decision list is an ordered list of rules where each rule consists of a pattern and a classification [8]. In this work, a classification is either a secondary or a tertiary structure assignment for the residue of interest. A pattern is the properties of residues that surround the residue of interest. We will be using the terms *frame* and *window* interchangeably to refer to the residues that surround a residue of interest.

One way to interpret a decision list is as a sequence of *if-then-else* constructs familiar from programming languages. Another way is to see the first rule as the default classification, the previous rule as specifying a set of exceptions to the default, the rule before that as specifying exceptions to those exceptions and so on. Table 1.1 illustrates how decision lists work.

| *Rule* | *Pattern* | *Classification* |
|---|---|---|
| **3** | Pattern 1 | Class 1 |
| **2** | Pattern 2 | Class 2 |
| **1** | Everything Else | Default Class |

**Table 1.1:** Illustration of a decision list with three rules.

Every rule has a pattern and a classification for that pattern. To classify an instance using this decision list, the instance is checked against the rules one by one. For example if an instance matches 'Pattern 1', it is assigned 'Class 1' (rule number 1). If an instance does not match 'Pattern 1' but it matches 'Pattern 2' it is assigned 'Class 2' (rule number 2). All instances that do not match 'Pattern 1' and 'Pattern 2' are classified as 'Default Class' (rule number 3 or the default rule).

In this work, frames of residues and their chemical or physical properties are used as patterns. Every position in a frame has a specific set of properties called *attributes* (or *features*). For example, "identity of the first residue to the left of the residue of interest" is an attribute. Every pattern is represented by a *conjunction* of these attributes. A conjunction (i.e. a logical conjunction) is a logical operator that results in true if all of its operands are true. The common name for this operator is 'and'. So a rule in a secondary structure prediction decision list may look like "If the identity of the first residue to the left of the residue of interest is Alanine **and** the identity of the first residue to the right of the residue of interest is Arginine, then the secondary structure of this residue is alpha-helix." In this case, the pattern is a conjunction of two attributes (the identities of left and right residues) and the class is alpha-helix. Each attribute in a rule defined this way can only have one value from a set of twenty residues. It is, however, also possible to have disjunctive attributes. A *disjunction* (i.e. a logical disjunction) is a logical operator that results in true if at least one of its operands is true. The common name for this operator is 'or'. When every attribute is a disjunction of attributes instead of a single attribute, a rule in a secondary structure prediction may look like "If the identity of the first residue to the left of the residue of interest is Alanine **or** Asparagine **and** the identity of the first residue to the right of the residue of interest is Arginine **or** Valine, then the structure of this residue is alpha-helix." In this case, the pattern is a conjunction of two attributes, where each attribute is a disjunction and the class is alpha-helix. Table 1.2 illustrates these rules more compactly.

| Rule Type | Pattern | | | Class |
|---|---|---|---|---|
| *Conjunction* | Left first residue=Alanine | **and** | Right first residue=Arginine | α-helix |
| *Conjunction of disjunctions* | Left first residue=Alanine **or** Asparagine | **and** | Right first residue=Arginine **or** Valine | α-helix |

**Table 1.2:** Rules based on conjunctions and on conjunctions of disjunctions.

Decision lists based on both a set of rules based on conjunctions of disjunctions and rules based on simple conjunctions have been utilized throughout this work. The tertiary structure predictions were performed using rules based on conjunctions and the secondary structure predictions were performed using rules based on conjunctions of disjunctions. Final version of decision lists adopted in this work uses rules based on conjunction of disjunctions.

There are a number of approaches to the problem of building a decision list given a set of instances and their classes. We have developed a novel learning algorithm named Greedy Decision List learner (GDL). A variant of the PREPEND [9] algorithm, GDL works by prepending one rule at a time to the front of a growing decision list. At each step, GDL searches for a rule that, when added to the decision list, maximizes the number of correctly classified instances in the training set. To cope with the large search space (more than $2^{180}$ possible rules for a 9 residue frame and 3 classes) a heuristic search algorithm had to be developed. The details of GDL algorithm are given in Section 2.5.

## *1.3 Outline*

In the light of the brief introduction we have given in this section on structure of proteins and decision lists, Section 2 concentrates on secondary structure prediction methods. We first give a more complete definition of secondary structure. Then materials and methods used to assess prediction methods are stated. The method of assessment becomes especially important, when one tries to compare the available prediction strategies objectively. Afterwards, the common strategies in the state-of-the-art prediction methods have been summarized, with a following brief introduction on the most commonly utilized methods. We describe our method in detail in Section 2.5 and show how we have implemented these common strategies. Then we give the results we obtained (i.e. the accuracy of our method) and discuss how consistent is the resulting model with the common knowledge and how each part in our prediction strategy contributes to the final result. We conclude by arguing that our algorithm yields results comparable to the state-of-the-art with a very simple model.

Section 3 concentrates on tertiary structure prediction based on backbone torsion angles. The data and data representation we have utilized in the decision list is given, followed by a description of the methods of accuracy measurement. Then we discuss the prediction results of various strategies by stating the accuracy of each strategy using common measures. We conclude the prediction studies with a brief description of our contributions to this field.

In the appendix, we give the results of a work similar to the one that Ramachandran [56] has conducted but this time expanding the study to further investigate restrictions on backbone torsion angles that stem from steric collisions.

# 2. Secondary Structure Prediction

In this section, we describe what secondary structure of a protein is and how it can be predicted from the proteins primary structure (i.e. from its sequence information). The common strategies in the state-of-the-art prediction methods are stated. How we implement these strategies, and how each part contributes to the accuracy of predictions follows. We conclude this section giving a possible ceiling for secondary structure prediction accuracies based on single sequence information and how accuracies can get past this ceiling by introduction of homology information to predictions.

## *2.1 Secondary Structure*

Protein chains form frequently observed structural motifs such as helices in their native state [10]. Common motifs (or patterns) are identified using various methods. The composition and sequence of these motifs in a protein is called the secondary structure. Most frequently observed secondary structure elements are α-helices and β-sheets. α-helices (Figure 2.1) are helix like structures and β-sheets (Figure 2.2) are pleated sheet like structures as their names suggest. Every region remaining in the protein after all the α-helix and β-sheet regions (and sometimes some other regions, depending on the method of secondary structure definition) are assigned is called a loop region. Loops do not have a specific shape like helices or sheets.

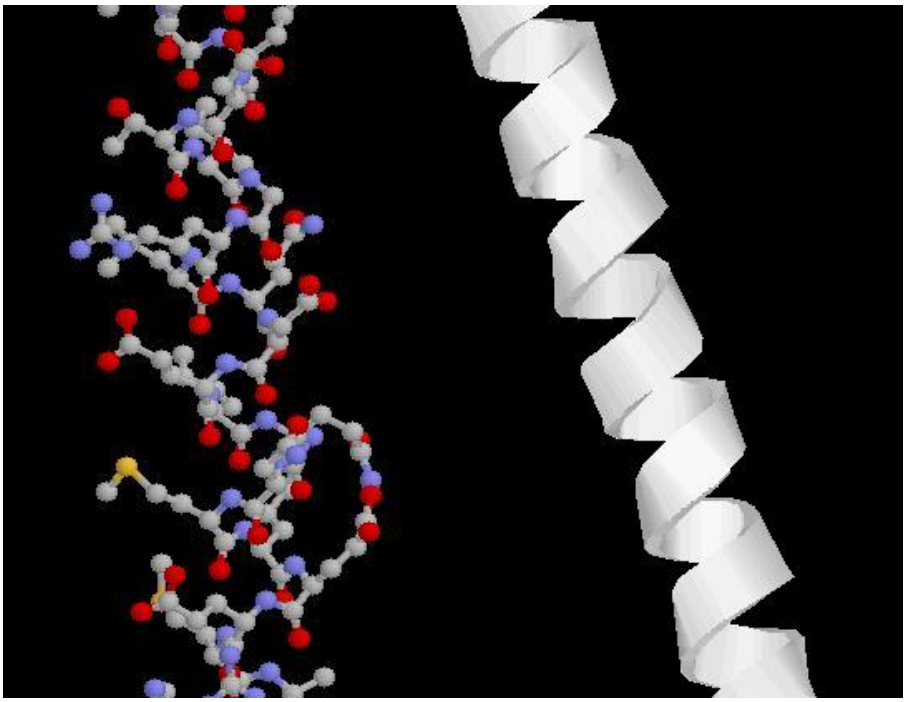

**Figure 2.1:** The α-helical secondary structure of Human Vimentin Coil 2B Fragment (PDB Code 1GK4). The amino acid backbone winds in a right-handed spiral. (*Created using Protein Explorer [57]*)

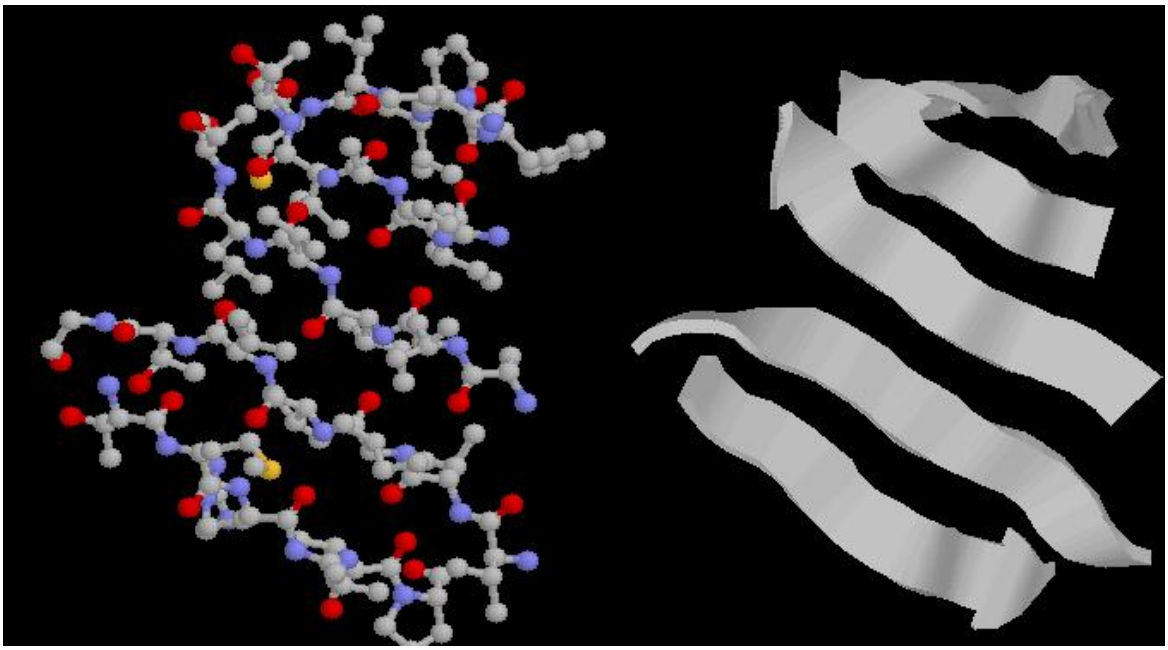

**Figure 2.2:** β-sheet secondary structure from Pyruvate Kinase (PDB Code 1PKN).
(*Created using Protein Explorer [57]*)

Determination of protein secondary structure is not an exact science. There are both laboratory based methods and theory based methods for determination of secondary structure. Laboratory based methods can be used to guess the secondary structure content without knowing the tertiary structure of the protein [5]. However, these methods are not very accurate. In another method, the scientist first determines the tertiary structure of the protein using a method like x-ray crystallography [6] and then specifies the regions of different secondary structure by careful inspection of the structure. If the tertiary structure of a protein is known, the secondary structure determination is a pattern recognition problem [11]. Various algorithms have been developed for determination purposes. There is no exact measure to define the secondary structure of a protein from its tertiary structure and different algorithms agree only about 71% of the time [12]. Common methods used in defining secondary structure of a protein are DSSP [11], STRIDE [13] and DEFINE [14].

Theory based methods rely on backbone geometry of the protein (DEFINE), the intermolecular hydrogen bonds (DSSP) or both (STRIDE). Backbone geometry based methods make use of the *phi* and *psi* angles of the residues. Phi and psi angles are dihedral angles defined on the backbone of a protein molecule (Figure 2.3). Certain combinations of these angles in consecutive residues result in a specific secondary structure motif [15]. These structures are stabilized by intermolecular hydrogen bonds (some exceptions to these are pure geometric definitions like bends [11]). DSSP assignment method has been the most frequently used method for secondary structure prediction studies. It is also the method adopted in this work.

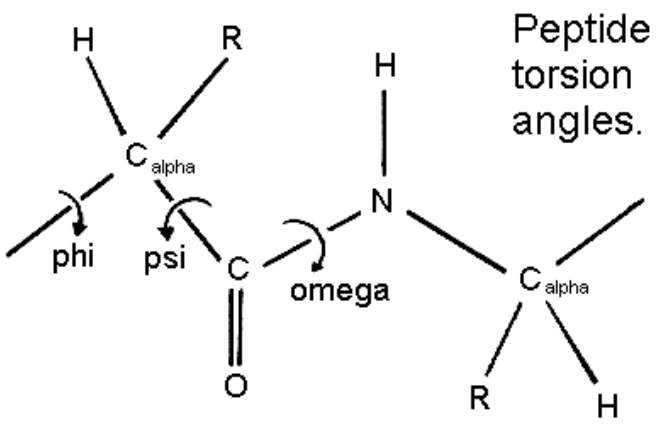


**Figure 2.3:** Definition of the phi ( $\phi$ ) and psi ( $\psi$ ) angles.

(From *http://www.cryst.bbk.ac.uk/PPS2/course/section3/helix1.html*)

## 2.2 Evaluation of Prediction Accuracy

DSSP defines 8 states of secondary structure. These states can be seen in Table 2.1. Since 8 states are considered hard to predict, these states are reduced to 3 groups. Structure prediction methods are evaluated on the three states, alpha-helix, extended-strand and coil (or loop). The remaining states are assigned one of these states. Prediction methods are evaluated on these three states.

There are different measures used to assess secondary structure prediction methods. The most common measures are $Q_3$ [16] and *Segment Overlap* [17]. $Q_3$ score is defined to be the percentage of the number of correctly estimated structures in the overall predictions. This measure depends on only the three states (helix/strand/coil), hence the name $Q_3$. Throughout this work three state per-residue accuracy definition ($Q_3$) is utilized to measure prediction performance.

| Reduction | DSSP Code | Description |
|---|---|---|
| H | H | alpha-helix |
| H | G | 3-helix (3/10 helix) |
| C | I | 5-helix (pi helix) |
| E | B | residue in isolated beta-bridge |
| E | E | extended strand, participates in beta ladder |
| C | T | hydrogen bonded turn |
| C | S | Bend |
| C | (no code) | Loop/coil |

**Table 2.1:** The 8 secondary structure states used by the DSSP method and their reduction to the $Q_3$ states. This reduction scheme has been adopted in this work and in JPred [12, 22]. (From ***http://www.cmbi.kun.nl/gv/dssp***)

### 2.2.1 Secondary Structure

There is a plethora of algorithms that predict protein secondary structure, each with a different accuracy. The choice of the data set and the secondary structure determination algorithm seems to have a greater effect on the accuracy figures than the particular learning algorithm used [12]. Therefore, in order to meaningfully compare the results of different algorithms it is important to test those using well known data sets and secondary structure determination conventions from the literature.

#### 2.2.1.1 Non-Redundancy

*Sequence identity* is the percentage of identical residues in a pair of aligned protein chains over the aligned length. This measure is used to estimate the homology relation between two proteins. There are other methods such as SD-Score [12] to measure this relation. In both methods, a pair of sequences is aligned globally using a dynamic programming algorithm (such as Needleman-Wunsch [19]) and a score is obtained from this alignment. The sequence identity is defined to be the number of exact matches in the aligned residue pairs (gaps not included). This measure has a drawback that there is not a single threshold for assigning two chains as homologues. This threshold usually changes with respect to the length of the proteins (i.e. the number of residues). For example 25% is the cut-off used for chains longer than 80 residues [20]. SD-Score, on the other hand, uses statistical techniques to normalize the score of the global alignment algorithm for length and compositional bias of the proteins [12]. Throughout this work SD-Score has been adopted as the measure of homology between two proteins.

Chothia et al. have shown that homologous proteins with more than 20% sequence identity have less than 2 Å *r.m.s. deviation* (Formula 2.1) of the backbone atoms of their common cores (i.e. regions of close or same fold and secondary structure content) [18]. If a protein of an unknown structure has a homologue of known structure, techniques based on aligning the two sequences would thus be expected to give good results. For objective comparison of prediction methods it is important to remove the test proteins and their homologues from the training set. Kabsch et al. showed that the proper selection of the test set could have 7%-56% effect on the reported results [21]. We also have found out-of-sample accuracy figures exceeding 80% using GDL before starting to use a test set free of homologues.

In this study the accuracy results were obtained using the CB513 dataset [12] and seven-fold cross validation. We added the multiple sequence alignments of each chain based on PSI-BLAST [27] to the respective training or test set. The data set contains 513 non-homologous chains and a total of 1,756,957 residues including the multiple sequence alignments. For training, the alignments are assumed to have identical secondary structure and are used as additional training data. For testing, the target chain and all its alignments are predicted and the final result at each position is decided by majority vote.

$$r.m.s.d(P_1, P_2) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|C_{i,1}^{\alpha} - C_{i,2}^{\alpha}\right|^2}$$

**Formula 2.1:** A formula for the root mean square deviation (rmsd) of the backbones of two protein chains. $C_{i,1}^{\alpha}$ and $C_{i,2}^{\alpha}$ are the $i^{th}$ alpha-carbons of protein chains $P_1$ and $P_2$ respectively and N is the number of residues in one chain. This formula can be extended to include other main chain atoms N and C.

### 2.2.1.2 Secondary Structure Definition

As stated in Section 0, even assigning secondary structure to a protein with known tertiary structure is not an exact science, and the different definitions (DSSP [11], STRIDE [13], DEFINE [14]) only agree with each other around 71% of the time [12]. A further complication is the number of states represented by a particular method. DSSP recognizes 8 different secondary structure states, and these need to be mapped to the three standard states (alpha-helix, beta-strand, and coil) for $Q_3$ accuracy evaluation. Application of the different published 8-to-3 state reduction methods shows variation of over 3% on apparent prediction accuracy [12]. In this study we use the DSSP definition and the reduction recommended by Cuff et al. given in Table 2.1 [12].

## *2.3 Prediction Strategies*

A plethora of machine learning approaches have been used for protein secondary structure prediction, including neural networks (e.g. PHD [20], JNet [32]), information theory (e.g. GOR [23]), and nearest neighbor algorithms (e.g. NNSSP [24]). All of these algorithms predict secondary structure from primary structure (i.e. sequence of residues). During the development of the decision list based approach we have found it instructive to consider the contributions of three components common to most approaches. Algorithms differ in how these three major components are implemented.

### 2.3.1 Sequence to Structure Component

Predicting secondary structure using the local sequence information forms the basis of all methods reviewed in this study. Using the sequence-to-structure component alone, earlier methods were able to achieve prediction accuracies around 60% [25]. To give some perspective, assigning every residue the loop classification gives 43% accuracy. Assigning every residue the most likely classification based on its amino acid gives 49% accuracy[1]. Thus, we can rate the contribution of the local sequence information at 10-15% depending on the baseline chosen.

[1] Based on CB513 data set.

### 2.3.2 Structure to Structure Component

A common method to improve performance is to filter the output of the sequence-to-structure component to correct unlikely structures, e.g. a single residue helix is unrealistic so it is turned into a loop. It is possible to achieve an additional 2-3% improvement over the plain sequence-to-structure prediction using manually constructed or machine learned structure-to-structure filters. In some approaches (e.g. SSPro [26]) the sequence-to-structure and structure-to-structure components are well integrated and their individual contributions are difficult to identify. We use two separate decision lists for these two components.

### 2.3.3 Multiple Sequence Alignments

The largest performance boost in prediction accuracy in the last twenty years was achieved by the introduction of evolutionary information in the form of multiple sequence alignments. Homologues of the target protein (even though their structure may also be unknown) bring in additional information that helps the prediction. The well known PHD algorithm exceeded 70% accuracy in 1993 with multiple sequence alignments accounting for 6-8% of the performance[2] [20]. Larger databases and improved search tools for multiple sequence alignment (e.g. PSI-BLAST [27]) are mainly responsible for the more recent improvements [28]. In some approaches multiple sequence alignment is an integral part of the procedure and its contribution is difficult to identify unless explicitly measured by the authors. For example, most neural network based methods construct their input using the frequency of occurrence of each of the 20 amino acids at each position in the alignment. In contrast, our decision list approach predicts the structure of each aligned protein separately and uses simple voting to decide the final results. We will use the term *single sequence prediction* to indicate results achieved without using multiple sequence alignments of the test proteins. A sample multiple sequence alignment is given in Figure 2.4.

```
1mcti-1-AUTO.1    : RICPRIWMECTRDSDCMAKCICVAGHCG : ITR2_CUCSA
ITR2_CUCSA        : MVCPKILMKCKHDSDCLLDCVCLEDICG : ITR2_BRYDI
ITR2_BRYDI        : RGCPRILMRCKRDSDCLAGCVCQKNGCG : ITR1_CITVU
ITR1_CITVU        : RRCPRIYMECKRDADCLADCVCLQHGCG : ITR3_CUCPE
ITR3_CUCPE        : RVCPKILMECKKDSDCLAECICLEHGCG : ITR3_LUFCY
ITR3_LUFCY        : .ICPRILMPCSSDSDCLAECICLENGCG : ITR1_MOMCH
ITR1_MOMCH        : RRCPRILKQCKRDSDCPGECICMAHGCG : ITR1_MOMRE
ITR1_MOMRE        : GICPRILMECKRDSDCLAQCVCKRQGCG : ITR3_CUCMC
ITR3_CUCMC        : RMCPKILMKCKQDSDCLLDCVCLKEGCG : ITR4_LUFCY
ITR4_LUFCY        : GICPRILMPCKTDDDCMLDCRCLSNGCG : TKTII
TKTII             : VACPRILMPCKVNDDCLRGCKCLSNGC. : CALU_CAVPO
CALU_CAVPO        : GSCPRVMIYCPARNKCTSDYDCPKPQCG : ELAF_HUMAN
ELAF_HUMAN        : GSCPIILIRCAMLNRCLKDTDCPGIKCG : APA1_HUMAN
```

**Figure 2.4:** The multiple sequence alignment of protein *1MCT* chain *I* from the CB513 data set. Dots indicate gaps in the alignment. (From ***http://www.compbio.dundee.ac.uk/~www-jpred/data***.)

[2] Based on RS126 data set.

## *2.4 Prediction Methods in Literature*

As stated previously, there are different methods of prediction with various accuracies. The methods differ in how they implement the three major components. Furthermore, they may have been trained with different data sets (i.e. set of proteins) and also different secondary structure assignments for that data sets. In this section some of the state-of-the-art secondary structure prediction methods will be discussed.

### 2.4.1 PHD

PHD [20] is the first method to break the 70% boundary on $Q_3$ accuracies of secondary structure prediction methods. The two-level neural network structure in this work has been adopted by several other methods later (such as JNet [32] and PSIPRED [29]).

In this work, the authors prepared a set of non-homologous proteins and named it the *RS126* set after the initials of their names (Rost and Sander) and the number of protein chains in it. If two proteins are at least 80 residues long and 25% of their sequences are identical, they are considered to be homologues and only one of them has been included in the set. For a reasonable measurement of the performance of the algorithm, it is necessary to have a non-redundant set of proteins (Section 2.2.1.1).

After a non-redundant set of sequences is compiled, the multiple sequence alignments and secondary structure assignments of each sequence have been retrieved from a database called HSSP [30]. The secondary structure definition algorithm used in this work is DSSP [11] (DSSP is the method of secondary structure definition in HSSP). The 8-to-3 state reduction scheme used is given in Table 2.2.

| Reduction | DSSP Code | Description |
|---|---|---|
| H | H | alpha-helix |
| H | G | 3-helix (3/10 helix) |
| H | I | 5-helix (pi helix) |
| C | B | residue in isolated beta-bridge |
| E | E | Extended strand, participates in beta ladder |
| C | T | Hydrogen bonded turn |
| C | S | Bend |
| C | (no code) | Loop/coil |

**Table 2.2:** The 8-to-3 state reduction scheme used in PHD method.

Also in PHD, minor corrections have been applied to the DSSP assignments (i.e. BC → BB, but BCB → CCC).

PHD uses a two layered *feed-forward neural network* [31] for sequence-to-structure prediction. The input to this network is a frame of 13 consecutive residues (See Section 1.2 for definition of frame). Each residue is represented by the frequencies of residues in the column of multiple sequence alignment which corresponds to that residue. That is to say, the residues in the homologous proteins that correspond to the residue in the query protein are selected and frequencies of each type of residue

are calculated and input to the network. This means each residue introduces 20 inputs to the neural network. Also, one more input is used for each residue in the frame for the cases that the frame extends over the N or C terminus of the protein. One final input is added for each residue called the *conservation weight* [20]. This weight represents the quality of a multiple sequence alignment (i.e. the number of aligned sequences and the similarity of the residues at that position in the alignment). So every residue is represented by *20+1+1=22* inputs, thus the sequence-to-structure network has *13x22* input nodes. The output of this network is 3 weights, one for each of the helix, strand and loop states.

The structure-to-structure prediction part of the algorithm is also implemented as a two layered feed-forward network. This time the input to the network is a frame of 17 consecutive residues. Each residue is represented by the 3 weights from the output of sequence-to-structure part plus one other weight for the cases that the frame extends over the N or C terminus of the protein. The conservation weights are added here too. This means each residue is represented by *3+1+1=5* nodes and this makes a total of *17x5* input nodes to the structure-to-structure network. Output of this step is again 3 weights for each of the possible states.

A number of different networks have been trained to reduce the bias that may stem from the order of representation of the inputs to the method and from the different encodings of the input. A third level is added for this purpose. This level simply takes the arithmetic average of the weights of each state over all trained networks. Then the state with the largest weight is assigned to be the final prediction for that position.

The contribution of each step is clearly stated for the PHD algorithm. Without utilizing the multiple sequence alignments (i.e. single sequence prediction), the $Q_3$ accuracy of the sequence-to-structure part is 61.7%. The structure-to-structure component adds a 0.9% to this performance for a total of 62.6% accuracy in single sequence predictions. Multiple sequence alignments add about 4.2% to the performance of the sequence-to-structure part for a performance of 65.9%. In presence of multiple sequence alignments, the structure-to-structure part adds up to 2.3% for a total of at most 68.2% accuracy. Different networks trained using different strategies result in somewhat lower accuracies. The jury decision adds up to 2% on the structure-to-structure network for a final of 70.2% $Q_3$ accuracy. In a separate work, PHD scored an average of 72.3% accuracy on the CB513 set with a new multiple sequence alignment procedure [12].

### 2.4.2 JNet

JNet [32] algorithm uses the same network structure used in PHD method (Section 2.4.1). The difference of this algorithm is that it utilizes an expanded set of protein chains, another 8-to-3 state reduction scheme and a number of new methods for generating multiple sequence alignments.

In this work, the authors prepared a set of non-homologous proteins and named it the *CB480* set after the initials of their names (Cuff and Barton) and the number of protein chains in it. This set is obtained from the CB513 data set [12] by removing chains shorter than 30 residues. The homology

measure used for preparation of CB513 set was SD-Score [12], which is more stringent than simple sequence identity.

The multiple sequence alignments in this method have been obtained by running PSI-BLAST searches on different databases and by aligning the sequences using different techniques (such as AMPS [33] and CLUSTALW [34]). The secondary structure definition algorithm used in this work is also DSSP [11]. The 8-to-3 state reduction scheme used is given in Table 2.3.

| Reduction | DSSP Code | Description |
|---|---|---|
| H | H | alpha-helix |
| C | G | 3-helix (3/10 helix) |
| C | I | 5-helix (pi helix) |
| E | B | residue in isolated beta-bridge |
| E | E | extended strand, participates in beta ladder |
| C | T | hydrogen bonded turn |
| C | S | Bend |
| C | (no code) | Loop/coil |

**Table 2.3:** The 8-to-3 state reduction scheme used in JNet method.

The sequence-to-structure part of this algorithm is, like PHD [20], a neural network. In this case the input frame is 17 residues long. At this step the various networks were trained which utilizes different representations of the columns of multiple sequence alignments (i.e. the query residue and its equivalents in the homologous proteins). These representations are:

- Frequencies of residues in the column of multiple sequence alignment (Same method with PHD).
- Weighted frequencies, where weights are the *BLOSUM62* [35] scores of each residue with respect to the query residue in the column.
- A position specific profile with position specific scores.

Residues in the frame are represented with one of these values and also for each residue in an alignment its conservation weight [20] is added. This level consists of one input, one hidden and one output layer. The hidden layer has nine nodes.

The output of the sequence-to-structure network is fed into a structure-to-structure network. This network also uses the conservation weights. The frame size at this part is 19 residues. This level also consists of one input, one hidden and one output layer. The hidden layer of this network also has nine nodes.

This algorithm utilizes one more level of neural network like the PHD method (In PHD, this was just an arithmetic average). If all the networks, which were trained on different data representations, agree on the final prediction than the residue is predicted to be of that structure. For the positions where there isn't a consensus on the final prediction (i.e. when all members of jury do not agree), a separately trained neural network is utilized (a network trained on no jury positions only).

All three parts of the algorithm combined gives an average $Q_3$ accuracy of 76.9%. The individual contributions of the three major components in JNet are not stated, but effect of multiple sequence alignment methodology is given. Selection of the homologue search method and the database searched can change the prediction accuracy around 1.1%. Different methods for aligning homologues and different representations of these multiple sequence alignments can change the accuracy around 7%. Finally, the third part (i.e. jury/no jury network) adds a 0.4% for a total of 76.9%.

### 2.4.3 PSIPRED

PSIPRED [29] is a neural network based method, which has three components. The difference of this method is that it conducts homology searches on a different database and uses a different set of proteins for training and testing. It also represents the multiple sequence alignments only as PSI-BLAST position specific scoring profiles. This method uses the same 8-to-3 state reduction scheme we have adopted (Table 2.1).

The network structure is simplified with respect to PHD and JNet methods (no jury networks or complex representations for multiple sequence alignment etc.). The sequence-to-structure part of the method is a back-propagation neural network. The input to this part is a frame of 15 residues. The residues are represented by the PSI-BLAST [27] scoring matrices. This neural network has 75 hidden nodes and 3 output nodes.

The output of the sequence-to-structure network is fed to the structure-to-structure network in frames of 15 residues. This network has 60 hidden nodes and 3 output nodes for the final prediction.

The performance of this method is not directly comparable to PHD or JNet since the same data set with those methods was not utilized during its development. Its $Q_3$ accuracy is 76.5%. This method has, however, proven to be more successful than the others in the third *Critical Assessment of Techniques for Protein Structure Prediction* (CASP) [36, 53] experiment [29].

### 2.4.4 GORV

GORV [23] is a secondary structure prediction method based on information theory and Bayesian statistics. Unlike other methods mentioned previously, this method does not use real valued encodings (such as frequencies or position specific scoring matrices etc.) of multiple sequence alignments.

GORV uses the CB513 [12] data set. Secondary structure assignments were taken from DSSP. The 8-to-3 state reduction scheme used is given in Table 2.4. This scheme does not take into account the 3/10 helices, which are not so rare[3] (3%). Thus the published results are not comparable with the other methods using CB513 set. We have checked to see that this reduction scheme may add at least 2.44% to the performance of a prediction using one of the other reduction schemes and exactly the same methods other than that (same training algorithm, same multiple sequence alignments etc.).

[3] Based on CB513 data set.

| Reduction | DSSP Code | Description |
|---|---|---|
| H | H | alpha-helix |
| C | G | 3-helix (3/10 helix) |
| C | I | 5-helix (pi helix) |
| C | B | residue in isolated beta-bridge |
| E | E | Extended strand, participates in beta ladder |
| C | T | hydrogen bonded turn |
| C | S | Bend |
| C | (no code) | Loop/coil |

**Table 2.4:** The 8-to-3 reduction scheme used in GORV method.

This method also applies some corrections to the reduced form (very short helices or strands are substituted with coils).

GORV method utilizes the three major parts of secondary structure prediction (mentioned in Section 2.3). The sequence-to-structure component depends on information theory, and specifically on the information function. Each residue is represented by a frame of 7 to 13 residues (depending on the sequence length). The predictions are based on the information function given in Formula 2.2 (details in GORV paper [23]). $P(S \mid R)$, which is the probability of a residue being of a secondary structure S, given the surrounding residues R (i.e. the frame), is approximated by the statistics of single residues, pairs of residues and triplets of residues in the frame. This means the frame is represented by the residues in specific positions, and residue pairs and triplets in specific positions. The probabilities of each secondary structure state (helix, strand or coil) are calculated using this method and each probability is normalized to [0, 1] interval. Then the most probable state is selected as the secondary structure prediction. Some thresholds for assigning a secondary structure are also applied at this step since the algorithm had a bias towards the coil structure (i.e. a considerable number of helix and strand states were predicted to be coil).

$$I(S;R) = \log\left(\frac{P(S \mid R)}{P(S)}\right)$$

**Formula 2.2:** Information function utilized in the GORV method.

R represents the frame of residues and S represents the secondary structure assigned to that frame. $P(S \mid R)$ is the probability of a residue has a secondary structure (one of H, E or C), given the frame of residues R.

Multiple sequence alignments are introduced to the predictions at the sequence-to-structure step. Basically each residue in the protein chains in the alignment of a query protein is assigned a probability for each of the states. Then these probabilities are averaged residue by residue and the most probable structure is assigned as the prediction. The thresholds are applied at this step when multiple sequence alignments are incorporated.

The structure-to-structure part of this algorithm is not a learner but simply a filter. At this step, only the unlikely estimates are eliminated. Very short helices and one-residue long strands are assigned to be loop. This is possible because the reduction scheme does not take into account the

isolated beta-bridges, which are assigned to be strands in most of the other works resulting in single residue strands.

The sequence-to-structure part of this method has 66.9% single-sequence $Q_3$ accuracy[4]. When multiple sequence alignments are incorporated to the algorithm, the accuracy rises to 73.4%. The individual contribution of the filtering part is not stated.

## 2.5 Structure Prediction with GDL

Different machine learning algorithms have been utilized to predict protein secondary structure. The most successful methods share a common core (Section 2.3). They represent the context of a residue by a frame of residues that surrounds it, they predict structure from this frame of residues and then they add another step of prediction to correct unlikely results by inspecting consecutive structure predictions. Finally, they all depend on multiple sequence alignments to represent evolutionary information into their prediction algorithms. Algorithms differ in how they implement each of these steps.

Although the algorithms mentioned before have very high accuracies (as high as 76%) their resulting models are difficult to interpret. Throughout this work, we have tried to build a method that predicts secondary structure with accuracy comparable to the state of the art (Table 2.6) and that yields a model interpretable by a researcher. We have selected to work with decision lists for this purpose. Table 2.5 shows a sample from one of the decision lists generated by our algorithm. These three rules result in 58.86% $Q_3$ accuracy[5].

| Rule | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Class |
|---|---|---|---|---|---|---|---|---|---|---|
| **3.** | l | al | dqe | adgp | CILFWYV | p | rqekp | lm | lm | Strand |
| **2.** | * | * | g | gp | ngps | gp | p | p | p | Helix |
| **1.** | * | * | * | * | * | * | * | * | * | Loop |

**Table 2.5:** A three rule decision list for secondary structure prediction.

The columns numbered 1-9 represent nine adjacent residue positions. Each uppercase single letter amino acid code indicates a residue that is allowed, and each lowercase code indicates a residue that is not allowed at a given position. A star indicates that all residues are allowed. For example, the first rule assigns every instance the loop classification. The second rule indicates that if the residues 3 to 6 are not Glycine, and the residues 4 to 9 are not Proline, and the center residue is not Asparagine or Serine then the conformation of the center residue (no. 5) is helix. The third rule requires, among other things, the center residue to be one of Cysteine, Isoleucine, Leucine, etc. for a strand prediction.

Decision lists have been described generally in Section 1.2. In this section, we will describe our novel method, Greedy Decision List learner (GDL) and how we have used this learner in secondary structure prediction problem.

[4] Based on CB513 data set.
[5] Based on CB513 data set. This is the result before structure-to-structure step is applied but including the multiple sequence alignments.

### 2.5.1 The Greedy Decision List Learner

To learn a decision list from a given set of training examples the general approach is to start with a default rule or an empty decision list and keep adding the best rule to cover the unclassified or misclassified examples. The new rules can be added to the end of the list [38], the front of the list [9], or other positions [39]. Other design decisions include the criteria used to select the 'best rule' and how to search for it.

The Greedy Decision List learner (GDL) starts with a default rule that matches all instances and classifies them using the most common class in the training data. Then it keeps prepending the rule with the maximum *gain* to the front of the growing decision list until no further improvement can be made. The gain of a candidate rule is defined as the increase in the number of correctly classified instances in the training set as a result of prepending the rule to the existing decision list (Formula 2.3). The algorithm is briefly described in Section 2.5.1.1.

GDL is a greedy algorithm since it tries to find the rules with the maximum possible gain at each step. It is not feasible to search all possible rules and select the best one since the search space is on the order of $2^{180}$ for even a 9 residue frame. So our algorithm tries to add rules which classify the largest number of misclassified instances correctly, but keeping in mind the previously correct classifications.

$$gain(r \mid d) = r_t d_f - r_f d_t$$

**Formula 2.3:** The gain of a candidate rule given a decision list.

$r$ represents the candidate rule and $d$ represents the decision list that has been built so far. $d_t$ and $d_f$ represent the sets of instances that have been correctly classified and misclassified by the decision list respectively. Likewise, $r_t$ and $r_f$ represent the sets of instances that the candidate rule correctly classifies and misclassifies respectively. So $r_t d_f$ is the number of new correct classifications introduced by the candidate rule and $r_f d_t$ is the number of new misclassifications imposed by the candidate rule.

#### 2.5.1.1 GDL Algorithm

1- Find the default rule $r_{def}$ that matches every possible instance in data set $D$ and assign the most common class to this rule. Calculate the gain $g_{def}$ of this rule assuming every instance in D was initially misclassified.
2- Set $g_{best}$ to $g_{def}$.
3- While $g_{best}$ is greater than zero
   a. Create a rule $r_0$ that matches no instances and set its gain $g_0$ to 0.
   b. For each possible class $c$
      i. Assign class $c$ to rule $r_0$.
      ii. Select a random instance $i$ from D.
      iii. Create a copy $r_1$ of rule $r_0$.
      iv. If $r_1$ matches $i$ but it misclassifies $i$ and $i$ is currently correctly classified, modify $r_1$ by removing one of the features that are available in both this rule and instance $i$ so that $r_1$ does not match instance $i$ anymore.
      v. Else, if $r_1$ does *not* match $i$ but it has the same class with $i$ and $i$ is currently misclassified, modify $r_1$ by adding the necessary features that are available in $i$ but not available in $r_1$ so that $r_1$ matches instance $i$.
      vi. Calculate gain $g_1$ of $r_1$.
      vii. If $g_1$ is greater than $g_0$, *set* $r_0$ *to* $r_1$ *and* $g_0$ *to* $g_1$.
      viii. Continue (b) until gain $g_0$ does not increase and a maximum number of consecutive attempts have been made.
   c. Prepend $r_0$ to the decision list and set $g_{best}$ to $g_0$.

### 2.5.2 Secondary Structure Prediction

In this section we describe the methods used for constructing and testing the decision list model. First the data is split into training, validation and test sets. A sequence-to-structure decision list is constructed from the training set. The decision list construction algorithm, GDL, was described in the previous section. Validation is performed by removing some of the rules that do not improve the performance on the validation set (i.e. rule pruning). Next, a structure-to-structure decision list is constructed using the same training and validation sets taking into account the errors made by the sequence-to-structure prediction. Finally, the two decision lists are used to predict the secondary structure of the test set proteins. When multiple sequence alignments are used during validation and testing, the decision list is applied to each sequence in an alignment separately, resulting in independent predictions and a majority vote determines the final decision at each position.

#### 2.5.2.1 Sequence to Structure

The sequence-to-structure decision list takes a window of nine adjacent residues as input[6] and classifies the center residue as helix, strand, or coil. For each protein chain in the training set, the window is shifted residue by residue generating $N$ instances for an $N$ residue chain. At the edges, special gap symbols are used to represent the positions that fall outside of the chain.

When multiple sequence alignments are used for training, each sequence (original or aligned) is treated as separate training data. All the aligned sequences are assumed to have identical structure. If

[6] Experiments with window sizes other than nine did not show significant improvement.

the alignments contain gaps, the windows with a gap in the center are ignored during training. Windows that are not significantly different than the query sequence sequence-wise are ignored too. The reason for this strategy will be mentioned in Section 2.7.3

Throughout training and testing, the inputs to the decision lists consist of nine individual residues. This is in contrast with neural network based approaches, where the multiple sequence alignment of a chain is used to construct a profile of amino acid frequencies at each position.

#### 2.5.2.2 Validation

GDL is used on the training set to construct the initial sequence-to-structure decision list. The training is followed by a validation procedure, where independent validation data is used to filter the rules that do not generalize well. The accuracy of the first $k$ rules of an $n$ rule decision list is computed for $k = 1 \ldots n$ on the validation set. The first $k_{max}$ rules that maximize the validation set accuracy are kept as the final sequence-to-structure decision list.

#### 2.5.2.3 Structure to Structure

The structure-to-structure decision list takes a window of 19 adjacent *secondary structure predictions* (helix, strand or loop) as input, and outputs a (possibly) new classification for the center residue. The validated sequence-to-structure decision list described above is applied to the original training set to get its predictions. Voting is used with multiple sequence alignments. These predictions, along with the correct classifications are given to GDL to construct the structure-to-structure decision list. A similar validation procedure is used to decide the final rule list.

### *2.6 Results*

To learn a decision list from a given set of training examples, we developed a novel learning algorithm named GDL that works by prepending one rule at a time to the front of a growing decision list. At each step, GDL searches for the rule that, when added to the decision list, maximizes the number of correctly classified instances in the training set. To cope with the large search space (more than $2^{180}$ possible rules) we have developed a heuristic search algorithm.

For the evaluation of the algorithm and performance comparison with other methods, it is essential to avoid some common pitfalls: comparing methods based on different data sets, different secondary structure definition methods, too few data points, training and test sets that contain homologues etc. do not give meaningful results. Our evaluation is based on the recommendations and the data set described in Cuff and Barton, 1999 [12]. To summarize, we ran a seven-fold cross validation test on a publicly available database of 513 non-homologous protein chains (CB513) using DSSP as the

secondary structure definition method and the 8-to-3 state reduction recommended in that work (Table 2.1) [12]. Table 2.6 compares the performance[7] of GDL with other algorithms using the same data set.

| **Method:** | PHD | DSC | PREDATOR | NNSSP | GDL |
|---|---|---|---|---|---|
| **Accuracy:** | 72.3 | 69.1 | 69.0 | 71.7 | 69.2 |

**Table 2.6:** Performance results for the set of CB513 proteins [12].

The $Q_3$ accuracy of single sequence predictions of GDL is 60.48% at the sequence-to-structure step. The structure-to-structure models add a 2.06% to this result for a total of 62.54% accuracy in single sequence predictions. The sequence-to-structure models yield 66.36% accuracy in the sequence-to-structure step. The structure-to-structure step adds a 2.38% to this result for a total of 69.21% accuracy. Clearly, multiple sequence alignments are the source for major improvement in prediction accuracy with a contribution about 7% (See Table 2.7).

| | **Single Sequence** | **Multiple Sequence** | **Improvement** |
|---|---|---|---|
| **Sequence-to-structure** | 60,48 | 66,36 | 5,88 |
| **Structure-to-structure** | 62,54 | 69,21 | 6,67 |
| **Improvement** | 2,06 | 2,85 | |

**Table 2.7:** Contributions of each step to the performance of our algorithm.

We regard the simplicity of the decision list based models as an asset. We have included a complete GDL model in Table 2.12. This model is sufficient to obtain 69% accuracy[8] on the CB513 data set and could be taken as a baseline for future work. (All models produced throughout this work is given under the folder '/1 - Secondary Structure Prediction/models' in the code base).

## *2.7 Discussion*

In the context of the results stated in the previous section, this section considers the following issues: the interpretation of the rules, the limits of single sequence prediction, and the contribution of multiple sequence alignments.

### 2.7.1 Classification of Amino Acids Based on Structural Preferences

Table 1.1 shows the first 3 rules of our sequence-to-structure model. These three rules result in 58.86% accuracy, when multiple sequence alignments are included[9]. This is a significant result since our whole sequence-to-structure model results in 66.36% accuracy. Actually these three rules are in accordance with some of the well known biologically inferred rules.

[7] We use the $Q_3$ performance measurements throughout this work, i.e. the percentage of residues correctly predicted as one of helix, strand, or loop.
[8] 69% is reached after the structure-to-structure step is applied to the results of this model.
[9] Based on CB513 data set. This is the result of only the sequence-to-structure step.

The base rule is "Assign every local frame loop". This is simply due to the fact that the most observed secondary structure is loop. If no structure can be assigned to a frame using the other rules, it is considered to be loop.

The second rule (Table 2.8) is "If the residue of interest is not Asparagine, Serine, Glycine or Proline, and the surrounding residues are not Glycine or Proline, assign this frame helix." This rule emphasizes the well-known fact that Glycine and Proline are helix breakers. Glycine is very flexible since it doesn't have a side-group (only a hydrogen atom). Proline causes kinks in the protein backbone. Thus, these two residues generally don't allow for a helix to form. Asparagine and Serine are polar residues [40, 41] and they have relatively small volume with respect to other residues. These may be the reasons these two residues do not prefer to be in helical structures.

The first rule (Table 2.9) is a little more complicated. We can interpret this rule roughly as "If the surrounding residues are hydrophobic, and the center residue is non-polar, assign this frame strand." Also a detail left out in this interpretation is, the close neighborhood of the center residue should not be Proline. This may, like in the helix case, stem from the kinked structure of Proline disrupting the strand structure. Other than that, the allowed residues here form a hydrophobic core for the protein. So strands may be more likely to occur in the relatively low solvent accessible areas of the protein.

| | | | | **HELIX** | | | | |
|---|---|---|---|---|---|---|---|---|
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| * | * | !GLY | !GLY | !ASN | !GLY | !PRO | !PRO | !PRO |
| | | | !PRO | !GLY | !PRO | | | |
| | | | | !PRO | | | | |
| | | | | !SER | | | | |

**Table 2.8:** The second rule in the sequence-to-structure decision list (Table 2.5)**.**

The exclamation marks indicate that those residues are not allowed at that position. The secondary structure assignment belongs to the 5$^{th}$ residue.

| | | | | **STRAND** | | | | |
|---|---|---|---|---|---|---|---|---|
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| !LEU | !ALA | !ASP | !ALA | CYS | !PRO | !ARG | !LEU | !LEU |
| | !LEU | !GLN | !ASP | ILE | | !GLN | !MET | !MET |
| | | !GLU | !GLY | LEU | | !GLU | | |
| | | | !PRO | PHE | | !LYS | | |
| | | | | TRP | | !PRO | | |
| | | | | TYR | | | | |
| | | | | VAL | | | | |

**Table 2.9:** The third rule in the sequence-to-structure decision list (Table 2.5).

| | |
|---|---|
| **Aliphatic** | Leu, Ala, Gly, Val, Ile, Pro |
| **Acidic** | Glu, Asp |
| **Small Hydroxy** | Ser, Thr |
| **Basic** | Lys, Arg, His |
| **Aromatic** | Phe, Tyr, Trp |
| **Amide** | Asn, Gln |
| **Sulfur** | Met, Cys |

**Table 2.10:** Amino acid classification based on the Swiss-Prot protein knowledgebase [42] release 47.8 statistics.

(From ***http://www.expasy.org/sprot/relnotes/relstat.html***)

Our rules re-confirm the fact that the classification of amino acids based on their chemical properties [58] does not always correlate with their structural preferences (Figure 2.5, Table 2.10). For example the acidic residues Aspartic acid and Glutamic acid, even though close to each other chemically, have very different alpha helix propensities (30% vs. 50%). In the other extreme, the aromatic residues Phenylalanine, Tyrosine, and Tryptophan have near identical secondary structure preferences (Figure 2.5).

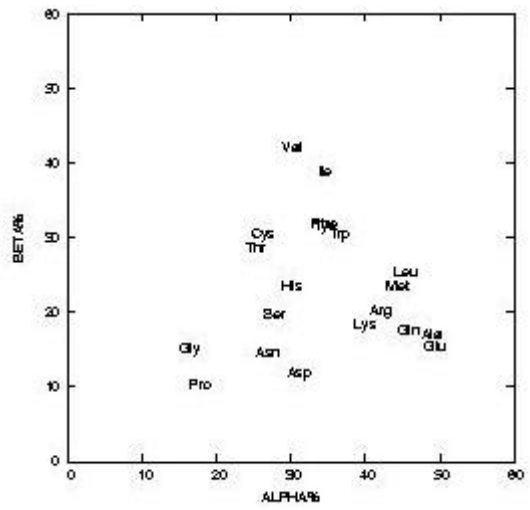


**Figure 2.5:** Frequency of each amino acid's alpha helix and beta strand conformations in the CB513 database.

For example, Alanine is observed in an alpha-helix conformation 48.82% of the time, beta-strand conformation 17.13% of the time, and is found in loops the rest of the time.

The amino acid substitution matrices that represent their relative replaceability in an evolutionary scenario seem to match most pairs with similar structural preferences. Table 2.11 shows the top 13 closely matched amino acid pairs with a score of 2 or higher from the BLOSUM50 matrix [35]. Most of these pairs are also close in Figure 2.5 except maybe the last column.

| Pair | BL50 | Dist | Pair | BL50 | Dist | Pair | BL50 | Dist |
|---|---|---|---|---|---|---|---|---|
| Phe-Tyr | 4 | 0.36 | Tyr-Trp | 2 | 2.40 | Tyr-His | 2 | 9.06 |
| Val-Ile | 4 | 5.54 | Glu-Gln | 2 | 4.31 | Ser-Thr | 2 | 9.23 |
| Leu-Met | 3 | 2.02 | Asp-Asn | 2 | 5.04 | Leu-Ile | 2 | 17.34 |
| Lys-Arg | 3 | 2.78 | Lys-Gln | 2 | 5.79 | Met-Ile | 2 | 18.26 |
| | | | | | | Asp-Glu | 2 | 18.72 |

**Table 2.11:** The closest matched pairs of amino acids in the BLOSUM50 matrix and their distance in the ALPHA%-BETA% plane given in Figure 2.5**.**

Our algorithm is successful in grouping amino acids of same structural preferences. For example, Alanine, Glutamic Acid and Glutamine have a high preference for alpha helices (Figure 2.5) and in all seven models generated by GDL in the cross-validation process, this preference can be observed in the first rule that assigns a helix (Table 2.8). Likewise Valine and Isoleucine prefers more to be in the strand structure and in all seven models they are grouped together in the first rules that assign strand to a residue (Table 2.9). (This is an expected result since every rule prepended to the list is less general than the previous ones, and the first three rules are the most general ones, and they reflect the most general preferences of the amino acids.) Phenylalanine and Tyrosine have similar structural preferences and this pair has a high score in the BLOSUM50 matrix (Table 2.11). These two residues are grouped together around 51% of the time in our models. Aspartic acid and Glutamic acid, which have different structural preferences and low match scores in BLOSUM50, are grouped together only 44% of the time. These statistics show that our algorithm is successful in capturing the nature of the data.

### 2.7.2 How Much We Can Get From Local Sequences?

Before the introduction of multiple sequence alignments into protein secondary structure prediction methods, the prediction accuracy peaked at around 60%[10] [28]. Our results obtained with simple rule sets also confirm this ceiling. Could we get past this limit using better algorithms or more data? Or is there a fundamental limit imposed by the data itself? To answer these questions we will propose a simple algorithm, look at its asymptotic behavior, and argue that the limits that apply to this algorithm apply to any learning algorithm based on local sequence information.

Consider a simple decision list model that takes a fixed length window centered at the target residue, tries to find exact matches to this window in the training set, and assigns the target residue the structure that is most frequently seen in these matches. Using smaller window sizes will increase the number of exact matches in the training set, but will also increase the variance of the target residue's structure. Using larger window sizes will provide more accurate estimates of the target residue's structure if exact matches are found, but the probability of finding such matches decreases exponentially with window size.

[10] In 1990s.

Figure 2.6 illustrates the decrease in the probability of finding a match and the increase in the accuracy of prediction as a function of the window size for two different data sets. The first data set contains homologues between the training and testing instances. Correlated proteins were eliminated in the second data set. In both cases we can see the expected decrease in the match probability as the window size is increased (more so in the non-homologous data set). More striking, however, is the difference in prediction accuracy between the two data sets when exact matches *are* found. In the first plot the accuracy increases with window size as expected, however in the second plot the accuracy peaks around 75%. In other words, even when we find an exact match for a 9 residue segment in a non-homologous protein with known structure, the center residue takes the same conformation only 75% of the time.

75% is not a limit that comes from data size limitations or algorithm choice, but is a fundamental limit reflecting the uncertainty in the data. It shows us that a significant part of the structure prediction puzzle must be solved by bringing in information not contained in the local sequence.

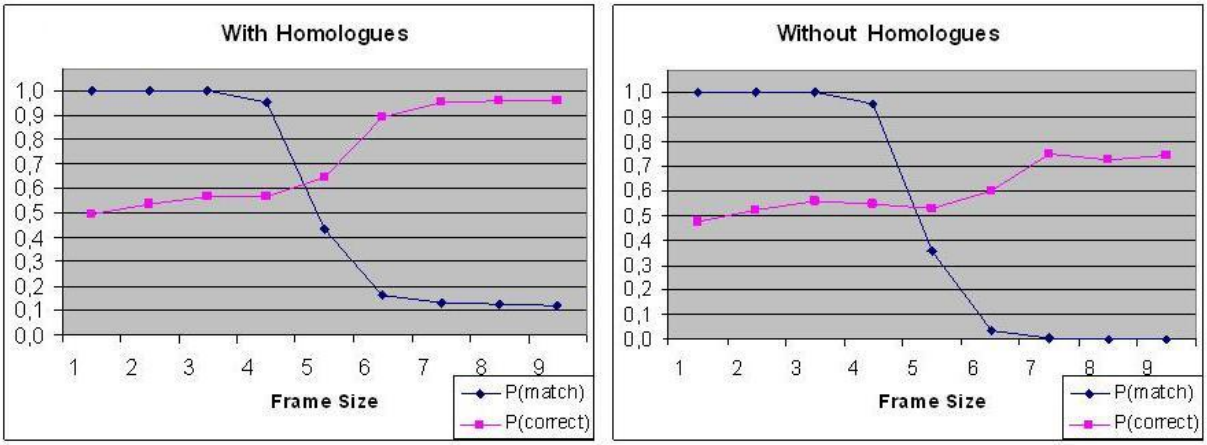


**Figure 2.6:** Probability of finding an exact match in the training set and probability of making a correct prediction when an exact match is found as a function of window size for two different data sets.

The first plot was obtained using 700,000 training instances and 60,000 testing instances from the union of PDB-Select [43] and WHAT-IF [44] databases. The second plot was obtained using 23,000 instances from the RS126 data set for testing. The training set consisted of 750,000 instances from WHAT-IF and PDB-Select databases after filtering for homologues to RS126 using the SD-Score cut-off 5 [12].

### 2.7.3 How does Multiple Sequence Alignment Help?

The most significant improvement in prediction accuracy has been due to the introduction of multiple sequence alignments [20]. Cuff et al. have shown that the method of generating multiple sequence alignments and the method of representing them has a significant effect on the accuracy of the prediction [12, 32]. In the previous section, we argued that there might be a ceiling for predictions

based on local sequence around 75%[11] in absence of homologous proteins. In this section we will try to explain how multiple sequence alignments help prediction methods.

Protein functional regions are expected to be more conserved than their sequence [18, 23]. Assume we are trying to predict the structure of a residue in a conserved region of a query protein and we have found *n* homologues of this protein. At every chain in the multiple sequence alignment, we expect almost all residues that correspond to the query residue to be in the same secondary structure. Assume the probability of a decision list assigning the correct structure to a single sequence is *p*. We assign a prediction to each of the *n* residues using the decision list. Assume that the decision list predictions, when they are not correct, are wrong for some different reason at each alignment (i.e. errors of sequences are independent). The probability of a correct prediction after a majority vote amongst the predictions of multiple sequence alignments is given in Formula 2.4.

$$P_{correct} = P\left(t > n/2\right) = 1 - P\left(t \le n/2\right) \cong 1 - P\left( Z \le \frac{\frac{n}{2} + 0.5 - (np)}{\sqrt{np(1-p)}} \right)$$

**Formula 2.4:** The probability of a correct prediction after a majority vote.

*t* is the number of correct predictions. *n* is the number of sequences in the alignment. At least *n/2* of the predictions should be correct for the majority vote to assign the correct prediction. (Actually there are other combinations with t < *n/2* that can result in a correct majority vote, but we discard them to simplify the formula for the sake of interpretability). The number of correct predictions is binomially distributed since we assume the errors of predictions are independent. Z is the normal approximation to the binomial distribution.

If an alignment of 10 sequences has been generated for a query sequence and the single sequence prediction accuracy is 70%, the probability of a correct prediction would be 95% from Formula 2.4. Likewise, if we can find 20 sequences, with again 70% prediction accuracy, the probability of a correct prediction would be 98%. In practice, this result is more like a ceiling since the residues in the multiple sequence alignment that correspond to the query residue will not be all of the same secondary structure with the query residue. Furthermore, the errors of different frames will not be independent all the time. We select only frames of low sequence identity with the query frame in the majority voting process to make this independence assumption more realistic. For example, a frame of residues may join the voting process if it has at least one residue that is not the same as the query frame. Other works have tried discarding chains in the alignment that have very high sequence identity to the entire query chain [12]. We, on the other hand have discarded frames with high sequence identity, not the entire sequence. Regions with low sequence identity but the same structures with the query protein are more likely to have independent errors, and we preserve this information by not discarding the entire sequence. (The results of this local frame discarding process can be seen in the code base under '/1 – Secondary Structure Prediction/models/results-q3.xls').

[11] This is a ceiling for local frames of 9 residues. Actually, the probability of finding a matching frame in absence of homologues is very low, thus the accuracies will be much lower than this.

Most common method to search for homologues from sequence is PSI-BLAST [27]. The most common algorithms to create multiple sequence alignments are PSI-BLAST (as in PSIPRED) and CLUSTAL [34] (as in GORV and in this work). There is still room for improvement in methods that search for homologous sequences in a database and methods that estimate homology from sequence. Thus, there is still room for improvement in secondary structure prediction.

### 2.7.4 Conclusion

We have developed a discrete and simple algorithm, GDL, for secondary structure prediction. The models GDL has generated are comparable to state of the art in secondary structure prediction in terms of accuracy (See Section 2.6). To emphasize the simplicity of the models, we present a human readable sequence-to-structure prediction model in Table 2.12. All of the models are also given in appendix (in code base, under '/1 – Secondary Structure Prediction/models') .

Unlike other methods, our models are human readable, and we believe this is the most important aspect of this algorithm. There are well known biological rules in literature on secondary structure prediction but these are very few. One commonly known rule is that Proline and Glycine are α-helix breakers. Glycine does not appear in alpha-helices frequently due to its flexible nature and Proline does not appear in α-helices frequently due its kinked structure. (We assign π-helices to be loops in the 8-to-3 state reduction scheme we utilize. Thus, Proline, which forms the π-helices, does not appear in helices). This property of α-helices is clearly reflected in the model (The $19^{th}$ rule in Table 2.12). We tried to interpret the most general three rules (the rules go from specific to general in decision lists) of the decision list biologically (Section 2.7.1). It may be possible for a biologist to interpret the rest of the rules and discover a new restriction on the secondary structure preferences of amino acids.

Furthermore, it is still possible to improve the accuracy of these models by modifying the rule searching methods. Although it is clear that most of the improvement in secondary structure prediction comes from the multiple sequence alignments, decision lists may be modified by introducing different aspects of the protein (some global properties of the protein) as an input. Amino acid composition (i.e. frequencies of the amino acids) in the chain is a good candidate for this purpose (We have tried incorporating amino acid compositions as an input to tertiary structure predictions and obtained a 0.2% improvement.).

One possible method for improvement at structure-to-structure step is to introduce the amino acid identities with the predictions from the sequence-to-structure step. While predicting the secondary structure of a chain from its sequence, every predicted structure may be used as an input for the next predictions in the same chain. Equivalently, using residue names as well as their structure predictions from the sequence-to-structure step as input to the structure-to-structure step may simulate this behavior. At the training phase, one can try both using the original secondary structures of residues, or try using the predicted secondary structures by the validated sequence-to-structure model.

There has to be a measure for confidence in the accuracy of a prediction. A reliability index has been developed in various works [32]. This is an index that is used to estimate how reliable a prediction is without knowing the real structure of the protein. Some similar measure may be investigated in GDL based predictions. A candidate measure that can be utilized is the difference between the numbers of votes in the multiple sequence alignment (i.e. If 5 out of 10 alignments are predicted to be helix and 3 of them are predicted to be strand, the reliability using this measure would be 5-3 = 2 for this specific prediction). Building a reliability scoring scheme is important to estimate how accurate a structure prediction of the protein would be in absence of the proteins tertiary structure.

| Rule | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Class |
|---|---|---|---|---|---|---|---|---|---|---|
| **20** | cpxo | ACILMFWYVX | cpvx | cilfwv | AILMFWYV | rndghkpstxo | cpwvx | ghp | dgpxo | **Helix** |
| **19** | qhx | mx | ailmtwvx | Xo | NDGP | fpxo | ilmwvx | rqmx | aqekmx | **Loop** |
| **18** | ifx | cmwx | clmswx | aelmxo | IFYV | pxo | CILFV | lmpx | rhkx | **Strand** |
| **17** | x | ilwvx | aqehiwvx | ailmwvxo | gixo | NDGSTY | cilmfvx | ilmfx | ailmx | **Loop** |
| **16** | delx | alfwx | qlo | delx | andcqgpxo | ILFWV | acgpwxo | CGILMFWYV | almwvx | **Strand** |
| **15** | ngpxo | ARNDQEKPS | ncgpvxo | ACHILMFWV | dghpxo | chilmfpwvx | degpxo | rdcegpswxo | cifwvx | **Helix** |
| **14** | aclmxo | CILFWVO | dpx | CILMV | ndgpxo | rqelkpxo | cgxo | celmxo | rcilmyv | **Strand** |
| **13** | rnghkpstxo | cgpxo | cgifptwvxo | ndghpstvxo | ARCILKMFWV | dgifpwvo | ifpv | * | cst | **Helix** |
| **12** | cwx | px | kpxo | wvxo | fwvxo | ndgpsto | AQILMFWYV | ARQELKMX | dcgilfptvxo | **Helix** |
| **11** | x | x | ailmfwvx | NDCGHKPST | gilfpwvxo | aqeilmpvxo | arqekmyx | ax | kx | **Loop** |
| **10** | elm | e | ILMFYV | ndgpwo | CILMFWYVX | acilmfpxo | hlo | aclf | ifx | **Strand** |
| **9** | cghifyvxo | cgpvx | ACILMFWYV | dcghpsxo | ARNDQEHKST | dgpxo | ndepstwxo | * | cfx | **Helix** |
| **8** | ckpxo | x | ciwvxo | cpxo | AEILMFWYV | dchpxo | cifpwvx | ndcegpstvxo | ndeghpstw | **Helix** |
| **7** | cmfx | mwx | acmx | qewxo | CILMFTWYV | CILFTYV | rndqehkpsxo | arqepxo | rlmxo | **Strand** |
| **6** | ex | qex | exo | gmxo | RNDCHKST | RNDQEGHKST | ailmfwxo | ilmx | px | **Loop** |
| **5** | gxo | ghpx | px | nghtxo | dghifptvxo | ndcgpstvo | dcpstxo | dcgpst | cvx | **Helix** |
| **4** | cl | e | am | CILFWYV | gpo | CILMFWYVX | lmo | q | * | **Strand** |
| **3** | w | acl | dqeo | adgpo | CILFWYV | pxo | arqekpxo | lmxo | lx | **Strand** |
| **2** | * | o | go | gpo | ngpso | gpo | gpo | pxo | p | **Helix** |
| **1** | * | * | * | * | * | * | * | * | * | **Loop** |

**Table 2.12:** A 20 rule decision list for secondary structure prediction.

This model performs predictions with 69% accuracy (after the structure-to-structure model is applied to the results of this model. The full model can be found in the code base under '/1 – Secondary Structure Prediction/models/cb513-not1.seq2str.model). The columns numbered 1-9 represent nine adjacent residue positions. Each uppercase single letter amino acid code indicates a residue that is allowed, and each lowercase code indicates a residue that is not allowed at a given position. A star indicates that all residues are allowed. For example, the 1st rule assigns every instance the loop classification. The 2nd rule indicates that if the residues 3 to 6 are not Glycine, and the residues 4 to 9 are not Proline, and the center residue is not Asparagine or Serine then the conformation of the center residue (no. 5) is helix. *x* represents unknown or non-standard amino acids and o represents the cases where the frame extends over the N or C terminus of the protein.

# 3. Tertiary Structure Prediction

Tertiary Structure is the native state, or folded form, of a single protein chain. This form is also called the functional form. Tertiary structure of a protein includes the coordinates of its residues in three dimensional space. Protein structure determination in laboratory conditions is a time and money consuming process. NMR and x-ray crystallography are the most commonly used methods to determine protein structure. There are also less frequently utilized techniques like electron microscopy [6]. There are certain drawbacks for each method. For example, during X-ray crystallography, the protein is crystallized, and on rare occasions this distorts portions of a structure [6]. Also, different laboratories using different methods may publish different structures for the same protein. This may stem from the laboratory conditions, the method and the resolution used during the determination process, the response of the protein to different media or simply by misinterpretation of the results by the researcher.

Despite their drawbacks, there is still no acceptable alternative to the laboratory based methods. The next best method for protein structure determination is molecular simulation. Molecular simulation programs try to generate the life-like conditions in a computer environment so as to simulate the folding process of the protein. These simulations require memory, CPU power and time. Although the first two conditions can be met to some extent, the last one, time, prevents the usage of this method from being practical. It has been shown that it is possible to fold a protein using molecular simulations to a reasonable deviation from its native structure. Recently a 36 residue long protein (Thermostable subdomain from chicken villin headpiece, PDB Code 1VII [45]) was folded to 4.5 Å rmsd (See Formula 2.1) from its native state. It took 4 months using a 256-CPU parallel computer to simulate 1 microsecond of molecular dynamics and the actual folding time of this protein is estimated to be 10 to 100 microseconds [46]. The running time of these simulations increase with the protein size. This CPU intensive nature of the problem stems from the fact that during the folding process thousands of atoms interact and most of these interactions have to be simulated.

The primary structure (i.e. amino acid sequence) of a protein is much easier to determine than its tertiary structure. The gap between the number of proteins of known sequence and the number of proteins of known tertiary structure is increasing in an accelerating manner (Figure 3.1). This stems from the limitations of laboratory based methods mentioned above, and in order to help close this gap, there has been many researches on how to determine the tertiary structure of a protein from its sequence. Tertiary structure prediction methods come into picture at this point.

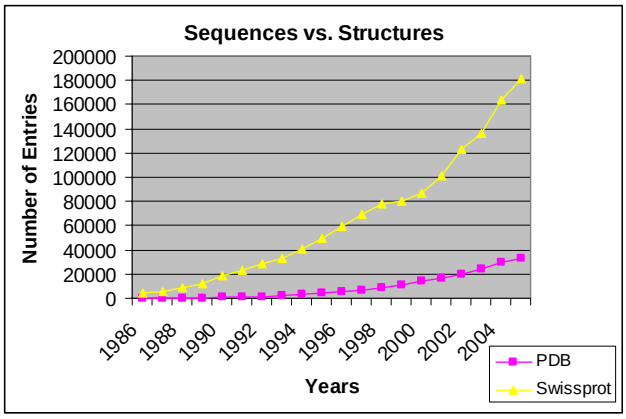


**Figure 3.1:** The number of known protein sequences (triangles) versus the number of known structures (rectangles).

(From *http://www.expasy.org/sprot/relnotes/* and *http://www.rcsb.org/pdb/holdings_table.html*)

Given the same environmental conditions, two proteins of the same sequence fold into the same structure. There are, however, cases where same sequence may lead to different structures, like in the prion protein case. $PrP^{c}$ (PDB Code 1AG2 or 1BWY [45]) and $PrP^{Sc}$ (which causes scrapie disease in sheep. PDB Code 1B10.) are two different versions of the same protein with same sequence and different three dimensional structures [47] (10% of the residues do not match in these pairs of PDB entries but this is a negligible amount). These cases are rare as a consequence of evolution and all structure prediction methods somehow incorporate sequence information of proteins in the prediction process.

Unfortunately, there is still no method that predicts tertiary structure accurately enough (Although some methods predict structures very close to the native state of the predicted protein, this is not the case for all proteins. The same methods predict some structures with very low accuracy for some proteins). The CASP (Critical Assessment of Methods of Protein Structure Prediction) experiment is an international platform where researchers try to blindly predict tertiary structures of proteins whose structures are not available to them before they submit their final predictions [53]. The best submitted models for different proteins have backbone *rmsd* values ranging from 0.66 to 22.63 Å, with a 5.65 Å mean and a 4.55 Å standard deviation (from the native states of the targets) [48]. Although there are very accurate predictions for specific targets, on the average the predictions are not very accurate and there is no way to know for sure whether a prediction is an accurate one or not before the real structure of a target is experimentally determined.

In this work we have concentrated on a number of ways to simplify the prediction problem. There are different ways to define tertiary structure. The obvious way to define tertiary structure is as a set of coordinates in the three dimensional space. Another way to define the tertiary structure is as a sequence of backbone torsion angles phi ($\phi$) and psi ($\psi$) (See Figure 2.3). The combination of $\phi$ and $\psi$ angles fully determine the backbone configuration of a protein [49]. Throughout this work we have

chosen to represent the tertiary structure of a protein by its backbone torsion angles, and the tertiary structure predictions are based on these angles.

## *3.1 Method*

We have based our tertiary structure predictions on phi and psi angles of the residues in a protein. As mentioned before, predicting these angles is equivalent to predicting tertiary structure. We utilized decision lists as the machine learning method for prediction (Section 2.5). The main advantages of using decision lists are that they are simple to implement and they yield human readable models. We have shown also that decision lists yield models that predict secondary structure within a reasonable accuracy compared to other methods in literature. The results we obtained in secondary structure predictions show that decision list, in particular GDL, is an appropriate method for use in protein structure prediction problems.

One important obstacle before utilizing decision lists in tertiary structure prediction is that a decision list requires both its inputs and outputs to be discrete since inputs should be a series of conjunctions and outputs should be classes assigned to these conjunctions (Section 1.2). In the tertiary structure prediction case, we tried to predict $\phi$ and $\psi$ angles from both residue names in the frame and the previously predicted angles of that frame. We utilized different methods to discretize these angles.

### 3.1.1 Data Discretization

Discretization is the process of mapping continuous data onto a discrete representation. In our case, we had to discretize $\phi$ and $\psi$ angles. There are several methods in literature to discretize continuous data [50]. In our case, a straightforward definition of the distance of two angles is simply the difference of the angles. In tertiary structure prediction case, the input is already discrete when the predicted angles are not used as input. Residue names are already discrete. However, the output (i.e. the predicted angles) is always a real value within [-180,180]. So some discretization schema must be used for output. Also some discretization schema must be used for input when predicted angles are used as input.

Discretization of the angles results in loss of information up to some extent. For example, assume we are splitting the range [-180, 180] into buckets of 10 degrees long. Each $\phi$ angle would be represented by the center of the bucket it is in. The average distance of the center of a bucket to the data points in it would be ½*(½*10) = 2.5 degrees. This means each $\phi$ angle would be represented with an error of 2.5 degrees on the average (if phi angles are equally distributed in the bucket), and with a maximum error of 5 degrees. If the predictions are accurate enough, the final tertiary structure of the protein may be found in a reasonable time and acceptable accuracy using methods like molecular dynamics simulation. For example, one of the structures predicted by ROSETTA server, which was the most successful tertiary structure prediction method in fourth CASP experiment [52], had a 0.26 Å backbone rmsd from the native state, and Fan et al. were able to improve this result to

0.17 Å rmsd and improvements in the predictions for other targets were also achieved by means of molecular dynamics [51].

Two different discretization methods have been utilized throughout this work. Initially, the [-180, 180] region was split into buckets of equal length. The bucket lengths were one of 15, 30, 60, 90 or 120 degrees. This way, either ϕ or ψ angles are discretized for prediction. The second method is to discretize these two angles together. The frequencies of different angle pairs have been subject to a number of previous studies. Ramachandran [54] has chosen to represent the distribution of angle pairs in a single chain or multiple chains with a scatter plot, which is today called *Ramachandran plot*. A sample Ramachandran plot is given in (Figure 3.2).

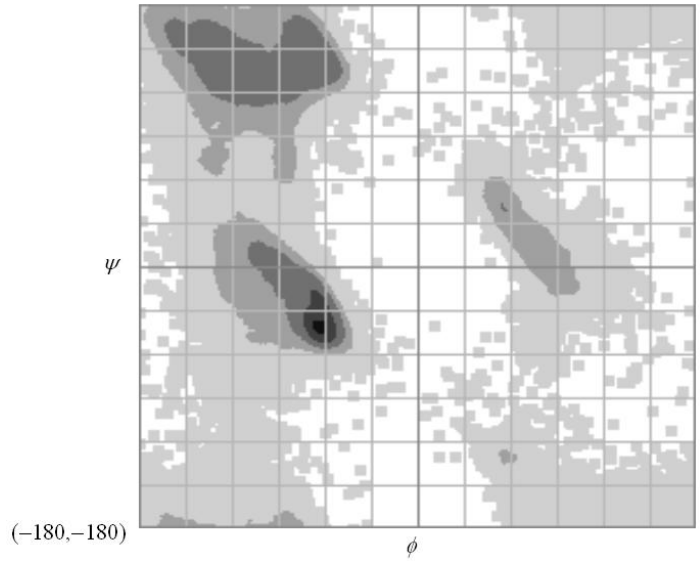


**Figure 3.2:** A sample Ramachandran plot. The x-axis shows the phi angles and the y-axis shows the psi angles. The left bottom vertex of the graph shows the ϕ and ψ angle combination (-180, -180) degrees. Grid lines are placed at every 30 degrees. The shades show the density of the angle combination at that point. Darker shades indicate that there are more pairs than others. This figure also shows a sample discretization with 30 degree long square buckets, i.e. 30 degrees for ϕ and 30 degrees for ψ angles (From ***http://home.ku.edu.tr/~dyuret/bio***).

All ϕ and ψ angle configurations are not observed in protein chains (white regions in Figure 3.2). Some of the regions are more populated than others and some of these regions are observed in secondary structure elements like α-helices or β-sheets. Actually, backbone angle configurations are the basis of secondary structure definition methods (See Section 0). Figure 3.3 illustrates some of these regions that correspond to specific secondary structures.

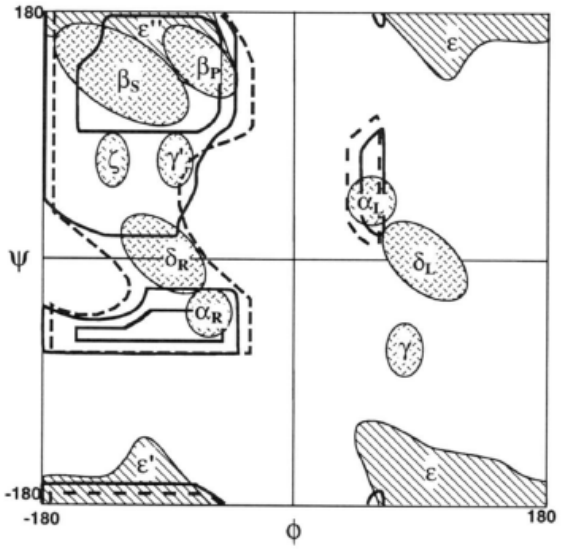


**Figure 3.3:** Most frequently occupied regions in a Ramachandran plot [15].

In the first discretization schema, all of the shaded regions were used. In the second discretization based on this plot, only the most common secondary structure regions were selected. These regions are $\alpha_R$ for α-helices (right-handed), $\beta_s$ for beta-sheets, γ and γ' for tight turns. The white regions are referred to as "strained" regions, since they are not occupied frequently, and mostly they are occupied only because the long range (global) interactions in the protein cause amino acids to prefer a different conformation than it would have if no long range interactions were present.

In this work, two different discretization schemes based on Ramachandran plots are utilized, first the most common regions are used as buckets and then the regions that correspond to the most common secondary structures (α-helices and β-sheets) are used (Figure 3.3).

### 3.1.2 Input to the Decision List

To predict the tertiary structure of a protein from its primary sequence, it is a common method to use amino acid names as input to that algorithm (See Section 2.4). Residues can be also grouped according to their physical and chemical properties (e.g. large amino acids, polar amino acids etc.) [55]. It is possible to feed this grouping information of the amino acids to the learning algorithm as well as their identities. This may be a good way of representing the amino acids, since their physical and chemical properties are factors in forming of the tertiary structure (by means of Van der Waals interactions, ionic interactions, hydrogen bond formation, etc.). This information has been incorporated into our tertiary structure predictions. However, later tests have shown that, this information does not improve the accuracy of predictions significantly. A comparison of results is given in Table 3.5.

We also used as an input the previously predicted angles. We have conducted tests, where the already predicted phi and psi angles in the local frame were used to predict the next angle. When predicting the structures of the residues in a protein chain, it is possible to go over the residues randomly or in an orderly fashion starting from the n-terminus to c-terminus of the chain (i.e. left to right) or vice versa. As the predictions are conducted, we obtain more information on the sequence we

have, the guessed structures of the surrounding residues. This information can be incorporated into the algorithm for a possible boost in prediction accuracy (Table 3.1).

| *Order of Prediction* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Sequence* | R | I | C | P | R | I | W | M | E | C | . | . | . | . |
| *$\phi$-Angle Prediction* | $\phi_1$ | $\phi_2$ | $\phi_3$ | $\phi_4$ | | | | | | | | | | |

**Table 3.1:** Incorporating previous predictions in a sequence to the current prediction.

The table shows an intermediate stage of a left to right (N terminus to C terminus) prediction. In the left to right prediction, structures of the residues indexed with 1, 2, 3 and 4 are already predicted. To predict the structure of the 4$^{th}$ residue, besides the identities of residues, the predicted structures of residues 1 to 4 are available as input.

To sum up, the input to our algorithm contains the amino acid identities in the local frame and their phi and psi angles if they are already predicted as well as their physical and chemical properties. These predicted angles are also discretized (since output is discretized). A frame size of 9 residues was chosen as a basis[12]. If a local frame extends over one of the termini, a special class (e.g. NAN) was used to represent those types of residues and angles. All tests in this section use data from a widely used set of proteins called PDB-Select [43] and the tertiary structures of these proteins have been taken from the PDB database [45].

### 3.1.3 Measuring Prediction Accuracy

The accuracies of predictions have been assessed using two different measures. The first measure is percentage of the number of correct predictions over all predictions. The other measure is the root mean square deviation (rmsd) of angles from their original position. The first measure is easy to calculate but it alone does not give enough insight into the quality of predictions. The percentage score does not give information on how tolerable the errors are. The ultimate goal of structure prediction is to find the final tertiary structure of a protein without using experimental methods if possible. Using molecular dynamics simulations the predicted models can be improved towards the native structure of the protein (See Section 3.1.1). However, improvements are not always possible on predicted models [51]. If the final predictions are very distant from the native state of the protein, or have some unnatural, strained conformation, molecular dynamics simulations may not help improve the model. In our case, for example, if an angle that corresponds to the middle of a chain is predicted with a 100 degree error, the structure of the protein would not be near the native state. Furthermore, if the false predictions are off their native state by a large distance, the resulting model may not be used for constructing the final tertiary structure, since atoms may be clashing with each other.

*Root mean square deviation* (shortly rmsd, See also Section 2.2.1.1) is a commonly used measure to assess accuracy of tertiary structure prediction problems. It simply gives information on how distant the atoms of the predicted model are to its native state. In tertiary prediction studies, this value is calculated from the difference of coordinates of backbone atoms in the native state and the predicted

[12] Increasing frame size didn't provide significant improvement.

form of the protein. In this study, since we are interested in the local frame, the rmsd values have been calculated by measuring the difference between the predicted angles and their values in the native state. When discretization methods have been used, the centers of the buckets (or regions) have been used to measure the distance from the original angle. Formula 3.1 shows how rmsd values have been calculated throughout our tertiary structure prediction studies.

$$a)\ \sqrt{\frac{\sum_{i=1}^{N}(\partial\phi)^2}{N}} \qquad b)\ \sqrt{\frac{\sum_{i=1}^{N}(\partial\phi)^2+(\partial\psi)^2}{2*N}}$$

**Formula 3.1:** Rmsd calculations of phi (ϕ) and psi (ψ) angle predictions.

Formula *a* shows how to calculate backbone rmsd of a prediction based on only ϕ angles. If both ϕ and ψ angles are predicted, i.e. discretization based on Ramachandran plots, formula *b* is utilized.

### 3.1.4 Discussion

A number of decision lists have been trained for this prediction with different discretization schemes and input features. Each trained model is than tested to check which method yields the best performance in terms of percentage accuracy (Table 3.2) and rmsd (Table 3.3).

| | | Phi | | | | | Psi | | | | | Combined | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **15** | **30** | **60** | **90** | **120** | **15** | **30** | **60** | **90** | **120** | **Region** | **Secondary** |
| **All** | **All** | 37.27 | 51.22 | 64.87 | 77.15 | 80.51 | 32.79 | 52.06 | 68.81 | 76.04 | 80.44 | 58.75 | 71.82 |
| | **Same** | 31.90 | 44.52 | 61.44 | 68.44 | 78.26 | 30.47 | 49.05 | 64.97 | 71.99 | 77.42 | 58.04 | 73.05 |
| | **Identical** | 31.56 | 43.89 | 61.37 | 66.81 | 78.12 | 29.84 | 48.39 | 64.72 | 72.00 | 76.64 | 56.40 | 71.40 |
| | **None** | 29.40 | 42.58 | 61.23 | 59.92 | 78.11 | 22.83 | 35.69 | 49.85 | 53.90 | 60.77 | 39.38 | 53.18 |
| **Right** | **All** | 36.64 | 50.36 | 63.57 | 76.84 | 79.51 | 31.08 | 48.29 | 65.22 | 71.74 | 75.58 | 57.20 | 69.60 |
| | **Same** | 31.64 | 43.81 | 61.66 | 67.24 | 78.19 | 29.89 | 47.41 | 63.74 | 70.16 | 74.81 | 56.47 | 69.51 |
| | **Identical** | 31.49 | 43.82 | 61.39 | 66.44 | 78.10 | 29.47 | 47.35 | 63.57 | 69.56 | 74.56 | 56.60 | 69.42 |
| **Left** | **All** | 32.33 | 46.78 | 62.27 | 68.14 | 79.05 | 31.60 | 49.87 | 67.67 | 75.57 | 79.78 | 54.96 | 69.21 |
| | **Same** | 31.23 | 44.29 | 61.48 | 66.72 | 78.16 | 28.47 | 45.96 | 63.34 | 70.83 | 76.26 | 54.60 | 68.66 |
| | **Identical** | 31.03 | 43.32 | 61.32 | 66.61 | 78.12 | 28.64 | 45.80 | 63.35 | 70.22 | 74.94 | 55.07 | 68.59 |

**Table 3.2:** The percentage of correct estimates for different input sets obtained from the proteins in PDB-Select. These results are obtained using 120,000 residues for training, 9,000 residues for validation and 45,000 residues for testing.

The columns in Table 3.2 show the results of different discretization methods. For example the third column shows results when the predicted ϕ angles are discretized using 15 degree buckets. The last two columns are results for combined discretization methods based on the frequently occupied regions in Ramachandran (See Section 3.1.1). “Region” column shows the results for discretization using most frequently observed regions in the Ramachandran plot. “Secondary” column shows results for the reduced version of the “Region” discretization, i.e. only the regions contributing to the formation of the most common secondary structure motifs like helices, sheets and turns are included (Figure 3.4).

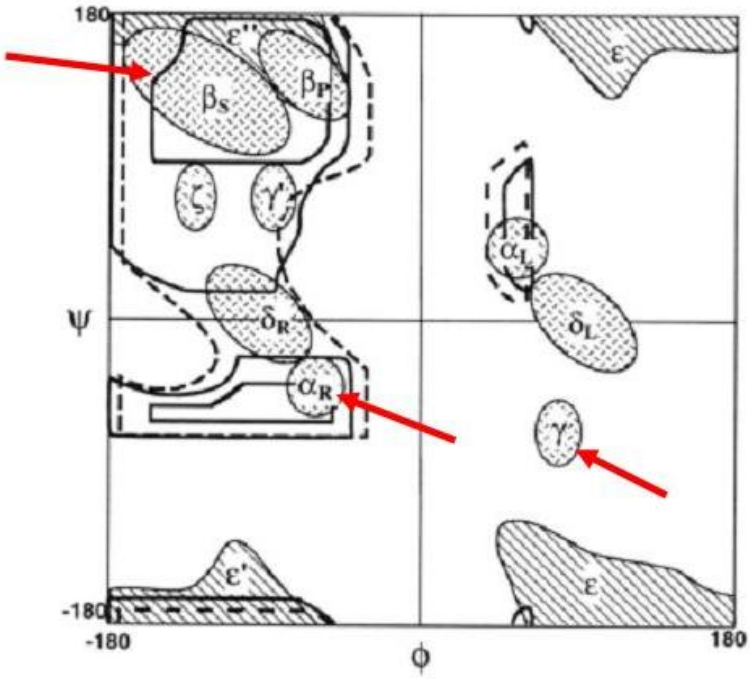


**Figure 3.4: "**Region" and "Secondary" discretization schemes.

In region based partitioning, the shaded areas in the plot has been used as buckets ($\beta_S$, $\beta_P$, $\zeta$, $\gamma'$, $\varepsilon$, $\varepsilon'$, $\varepsilon''$, $\alpha_R$, $\alpha_L$, $\delta_R$ and $\delta_L$). All other regions (the white, unshaded parts) are labeled strained. In the secondary structure portioning scheme, only those angle pairs that occur in the most frequent secondary structure elements (helix, sheet and tight turns) are taken. These regions are marked by arrows in the figure. All other combinations, in this case are taken to be loop/coil.

The rows in Table 3.2 show different input schema after discretization methods are selected. The first column shows if a prediction from left to right or right to left is conducted (See Section 3.1.2). "All" means all angles other than the angle of interest are assumed to be known beforehand. "Right" means the angles are predicted starting from right end of (C terminus) the protein to the left end. And "Left" means the angles are predicted starting from left end of the protein to the right end. The second column shows which angles ($\phi$, $\psi$ or both) are used and in which discretization schema. "All" means all possible angles and all possible discretization methods are used as an input. "Same" means only the predicted type of angles are used but with any possible discretization schema (i.e. if $\phi$ is predicted only $\phi$ angles are used as input) "Identical" means only the predicted type of angles discretized with the predicted type of discretization are used. "None" means prediction based on just sequence information (i.e. No angle information is used). "None" overrides the first column. For example, the performance (61.66%) at the intersection of the row "right", "same" and the column "phi", "60" is the result for a prediction of $\phi$ angles discretized at 60 degree buckets. The inputs to this prediction are the amino acid identities and the $\phi$ angles that remain at the right of the angle of interest (i.e. right to left prediction.). Not only the 60 degree buckets but also the different bucket sizes (15, 30, 90, and 120) are input to this test. The performance (70.22%) at the intersection of the row "left", "identical" and the column "phi", "90" is the result for $\psi$ angles discretized at 90 degree buckets. The inputs to this

prediction are the amino acid identities and ψ angles that remain at the left of the angle of interest (i.e. left to right prediction). Only the 90 degree discretization of the psi angles is used for this test.

In Table 3.2, all of the results have been obtained by using already known angles as input in the prediction stage (not the predicted angles). These results are obtained only to have a loose estimate on the upper bound of backbone angle predictions. The "Secondary" column is especially important in this graph since it shows significant correspondence with the results obtained in our secondary structure prediction studies (See Section 2.6). The average three state (helix/sheet/coil) accuracy in secondary structure prediction using single sequence information is around 62%. The closest analogue of secondary structure prediction in tertiary structure predictions is the ones based on the discretization scheme "Secondary" (Figure 3.4). As mentioned before, secondary structure definition algorithms depend on backbone angle configurations to assign a secondary structure to a residue [14]. Consecutive occurrence of a specific ϕ-ψ combination results in a specific secondary structure. So the tertiary structure predictions using secondary structure based discretization schemes resemble the secondary structure predictions closely. The backbone angle predictions that do not use the previously predicted angles in the local frame are analogous to the sequence-to-structure step in the secondary structure predictions. The ones that incorporate the predicted angles in the local frame as input to the predictions are analogous to the combination of sequence-to-structure and structure-to-structure predictions. The best result in the latter type of tertiary structure prediction is 69.42% (the "Secondary" column intersected with the "right, identical" row in Table 3.2). Although in this case we predict four states, the accuracy is close to the 62% three state accuracy in the single sequence predictions. A better accuracy in this type of tertiary structure prediction is not unexpected since secondary structure is defined when consecutive angle pairs lie in specific angle combinations. One has to predict the regions of consecutive residues correctly to assign a secondary structure to a residue in that region. The probability of predicting two consecutive regions correctly is lower than predicting one region correctly. Also, in the tertiary case, the stated results are obtained using real angle values surrounding the residue of interest. So the actual results will be lower. One important fact is that the data set used in secondary structure predictions is different from the one used in tertiary structure predictions, which may lead to incomparable accuracies.

| | | Phi | | | | | Psi | | | | | Combined | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **15** | **30** | **60** | **90** | **120** | **15** | **30** | **60** | **90** | **120** | **Region** | **Secondary** |
| **All** | **All** | 42.95 | 44.37 | 46.67 | 51.13 | 52.98 | 66.38 | 66.01 | 71.23 | 70.84 | 53.07 | 56.93 | 48.11 |
| | **Same** | 47.68 | 46.90 | 50.95 | 57.37 | 55.95 | 71.30 | 69.56 | 75.38 | 75.21 | 57.02 | 57.56 | 47.09 |
| | **Identical** | 47.70 | 47.42 | 51.19 | 58.07 | 56.13 | 72.25 | 70.15 | 75.80 | 75.19 | 57.99 | 60.64 | 48.60 |
| | **None** | 49.24 | 48.45 | 51.60 | 61.11 | 56.15 | 99.69 | 101.12 | 103.49 | 106.83 | 75.16 | 79.98 | 65.34 |
| **Right** | **All** | 43.41 | 44.95 | 47.71 | 51.49 | 54.31 | 73.97 | 72.02 | 75.54 | 75.64 | 59.30 | 59.31 | 50.37 |
| | **Same** | 48.17 | 47.06 | 50.24 | 58.33 | 56.04 | 74.24 | 74.37 | 78.21 | 78.64 | 60.23 | 58.96 | 50.50 |
| | **Identical** | 48.26 | 47.72 | 51.08 | 58.73 | 56.16 | 74.13 | 73.57 | 78.58 | 80.03 | 60.52 | 58.94 | 50.61 |
| **Left** | **All** | 45.41 | 45.48 | 49.19 | 56.92 | 54.92 | 69.45 | 67.86 | 73.14 | 71.45 | 53.96 | 59.59 | 50.62 |
| | **Same** | 47.27 | 47.06 | 50.80 | 58.94 | 56.08 | 73.92 | 72.82 | 77.88 | 77.72 | 58.47 | 60.26 | 51.24 |
| | **Identical** | 47.37 | 48.56 | 51.39 | 59.01 | 56.13 | 73.15 | 72.47 | 78.00 | 78.72 | 60.07 | 59.73 | 51.32 |

**Table 3.3:** Rmsd values (in degrees) for the tests whose performances are given in Table 3.2. The notation is the same. Rmsd values are calculated using Formula 3.1.

As mentioned before, the percentage of correct estimates is not sufficient to fully assess the quality of a prediction method in terms of three dimensional distance estimates. For example, in Table 3.2 it is clear that, as the bucket size increases the percent of correct estimates increase. This is due to the fact that as the bucket size increases the number of classes decrease, and the probability of the algorithm to randomly select a correct class increases. For the 120 degree case, for example, the algorithm has only three choices, and a random predictor would have 33% accuracy on the average. So the learning algorithm results in 45% accuracy gain (for a total of 78%). However, in the 15 degree case, there are 24 buckets for the algorithm to choose from. A random predictor would have 4% accuracy in this case, and the gain of the algorithm is 27% (for a total of 31%).

Rmsd is more suitable for assessing the quality of a prediction than percentage accuracy. In the "region" based discretization scheme, the "strained" region is a relatively large region with respect to all of the other regions. A representative point is necessary for each class to make an rmsd calculation and in the strained case this representative is hard to select since the angle pairs are scattered among the relatively large strained region. So instead of selecting a representative, we have used 90 degree average distance for every misclassified instance. In the region [0, 360], the average distance between two randomly selected points is 90 degrees.

The best rmsd value (43.41 degrees) for phi angles are from the predictions that use all angles to the right of the phi angle of interest with all possible discretizations of those angles (right to left prediction using both real phi and psi angles in the input). For psi angles, the best value (53.96 degrees) is obtained by using all angles with all their discretizations, however, this time with a left to right prediction. As one can see the psi angles are harder to predict, most probably because they are more flexible than phi angles, hence the higher rmsd values. These results are loose upper bound estimates since they are obtained by using the real values of angles in the prediction stage.

| Test | Average Accuracy | Average RMSD (Degree) | Percent of chains with less than 10Å RMSD | Test Info |
|---|---|---|---|---|
| Phi 15 All All, Psi 120 All All | 58.85 | 48.01 | 9.18 | Best RMSD (Degree) |
| Phi 90 All None, Psi 90 All None | 56.91 | 83.97 | 6.46 | Worst RMSD (Degree) |
| Phi 120 All All, Psi 120 All All | 80.47 | 53.02 | 2.72 | Best Accuracy |
| Phi 15 All None, Psi 15 All None | 26.11 | 74.46 | 6.12 | Worst Accuracy |

**Table 3.4:** The percentage of chains with backbone RMSD values less than 10 Å for the given set of tests (See Table 3.2 and Table 3.3) [59].

Table 3.4 shows the percentage of chains whose predicted structures have less than 10Å backbone rmsd to their native structures. The values are calculated from the coordinates of backbone atom $C^{\alpha}$ for the given test runs. This table clearly shows that high percentage accuracy does not necessarily mean low angstrom rmsd. The best performing models in terms of accuracy are selected (Phi 120 All All, Psi 120 All All) which perform with 80% on the average. But the percent of predictions less than 10Å rmsd are less than the models with worst accuracy (Phi 15 All None, Psi 15 All None). This may be due to the fact that by representing a 120 degree interval with its midpoint, even when the prediction is correct we represent the real angles with a 30 degree error on the average.

| | Accuracy | RMSD (Degree) |
|---|---|---|
| **Only residue names** | 50.88 | 44.66 |
| **Residue names and composition** | 50.90 | 44.63 |
| **All** | 50.97 | 44.45 |

**Table 3.5:** The effect of input feature set on the accuracy.

These values are obtained from variations of the 'Phi 30 All All' test (See Table 3.2, Table 3.3).

Table 3.5 shows the effect of input representation on the performance. Using chemical and physical properties of residues (in the frame that represents the query residue) as well as their names and angles, the accuracy was 51.22% (Table 3.2). When the chemical properties were discarded as attributes, the accuracy decreased by 0.34% to 50.88%. This shows that representing residues by their names are enough, since the gain obtained by using their chemical properties is relatively low. Using both residue names and amino acid compositions the results increase by a 0.2% to 50.90%. So introducing residue composition to the predictions doesn't improve the results.

| Test | Average Accuracy | Average RMSD (Degree) | Accuracy using Real Angles | RMSD using Real Angles |
|---|---|---|---|---|
| Phi 15 left same, Psi 15 left same | 24.63 | 82.93 | 29.85 | 60.59 |

**Table 3.6:** Prediction accuracy for a real prediction.

Table 3.6 shows the performance of a real prediction. This time the predicted angles are incorporated as input to the next predictions instead of their real values. As mentioned before, using

real values for the surrounding angles give a ceiling for the prediction performance. Real predictions in this case are about 5% lower than the corresponding performance ceiling. The method performed much worse in terms of rmsd in the real predictions.

### 3.1.5 Future Work

The data set, test results and the scripts used to replicate these results are given in the appendix. The results are not very accurate considering that an even 10 degree error on the average may cause models that don't resemble the native state of the protein. But the prediction method is open to improvement, by changing discretization methods and/or learning algorithm. At the very beginning of this project, we have been utilizing rules including only a conjunction in the decision list to make the rule searching time feasible. This allowed for a complete search over all possible rules in a reasonable time. Afterwards, the algorithm has been modified to search for rules that are conjunction of disjunctions. This meant now the decision lists could include the *logical not* of the attributes (e.g. not Alanine, not Valine). For this search would take $2^n$ time for n binary attributes, it wasn't possible to conduct a full search over the rule space. We have altered the algorithm to conduct the searches using a heuristic function rather than searching the complete rule space (See Section 2.5.1). Although simple conjunction rules are the best possible ones within the rule space that explains the training data, the restrictions imposed by the lack of negative attributes (*logical not*s) tends to be more effective in the performance of the algorithm. The new algorithm may be applied to the tertiary structure problem to see if one could get better results.

The physical and chemical properties of amino acids have been incorporated into the input to the algorithm as well as the amino acid identities and the angle information. However, later in the secondary structure prediction studies, we did not incorporate this data as input. Instead, we tried to induce biological rules such as secondary structure preferences of amino acid groups. Any further studies in tertiary structure prediction should also consider rebuilding models using only amino acid identities and angles as input to the learning method.

A common method in secondary structure prediction is to incorporate multiple sequence alignment information to the prediction algorithm. The same approach may be applied to the tertiary structure prediction. The two stage method we have utilized in the secondary structure predictions i.e. a sequence-to-structure and a structure-to-structure step, can also be utilized in the tertiary structure prediction studies.

Last but not least, one has to check the output of the prediction algorithm to assess the performance of the final tertiary structure. The final results may be fed to a molecular dynamics tool to see if the prediction leads to a feasible structure. It may be the case that the final predicted structure is at least partially acceptable. For example, there may be regions, which are correctly classified and possibly those estimates may be used in function determination or determination of other function related information.

# 4. Contributions

There have been numerous attempts to predict protein structure from its sequence. We have developed another method that is simple and modular. The methods in literature yield models that predict structure fairly well. But the models they yield are hard to interpret. For example, in the secondary structure prediction case, the contribution of sequence-to-structure and structure-to-structure steps are not clearly stated in some of the prediction methods we have mentioned. In our case, each step of the algorithm is clearly distinguished. The structure-to-structure model may even be used in filtering the predictions of other algorithms.

Structure prediction is only a step in protein structure determination and finally in function determination. The current prediction algorithms may be used to predict structure but most of them give little insight into the nature of the prediction problem. Decision lists, on the other hand, yield human readable models. What is more is, as we have shown in Section 2.7.1, the models are in correspondence with some biological rules. We have only evaluated three rules in a secondary structure decision list. There are more rules in that list that may yield invaluable knowledge in the structure prediction domain.

We have studied predictions that depend on local information, and have shown that local information is not enough to predict structure with complete accuracy (Section 2.7.2). Evolutionary data comes into picture at this point. Multiple sequence alignments increase the accuracy of predictions, even without knowing the three dimensional structure of the proteins in the alignment. We have incorporated multiple sequence alignments only to the secondary structure predictions. They may also help in predicting the tertiary structure. The outline of our prediction model is then:

- An initial prediction from sequence information
- Incorporating multiple sequence data
- A final step to relate or filter predicted structures.

In literature, this is a commonly utilized scheme and we have utilized the same scheme with a different algorithm, Greedy Decision List learner, which we think may help biologists find out information on the nature of prediction problems.

Currently, there are various methods that predict secondary and tertiary structure. Each type of prediction has associated difficulties. For example, even the assignment of secondary structure from known tertiary structure is not an exact science. For the tertiary structure case, there may be cases where the same sequence leads to completely different structures [47]. There is still more to discover in the nature of proteins and we have tried to help in this discovery by means of introducing a new algorithm in the domain to simplify the structure prediction process. We have also introduced a model to predict secondary structure, which researchers can even use for a quick back-of-the-envelope prediction. There is still room for improvement in our method and in all available prediction methods.

# 5. Appendix

## *5.1 Steric Collisions*

### 5.1.1 Introduction

Ramachandran has shown that all combinations of phi and psi angles cannot exist in nature, simply because of steric collisions of residue atoms [56]. Ramachandran has taken into account only $\phi-\psi$ angles of Alanine dipeptides and has shown that there are certain conformations an Alanine dipeptide cannot conform to [15]. These restrictions stem from the collision of two atoms in the disallowed conformations.

We have extended the study of Ramachandran to a set of Alanine chains ranging from 1 to 9 residues long. Every residue added to a protein chain introduces new restrictions on the backbone conformation space.

We have searched which atoms can collide when a short protein chain is randomly perturbed. The bond lengths within a protein are taken to be constant and the search is conducted by simply altering the $\phi$ and $\psi$ angles (and side chain angles when possible). The outcome of this search is a set of simple rules like "the oxygen atom of the second residue in a four residue long chain can only collide with the oxygen atom of the third residue." We have only tried to find out possible collisions on alanine chains two to nine residues long. This way we had very flexible chains, since Alanine has a small side chain. We call altering the $\phi$ and $\psi$ angles of residues in a protein to arbitrary values "steering" the molecule and the collisions detected this way "steric collisions".

### 5.1.2 Method

There are a number of ways to detect possible steric collisions in a protein chain. The complete search would be achieved by assigning every possible angle combination to the $\phi$ and $\psi$ angles. However, this is not possible since not only the angles are continuous but also the number of possible conformations increases exponentially with the number of residues in the chain. So there are two major problems in this search that needs attention: the how to select the amount of perturbation of an angle at each step, and how, i.e. in what order, these perturbations are applied to the angles.

We have performed different tests using two different search strategies. First, we have selected a discretization of angles and tried to check the proteins for collisions at all possible combinations of these angles. One possible discretization scheme is to divide the (-180, 180) degree interval into segments and represent angles by the segment in which it resides (See also Section 3.1.1). Different segment lengths are possible and this choice also effects the running time of the algorithm. For example, a segment length of 30 degrees means each angle has 360/30=12 possible values and each perturbation of an angle would mean a change of 30 degrees at least. A chain with 4 residues will have 3 $\phi$ and 3 $\psi$ angles to work with. This means the search would take $12^6 \approx 2$ million steps. Although

the application of this method is possible for very short protein chains, it is not feasible with longer chains. One more restriction is that, the algorithm we use does not check the path the atoms traverse as the angles are modified. That is to say, we have the initial and final positions of the atoms after a modification, not the path that results in that position. Thus, the amount of perturbation of an angle should be selected in such a way that the path that there should not be another atom in the path that any atom traverses. If a path contains another atom, which we do not check in our method, the protein may assume unnatural conformations, since in nature that atom will prevent the protein to assume that conformation. Also throughout the search, if any two atoms collide in their final position, the method goes back one step and continues with a different direction.

A better solution, which is the solution we adopted, is to search the space with random perturbations for a reasonable number of steps. This way, the time for search is reduced and most of the possible collisions can still be detected. This method still needs the perturbation amounts selected such that the path of any moving atom does not contain another atom.

Consider a 9 residue long protein chain. Assume all angles are planar initially. The maximum displacement of a residue at one step is achieved by rotating the $\phi$ or $\psi$ angle of the middle residue in the chain. The most significant displacement would be in the left-most atom of the left-end residue or the right-most atom of the right-end residue. Thus, the perturbation amount must be selected such that this residue cannot move past another atom (Figure 5.1). The displacement with respect to perturbation of an angle is illustrated in Figure 5.2. The maximum allowed perturbation angle can be calculated using Formula 5.1.

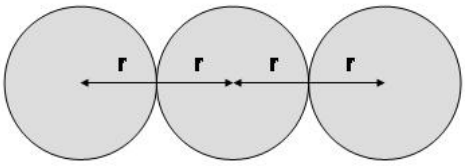


**Figure 5.1:** To move an atom past another one without detecting a collision, at least 4 times the radius of the atom should be traversed.

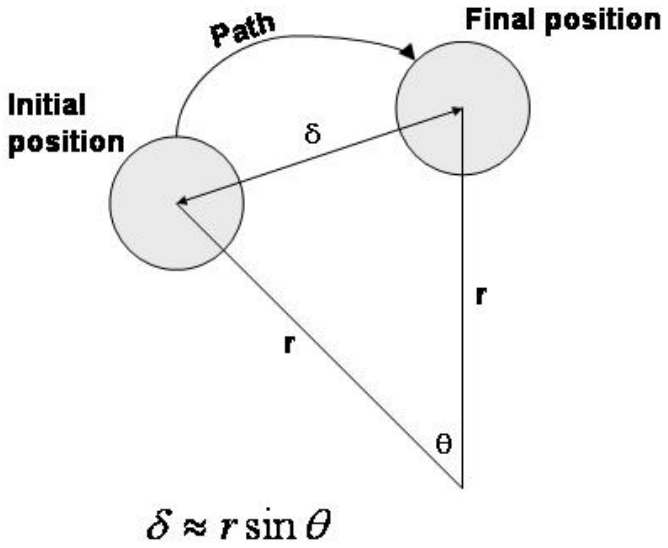


**Figure 5.2:** The displacement, $\delta$, with respect to an angle change of $\theta$ degrees.

$r$ is the distance of the atom from the residue in the middle of the chain. The formula shows the relation between the displacement and the perturbation angle for small $\theta$.

$$\sin\theta_{\max} = \frac{\sin\theta \times (4r_{\min})}{\delta}$$

**Formula 5.1:** Calculation of maximum allowed perturbation angle in a protein chain.

When one of the angles in the middle of a protein is perturbed θ degrees, the atoms in one end of the protein moves δ angstroms. Assuming that the other residues in the chain are rigid, the distance of the atom ($r$ in Figure 5.2) to the middle residue does not change. And since the maximum allowed displacement is $4r_{min}$ (Figure 5.1), where $r_{min}$ is the radius of a Hydrogen atom (0.79 Å) in our case, the formula follows.

Once the maximum allowed perturbation angle, $\theta_{max}$, is calculated, each angle is perturbed with an angle chosen randomly from the range ($-\theta_{max}$, $\theta_{max}$). During each pass through all of the angles in the protein, at each perturbation of an angle, every collision that occurs is recorded. When this process is repeated for a reasonably large number of times, possible atom pairs that can collide will be found.

During this part of the work, the Protein Molecule Library, libmol, by Deniz Yuret has been used to simulate protein behavior (in the code base, under '/3 – Steric Collisions'). Libmol is a C library that is used to represent a protein chain and to perturb it in various physical ways to observe how it behaves. The collision detection codes, as well as a sample set of these possible collisions for a number of different lengths of Alanine chains are given in appendix (in the code base, under '/3 – Steric Collisions').

### 5.1.3 Results

The possible atom pairs that can collide are detected. The detailed results are given in the appendix. Other than the possible atom pairs, we have checked the possible angle pairs. A Ramachandran plot for three cases is given in Figure 5.3. It can be clearly seen that as the chain size increases the flexibility of the angle decreases. This may stem from the spatial exclusion restrictions each residue adds to the chain. (Results are given in the code base, under '/3 – Steric Collisions').

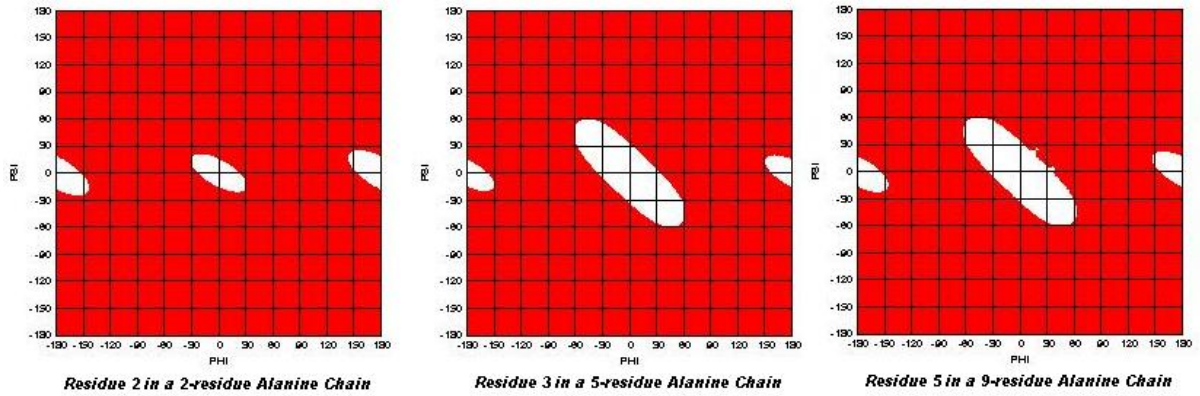


**Figure 5.3:** Ramachandran plots of short Alanine chains.

As the chain length increases the flexibility of the middle residues decreases (The plots show 2nd residue in a 2-residue chain, the 3rd residue in a 5-residue chain and the 5th residue in a 9-residue chain.) The shaded regions show the allowed conformations at 20 million steps of random perturbation of phi and psi angles. There are more allowed regions than Ramachandran most probably since the radii of residue atoms we have chosen for the model are smaller than Ramachandran has chosen which results in increased flexibility.

## *5.2 Code Base*

Every piece of code that has been used to generate the results stated throughout this work is present in the code base. Each folder and file has ample description in plain text format, allowing the replication of each of these results. Also some sample data (if not all) is presented in the code base so that immediate use of the code library is possible.

See the attached CD for the code base.

# 6. References


1. Protein, *Wikipedia, the Free Encyclopedia*. Retrieved August 28, 2005, from: **http://en.wikipedia.org/wiki/Protein.**
2. Selenocysteine, *Uniprot Knowledgebase*. Retrieved August 28, 2005, from: http://ca.expasy.org/cgi-bin/get-entries?KW=Selenocysteine.
3. Pyrrolysine, *Uniprot Knowledgebase*. Retrieved August 28, 2005, from: **http://ca.expasy.org/cgi-bin/get-entries?KW=Pyrrolysine.**
4. Protein Mass Spectrometry, *Mass Spectrometry Facility Home*. Retrieved August 28, 2005, from: **http://www2.musc.edu/Pharm/ms_one.html.**
5. Circular Dichroism Spectroscopy of Biomolecules, *Department of Chemistry at Rutgers University*. Retrieved August 28, 2005, from: **http://www.newark.rutgers.edu/chemistry/grad/chem585/lecture1.html.**
6. Nature of 3D Structural Data, *Research Collaboratory for Structural Bioinformatics*. Retrieved August 28, 2005, from **http://www.rcsb.org/pdb/experimental_methods.html.**
7. Jones, D. T., Threader, *Bioinformatics Unit, Department of Computer Science, University College*. Retrieved August 29, 2005, from http://bioinf.cs.ucl.ac.uk/threader/threader.html.
8. Rivest, R. L. (1987). Learning decision lists. *Machine Learning*, 2:229-246.
9. Webb, G. I. and Brkic, N. (1993). Learning decision lists by prepending inferred rules. In *Proceedings of the AI 93 Workshop on Machine Learning and Hybrid Systems*, pages 6-10, Melbourne.
10. Pauling, L., Corey, R. B and Branson, H. R (1951). The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain. In *Proceedings of National Academy of Sciences* (vol.37, is.4), pages 205-211, USA.
11. Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577-2637.
12. Cuff, J. A. and Barton, G.J. (1999). Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins*, 34: 508-519.
13. Heinig, M. and Frishman, D. (2004). STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins. *Nucleic Acids Research,* 32, web server issue.
14. Richards, F.M. and Kundrot, C.E. (1988). Identification of structural motifs from protein coordinate data: secondary structure and first-level super-secondary structure. *Proteins,* 3:71-84.
15. Karplus, P.A. (1996). Experimentally observed conformation-dependent geometry and hidden strain in proteins. *Protein Science,* 5:1406-1420.
16. Schulz, G.E. and Schirmer, R.H. (1979). In *Principles of protein structure*, Springer-Verlag, pages 1-314.
17. Rost, B., Sander, C. and Schneider, R. (1994). Redefining the goals of protein secondary structure prediction. *Journal of Molecular Biology,* 235:13-26.
18. Chothia, C. and Lesk, A.M. (1986). The relation between the divergence of sequence and structure protieins. *The EMBO Journal*, 5 (4):823-826.
19. Needleman, S.B. and Wunsch, C.D. (1970). A general method applicable to search for similarities in the amino acid sequence of two proteins.*Journal of Molecular Biology*, 48:443-453.
20. Rost, B. and Sander, C. (1993). Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology,* 232:584-599.

**21.** Kabsch, W. and Sander, C. (1983). How good are predictions of protein secondary structure? *FEBS Letters,* 155:179-182.

**22.** Cuff J. A., Clamp, M. E., Siddiqui, A. S., Finlay, M. and Barton, G. J. (1998). Jpred: A Consensus Secondary Structure Prediction Server. *Bioinformatics,* 14:892-893.

**23.** Kloczkowski, A., Ting, K.L., Jernigan, R.L. and Garnier, J. (2002). Combining the GORV algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence. *Proteins,* 49:154-166.

**24.** Salamov, A.A. and Solovyev V.V. (1995). Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *Journal of Molecular Biology,* 247:11-15.

**25.** Rost, B. and Sander, C. (2000). Third generation prediction of secondary structure. In Webster, D. (ed.), *Protein Structure Prediction: Methods and Protocols*, pages 71-95. Humana Press, Clifton, NJ.

**26.** Baldi, P., Brunak, S., Frasconi, P., Soda, G. and Pollastri, G. (1999). Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15:937-946.

**27.** Altschul, S. et al. (1997). Gapped blast and psi-blast: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3389-3402.

**28.** Rost, B. (2001). Review: Protein secondary structure prediction continues to rise. *Journal of Structural Biology*, 134:204-218.

**29.** Jones, D.T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292:195 -202.

**30.** Sander, C. and Schneider, R. (1991). Database of homology-derived structures and the structural meaning of sequence alignment. *Proteins,* 9:56-68.

**31.** Minsky, M. and Papert, S. (1988). *Perceptrons,* MIT Press, Cambridge MA.

**32.** Cuff, J.A. and Barton G.J. (2000). Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins,* 40:502-511.

**33.** Barton G.J. (1990). Protein multiple sequence alignment and flexible pattern matching. *Methods in Enzymology,* 183:403-428.

**34.** Thompson J.D., Higgins, D.G. and Gibson T.J. (1994). Clustal W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research,* 22:4673-4680.

**35.** Henikoff, S. and Henikoff, J.G. (1992). Amino acid substitution matrices from protein blocks. In *Procedings of National Academy of Sciences* (vol.89, is.22), pages 10915-10919, USA.

**36.** Moult, J., Hubbard, T., Bryant, S.H., Fidelis, K. and Pedersen, J.T. (1992).Critical assesment of methods of protein structure prediction (CASP): round II. *Proteins*, S1, 2-6.

**37.** Garnier, J., Gibrat J.F. and Robson, B. (1996). GOR method for predicting protein secondary structure from amino acid sequence. *Methods in Enzymology,* 266:540-553.

**38.** Clark, P. and Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, 3:261-283.

**39.** Newlands, D. and Webb, G. I. (2004). Alternative strategies for decision list construction. In *Proceedings of the Fourth Data Mining Conference (DM IV 03)*, pages 265-273.

**40.** Kyte, J. and Doolitle, R. (1982). A simple method for displaying the hydrophobic character of a protein. *Journal of Molecular Biology*, 157:105-132.

**41.** Zimmerman, J.M., Eliezer, N. and Simha, R. (1968). *Journal of Theoretical Biology*, 21:170-201.

**42.** Swiss-prot knowledgebase*, Swiss Institute of Bioinformatics.* Retrieved September 01, 2005, from: **http://www.expasy.org/sprot/**.

**43.** Hobohm, U. and Sander, C. (1994). Enlarged representative set of protein structures. *Protein Science,* 3:522-524.

**44.** Vriend, G. (1990). WHATIF: A molecular modeling and drug design program. *Journal of Molecular Graphics,* 8:52-56.

**45.** Protein Data Bank, *Research Collaboratory for Structural Bioinformatics*. Retrieved September 03, 2005, from **http://www.rcsb.org/pdb/.**

**46.** Duan, Y. and Kollman, P.A. (1998) Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science*, 282:740-744**.**

**47.** Pan, K.M. et al. (1993). Conversion of α-helices into β-sheets features in the formation of the scrapie prion proteins. In *Proceedings of National Academy of Sciences* (vol.90, is.23), pages 10962-10966, USA**.**

**48.** Zhang, Y. Automated assessment of CASP6 predictions., *University at Buffalo Center of Excellence in Bionformatics*. Retrieved September 03, 2005, from **http://www.bioinformatics.buffalo.edu/new_buffalo/people/zhang6/casp6/.**

**49.** Keskin, O., Yuret, D., Gursoy, A., Turkay, M. and Erman, B. (2004). Relationship between amino acid sequence and backbone torsion angle preferences. *Proteins*, 55:992-998**.**

**50.** Liu, H., Hussain, F., Tan, C.L. and Dash, M. (2002). Discretization: an enabling technique. *Data Mining and Knowledge Discovery*, 6 (4): 393-423**.**

**51.** Fan, H. and Mark, A.E. (2004). Refinement of homology-based protein structures by molecular dynamics simulation techniques. *Protein Science*, 13:211-220**.**

**52.** HHMI News: Rosetta may hold key to predicting protein folding. *Howard Hughes Medical Institute*. Retrieved September 04, 2005, from http://www.hhmi.org/news/baker.html.

**53.** Protein Structure Prediction Center. *Genome Center, University of California, Davis.* Retrieved September 04, 2005, from **http://www.predictioncenter.org**.

**54.** Ramachandran, G.N., Ramakrishnan, C. and Sasisekharan, V. (1963). Stereochemistry of polypeptide chain conformations. *Journal of Molecular Biology*, 7:95-99**.**

**55.** Kawashima, S., Ogata, H. and Kanehisa, M. (1999). AAIndex: amino acid index database. *Nucleic Acids Research*, 27:368-369**.**

**56.** Ramachandran, G.N. and Sasisekharan, V. (1968). Conformation of polypeptides and proteins. *Advances in Protein Chemistry*, 23:283-438**.**

**57.** Molecular Visualization Freeware, Microbiology Department of University of Massachusetts. Retrieved September 11, 2005, from http://www.umass.edu/microbio/rasmol/.

**58.** Creighton, T.E. (1992). *Proteins* (2nd ed.). W.H.Freeman, New York.

**59.** Martin, A.C.R. ProFit: Protein least-squares fitting, Bioinformatics group at University College London. Retrieved September 12, 2005, from http://www.bioinf.org.uk.

# Vita

I was born in Izmir, 1981. After attending a number of elementary schools, I have ranked exceptionally in nationwide exams; first to attend Icel Anatolian High School, a highly respected high school from where I graduated in 1992 and to Sabanci University from where I obtained my Bachelor of Science degree in Computer Science and Engineering in 2003. My graduation project was titled "Secure Communication over Bluetooth Personal Area Networks" and I have designed and implemented an application layer secure protocol. Later, I have decided to conduct a thesis study on structural bioinformatics, a subject at which I have always marveled. This booklet is a review of my studies on "Protein Structure Prediction using Decision Lists", as a result of which I obtained my Master of Science degree from Koc University in 2005. I can be reached anytime at volkan@su.sabanciuniv.edu.