UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

PROTEIN SECONDARY STRUCTURE PREDICTION USING AMINO ACID REGULARITIES

by

Frederick Petrus Senekal

Submitted in partial fulfilment of the requirements for the degree
Master of Engineering (Computer Engineering)
in the
Faculty of Engineering, the Built Environment and Information Technology
UNIVERSITY OF PRETORIA

Advisor: Professor E. Barnard

July 2008

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Protein secondary structure prediction using amino acid regularities

by

Frederick Petrus Senekal

Promotor:	Professor E. Barnard
Department:	Electrical, Electronic and Computer Engineering
Degree:	Master of Engineering (Computer Engineering)

SUMMARY

The protein folding problem is examined. Specifically, the problem of predicting protein secondary structure from the amino acid sequence is investigated. A literature study is presented into the protein folding process and the different techniques that currently exist to predict protein secondary structures. These techniques include the use of expert rules, statistics, information theory and various computational intelligence techniques, such as neural networks, nearest neighbour methods, Hidden Markov Models and Support Vector Machines.

A pattern recognition technique based on statistical analysis is developed to predict protein secondary structure from the amino acid sequence. The technique can be applied to any problem where an input pattern is associated with an output pattern and each element in both the input and output patterns can take its value from a set with finite cardinality. The technique is applied to discover the role that small sequences of amino acids play in the formation of protein secondary structures.

By applying the technique, a performance score of $Q_8 = 59.2\%$ is achieved, with a corresponding Q_3 score of 69.7%. This compares well with state of the art techniques, such as OSS-HMM and PSIPRED, which achieve Q_3 scores of 67.9% and 66.8% respectively, when predictions on single sequences are made.

KEYWORDS

protein, amino acid, secondary structure, bioinformatics, pattern recognition, protein folding problem, protein secondary structure prediction, amino acid sequence, classification, neural network

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Voorspelling van proteïen sekondêre struktuur deur aminosuur-reëlmatigheid

deur

Frederick Petrus Senekal

Promotor:	Professor E. Barnard
Departement:	Elektriese, Elektroniese en Rekenaar-Ingenieurswese
Graad:	Meester van Ingenieurswese (Rekenaar-Ingenieurswese)

OPSOMMING

Die probleem van hoe proteïne vou word ondersoek. Daar word in besonder gekyk na hoe om die sekondêre struktuur van 'n proteïen te voorspel, gegee die aminosuur sekwensie van die proteïen. 'n Literatuurstudie word voorgelê oor die proses van proteïenvouing en die tegnieke wat bestaan om proteïen sekondêre strukture mee te voorspel. Tegnieke soos heuristieke, statistiek, inligtingsteorie en kunsmatige intelligensie word gebruik. Die kunsmatige intelligensie tegnieke sluit in neurale netwerke, "nearest neighbour" metodes, "Hidden Markov Models" en "Support Vector Machines."

'n Patroonherkenningstegniek word onwikkel om proteïen sekondêre struktuur te voorspel, gegee die aminosuur sekwensie van die proteïen. Die tegniek is geskool op statistiese analise en is van toepassing op enige probleem waar 'n insetpatroon assosieer word met 'n uitsetpatroon en elke element in beide die inset- en uitsetpatroon uit 'n eindige versameling gekies word. Die tegniek word aangewend om die rol wat klein aminosuur sekwensies speel in die formasie van proteïen sekondêre strukture te bepaal.

'n Doeltreffendheidsvlak van $Q_8 = 59.2\%$ word behaal deur die tegniek uit te voer. Die ooreenskomstige Q_3 waarde is 69.7%. Dit vergelyk goed met van die beste bestaande tegnieke, soos OSS-HMM en PSIPRED wat onderskeidelik Q_3 waardes van 67.9% en 66.8% behaal op die voorspelling van enkel sekwensies.

SLEUTELWOORDE

proteïen, aminosuur, sekondêre struktuur, bio-informatika, patroonherkenning, proteïen-vouingsprobleem, proteïen sekondêre struktuur voorspelling, aminosuur sekwensie, klassifikasie, neurale netwerk

ACKNOWLEDGEMENTS

This research was conducted over a period of four years. This was a learning experience for me, one that took a fair amount of reading, researching, experimenting, thinking, writing and lively discussions. I would like to express my gratitude to a number of people who stood by me during this time:

- My promotor, Professor Barnard, for his guidance and the considerable amount of time he invested in me.
- My wife, Adéle, for her love, continued support and inspiration.
- My parents, Fred and Liza, who through their love, provided me with opportunities in life, a good education, and a value system to strive for.
- Gerard Finnie and Ajith Gopal who proofread my dissertation and provided me with valuable feedback, suggestions and corrections.
- The CSIR, for their financial contribution towards my Master of Engineering degree.
- Our Heavenly Father, for the talents He gave me. *Soli deo gloria!*

Contents

Abbreviations

3D	Three-dimensional
ANN	Artificial Neural Network
BLAST	Basic Local Alignment Search Tool
BLOSUM	Blocks Amino Acid Substitution Matrices
BPTI	Bovine Pancreatic Trypsin Inhibitor
DNA	Deoxyribonucleic Acid
DSSP	Dictionary of Protein Secondary Structure
GOR	Garnier-Osguthorpe-Robson (Method)
HMM	Hidden Markov Model
HMM-STR	HMM for Sequence-Structure Correlations
HSP	High-Scoring Pair
OSS-HMM	Optimal Secondary Structure Hidden Markov Model
NMR	Nuclear Magnetic Crystallography
NNSSP	Nearest Neighbour Secondary Structure Prediction
PDB	Protein Data Bank
PHD	Profile Network from Heidelberg
PROF	Profile-based Neural Network Prediction
PSI-BLAST	Position Specific Iterated BLAST
PSIPRED	Position Specific Iterated Predict Secondary Structure
PSSM	Position-specific Scoring Matrices
RNA	Ribonucleic Acid
SOV	Segment Overlap (Score)
SVM	Support Vector Machine

Chapter 1

INTRODUCTION

Life is one of the greatest mysteries in the universe. It seems to possess a beauty and complexity that can only be appreciated by living things themselves.

Through the centuries, man has tried to understand this mystery, a mystery that would explain his own origin. He has asked inquisitive questions, questions that life itself has weaved into the very fabric of his existence. In his quest for understanding, he has turned to religion, with the hope of understanding the greatness of life. He has philosophized greatly about the meaning of life, trying to make sense of it all. And now, through the scientific and engineering tools available to him, he has made great discoveries about the intricate details of life, which fuels the hope that many more discoveries will still be made.

With the recent completion of the human genome project, man is one step closer in understanding his origin. For the first time in human history, the blueprint of the human race is available - it is now up to scientists and engineers to analyse and interpret its meaning.

As a consequence of the human genome project, we now know of the existence of a large number of proteins as well as the sequence of amino acids from which they are composed. What remains unknown is the function of the majority of these proteins. The function of a protein is mostly determined by its three dimensional structure.

The amazing thing is that given a specific sequence of amino acids, there is a seemingly infinite number of three dimensional structures that can be created; however, a protein will almost always fold into the same three dimensional structure! Life on earth has the ability to manufacture proteins that are always the same.

The central question addressed in this dissertation, is one that investigates the way in which amino acids contribute to protein structure, which in turn determines the function of a protein. By understanding these assembly units of life on earth, we will gain insight into evolution, the functioning of the body and perhaps most importantly, we will be in a better position to develop treatments and cures for certain diseases.

I invite you now on a scientific journey that aims to discover the exciting principles that underlies the foundations of life. It is only once we understand how life functions, that we will be in a position to touch on the greatest mystery of all - the reason there is life...

1.1 BACKGROUND

Proteins are organic macromolecules that are essential for the structure, function and regulation of the body's cells, tissues and organs. They are composed of a linear sequence of amino acids linked together by peptide bonds to form a polypeptide. This sequence of amino acids, without regard to spatial arrangement, is known as the primary structure of the protein.

There are 20 different types of commonly occurring amino acids in proteins. Each amino acid is composed of a central carbon atom (known as the C_α atom), attached to a hydrogen atom (H), an amino group (NH_2), a carboxyl group (COOH) and a side chain, also known as a residue (R). This residue can range from a single hydrogen atom in the case of the amino acid glycine, to a compound of 19 atoms in the case of the amino acid arginine. It is this residue that gives each amino acid its unique properties.

Two amino acids can link together via a peptide bond, a reaction in which the amino group of one amino acid reacts with the carboxyl group of another amino acid. A water molecule is released as a by-product of the reaction. Of course, multiple amino acids can link together in the same way to form a polypeptide. This polypeptide has a repeating backbone structure of N-C$_\alpha$-C atoms (known as the main-chain atoms) all linked together by covalent bonds.

The local spatial arrangement of the main-chain atoms of a segment of a polypeptide chain is known as its secondary structure. This definition disregards the conformation of side chains or the relationship with other segments. Regular patterns have been observed in this spatial arrangement. For instance, alpha helices and beta sheets are secondary structure patterns frequently observed in a polypeptide chain. Within these structures, hydrogen bonds between the amino acids at regular intervals within the chain add to the stability of the structure.

The tertiary (or three-dimensional) structure of a protein, is the arrangement of all its atoms in space. The amazing thing about proteins is that for a specific primary structure, there is almost always a single associated tertiary structure in its native state. Research has shown that there is a strong correlation between the tertiary structure of a protein and its function. For instance, hemoglobin, the protein that carries oxygen in the body, has a specific globular shape that is able to trap oxygen. Another protein, collagen, has a rod-like form and is commonly found in cells. This rod-like feature gives form and stability to cells. It is reasonable to assume that proteins with similar structures are likely to have similar functions.

With the completion of the human genome project, it is now known that there are about 20000 to 25000 different human proteins (one study suggests that 19599 protein-coding genes have been identified and another 2188 DNA segments are predicted to be protein-coding genes [25]). For each of these proteins, the primary structure is known. However, the tertiary structure and function of the majority of these proteins are currently unknown.

Scientists are faced with the challenge to predict the tertiary structure of a protein in its functional environment from its known primary structure in order to determine the possible function of the protein. This is known as the “protein-folding problem” and is an active research field.

Current research focuses on predicting the secondary structures that form from sequences of amino acids and how these secondary structures combine to form the tertiary structure of a protein. Different approaches have been applied to the problem of protein secondary structure prediction. These approaches include use of statistics and expert rules, information theory and computational intelligence techniques. The bulk of the methods are in the domain of computational intelligence. These techniques include neural networks, nearest neighbour methods, Hidden Markov Models and Support Vector Machines.

1.2 MOTIVATION

The protein folding problem is one of the central unanswered questions in biology. It has been studied by many, yet the exact mechanisms involved remain elusive.

Apart from the intellectual quest, an understanding of the protein folding process is of significant practical importance. Diseases such as cystic fibrosis, Bovine spongiform encephalopathy (mad cow disease) and its human counterpart (Creutzfeldt-Jacob disease) and certain strains of Alzheimer's disease are now known to be caused by proteins that fold incorrectly. If the process is better understood, it may become possible to manufacture drugs to treat these diseases. Insights into the process will also lead to a valuable understanding of evolution. The folding process provides insight into the way different proteins are related, making it possible to trace the evolutionary paths of proteins and enabling a taxonomy of organisms to be created. Other areas, such as that of food manufacturing and preservation will also benefit from a better understanding of the protein folding process.

One of the key areas of research into protein folding is predicting protein secondary structure from the amino acid sequence. Secondary structure prediction techniques have improved considerably during the last 20 to 30 years. The reason for this improvement is twofold: the employment of advanced computational intelligence techniques and the availability of larger databases of solved protein structures (that serve as training examples to the computational intelligence techniques).

Depending on one's viewpoint, it may be argued that the availability of advanced techniques and a large amount of data does not contribute to the understanding of the protein folding process *per se*. The better prediction accuracy is not an indication of a better understanding of the protein folding process, but an indication of the ability of computational intelligence techniques to capture the mapping between the primary and secondary structure of a protein. What would contribute to the understanding of the protein folding process, is if the fundamental rules or mapping could be extracted from the computational intelligence techniques.

Others argue that the protein folding processes are well understood. Indeed, detailed simulations of the underlying physics and chemistry exist (refer to Section 2.2.3 on protein folding simulation). Although the simulations take an immense amount of time, they very accurately simulate the actual folding process. However, these low-level descriptions provide little by way of intuitive understanding, just as a quantum-mechanical description of doped silicon is not suitable to give insight into the operation of, for example, a microprocessor.

It is the author's viewpoint that research and scientific discovery is after all a human activity. It is not only the end destination that matters, but also the journey taken to get there. Although the final (simulated) protein structure is important, it is in human nature to *want* to understand the fundamental principles involved. Such an understanding is crucial for both synthesis and high-level analysis.

A description of such understandable principles is to some extent lacking in the advanced computational intelligence techniques. The aim of the dissertation is to make a contribution to this understanding.

1.3 OBJECTIVES

This dissertation aims to be a thorough investigation into the contributions of single amino acids or small sequences of amino acids to protein secondary structure.

Specific research questions that will be addressed include:
- Do certain amino acid sequences have a preference to form specific secondary structures?
- Could certain amino acid sequences serve as substitutes for other amino acid se-

quences (i.e. could one sequence be substituted with another whilst maintaining the same secondary structure)?
- What properties of amino acid sequences contribute to the formation of secondary structures and how should these properties be used in developing a method for secondary structure prediction?

Methods will be developed to answer these questions and will be implemented as computer programs capable of predicting protein secondary structure from their sequence.

1.4 CONTRIBUTION

The dissertation contributes through the development of a new protein secondary structure predication algorithm. The predication algorithm achieves a performance value of $Q_8 = 59.2\%$, with corresponding Q_3 value of 69.7% (these measures are defined in Section 2.4.1). This is comparable to performance values achieved using current state of the art techniques, such as OSS-HMM and PSIPRED which achieve Q_3 scores of 67.9% and 66.8% respectively, when predictions on single sequences are made. Through additional work, the algorithm can be further developed and it is believed that even better performance can be achieved.

The algorithm in itself can also be applied to a broader range of applications. In particular, pattern recognition problems where there exist a mapping between input sequences and output sequences, where each element in the input and output sequences are from a finite set, can benefit from this algorithm.

The algorithm is applied, together with other tests, to discover the role that small sequences of amino acids play in the formation of protein secondary structures. A number of key findings are made and are described in Section 6.1.

1.5 OVERVIEW

Chapter 2 gives comprehensive background information on proteins, amino acids, peptide bonds, etc. An understanding of the concepts and terminology introduced in this section is fundamental in understanding the rest of the dissertation. Readers new to the field of bioinformatics are encouraged to read through this chapter, whilst those more familiar with the field may choose to browse through it.

Chapter 3 describes the pattern recognition algorithm that was developed to predict protein secondary structure from protein primary structure. It should be noted that the pattern recognition algorithm can be applied to a broader range of problems, namely those problems which are structured in such a way that the input and output sequences are defined over two possibly different alphabets. The chapter is supplemented with a detailed example.

Chapter 4 describes the pattern recognition algorithm mathematically. It also formalises the way in which some of the other results were obtained.

The results achieved with the algorithm as well as the results of a number of other tests are presented in chapter 5. The chapter is broken down into a number of experiments, each of which describe the objective of the experiment, the setup and execution, the results obtained and relevant conclusions reached.

The dissertation is concluded in chapter 6.

The proteins that the research is based on are listed in appendix A.

Chapter 2

BACKGROUND

This chapter provides background information on proteins, their structure and the protein folding process. It discusses ways of classifying protein secondary structure and describes the methods that exist to predict secondary structure. Comprehension of these concepts is necessary for understanding the rest of the dissertation.

Section 2.1 gives an overview of proteins and how they are constructed from amino acids through peptide bonds. It also describes the genetic code and how proteins are synthesized. In Section 2.2 the protein folding process is discussed. Of particular interest are the regular local structures that are formed during the folding process, called secondary structures. The section also discusses different theories that exist to describe the protein folding process. Section 2.3 describes the formation of protein secondary structure and introduces the DSSP code for classifying secondary structures. The chapter is concluded in Section 2.4 which introduces the methods currently in use to predict secondary structures as well as the measures of performance that are used to quantify their success.

2.1 PROTEINS

2.1.1 Brief Historical Overview

Proteins are organic macromolecules essential for the structure, function and regulation of the body's cells, tissues and organs. They are composed of a linear sequence of amino acids linked together by peptide bonds to form a polypeptide. Although these facts are now widely known, it is useful to understand how these concepts came into existence.

Up until the early nineteen hundreds, scientists described animal and vegetable materials in terms of the general properties they possessed. By 1815, it was known that animal and plant materials are composed of the elements carbon (C), hydrogen (H), oxygen (O) and nitrogen (N). Methods based on the oxidation of materials were developed by Jöns Jakob Berzelius in Stockholm and Joseph Louis Gay-Lussac in Paris to determine the relative quantities of C, H, O and N in organic materials.

In 1820, Henri Braconnot was studying the effect of sulfuric acid on animal substances. When applied to gelatin, it would yield what he called "gelatin sugar" which was later renamed as glycine. When applied to muscle fibres and wool, it would yield a white substance he named leucine. Glycine and leucine were the first two amino acids to be discovered. At the time, it was not known that these were the essential building blocks of proteins. The term "amino acid" was only proposed that same year by Berzelius for nitrogen-containing organic acids. The discovery of the other amino acids which naturally occur in proteins (proteinogenic amino acids) continued from 1849 when tyrosine was discovered, to 1936 with the discovery of threonine.

In a paper [1] by Gerardus Johannes Mulder in 1839, he described the chemical composition of some substances, and was the first to use the term "protein" to describe these substances. He states that this term was a suggestion by Berzelius from a letter dated 1838. In the period that followed, more amino acids were discovered and proteins were characterized in terms of the amino acids they are composed of. As early as 1872, Karl Ritthausen (who also discovered glutamic acid and aspartic acid), published a book [2] which analyzed the three main types of protein contained in cereals, legumes and oilseeds in terms of amino acid composition.

The next great advance came on 22 September 1902, at the 74th Annual Meeting of the Gesellschaft der Deutschen Naturforschen und Ärzte (Society of German Naturalists and Physicians). At the meeting, Franz Hofmeister [3] and Hermann Emil Fischer [4] independently suggested that amino acids link with each other via peptide bonds to form a polypeptide. Fischer won the 1902 Nobel Prize in Chemistry for his work on sugar and purine synthesis.

The polypeptide theory became widely accepted and the question now naturally arose as to which amino acids existed and how a protein could be characterized in terms of amino acids. In 1941, Hubert Bradford Vickery published a paper [5] in which he grouped the amino acids into four groups. One of these groups contained 18 amino acids, 17 of which were proteinogenic.

In 1942, Archer John Porter Martin and Richard Laurence Millington Synge invented partition chromatography [6] (for which they received the Nobel Prize in Chemistry in 1952). This brought about a revolution in the task of decomposition of proteins into amino acids. It enabled Synge to draw up a list of amino acids [7]. Later column-chromatographic methods were invented by Moore and Stein (Nobel Prize for Chemistry, 1972), which made the complete automation of decomposition of proteins into amino acids possible.

The challenge now turned to determining the amino acid sequence (not just composition) of a protein. Frederick Sanger managed to identify the N-terminal of proteins by the formation of dinitrophenyl derivatives and succeeded to identify the sequence of amino acids and disulfide bonds in insulin [8]. This breakthrough earned him the 1958 Nobel Prize for Chemistry and was significant in that it proved the polypeptide theory of Hofmeister and Fischer.

The next big breakthrough came in the determination of the three dimensional structure of proteins through the X-ray study of protein crystals. In 1959 Max Ferdinand Perutz managed to determine the molecular structure of hemoglobin [9] and John Cowdery Kendrew managed to determine the structure of myoglobin [10]. They received the 1962 Nobel Prize for their work.

During the same decade, in the period from 1951 to 1953, James Dewey Watson and

Francis Harry Compton Crick discovered the double-helical structure of DNA [11] (which earned them the Nobel Prize for Physiology or Medicine in 1962, shared with Maurice Wilkins). The publication of the discovery in Nature magazine in 1953, led George Gamow to the idea that perhaps the nucleotides in the DNA structure could serve as instructions on how to manufacture proteins [12].

Gamow's theory turned out to be correct. It is now known that sections of the DNA strand are transcribed to an RNA strand. Sections of the RNA sequence (known as codons) are then translated to amino acids through what is known as the "genetic code". This process, whereby DNA is used as the blueprint to manufacture proteins, is known as the "central dogma of molecular biology".

In 1961, Marshall Warren Nirenberg and Heinrich J Matthaei performed the Nirenberg-Matthaei experiment that would be the first step in the determination of the genetic code [13], [14]. Their work was supplemented by the Nirenberg-Leder experiment and later by work of Har Gobind Khorana [15]. Through their work, they determined the correspondence between codons and the amino acids they code - the genetic code was solved. Nirenberg and Khorana (together with Robert W Holley) received the 1968 Nobel Prize in Physiology or Medicine for their work. The establishment of the genetic code also meant that it was now known that only 20 amino acids were naturally manufactured through the process of translation.

In 1976 Frederick Sanger and Walter Gilbert independently developed methods for determining nucleic acids base sequences in DNA. Sanger used his method, known as the chain or dideoxy termination method [18], to sequence the genome of the Phage Φ-X174 in 1977 [19] [20], the first fully sequenced genome. Sanger and Gilbert (together with Paul Berg) received the 1980 Nobel Prize for Chemistry for their efforts.

The methods developed by Sanger and Gilbert made it possible to automate the process of determining base sequences in DNA. This led to the establishment of the Human Genome Project in 1986 [21]. The objective of the project is to map and sequence the estimated 2.85 billion (2851330913 according to [23]) nucleotides in the human genome and to identify the genes present in it. It was headed by James Watson from 1988 and initially 16 institutions from 5 countries participated.

In 1995 the entire 1.8 million base pairs of the bacterium Haemophilus influenzae was published [17]. On 26 June 2000, it was jointly announced by Bill Clinton and Tony Blair that an initial working draft of the entire human genome was finished. The working draft was published in 2001 and made freely available [22]. A major milestone was reached in May 2006, when the sequence of the final chromosome of the human genome was published in the journal Nature [24]. It is also of significance that there are an estimated 20000 to 25000 protein-encoding genes in the human genome [23]. Although the exact number is not known, 19599 protein-coding genes have been identified and another 2188 DNA segments are predicted to be protein-coding genes [25].

The human genome project has made major contributions to the understanding of the biological principles that underpin life. Research is under way to identify genes and the proteins they encode. However, a protein's function is not directly determined through its amino acid composition; its three dimensional structure is a more appropriate framework for understanding functionality. The majority of proteins' three dimensional structure continue to be determined through X-ray crystallography. A smaller percentage of structures are also determined through nuclear magnetic resonance (NMR) and mass spectrometry. These methods are however laborious and expensive and new techniques are sought to determine or predict the 3D structure of proteins. The structures are shared through internet resources such as the Protein Data Bank (PDB) [114]. In July 2006, the PDB contained 34577 protein structures of various organisms.

New discoveries continue to be made. In 1986 selenocysteine and in 2002 pyrrolysine were discovered. These are coded from the stop codons UGA and UAG (refer to Section 2.1.4) respectively of some organisms.

2.1.2 Amino Acids

In chemistry, an amino acid is any molecule that contains both an amino and carboxyl functional group. In biochemistry however, the term amino acid is often used to mean alpha amino acid - a molecule where the amino and carboxyl functional groups are attached to the same carbon atom. For the remainder of this dissertation, the term amino acid will be used to refer to alpha amino acids.

Figure 2.1 shows the structure of an alpha amino acid. Each amino acid is composed of a central carbon atom (known as the C_α atom), attached to a hydrogen atom (H), an amino group (NH_2), a carboxyl group (COOH) and a side chain, also known as a residue (R). All the atoms in an amino acid are attached by covalent bonds.

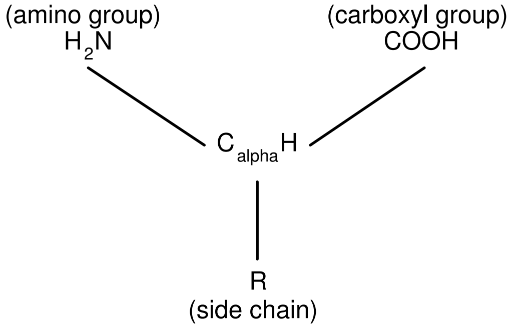


Figure 2.1: Structure of a Single Amino Acid

The residue can vary from a single hydrogen atom (in the case of amino acid glycine), to a large compound of different atoms (for instance arginine contains 19 atoms in its residue). It is this residue that gives each amino acid its unique properties. In nature, only 20 different amino acids (i.e. 20 different residues) are used to synthesize proteins. These are known as the proteinogenic or standard amino acids, and are listed in Table 2.1. Each of the proteinogenic amino acids contain carbon (C), hydrogen (H), oxygen (O) and nitrogen (N), whilst some (cysteine and methionine) also contain a sulphur (S) atom in their residue chains.

A large number of other non-standard amino acids also exist in nature or can be synthesized through artificial processes. Of note are selenocysteine and pyrrolysine, two amino acids that are sometimes manufactured by some organisms. Other amino acids, such as hydroxyproline, norvaline and hydroxylysine also sometimes occur. These are manufactured though a process known as post-translational modification, i.e. modification of the amino acid chain after translation (protein synthesis).

From a geometrical point of view, all amino acids have four different groups attached to the C_α atom. These groups can can be attached in two different configurations, known as the levo (L) and dextro (D) configurations. These two configurations are optical isomers of each other, meaning that they are non-superimposable mirror images of each other. Figure 2.2 illustrates the two different isomers (imagine looking down onto the

Table 2.1: The 20 Proteinogenic Amino Acids

Amino Acid	**Abbreviation**	**Linear Structure Formula**
Alanine	ala or a	CH_3-CH(NH_2)-COOH
Arginine	arg or r	HN=C(NH_2)-NH-(CH_2)$_3$-CH(NH_2)-COOH
Asparagine	asn or n	H_2-CO-CH_2-CH(NH_2)-COOH
Aspartic Acid	asp or d	HOOC-CH_2-CH(NH_2)-COOH
Cysteine	cys or c	HS-CH_2-CH(NH_2)-COOH
Glutamine	gln or q	H_2N-CO-(CH_2)$_2$-CH(NH_2)-COOH
Glutamic Acid	glu or e	HOOC-(CH_2)$_2$-CH(NH_2)-COOH
Glycine	gly or g	NH_2-CH_2-COOH
Histidine	his or h	NH-CH=N-CH=C-CH_2-CH(NH_2)-COOH
Isoleucine	ile or i	CH_3-CH_2-CH(CH_3)-CH(NH_2)-COOH
Leucine	leu or l	(CH_3)$_2$-CH-CH_2-CH(NH_2)-COOH
Lysine	lys or k	H_2N-(CH_2)$_4$-CH(NH_2)-COOH
Methionine	met or m	CH_3-S-(CH_2)$_2$-CH(NH_2)-COOH
Phenylalanine	phe or f	Ph-CH_2-CH(NH_2)-COOH
Proline	pro or p	NH-(CH_2)$_3$-CH-COOH
Serine	ser or s	HO-CH_2-CH(NH_2)-COOH
Threonine	thr or t	CH_3-CH(OH)-CH(NH_2)-COOH
Tryptophan	trp or w	Ph-NH-CH=C-CH_2-CH(NH_2)-COOH
Tyrosine	tyr or y	HO-p-Ph-CH_2-CH_2-CH(NH_2)-COOH
Valine	val or v	(CH_3)$_2$-CH-CH(NH_2)-COOH

C_α atom with the H atom closest to you).

The standard amino acids are mostly found in the levo configuration. The dextro configuration has been found in some sea-dwelling creatures and in the cell walls of some bacteria. A useful way of remembering the levo configuration is by means of the CORN rule (suggested by Richardson [51]). When looking at the C_α atom with the H atom closest to you, the other functional groups spell CORN ($\underline{\text{CO}}$OH - $\underline{\text{R}}$ - $\underline{\text{N}}H_2$) when read clockwise. Note that in the case of glycine, where the residue is a single H atom, two of the groups attached to the C_α atom are identical and therefor the levo and dextro configurations are the same.

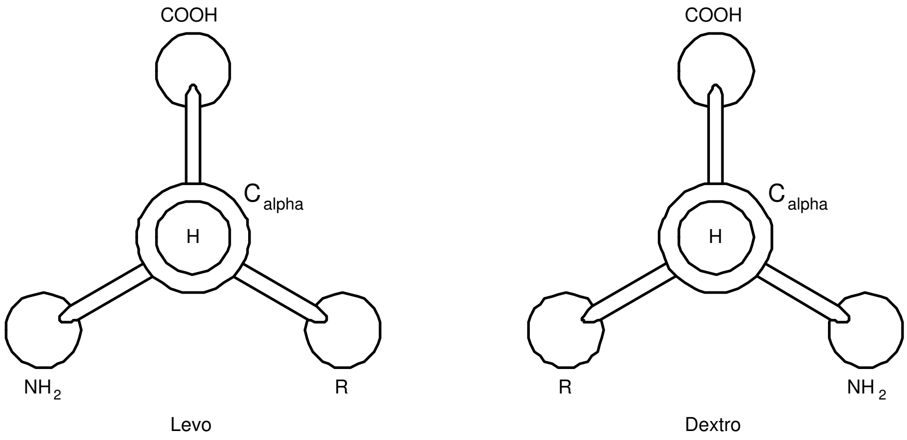


Figure 2.2: Levo and Dextro Configurations

2.1.3 The Peptide Bond

Two amino acids can “link” together through the formation of a peptide bond. The amino group of one amino acid reacts with the carboxyl group of the next amino acid as illustrated in Figure 2.3. In the process, a water molecule is released (i.e. dehydration synthesis). The resulting peptide bond is a strong covalent bond.

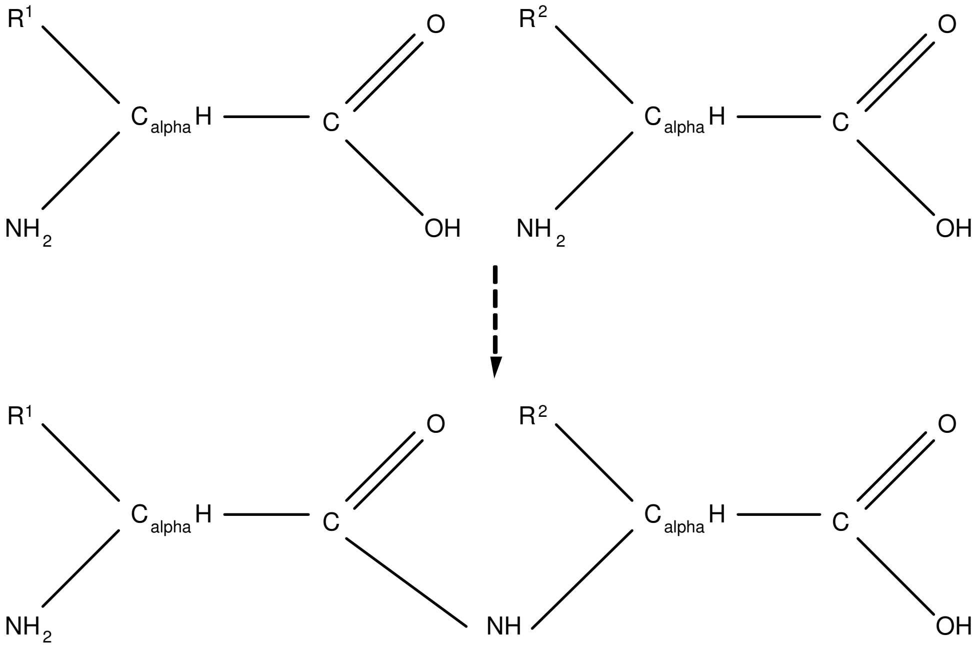


Figure 2.3: Formation of the Peptide Bond

Multiple amino acids can link together in the same way to form a polypeptide. In this polypeptide, there is always an uncomplexed ("free") amino group at the one end (known as the N-terminus) and an uncomplexed carboxyl group at the other (known as the C-terminus). By convention, the amino group indicates the start of the chain and the carboxyl group the end. The acute reader will note that the "backbone" of the polypeptide chain is formed by a repeating sequence of N-C_α-C atoms. These atoms are known as the main-chain atoms.

A dipeptide contains two amino acids and a tripeptide three. The terms peptide, polypeptide and oligopeptide are roughly equivalent, although peptide and oligopeptide are sometimes used in conjunction with "smaller" sequences and polypeptide with "larger" sequences of amino acids.

During dehydration synthesis, a water molecule is released to form a peptide bond between two amino acids. This process can be reversed through a process known as hydrolysis. Through the addition of a water molecule, the peptide bond can be broken and amino acids separated.

It is interesting to note that the six atoms from one C_α atom to the next C_α atom (C^i_α, CO^i, NH^{i+1} and C^{i+1}_α) all lie in a plane as illustrated in Figure 2.4. This is due to the double bond character of the peptide bond. The backbone N-C_α-C angle, τ, as well as the dihedral angles, ϕ around the N-C_α bond, ψ around the C_α-C bond and ω around the C-N bond are shown as well.

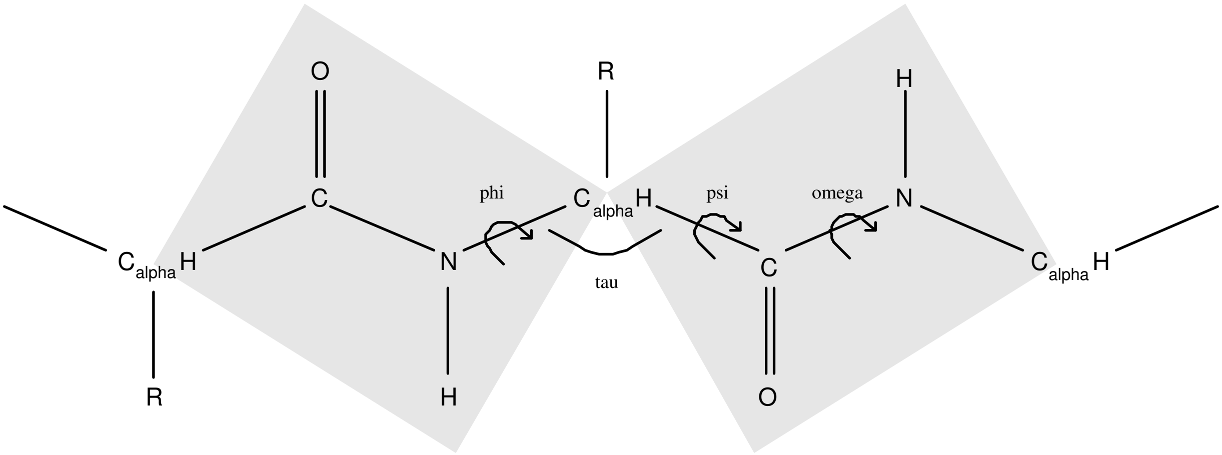


Figure 2.4: Bond Angles

Since the C_α atom is tetrahedral, τ is about 109.5°, although it has been noted that this angle can change to accommodate other strains in the structure. The peptide bond is almost always found in the trans configuration, implying that ω is 180°, although it is sometimes found in proline with the cis configuration. Figure 2.5 illustrates the difference between the trans and cis configurations. The source of essentially all the interesting variability in protein conformation are the ϕ and ψ angles. Although there is much freedom as to the values these angles can take, they are constrained geometrically by the amino acid residues and other factors. The distribution of these two angles for the amino acids in a particular protein is often plotted on a graph called a Ramachandran plot [16].

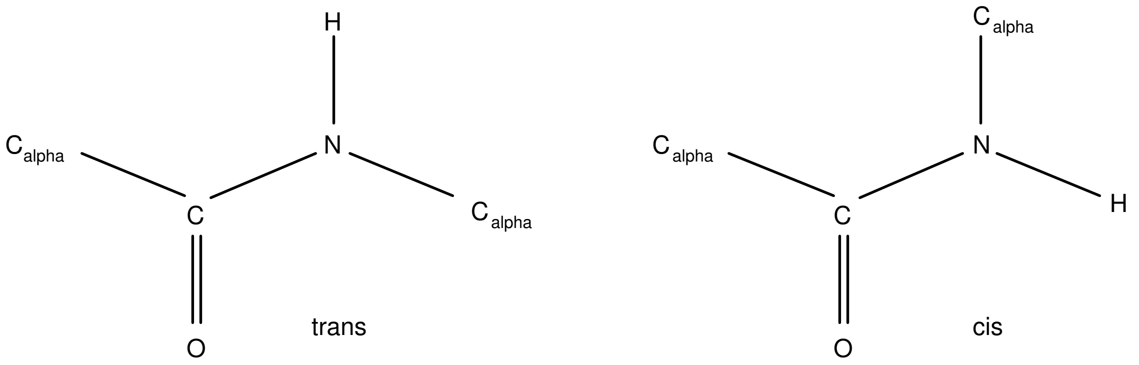


Figure 2.5: Trans and Cis Configurations

2.1.4 Protein Synthesis

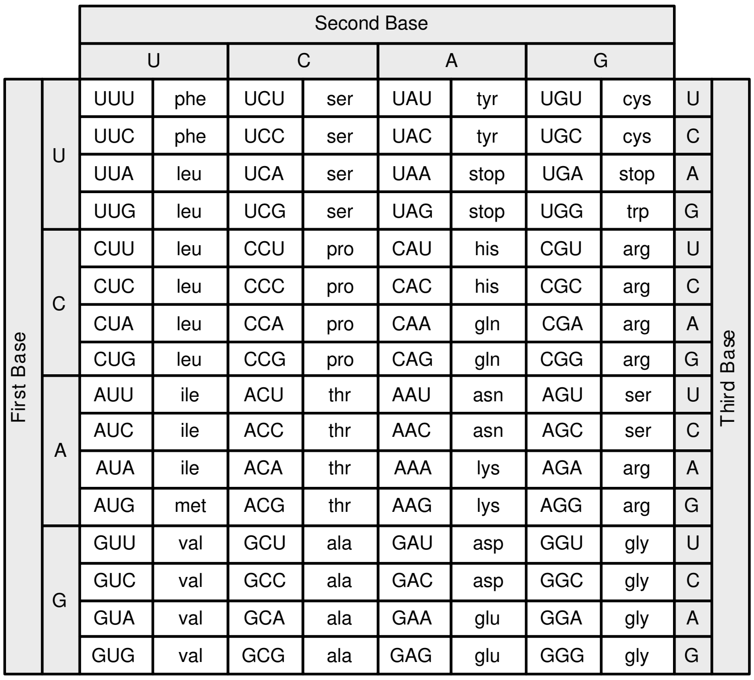
Proteins are manufactured in the ribosomes. The processes that play a role are transcription and translation.

The instructions to manufacture proteins are contained in the deoxyribonucleic acid (DNA) of an organism. A DNA strand is not a single molecule, but rather two molecule strands which are linked together through hydrogen bonds. Each strand is made up of a long sequence of nucleotides. There are four types of nucleotides or bases: adenine (A), cytosine (C), guanine (G) and thymine (T). Between the two strands of DNA, different bases pair up with each other: A with T and C with G. Note that the two strands are aligned, i.e. consecutive bases pair up with one another. This implies that a single strand contains all the information of the whole DNA molecule, or put differently, that one of the strands could be manufactured from knowledge of the other. DNA strands are tightly pack around proteins. This packaging is known as a chromosome. Human

DNA is packed into 46 chromosomes - two sets of 23.

The term genome refers to all the hereditary information contained in the DNA (both genes and non-coding regions). A gene is a section of a DNA strand that will code for a specific protein. A messenger ribonucleic acid (mRNA) strand is constructed from the part of the DNA strand where the gene is located. Adenine in DNA codes for uracil (U) in RNA, cytosine for guanine, guanine for cytosine and thymine for adenine. The constructed mRNA then travels from the nucleus where the DNA is contained to the ribosomes in the cytoplasm. This process, whereby a mRNA molecule is created, is known as transcription.

In the ribosome, each sequence of three nucleotides in the mRNA is interpreted as an instruction (known as a codon) to manufacture a specific type of amino acid. The process by which this takes place is known as translation. The pairing between codons (of which there can be $4^3 = 64$) and the 20 amino acids is known as the genetic code and is illustrated in Figure 2.6.



		Second Base								
First Base		U		C		A		G		Third Base
	U	UUU	phe	UCU	ser	UAU	tyr	UGU	cys	U
		UUC	phe	UCC	ser	UAC	tyr	UGC	cys	C
		UUA	leu	UCA	ser	UAA	stop	UGA	stop	A
		UUG	leu	UCG	ser	UAG	stop	UGG	trp	G
	C	CUU	leu	CCU	pro	CAU	his	CGU	arg	U
		CUC	leu	CCC	pro	CAC	his	CGC	arg	C
		CUA	leu	CCA	pro	CAA	gln	CGA	arg	A
		CUG	leu	CCG	pro	CAG	gln	CGG	arg	G
	A	AUU	ile	ACU	thr	AAU	asn	AGU	ser	U
		AUC	ile	ACC	thr	AAC	asn	AGC	ser	C
		AUA	ile	ACA	thr	AAA	lys	AGA	arg	A
		AUG	met	ACG	thr	AAG	lys	AGG	arg	G
	G	GUU	val	GCU	ala	GAU	asp	GGU	gly	U
		GUC	val	GCC	ala	GAC	asp	GGC	gly	C
		GUA	val	GCA	ala	GAA	glu	GGA	gly	A
		GUG	val	GCG	ala	GAG	glu	GGG	gly	G

Figure 2.6: The Genetic Code

The construction of a protein is started when the codon AUG appears in the mRNA sequence. AUG codes for the amino acid methionine. Construction of a protein is stopped when one of the codons, UAA, UAG or UGA is found in the mRNA sequence.

2.2 PROTEIN FOLDING

2.2.1 Overview

After a protein is manufactured in the ribosomes, it spontaneously folds into a 3-dimensional structure. It is this 3D structure of a protein that determines its function. The seemingly amazing thing is that a specific protein will almost always fold in more or less the same way and will end up with the same 3D structure called its native state.

A convincing argument is that this is due to evolution - if the same sequence of amino acids would lead to different structures in proteins, their proper functioning could not be guaranteed. It is thus conceivable that evolution has produced proteins where multiple native states are unlikely.

Exactly how proteins fold from the sequence of amino acids (primary structure) remains an open question and has been a topic of much research since the protein folding problem was first posed. It is now accepted that proteins first form smaller local structures called secondary structures, before (or as some theories suggest, simultaneously) folding into its 3D structure (tertiary structure).

In 1951, Linus Carl Pauling analyzed the geometry and dimensions of peptide bonds. His research revealed the bond lengths and angles involved in the peptide bond molecules. Together with Robert B Corey, he predicted the existence of two regular secondary structures that are formed in proteins, namely alpha helices [49] and beta sheets [50] (and also falsely hypothesized other structures). Note that this work was done before protein structure has been experimentally determined. Their predictions turned out to be correct and earned them the Nobel Prize for Chemistry in 1954. These were the first secondary structures to be discovered.

Proteins can be unravelled or “denatured”. This can be achieved through the application of heat and certain chemicals. Christian Boehmer Anfinsen denatured a protein called ribonuclease and showed that it lost its shape and function (1961) [27]. By removing the denaturing substance, ribonuclease regained its function. Through chemical analysis and deductive reasoning, he was able to show that ribonuclease regained its original shape as well. This is a significant result, since it shows that all the "knowledge" required for protein to fold into its native state is contained in its amino acid sequence (and thus in the DNA sequence that codes for that protein), i.e. no folder or shaper is needed. Anfinsen's work led to him being awarded the Nobel Prize for Chemistry in 1972.

It is now known that in certain cases proteins can indeed fold into a wrong shape. Although the folding knowledge lies primarily in the amino acid sequence, proteins, known as chaperones, are sometimes used to keep their target proteins from folding incorrectly. Other factors, such as temperature, solvent viscosity and acidity, can also influence the folding process.

As could be expected, proteins that misfold are the cause of certain diseases [28] [29]. Even a single amino acid that is missing or incorrect could cause such a misfold. Since a protein's function is largely determined by its structure, a misfold implies that a protein does not function correctly or does not function at all. In the worst case, the misfold could lead to a situation where the protein influences substances around it in a detrimental way and as such "poisons" a cell. Diseases such as cystic fibrosis, Bovine spongiform encephalopathy (mad cow disease) and its human counterpart (Creutzfeldt-Jacob disease) and certain strains of Alzheimer's disease [30] are now all attributed to protein misfolding. By understanding the folding process, and perhaps more importantly the factors that cause misfolding, cures could be developed for these diseases.

Another interesting aspect is the time it takes for a protein to fold into its native state. It typically takes a anything from a number of milliseconds to a number of seconds for a protein to assume its native state. The fastest folders complete this process in a couple of microseconds whilst some proteins could take a number of minutes. In 1968, Cyrus Levinthal showed that the total number of conformations a protein could take is astronomical [32]. Even if a protein could sample a conformation in a nano- or picosecond, it would take more than the age of the universe to sample all configurations. It can thus be concluded that a random conformational search does not occur in folding, but rather that one or more mechanisms exist which allow a protein to fold via some pre-determined path. Theories regarding the exact way in which this is accomplished are discussed in Section 2.2.3.

2.2.2 Levels of Protein Structure

This section defines a number of terms that are used to describe the level of protein structure.

2.2.2.1 Primary Structure

The primary structure of a protein (or segment of polypeptide chain) is the sequence of amino acid residues, without regard to spatial arrangement. Note that in the primary structure of a protein, all the atoms are held together by covalent forces.

2.2.2.2 Secondary Structure

The secondary structure of a segment of polypeptide chain is the local spatial arrangement of its main-chain atoms without regard to the conformation of its side chain or to its relationship with other segments. Note that a secondary structure is locally defined, i.e. there can be multiple secondary structures within a single protein. The secondary structures form due to hydrogen bonds that form between amino acids at regular intervals within the chain. The reader is referred to Section 2.3 for a detailed discussion of secondary structures.

2.2.2.3 Supersecondary Structure

It is sometimes observed that certain structural components comprising a number of secondary structures are frequently repeated within proteins, e.g. two alpha helices joined by a loop region. These are termed supersecondary structures. Some of these structures are associated with certain biological functions, whilst others are part of larger structural or functional units.

2.2.2.4 Tertiary Structure

The tertiary structure of a protein molecule, or of a subunit of a protein molecule, is the arrangement of all its atoms in space, without regard to its relationship with neighbouring molecules or subunits. The tertiary structure of a protein is kept in place through hydrophobic interactions, hydrogen bonds, ionic interactions and disulfide bonds.

2.2.2.5 Quaternary Structure

Some proteins, termed multimeric proteins, consist of a number of subunit proteins or polypeptide chains. The quaternary structure of a protein molecule is the arrangement of its subunits in space and the ensemble of its intersubunit contacts and interactions, without regard to the internal geometry of the subunits.

2.2.2.6 Protein Conformation

The process by which higher structures form from the primary structure is called protein folding. A folded protein can have more than one stable folded state or conformation. Each conformation has its own biological activity. At any stage, only one conformation is active. The most common state is called the native conformation. The transitions between different conformations are called conformational changes.

2.2.3 Theories of Protein Folding

The resulting tertiary structure that forms when a protein folds is a stable conformation. It is generally accepted that proteins fold to reach a state of lower energy. The open question is whether it reaches a global (stable) or local (meta-stable) minimum in its native conformation.

The thermodynamic hypothesis of protein folding was proposed by Epstein in 1963 [33] after earlier work by Haber and Anfinsen [31]. According to the thermodynamic hypothesis, the native state of a protein is reached when it is has reached a global minimum in its energy state. In opposition to the thermodynamic hypothesis is the kinematic hypothesis of protein folding. As proposed by Wetlaufer in 1973 [34], [35], the kinematic hypothesis states that a protein could become trapped with a local minimum in its energy state, unable to overcome the energy barriers that will enable it to reach a global minimum. The native state of a protein correspond to this local minimum. It is conceivable that these meta-stable states could be vastly different from the true stable (minimum energy) conformation.

Initially, the unfolded protein is in a random coil state. The changes that occur during the initial phase of the folding process could thus appear to be somewhat random in nature. Levinthal showed that if only random changes were made to the conformation of a protein, with the expectation that a minimum energy state will be reached in which the native state is always the same, it would take an astronomical amount of time [32].

Levinthal's work led to the conclusion that there exist folding pathways and intermediates - states and partially folded chains that a protein necessarily undergo during the folding process. Such intermediates were observed by Ikai and Tanford [36] and Tsong and Baldwin [37] in 1971.

Different views persist as to how the folding process gets started. One view is that folding is hierarchic - local backbone structures are formed and persist until the native state emerges. The other view is that folding is started through a tertiary interaction - distant clusters of side chains are then drawn together.

2.2.3.1 Framework Model

The framework model [38] [39] [40] suggests a hierarchical mechanism whereby local secondary structures are formed based on primary sequences, but independent of tertiary structure. Once these secondary structures collide, they coalesce to form tertiary structure. One problem with the theory is that peptides do not generally form stable secondary structures in solution.

2.2.3.2 Hydrophobic Collapse Hypothesis / Molten Globule Hypothesis

Proteins are normally found in a configuration where the hydrophobic amino acids are buried toward the inside of the folded protein, whilst hydrophilic amino acids are found more towards the surface of the protein. The hydrophobic collapse hypothesis [41] [42] [43] states that a protein assumes its native conformation through the formation and rearrangement of a compact collapsed structure known as a molten globule. This step constitutes an early step in the folding pathway. The framework and hydrophobic collapse models suggest the formation of kinematic intermediates.

2.2.3.3 Nucleation model

The nucleation model [44] [34] states that tertiary structure forms as an immediate consequence of the formation of secondary structure. A few amino acid residues form secondary structures which serve as a nucleus. Further structure then propagates from this nucleus. Note that the nucleation model does not necessarily lead to the formation of kinematic intermediates.

2.2.3.4 Directed Folding Model

The directed folding model suggests that specific interactions could direct the folding pathway by stabilizing folded conformations. For instance, in bovine pancreatic trypsin inhibitor (BPTI) it has been shown that the formation of disulphide bonds stabilize secondary structure and leads to specific pathways [45].

2.2.3.5 Folding Funnel Model

One of the more recent theories is that of the folding funnel model. The theory represents the energy surface of the protein folding pathway as a funnel. Different unfolded conformations are at the rim of this funnel, with a single global minimum representing the native conformation. Different folding paths exist from the unfolded states to the native state. The protein could fold by means of steepest decent (fastest folding) or follow other paths through local minima (intermediates) and maxima (transition states) [46] [47].

The principle of minimum frustration, hypothesized by Peter Wolynes, states that through evolutionary processes, natural proteins are composed of amino acid sequences that interact with one another in such a way as to be directed towards the native state, i.e. the energy landscape is mostly smooth.

2.2.3.6 Simulations of Protein Folding

De novo or ab initio techniques for computational protein structure prediction employ simulations of protein folding to determine the protein's final folded shape.

An example of such a simulation is LINUS by Rose and Srinivasan [48]. LINUS implements elements of the framework model, hydrophobic collapse and the nucleation model and allows for the fact that the native state could be a local minimum (kinematic hypothesis). LINUS was executed against 7 proteins. The authors claim that 99% of the secondary structures were correctly predicted and 6 out of the 7 proteins had the correct shape through visual inspection.

One problem with protein folding simulation is that it takes a tremendous amount of computational power (and thus time) to simulate even a small amount of time during the folding process. As such many distributed initiatives have seen the light since 2000. These include Folding@home [119], Human Proteome Folding Project, Predictor@home [120], Rosetta@home [121] and TANPAKU. Another approach is to use supercomputers to perform the simulation. IBM's BlueGene [122] is an attempt to construct a petaflop supercomputer dedicated to protein folding.

2.3 SECONDARY STRUCTURE

2.3.1 Secondary Structure Classification

2.3.1.1 The DSSP Code

Although different schemes exist or could be created to classify secondary structures, one scheme is currently predominant - the “Dictionary of Protein Secondary Structure” commonly referred to as the DSSP code. This code was developed by Kabsch and Sander in 1983 [52] and aims to unambiguously define secondary structures based on their physical and geometrical features. It thus provides a method to define secondary structures objectively (previously subjective classifications had to be made by crystallographers and structural biologists).

The code defines eight protein secondary structures. These are listed in Table 2.2. It is customary to associate one of the eight secondary structures with each amino acid in a protein. There is thus a one-to-one correspondence between each amino acid and its associated secondary structure. If no such association can be made, the coil (C) structure is assumed.

Table 2.2: The DSSP Code

Abbreviation	**Secondary structure**
G	3 turn helix (3_{10}-helix)
H	4 turn helix (α-helix)
I	5 turn helix (π-helix)
E	β-sheet (extended strand)
B	β-bridge
T	Hydrogen bonded turn
S	Bend
C	Coil (also known as loop - L)

Note that other secondary structures such as sharp loops and omega turns have been suggested. These structures have however not been used widely.

Table 2.3: Reducing the 8 DSSP classes to 3 classes

DSSP (8-class)	**3-class**
α-helix (H), 3_{10}-helix (G)	Helix (H)
β-sheet (E), β-bridge (B)	Strand (E)
π-helix (I), Turn (T), Bend (S), Coil (C)	Coil (C)

2.3.1.2 3-Class Classification

In pattern recognition and statistical terminology, the word "class" is used to designate a discrete set of values (or class labels) which a variable can be assigned. In the problem of secondary structure classification, the word class is often used interchangeably with the (secondary) structure that is being predicted. This convention is used throughout the dissertation.

Apart from the DSSP code, secondary structures are often classified according to only three classes: helices (H), sheets (E) and coils (C). This is probably due to the fact that after Pauling discovered alpha helices and beta sheets, these were the only known structures. If an amino acid did not form part of one of these two structures, it was classified as a coil. This classification scheme persisted and is useful in that it provides a common framework by which to compare the success of secondary structure prediction techniques.

It should be immediately apparent that there exist different schemes by which the eight classes in the DSSP code can be mapped to the three-class scheme. The scheme that is now in widespread use, has been suggested by Rost and Sander [74]. This mapping scheme, listed in Table 2.3, maps the H and G structures to helix (H), the E and B structures to strand (E), and all the rest (I, T, S and C) to coil (C).

This standard mapping scheme has since been used by most authors [78], [105], [65], although other mapping schemes have also been tried out [78], [110]. Rost, in a more recent article [65], has however pointed out that that this standard mapping provides a way to compare different secondary structure prediction methods. He also noted that other mapping schemes may lead to overly optimistic classification results.

2.3.2 Types of Secondary Structures

Secondary structures form due to hydrogen bonds that form between amino acids at regular intervals within the chain. The only exception is the bend secondary structure which does not form due to hydrogen bonds. The formation of secondary structures leads to regular patterns in the ϕ and ψ angles where these structures occur. A good discussion of secondary structures can be found in the work of Richardson [51].

2.3.2.1 Alpha helices

The alpha helix (also known as 4-turn helix or 3.6_{13}-helix) is the most commonly occurring type of secondary structure in proteins. Its existence was first predicted by Pauling et al in 1951 [49]. The amino acids are arranged in a helical structure about 5Å wide. Each amino acid contributes a 100° turn in the helix, i.e. there are 3.6 amino acids per turn. The translation along the helical axis from one amino acid to the next is about 1.5Å. The average length of an alpha helix is about 10 amino acids. At least 4 amino acids are required for a structure to be classified as an alpha helix [52]. Alpha helices are usually found in a right-handed configuration, although left-handed configurations sometimes occur. The backbone conformation angles in the right-handed configuration are $\phi = -63°$ and $\psi = -43°$ [51].

In general, alpha helices are found at the surface of protein cores where they provide an interface with the aqueous environment. The inner facing side of the helix tends to have hydrophobic amino acids and the outer facing side hydrophilic amino acids. Every third or fourth amino acid tends to be hydrophobic, a pattern that can be detected [55]. Alpha helices are sometimes found in protein cores in which case they have a higher distribution of hydrophobic amino acids ([53], pp. 378-388). They also contribute the most to the stability of a protein of all the secondary structure types [51].

Different amino acids have different preferences for forming alpha helices. Alanine, glutamic acid, leucine and methionine are readily found in alpha helices while proline, tyrosine, serine and glycine are rare in this structure [54].

The alpha helix arises because of hydrogen bonds forming between the C=O group of the nth amino acid and the NH group of the (n+4)th amino acid. The alpha helix and the corresponding bonds that form are illustrated in Figure 2.7.

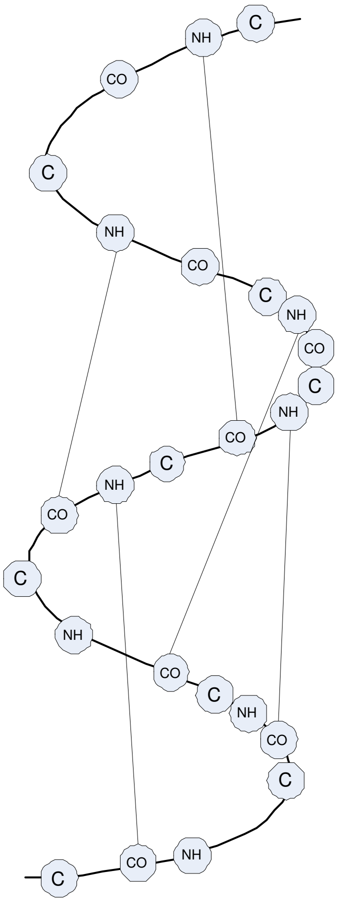


Figure 2.7: Hydrogen bonds in an alpha helix

2.3.2.2 Beta sheets

The beta sheet (also known as extended strand) is the second most commonly occurring type of secondary structure. Its existence was predicted by Pauling and Corey in 1951 [50], shortly after the existence of alpha helices was predicted.

A beta sheet consists of two or more amino acid sequences (beta strands) in the same protein that bond together through hydrogen bonds. These strands typically contain 5 to 10 consecutive amino acids and can bond with adjacent strands in a parallel or antiparallel configuration (or a mixture of the two in the case of three or more stands) as illustrated in Figure 2.8. The hydrogen bonding patterns are different in the parallel and antiparallel configurations. Note that the strands could be near each other in the amino acid sequence (typically separated by a short loop region) or far apart.

Parallel sheets and the parallel parts of mixed sheets tend to be buried in proteins, whilst antiparallel sheets tend to have one side exposed to solvents and the other buried in the core of the protein [51].

An interesting feature of sheets are that they twist [56]. A single beta strand is rarely perfectly extended, but rather exhibits a slight twist due to the chirality of the component acids. This can be attributed to the fact that the energetically preferred dihedral angles ($\phi = -135°$ and $\psi = 135°$) diverge from the fully extended conformation ($\phi = -180°$ and $\psi = 180°$). There are oftentimes alternating fluctuations in the dihedral angles to prevent the individual strands in a sheet from spraying apart. Note that if the twist of the hydrogen bonding direction or of the peptide planes is viewed along a strand, it would appear right-handed in most cases. The dihedral angles are about $\phi = -140°$ and $\psi = 135°$ in antiparallel sheets and $\phi = -120°$ and $\psi = 115°$ in parallel sheets.

2.3.2.3 Turns

The third of the three classical secondary structures is the hydrogen bonded turn. Turns serve the function of reversing the direction of the local segment of the polypeptide chain.

Turns were first recognized by Venkatachalam [57] through theoretical conformational analysis. Three types of turns were suggested by Venkatachalam and another five by Lewis [58]. Turns are given structure through hydrogen bonds between the CO atoms of amino acid i and the NH atom of amino acid $i + n$, where $n \in 3, 4, 5$.

Turns tend to be hydrophilic, which could be a result of the fact that a typical turn joins or interrupts secondary structures that are more internal [59] [60]. Turns are commonly found joining beta-strands or at the end of alpha-helices. Glycine and proline are common constituents of turns.

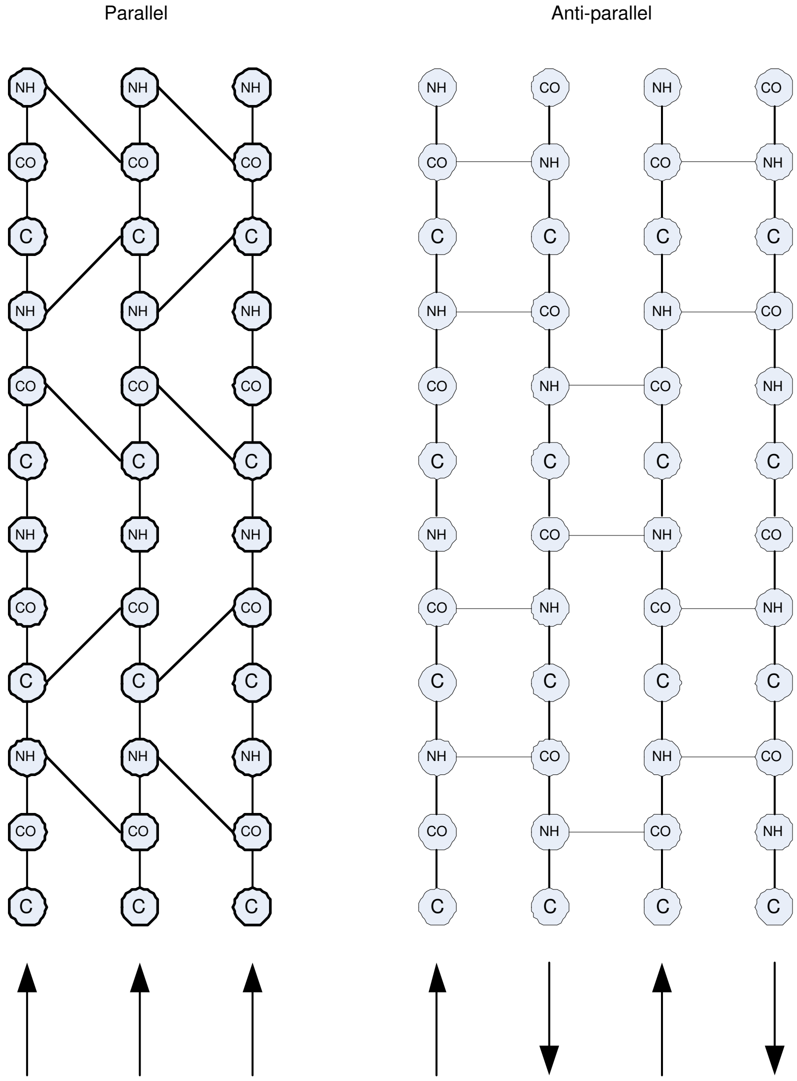


Figure 2.8: Hydrogen bonds in beta sheets

2.3.2.4 Other secondary structures

The 3_{10}-helix (also known as 3-turn helix) is another helix type that is frequently observed. Similarly to the alpha-helix, it forms due to hydrogen bonds, this time between amino acids at residues i and $i + 3$. A minimum of 3 consecutive amino acids are required to define a structure as a 3_{10}-helix. The backbone conformation angles are about $\phi = -70^\circ$ and $\psi = -20^\circ$ [51]. 3_{10}-helices are typically much shorter that alpha helices.

The π-helix (5-turn helix) forms due to hydrogen bonds between amino acids at residues

i and $i+5$ and five consecutive amino acids are required to define a structure as such. The π-helix is the least frequently occurring secondary structure - it requires that $\tau = 114.9°$, instead of the normal $\tau = 109.5°$ and the conformation angles $\phi = -57.1°$ and $\psi = -69.7°$ lie at the edge of the allowed minimum energy region on the Ramachandran plot. Both the 3_{10} and π-helices are sometimes found at the edge of regular alpha helices.

Note that in the case of all the helices (α, 3_{10} and π) the requirement for a hydrogen bond need not be mandatory. Rather, the conformation angles should be within the acceptable range.

A β-bridge is a single pair β-sheet, i.e. a hydrogen bond forms between two distant amino acids.

The bend is the only secondary structure that is not based on a hydrogen bond. A bend is a region with high curvature. For a bend at position i, the angle formed between C_α^{i-2}, C_α^{i} and C_α^{i+2} should be larger than 70°.

Coils (also known as loops) are used to describe two types of regions: those areas that are well-organized but non-repetitive, as well as those areas that are truly disorganized. Disorganized here means that the amino acids are not observed to be in any of the other regular secondary structures.

2.4 PREDICTION OF SECONDARY STRUCTURE

The assumption on which secondary structure prediction methods are based is that there is a correlation between amino acid sequence and secondary structure. This assumption follows *necessarily* from Anfinsen's work [27] that states that all knowledge of the final structure (and hence secondary structure) is contained in the amino acid sequence.

Secondary structure prediction was first attempted as early as 1957 [66]. Note that this was before the claim of the existence of alpha-helices and beta-sheets was even verified through X-ray structures. Since then, 3 generations of protein secondary structure prediction methods have seen the light [65] [64].

The first generation of methods were based on expert rules and statistics of the physico-chemical properties of single amino acids. These methods took into account only single amino acids at a time and achieved Q_3 scores in the order of 50% (the Q_3 score is the percentage of correctly predicted secondary structures and is explained in Section 2.4.1.1). The next generation of methods improved on this by also taking into account the window of amino acids adjacent to the central amino acid (the one for which a secondary structure is being assigned). Since the local structure influences the formation of the secondary structure at the central amino acid and these relationships were being taken into account, these methods achieved Q_3 scores in the order of 60%.

Since the conception of the second generation methods, the number of proteins for which the structures have been solved has increased considerably. This made it possible to identify evolutionary information in these databases. The third generation of methods is based on taking multiple sequence alignments as inputs instead of a single amino acid sequence. As such, they are able to consistently achieve Q_3 scores of about 70% (the best algorithms, such as PSIPRED, PROF and SSpro achieving an accuracy of about 76% [65]).

Another useful way of classifying secondary structure prediction algorithms is in terms of the method they employ. There are three main classes: Methods that use expert rules and statistics, such as the Chou-Fasman method, methods based on information theory, such as the Garnier-Osguthorpe-Robson method and methods based on computational intelligence. Various computational intelligence methods such as neural networks, recurrent neural networks, nearest neighbor methods, Hidden Markov Models and Support Vector Machines have been tried.

2.4.1 Methods to measure the accuracy of prediction

In order to compare the accuracy of different secondary structure prediction techniques with one another, the same data sets as well as the same measure of performance should be used in the comparison. This section discusses the different measures of performance.

Apart from the measures of performance used, it should also be noted that certain secondary structures (such as alpha helices) are more readily predicable than others. It thus implies that the set of test proteins could strongly influence the observed accuracy. In practise, standard data sets are often used and are selected so as to have a low sequence similarity.

2.4.1.1 Q-score

The Q-score is probably the most widely used measure of performance [62].

A secondary structure is associated with each amino acid in the sequence. The Q-score is simply the fraction of correctly identified secondary structures and is usually expressed as a percentage. It is given by

$$Q = \frac{\text{number of correctly classified secondary structures}}{\text{total number of amino acid residues}} \times 100\%. \qquad (2.1)$$

A subscript is usually used to indicate the number of classes a secondary structure can be assigned to. Thus, if the DSSP code is used, the score is referred to as Q_8. If the 3-class scheme is used, the score is referred to as Q_3.

Note that the Q score tends to favour methods overpredicting the secondary structure with the highest prior probability of occuring [65]. For instance, in the 3 class problem, methods that overpredict the C structure (as opposed to the H and E structures) are likely to have a higher Q_3 score. Another objection is that even a random assignment of secondary structures could have a relatively high Q score.

In cases where a secondary structure prediction is not made for every amino acid, it is sometimes convenient to use an adapted version of the Q score, namely the Q^* score. The score simply calculates the percentage of correctly classified secondary structures as a percentage of those for which a prediction was attempted. It is given by

$$Q^* = \frac{\text{number of correctly classified secondary structures}}{\text{total number of amino acid residues predicted}} \times 100\%. \tag{2.2}$$

2.4.1.2 Q-score for secondary structure types

The Q-score is sometimes adapted to serve as a per-residue accuracy measurement for secondary structure types. The per-residue accuracy [104] is calculated as

$$Q_x = \frac{\text{number of residues correctly predicted in state x}}{\text{number of residues observed in state x}} \times 100\%, \tag{2.3}$$

and the per-residue prediction accuracy as

$$Q_x^{pre} = \frac{\text{number of residues correctly predicted in state x}}{\text{number of residues predicted in state x}} \times 100\%, \tag{2.4}$$

where x represent the type secondary structure.

2.4.1.3 Matthews correlation coefficient

The Matthews coefficient [61] is calculated for each type of secondary structure and is given by

$$C_x = \frac{p_x n_x - u_x o_x}{\sqrt{(n_x + u_x)(n_x + o_x)(p_x + u_x)(p_x + o_x)}}, \tag{2.5}$$

where x represents the type of secondary structure, p_x is the number of correct positive predictions, n_x is the number of correct negative predictions, o_x is the number of overpredicted positive predictions (false positives) and u_x is the number of underpredicted residues (false negatives). The closer the coefficient is to 1, the better the success of the prediction algorithm in predicting the type of secondary structure.

2.4.1.4 Segment Overlap Measure

The segment overlap (SOV) measure [62] [63] is based on secondary structure elements and not on individual amino acid residues. It aims to quantify how well a prediction method predicts each secondary structure element. It takes into account the starting and ending residues of each secondary structure element and the length of each element.

Consider for example the case where two helices joined by a short turn are predicted as a helix. The Q_3 measure would penalize only on the short turn section. The SOV measure penalizes for predicting only one structure instead of two as well as missing the correct ending position of the first helix and the correct starting position of the second.

The SOV measure for a single secondary structure type is defined as

$$SOV_x = \frac{1}{N_x} \sum_{S_x} \frac{minOV(S1,S2) + \delta(S1,S2)}{maxOV(S1,S2)} \times \text{len}(S1) \times 100\%, \tag{2.6}$$

where $S1$ and $S2$ are the observed and predicted secondary structure segments of type x respectively, S_x is the number of all segment pairs $(S1, S2)$ where $S1$ and $S2$ have at least one residue of type x in common, len$(S1)$ is the number of residues in segments $S1$, $minOV(S1, S2)$ is the length of overlap of $S1$ and $S2$, i.e. the number of residues where both $S1$ and $S2$ are in state x and $maxOV(S1, S2)$ is the length of the total extent for which either of the segments $S1$ and $S2$ has a residue in state x and N_x is the total number of residues observed in state x. $\delta(S1, S2)$ is defined by

$$\delta(S1,S2) = min \left\{ \begin{array}{l} maxOV(S1,S2) - minOV(S1,S2) \\ minOV(S1,S2) \\ int(\frac{1}{2} \times len(S1)) \\ int(\frac{1}{2} \times len(S2)) \end{array} \right\}. \tag{2.7}$$

The segment overlap score for all the different types of secondary structure types is defined as

$$SOV = \frac{1}{N} \sum_{x \in C} \sum_{S_x} \frac{minOV(S1, S2) + \delta(S1, S2)}{maxOV(S1, S2)} \times \text{len}(S1) \times 100\%. \qquad (2.8)$$

Here N is the total length of the amino acid residues being observed and C is the set of secondary structure types.

2.4.2 Chou-Fasman Method

The Chou-Fasman method [67] is based on analysis of the frequency with which single amino acids are found to create different secondary structures. For instance, alanine, glutamic acid, leucine and methionine are strong predictors of alpha helices, whilst proline and glycine are predictors of a break in a helix.

The method is based on heuristics. Helices and sheets are predicted if amino acids that are indicative of that structure are found in sequence a number of times. Turns are modelled as tetrapeptides and two probabilities are calculated. If more than one secondary structure is predicted for a specific region, the structure with the highest probability is assigned. In the end, regions for which no prediction is made are assigned as coils.

The Chou-Fasman method achieved Q_3 scores in the region of 50-60% on standard test databases.

2.4.3 Garnier-Osguthorpe-Robson Method (GOR)

The GOR-method [68] [69] [70] extends the Chou-Fasman method by incorporating the idea that amino acids that flank the central amino acid influence the secondary structure that the central amino acid is likely to adopt. The GOR-method also uses principles from information theory to derive predictions.

The 8 amino acids prior and the 8 amino acids after the central amino acid are used to create three scoring matrices. These scoring matrices correspond to the central amino acid being found in an alpha helix, beta sheet or coil configuration. The columns of the scoring matrices indicate the probabilities of finding each of the amino acids in one of the 17 positions. These probabilities are calculated based on information theory concepts.

A prediction of a candidate sequence is made through a sliding window of 17 amino acid residues. The sequence is then compared with the matrices, the one with the highest score predicting the secondary structure associated with the central amino acid. Four residues in a row have to be predicted as an alpha helix and two in a row as a beta sheet for the prediction to be validated.

The GOR-method has been shown to achieve a Q_3 score of 64%. It is also known that the method underpredicts the number of residues with the sheet structure.

2.4.4 Neural Network Methods

Neural network methods have been used widely to predict protein secondary structure [71] [72] [73] [74] [75] [76] [77] [78] [79] [81] [82] [83]. It has been shown that the neural network models are theoretically able to extract more information from sequences than methods based on information theory such as the GOR-method [71].

In the neural network approach, a training phase is used to set weight values in the neural network. A sliding window of length n is moved along the amino acid sequence and the associated secondary structure of the central amino acid noted. This input-output mapping is then used to train the network using a method such as the back-propagation algorithm.

Usually, the classical 3-layer neural network is used. Each of the n amino acid residues is usually encoded using 21 input nodes (i.e. $n \times 21$ input nodes in total) - one node for each of the 20 different types of amino acid residues and an additional node to indicate if the position in the window is an edge. In each set of 21 input nodes, only one input node is thus triggered at a time. The output is usually encoded using m output nodes, where m represents the number of secondary structure classes. A sufficient number of hidden nodes is required to capture the input-output mapping. Various numbers of hidden nodes have been suggested, from 2-40 [76] to 60 [71]. The studies also suggest that a window length (n) of 13-17 gives optimum performance.

Once a neural network has been suitably trained, it can be used to predict the secondary structures associated with a protein of unknown secondary structure. If the neural network has been structured as explained above, it will present m outputs for each input sequences. Each of these m outputs represent the probability that the secondary structure to be associated with the central amino acid is of a specific type. Based on these probabilities, criteria such as the maximum-likelihood function or other smoothing rules can be applied to assign a secondary structure to each amino acid residue.

The best known methods are PHD by Rost [74] and PSI-PRED by Jones [78] which achieve an average prediction accuracy of 75-76 % (Q_3). These prediction methods do not use amino acids sequences directly as input to a neural network but rather make use of multiple sequence alignments and position specific scoring matrices (PSSM) generated by algorithms such as Basic Local Alignment Search Tool (BLAST) [85] and Position Specific Iterated (PSI) BLAST [86]. Without such multiple alignments, the accuracy achieved is typically about 67% [96].

BLAST and PSI-BLAST are used to compare a query (target) sequence to all sequences in a specified database (sequence database). The objective is to find subsequences in the sequence database that are similar to the target sequence. The idea is that the target sequence will exhibit similar structural attributes as those proteins with similar sequence. This fact can be exploited in the design of the neural networks.

The Blocks Amino Acid Substitution Matrices (BLOSUM) [87] represent frequencies of amino acid substitutions observed in a large number of related proteins. The BLOSUM62 matrix is tabulated in Table 2.4. Each position in the matrix represents the log odds score for the substitution of a particular amino acid with another amino acid.

The BLAST algorithm starts by creating a list of amino acid patterns (words) of length (W) 3 in the target sequence. It starts at positions 1, 2 and 3, followed by 2, 3 and 4, and so forth. The output of this stage is a list of unique patterns of length 3 in the target sequence. The algorithm then determines which words are likely substitutions

Table 2.4: BLOSUM62 Substitution Matrix

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	12																			
S	0	2																		
T	-2	1	3																	
P	-3	1	0	6																
A	-2	1	1	1	2															
G	-3	1	0	-1	1	5														
N	-4	1	0	-1	0	0	2													
D	-5	0	0	-1	0	1	2	4												
E	-5	0	0	-1	0	0	1	3	4											
Q	-5	-1	-1	0	0	-1	1	2	2	4										
H	-3	-1	-1	0	-1	-2	2	1	1	3	6									
R	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6								
K	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5							
M	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6						
I	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5					
L	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6				
V	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4			
F	-4	-3	-3	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9		
Y	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10	
W	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17

to the target words through evaluation using the BLOSUM62 matrix. For instance, consider the word PQG. The word PEG would score a value of 15 using the BLOSUM62 matrix (summing the log odds values of 7 for a P-P match, 2 for a Q-E match and 6 for a G-G match). A score threshold T is used to limit the number of possible words that can match to the target words. These words are organized into an efficient search tree for comparing them rapidly to the database sequences.

The database is now scanned for these remaining words that are likely substitutions for the target word. When such a word is found, the target sequence and the sequence from the database are aligned through the matching substitution words. The alignment is extended in both directions by evaluating the BLOSUM62 values for substitutions at corresponding locations in the sequences. The alignment is extended as long as the accumulated score does not decrease. This portion of the alignment is known as the high-scoring segment pair (HSP). All such HSP scores are calculated against the whole sequence database and HSP's with a score larger than a cutoff score S are noted. The statistical significance of the HSP score is calculated as an E-value. If it is significant the alignment is reported.

PSI-BLAST uses a series of iterated steps. This is done to identify a family of related proteins for a given target sequence. Once an initial set of related proteins are found for a given target sequence, these proteins are used to identify additional proteins that are related to the target sequence. PSI-BLAST generates PSSMs (sequence profiles) as part of the search process. In a PSSM, each row is associated with a specific amino acid in the target sequence and each column with one of the amino acid types (thus 20 columns). Each element in the matrix indicates the log likelihood of a substitution of the amino acid in the target sequence with the amino acid type specified by the column.

In PHD (year 1993), BLAST (1990) was used to create multiple sequence alignments and train the neural network. With the development of PSI-BLAST (1997) and the ease with which scoring matrices could be extracted, PSI-PRED (1999) used these intermediate profiles as input to the neural network. This eliminated the need for the time consuming multiple sequence alignment stage in PHD.

In terms of neural network architecture, PSI-PRED uses a window length (n) of 15, with 21 inputs associated with each amino acid in the window, similar to the classical neural network design for predicting secondary structures. The difference is that the each of the 20 inputs associated with each amino acid is the log odds value for a residue substitution as given by the PSSM (and filtered though the standard logistic function $\frac{1}{1+e^{-x}}$ to scale it to the range [0,1]). The additional input is used in the case where an edge is present. The rest of the neural network structure is similar to the structure described earlier in this section, with 75 hidden units and 3 outputs (m). Each output is the probability that the predicted secondary structure is either a helix, strand or coil.

A second neural network is used to filter the results from the first network. This network has 60 inputs (a window of 15 with 4 inputs each, indicating the probability of helix, strand or coil as calculated by the first network, or the presence of an edge), 60 hidden units and 3 outputs. The outputs represent the final 3-state predication.

Web servers exist that allow online prediction of protein secondary structure using PHD [115] and PSIPRED [116].

2.4.5 Nearest Neighbour Methods

Nearest neighbour methods [88] [89] [90] [91] [92] [93] [94] [95] predict the secondary structure of an amino acid in a query sequence by identifying sequences of known structures that are similar to the query sequence.

A database of training sequences is built in the same way as with neural network techniques, i.e. a sliding window of size n is moved across the training set and the secondary structure of the central amino acid observed.

For the query sequence, the best matching sequences in the training database are identified. The frequencies of occurrence of the different secondary structures are then used to predict the associated secondary structure for the query sequence.

The different algorithms in existence differ in the way sequences are compared. Amino acid scoring matrices such as BLOSUM [90], distances between sequences based on statistical analysis of the training sequences [89] and scoring matrices based on the categorization of amino acids into local structural environments [91] [92] have been used.

Programs such as PREDATOR [95] and NNSSP [93] have achieved accuracies of 75 % and 73.5 % (Q_3) respectively. Web servers exist that allow online prediction of protein secondary structure using these methods [117] [118].

2.4.6 Hidden Markov Models

Hidden Markov Models (also known as discrete space models) have been applied to the problem of protein secondary structure prediction by a number of researchers [96] [97] [98] [99] [100] [101] [102] [103].

A Hidden Markov Model (HMM) is a probabilistic finite state machine used to model stochastic sequences. A HMM contains states and connections between states as well as state transition probabilities. HMM's could be designed by hand, or designed algorithmically. Once a suitable HMM has been designed, it is used to predict the most likely output sequence (secondary structure) to be associated with the input sequence (primary structure). The HMMSTR model [103] claims an accuracy of 74 % (Q_3). In a recent result, OSS-HMM (Optimal Secondary Structure Hidden Markov Model) [96] achieved a Q_3 score of 68.8% when applied to single sequences, and 75.5% when multiple sequence alignments are used.

2.4.7 Support Vector Machines

Support Vector Machines (SVM) are some of the latest computational intelligence techniques that have been applied to the problem of protein secondary structure prediction [104] [105] [106] [107] [108] [109].

In the SVM approach, the input space (primary sequence) is mapped to a higher-dimensional feature space through the use of a kernel function. The idea is that the kernel function is such that the features are linearly separable in the higher-dimensional space. As such, SVM's are able to represent complex nonlinear functions. The other advantage of SVM's are that efficient training algorithms exist. Accuracies of up to 77 % (Q_3) accuracy have been achieved [105] using SVM's and multiple sequence alignments.

Chapter 3

PATTERN RECOGNITION ALGORITHM

This chapter and the next describe the pattern recognition algorithm that was developed to solve the problem of predicting protein secondary structure from protein primary structure. In this chapter, the method is outlined and discussed, whilst the next chapter describes the method formally (mathematically).

A pattern recognition method was developed that associates an output string with an input string, where the elements of the input and output strings are defined over two (possibly different) alphabets.

In the rest of this document, the method that was developed is described based on its applicability to protein secondary structure prediction. However, the method is independent of this particular problem and can be applied to other problems with a similar structure as well.

3.1 APPROACH

The aim of the pattern recognition algorithm (also referred to as the technique, method or predictor) is to accurately predict the unknown secondary structure of a protein for which only the primary structure is known. Although it is useful to have such a algorithm, the end goal is to discover and gain insight into the role that single amino acids or small sequences of amino acids play in the formation of protein secondary structures. The algorithm is an enabler for this discovery process.

Before the algorithm can be used to make predictions, it is trained on a set of proteins for which both the primary structure and secondary structure is known. This set of proteins is known as the training set. The training phase occurs only once, before any predictions are made.

Once the training phase is completed, the algorithm can be used to predict the secondary structures of proteins with known primary structure. This is known as the prediction set. Since a prediction can be made for proteins with unknown secondary structure, this application of the algorithm is of practical importance.

In order to establish the performance of the system, a prediction set is used as a training set (both the primary and secondary structures for the proteins in the training set are known). The secondary structures of the training set are compared to the predicted secondary structures as given by the system. The percentage of correctly predicted secondary structures is used as an indication of the performance of the predictor (as defined in Section 2.4.1.1).

The pattern recognition algorithm is based on extracting statistical information regarding the protein input-output mapping (the primary structure serves as input and the associated secondary structure as output). Clearly, it is possible for the same input pattern to map to different output patterns. It is also possible for different input patterns to map to the same output pattern.

For an input pattern of length N, 20^N (21^N if edges are included) different input patterns exist. As N increases, a large amount of training data is required to cover the complete input space (the “curse of dimensionality” [80]).

The algorithm tries to eliminate the need for such a large amount of data in two ways:

- It groups together input patterns that behave similarly. If there are m such groups, a total of m^N different input patterns exist. In the case that $m < 20$, less training data is required.

- If an output pattern should be predicted for an input pattern that does not exist in the database, the algorithm tries to find input patterns in the database that are somehow "similar" to the input pattern in question. To do this, a metric needs to be defined that indicates the distance between patterns.

The different steps during training, prediction and evaluation are illustrated in Figure 3.1 and will be discussed in the sections that follow.

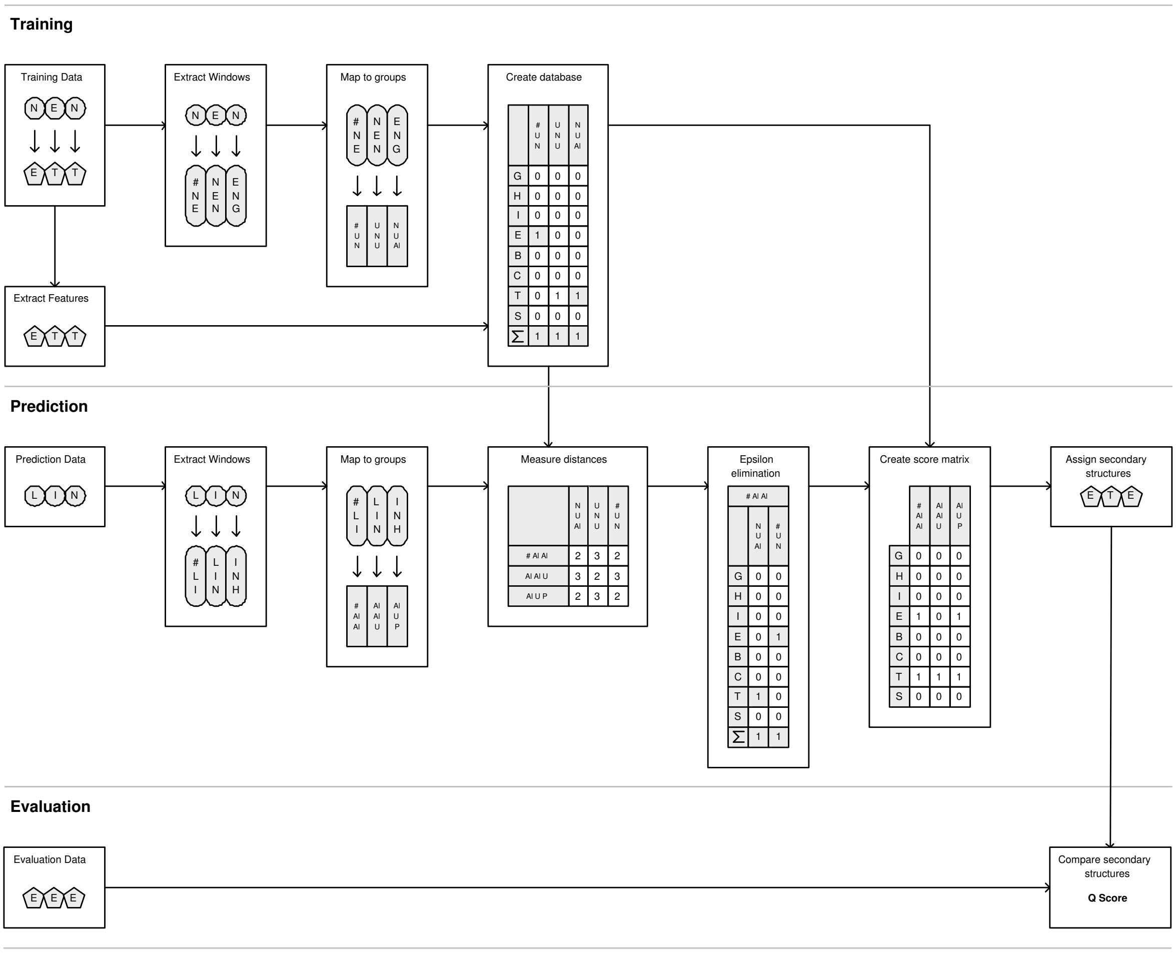


Figure 3.1: Steps in the pattern recognition algorithm

3.2 TRAINING

In the training phase, the objective is to build a database with relevant information that can be used for prediction.

The method is perhaps best illustrated by means of an example. Figure 3.2 shows the primary and associated secondary structure of the last 21 amino acids (number 381 to 401) in the molecule Creatine Amidinohydrolase. The primary structure of these amino acids are NENGAENITKFPYGPEKNIIR (each letter indicates an amino acid residue) with associated secondary structure ETTEEEECCCSCCSHHHHEEC (each letter indicates a secondary structure type). This small set of data will be used to construct the database. In practise, this process will be applied to all amino acids in all the proteins in the training set.

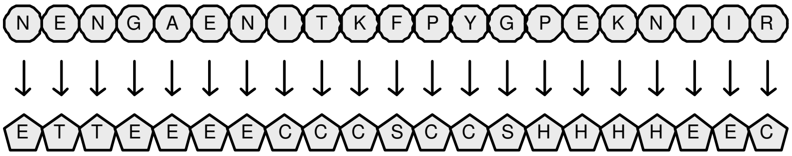

N E N G A E N I T K F P Y G P E K N I I R
E T T E E E E C C C S C C S H H H H E E C

Figure 3.2: Primary and secondary structure of amino acids 381 to 401 in the molecule Creatine Amidinohydrolase

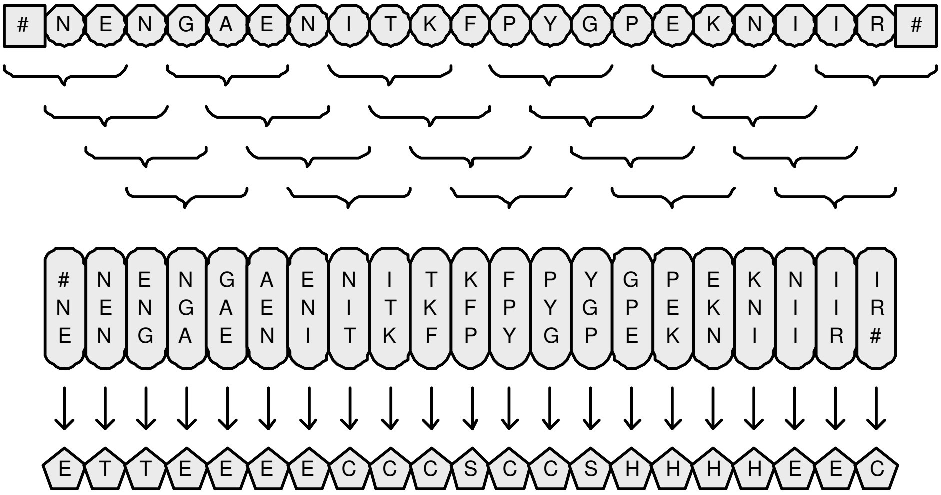
3.2.1 Step 1: Extracting Windows

The first step is to extract “windows” of amino acid residue sequences. These windows represent the input sequences that will be processed in order to create the patterns in the database and which will subsequently be used in the prediction process. This process of extracting windows is used by other prediction algorithms as well [74] [78].

The first decision is the size of the window (N). In the case of the method described here, smaller window sizes are less computationally expensive than larger window sizes. Although one would expect that larger window sizes would in general lead to better prediction, the truth is that the prediction accuracy is not only influenced by the window size, but also by a variety of other variables. One of the aims of this dissertation is to study the interplay of these variables.

Figure 3.3 illustrates how windows are extracted from the primary sequence. Note that each window is still associated with one of the secondary structures.

Once a suitable selection of the window size has been made, the association between the windows and secondary structures can be made in various ways (which window should be assigned to which secondary structure? - the problem should become immediately apparent when trying to imagine windows with even sizes). In the case of the example, each window consists of three ($N = 3$) amino acids: the original amino acid that was associated with the secondary structure (which for the purposes of discussion will be called the central amino acid) and the amino acids directly to the left ($l = 1$) and to the right ($r = 1$) of it. However, three different configurations are possible: $(l = 2, r = 0)$, $(l = 1, r = 1)$ and $(l = 2, r = 0)$.



# N E N G A E N I T K F P Y G P E K N I I R #
#NE NEN ENG NGA GAE AEN ENI NIT ITK TKF KFP FPY PYG YGP GPE PEK EKN KNI NII IIR IR#
E T T E E E E C C C S C C S H H H H E E C

Figure 3.3: Extracting windows of amino acids from the data. The example illustrates a window with $N = 3$ $(l = 1, r = 1)$

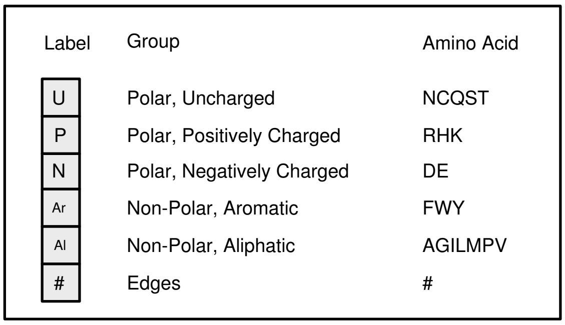
The question also arises how to treat windows on the "edges" of the residue sequence. One solution is to replace all residues that are "missing" with a placeholder. For the purpose of this discussion, the placeholder will be called an edge and will be denoted by the # symbol. Conceptually, an edge behaves exactly like a 21^{st} residue (with the restriction that a sequence of consecutive edges will never be found in a configuration where both the two residues to its sides are not edges). In the example, the first (#NE) and last (IR#) windows are examples where edges occur. In general, all windows will contain at least one non-edge (the central amino acid) and up to $N-1$ edges (although this is rarely the case).

3.2.2 Step 2: Assigning Groups to Windows

The second step is to decide on a relevant “grouping strategy”. Conceptually, the grouping strategy represents a mapping from an “input space” (residue space) to an “output space” (group space). The idea is that the problem is transformed to a space where the complexity in solving the problem is reduced.

Each window is mapped to a group vector. Each group vector consists of a number of group labels. The requirement is that there is at least one group label in a group vector (it is imperative to understand the difference between a group vector and group label). The mapping (L) can be as simple as an identity mapping ($\overline{y} = L(\overline{x}) = \overline{x}$, where $\overline{x}$ represents the input sequence and $\overline{y}$ represents the group vector), in which case each amino acid residue type is mapped to a group label and the group vector is exactly the window. This case may be useful when other parameters in the algorithm are compared, in which case this step can be omitted.

Figure 3.4 shows an example mapping that will be used for discussion. In this example, residues with similar characteristics are grouped together in six different groups. For instance, the amino acid residues that are both polar and uncharged (N, C, Q, S an T) are all assigned the same group label (U). Likewise the other group labels are assigned to different amino acid types, namely positively charged (P), negatively charged (N), aromatic (Ar) and aliphatic (Al). In this example, edges belong to their own group (#).



Label	Group	Amino Acid
U	Polar, Uncharged	NCQST
P	Polar, Positively Charged	RHK
N	Polar, Negatively Charged	DE
Ar	Non-Polar, Aromatic	FWY
Al	Non-Polar, Aliphatic	AGILMPV
#	Edges	#

Figure 3.4: A grouping strategy example

Figure 3.5 illustrates how the windows are mapped to group vectors. Each residue is replaced by its corresponding group label. As an example, the window ENG is replaced with the group vector NUAl. Note that each group vector is still associated with a secondary structure.

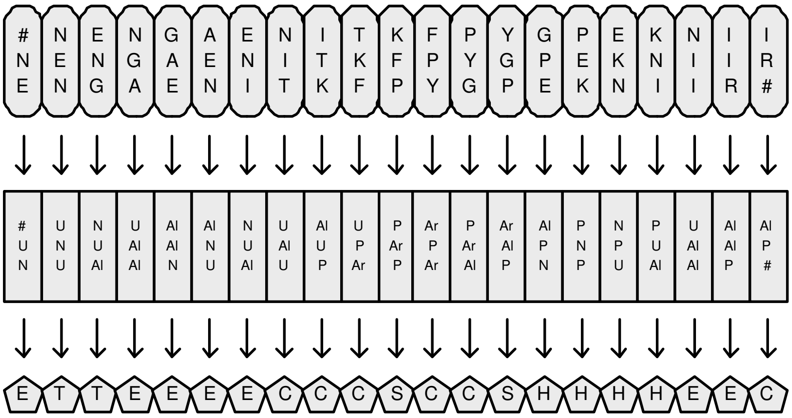


Figure 3.5: Assignment of group labels to windows

It is important to note that this is just one scheme whereby windows are mapped onto group vectors. Similar grouping schemes could easily be defined (consider for instance schemes where residues with similar molecular weights are grouped together). In fact, much more complex grouping strategies could be created where the residues in a window are not individually mapped to group labels but are used to create more complex group vectors. This makes it possible to have more (or fewer) group labels than original residue types. One objective of this dissertation is to find a mapping function that optimizes performance of the prediction algorithm.

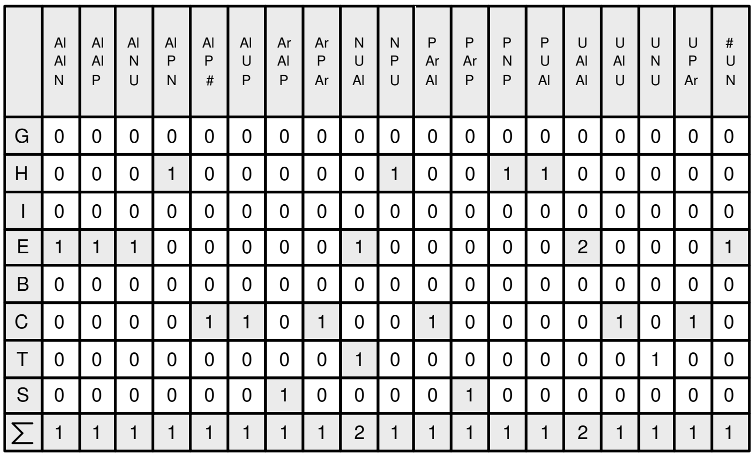
3.2.3 Step 3: Deciding on a Feature Variable

The next step is to decide on the feature variables that will be associated with each group vector - secondary structure pairing. The feature variables represent those distinguishing features in the training set that the prediction of secondary structures in the testing set will be based on.

Although any number of different types of feature variables can be used, it was decided that the only feature variable that will be considered is the secondary structures that occur in the training set (or more precisely, the number of times that a given secondary structure occurs in association with a group vector).

3.2.4 Step 4: Creating the Database

In this step, a database is created which associates each unique group vector with the set of feature variables (in this case, the number of occurrences of each secondary structure for the particular group vector). Figure 3.6 shows the corresponding database that results for this example. Since there is only a small amount of training data, most group vectors are associated with a single secondary structure. The exceptions are NUAl, where both the secondary structures E and T are found once, and UAlAl, where the secondary structure E occurs twice (and is thus considered "strong" evidence, relative to the other data in the database). If more training data were available, it should be obvious that the database matrix would be much less sparsely populated (depending on the grouping strategy of course).



	Al Al N	Al Al P	Al N U	Al P N	Al P #	Al U P	Ar Al P	Ar P Ar	N U Al	N P U	P Ar Al	P Ar P	P N P	P U Al	U Al Al	U Al U	U N U	U P Ar	# U N
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	1	1	1	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	1
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0
T	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
S	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
$\sum$	1	1	1	1	1	1	1	1	2	1	1	1	1	1	2	1	1	1	1

Figure 3.6: Database of scoring values

3.2.5 Decisions required in the training phase

This concludes the training phase. The decisions that need to be taken in the training phase are:
- choice of window structure (N and corresponding l and r),
- the grouping strategy and mapping scheme (L), and
- the choice of feature variables.

3.3 PREDICTION

In the prediction phase, the database that was created in the training phase is used to predict the secondary structures that should be associated with a sequence of amino acid residues.

The example that was started in the previous section will be continued in this section. A prediction will be made for the residue sequence LINHA. Note that amino acid residues L and H did not occur in the training data, yet a prediction will be made for the sequence.

3.3.1 Steps 1 and 2: Extracting Windows and Assigning Groups to Windows

As was the case with the training data, the first two steps are to extract windows and assign group vectors to the windows. Figure 3.7 illustrates how this process would take place for the residue sequence LINHA.

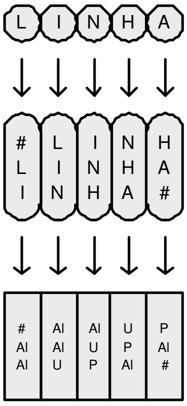


Figure 3.7: Extracting windows and assigning group vectors to an example input pattern

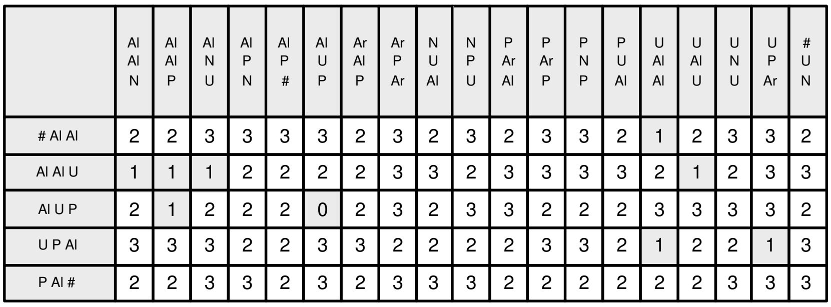
Note that the same window structure (N, l and r), grouping strategy and mapping function (L) is used in the training and prediction phases. Also note the insertion of edges in the windows of the prediction data.

3.3.2 Step 3: Distance Metric

In order to compare the group vectors in the database with the group vectors that are assigned in the prediction phase, a distance metric (d) is required. The distance metric gives an indication of how "near" or similar one group vector is to another group vector. The idea is that group vectors that are near each other in the group space should prefer to form the same secondary structures.

An example of an elementary distance metric is one that simply counts the number of differences in corresponding group labels in the group vector. The minimum distance between two group vectors is 0 (in the case that the two group vectors are exactly the same) and the maximum distance is equal to the number of group labels in the group vector (in the case that the two group vectors differ in every group label).

The distance between each group vector in the prediction data to every group vector in the database is now calculated based on the metric. Figure 3.8 tabulates the distances for the example data using the elementary distance metric defined in the previous paragraph.



	Al Al N	Al Al P	Al N U	Al P N	Al P #	Al U P	Ar Al P	Ar P Ar	N U Al	N P U	P Ar Al	P Ar P	P N P	P U Al	U Al Al	U Al U	U N U	U P Ar	# U N
# Al Al	2	2	3	3	3	3	2	3	2	3	2	3	3	2	1	2	3	3	2
Al Al U	1	1	1	2	2	2	2	3	3	2	3	3	3	3	2	1	2	3	3
Al U P	2	1	2	2	2	0	2	3	2	3	3	2	2	2	3	3	3	3	2
U P Al	3	3	3	2	2	3	3	2	2	2	2	3	3	2	1	2	2	1	3
P Al #	2	2	3	3	2	3	2	3	3	3	2	2	2	2	2	2	3	3	3

Figure 3.8: Distance table between the group vectors in the database and the group vectors in the prediction set

There is potentially much to gain by using more complex distance metrics. Such distance metrics may for instance be based on a matrix that defines distances between individual group labels and/or assigns weights to contributions of group labels at different positions within a group vector. One of the objectives of this dissertation is to find a suitable distance metric.

For each group vector for which a prediction needs to be made, the group vectors in the database that are near enough to it are retained. This is done by eliminating all the group vectors in the database that have a distance greater than a certain value. This value will be called epsilon (ϵ). The features of the group vectors that survive the elimination process will be used to classify the secondary structures of the prediction group vectors. Figure 3.9 shows the group vectors that were retained for the example case, with their associated feature variables (which the reader would recall is the number of occurrences of each secondary structure). An epsilon value of 1 was used.

Of particular interest in the example is that there is no group vector in the database that is within a distance 1 from the group vector PAl#. Also note from Figure 3.8 that the group vector AlUP for which a prediction needs to be made also occurs in the database.

# Al Al	
	U Al Al
G	0
H	0
I	0
E	2
B	0
C	0
T	0
S	0
∑	2

Al Al U				
	Al Al N	Al Al P	Al N U	U Al U
G	0	0	0	0
H	0	0	0	0
I	0	0	0	0
E	1	1	1	0
B	0	0	0	0
C	0	0	0	1
T	0	0	0	0
S	0	0	0	0
∑	1	1	1	1

Al U P		
	Al Al P	Al U P
G	0	0
H	0	0
I	0	0
E	1	0
B	0	0
C	0	1
T	0	0
S	0	0
∑	1	1

U P Al		
	U Al Al	U P Ar
G	0	0
H	0	0
I	0	0
E	2	0
B	0	0
C	0	1
T	0	0
S	0	0
∑	2	1

P Al #	
G	
H	
I	
E	
B	
C	
T	
S	
∑	0

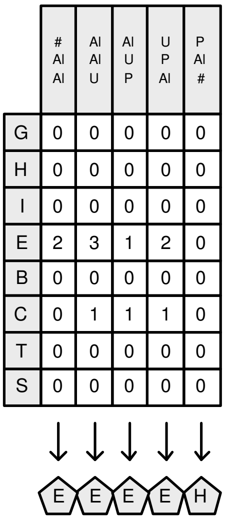
Figure 3.9: Feature variables in the database that contribute to the prediction

3.3.3 Step 4: Classification Function

From the set of retained group vectors in the database that are “near” enough to the group vector for which a prediction needs to be made, a score matrix is created. The scores in the matrix are an indication of the belief that a certain feature (in this case secondary structure) is associated with the prediction group vector.

The function that assigns the scoring matrix is known as the classification function (ϕ). The classification function can be based on a number of attributes of the retained group vectors: the number of times a particular group vector occurs, the number of times a particular group vector has a certain feature variable and/or the distance of of these group vectors to the prediction group vector. The rationale behind the classification function is that it allows different aspects of the feature variable and group vectors to be included in the creation of the score vector.

An example of a scoring matrix that results from an elementary classification function is shown in Figure 3.10. The classification function in the example simply adds the occurrences of all the secondary structures over all the group vectors that qualify. One drawback of such a scheme is that it does not take into account the distances from the retained group vectors to the prediction group vector. For instance, note in the scoring matrix that the score for both E and C for the group vector AlUP is 1 (which could mean that they are equally likely to occur). Analysis reveals that the "vote" for E was contributed by the group vector AlAlP in the database which is a distance 1 away from the AlUP while the "vote" for C was generated from the group vector AlUP in the database is obviously a distance of 0 away. It could thus be argued that it would be more probable for the secondary structure C to occur than the secondary structure E.



	# Al Al	Al Al U	Al U P	U P Al	P Al #
G	0	0	0	0	0
H	0	0	0	0	0
I	0	0	0	0	0
E	2	3	1	2	0
B	0	0	0	0	0
C	0	1	1	1	0
T	0	0	0	0	0
S	0	0	0	0	0

Figure 3.10: Assignment of secondary structures by means of a score matrix

A solution is to favour the contributions of group vectors which are nearer to the prediction group vector, perhaps by means of some weighting system. In such a case, the previous step of filtering out samples from the database above a certain ϵ value may become unnecessary, since it could be taken care of by the weighting system. Different classification functions will be considered in the dissertation.

3.3.4 Step 5: Assignment Function

The final step is to assign secondary structures to each group vector based on the score matrix. This is done by means of an assignment function (ψ).

An elementary assignment function which simply assigns the secondary structure with highest score to a group vector is illustrated in Figure 3.10. In the case where several secondary structures have the same (non-zero) score, the one with the highest prior probability of occurring is selected (as is the case where E is assigned to vector AlUP, since it has a higher prior probability of occurring than C). In the case where all secondary structures have a score of zero, the H structure is assigned, since it has the largest prior probability of all secondary structures (as is the case for group vector PAl#). An alternative to assigning H, is to flag the situation and to make no prediction.

Note that the assignment function could be made more complex. For instance, the assignment of a secondary structure to a particular group vector could depend on the scores for secondary structures next to it. For instance, the alpha helix secondary structure requires four consecutive residues to form part of the helix. In the case where a single helix secondary structure is predicted with non-helix neighbors, it may be possible to “filter out” the helix structure and replace it with another structure. Such techniques have been applied successfully in [111].

3.3.5 Decisions required in the prediction phase

The decisions that need to be taken in the prediction phase are:
- choice of distance metric (d),
- value of epsilon (ϵ),
- classification function (ϕ), and
- assignment function (ψ)

3.4 EVALUATION

In order to determine the performance of the algorithm, an evaluation phase is required. A testing set of proteins, where both the primary and secondary structure are known, is used for this purpose.

The secondary structures that are predicted in the prediction phase are compared to the actual known secondary structures of the testing set. The percentage of correctly predicted secondary structures is used as an indicator of the performance of the predictor.

3.4.1 Prediction Region

For the purposes of the dissertation, the concept of a "prediction region" will be defined. The prediction region is the set of secondary structures that will be predicted by the prediction algorithm and is determined by whether or not edges (see Section 3.2.1) are used in the prediction process.

In the case that edges are used, a secondary structure prediction will be made for every amino acid residue in the data set.

Note that the leftmost l and rightmost r windows for every protein primary sequence will contain edges. Thus, if edges are not used, those windows cannot be constructed and no secondary structure prediction can be made for the corresponding residues. If the length of the primary structure sequence is n, the prediction process for the innermost $n - l - r$ residues will remain unaffected. In this case, the prediction region is defined as this innermost $n - l - r$ secondary structures that will be predicted.

3.4.2 The Q-score

The Q-score is defined as the percentage of correctly predicted secondary structures in the prediction region (see Section 2.4.1.1).

Suppose that in the example, the real secondary structure sequence associated with the sequence LINHA is TEEHH. The predicted secondary structure sequence in the example is EEEEH. The algorithm correctly predicted secondary structure elements in positions 2, 3 and 5 and incorrectly predicted the secondary structure elements in positions 1 and 4. The algorithm thus correctly predicted 60% of the secondary structures. Since this prediction was made over 8 classes, it follows that $Q_8 = 60\%$.

3.5 PRACTICAL IMPLICATIONS

The algorithm presented in this chapter is conceptually easy to understand and should be straightforward to implement. It should be pointed out that programmatically a number of considerations should be taken into account.

For instance, the step described in Section 3.3.2 requires that every group vector pattern in the database is compared to every group vector in the prediction set. This can be a computationally expensive step, especially if there are thousands of patterns or if the distance metric is complex.

Although it will not be discussed here, it should be noted that programmatic optimizations can be made to reduce the amount of computational power required to complete certain steps in the algorithms. Certain steps can also be combined with the same effect.

Chapter 4

MATHEMATICAL FORMALISATION

This chapter provides a mathematical formalisation of the concepts and algorithm described in the previous chapter.

4.1 PROTEIN STRUCTURE

Let R represent the set or alphabet of residue labels, defined by

$$R = \left\{ \begin{array}{l} \text{ala, arg, asn, asp, cys, gln, glu, gly, his, ile, leu,} \\ \text{lys, met, phe, pro, ser, thr, trp, tyr, val, edge} \end{array} \right\}. \tag{4.1}$$

For the purposes of the mathematical explanation, the three letter abbreviations for the amino acid residues will be used in order to distinguish the residue class labels from the variables defined. In the rest of the document, the single letter abbreviations may be used as class labels, given that they are clearly distinguishable from other variables when read in context. Note that the "edge" was also defined as one of the possible residue class labels.

Let P represent the primary structure of a protein. P is a vector or string over R defined as

$$P = \overline{x} = [x_1, x_2, ...x_n], x_i \in R, \tag{4.2}$$

where x_i is the ith amino acid residue in the protein and n is the number of amino acids in the protein.

Let K represent the alphabet of secondary structure class labels. Two special instances of K are defined as

$$K^8 = \left\{ \begin{array}{c} 3_{10}\text{-helix (G)}, \alpha\text{-helix (H)}, \pi\text{-helix (I), Sheet (E)}, \\ \text{Bridge (B), Turn (T), Bend (S), Coil (C)} \end{array} \right\}, \tag{4.3}$$

and

$$K^3 = \{\text{Helix (H), Sheet (E), Coil (C)}\}. \tag{4.4}$$

The analysis in the dissertation is mostly performed with $K = K^8$. Cases where $K = K^3$ is used will be highlighted and are used mostly to compare results with the published literature.

Let S represent the secondary structure of a protein. S is a string over K defined as

$$S = \overline{y} = [y_1, y_2, ...y_n], y_i \in K, \tag{4.5}$$

where y_i is the secondary structure associated with the ith amino acid residue (x_i) in the protein.

For the sake of the analysis, a protein is thus completely characterized by the variables P and S. The notation $\overline{P}$ and $\overline{S}$ will be used to indicate the primary and secondary structures for a set of proteins, where

$$\overline{P} = [P_1, P_2, ...P_X], \tag{4.6}$$

and

$$\overline{S} = [S_1, S_2, ...S_X], \tag{4.7}$$

and X is the number of proteins in the set. The notation $\overline{P}^{train}$, $\overline{S}^{train}$, $\overline{P}^{test}$ and $\overline{S}^{test}$ will be used to indicate the primary and secondary structures of a training and testing set respectively.

Where necessary, a double subscript is used to indicate a construct associated with a specific element in a specific protein. Thus $x_{i,j}$ is the j^{th} amino acid residue of the i^{th} protein in the set and $y_{i,j}$ the j^{th} secondary structure of the i^{th} protein in the set. n_i denotes the length of the i^{th} protein.

4.2 WINDOW

Define the window $\overline{w}_i$ with leftward extension l and rightward extension r around amino acid residue x_i in primary structure P as

$$\overline{w}_i^{(l,r)} = \omega(P, i, l, r) = [x'_{i-l}, ...x'_{i+r}]; l, r \in \mathbb{N}_0, \tag{4.8}$$

where

$$x_j^{'} = \begin{cases} x_j & j \in [1, n] \\ \text{edge} & j < 1 \text{ or } j > n \end{cases} \tag{4.9}$$

The length of the window, N, is defined by

$$N = l + r + 1. \tag{4.10}$$

4.3 GROUP ASSIGNMENT

Let G represent an alphabet of group labels, defined by

$$G = \{G_1, G_2, ...G_m\}. \tag{4.11}$$

where m is the number of group labels. Let $\overline{g}$ denote a group vector, given by

$$\overline{g} = [g_1, g_2, ...g_p], g_i \in G. \tag{4.12}$$

p denotes the length of the group vector. Let L be an operator that maps a window $\overline{w}_i^{(l,r)}$ to a group window $\overline{g}_i$

$$\overline{w}_i^{(l,r)} \rightarrow^L \overline{g}_i. \tag{4.13}$$

The notation card() will be used to indicate the cardinality or number of items in a set. There can be $\text{card}(R)^N = 21^N$ different window patterns of length N and m^p different types of group vectors of length p. It is believed that a good choice is to choose m and p such that $m^p < \text{card}(R)^N$ (see Section 3.1). In most cases, p will be chosen such that $p = 1$ or $p = N$. With $p = N$ it follows that $m < \text{card}(R)$ represents a good choice.

With $p = 1$ it follows that $m < \text{card}(R)^N$ is a good choice. In this case it also expected that $m > \text{card}(R)$, thus $\text{card}(R) < m < \text{card}(R)^N$ is a likely choice.

4.4 DATABASE

The output of the training phase is a "database" of group vectors that occur in the training set with an associated set of features. Windows $\overline{w}_{i,j}^{train}$ are extracted using the window extraction function ω and mapped to group vectors $\overline{g}_{i,j}^{train}$ using the mapping operator L. The feature associated with each $\overline{g}_{i,j}^{train}$ is the secondary structure $y_{i,j}^{train}$.

Let $\overline{O}_{i,j}$ be a count vector with length equal to the cardinality of K

$$\overline{O}_{i,j} = [O_{i,j,1}, O_{i,j,2}, ...O_{i,j,\text{card}(K)}], \tag{4.14}$$

where

$$O_{i,j,k} = \sum_{r=1}^{X} \sum_{s=1}^{n_r} v_{r,s,k}, \tag{4.15}$$

and

$$v_{r,s,k} = \begin{cases} 1 & \text{if} \overline{g}_{r,s}^{train} = \overline{g}_{i,j}^{train} \text{ and } \text{index}_K(y_{r,s}^{train}) = k \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

$\overline{O}_{i,j}$ thus represents the number of times group vector $\overline{g}_{i,j}$ is found in a configuration where it is associated with the different types of secondary structures.

Let A represent the database of *unique* group vectors in the training set with their associated count vectors

$$A = \{(\overline{g}_{i,j}, \overline{O}_{i,j})\}. \tag{4.17}$$

The construction of the database concludes the training portion of the algorithm.

4.5 DISTANCE METRIC

Let δ be a distance metric that measures the distance between two group windows $\overline{g}_a$ and $\overline{g}_b$, that is

$$d_{a,b} = \delta_a(\overline{g}_b) = \delta(\overline{g}_a, \overline{g}_b). \tag{4.18}$$

The distance metric should be such that $d_{a,b} \geq 0$, $a = b \rightarrow d_{a,b} = 0$ and $d_{a,b} = d_{b,a}$. Distance metrics of interest that are considered in the dissertation are defined in the sections that follow.

4.5.1 Distance Metric 1

The distance between two group vectors, $\overline{g}_a$ and $\overline{g}_b$ is defined by

$$d^{(1)}_{a,b} = \delta^{(1)}(\overline{g}_a, \overline{g}_b) = \sum_{i=1}^{p} h_i, \tag{4.19}$$

where

$$h_i = \begin{cases} 0 & g_{a,i} = g_{b,i} \\ 1 & \text{otherwise} \end{cases} \tag{4.20}$$

4.5.2 Distance Metric 2

The distance between two group vectors, $\overline{g}_a$ and $\overline{g}_b$ is defined by

$$d_{a,b}^{(2)} = \delta^{(2)}(\overline{g}_a, \overline{g}_b) = \sum_{i=1}^{p} w_i h_i, \tag{4.21}$$

where h_i is defined by Equation 4.20 and w_i is a weight associated with h_i. Without loss of generality, w_i can be restricted to

$$w_i \in [0, 1]. \tag{4.22}$$

It should also be clear that metric 1 is a special case of metric 2. By letting $w_i = 1$ for all i, metric 1 is derived from metric 2.

4.5.3 Distance Metric 3

Let U be a matrix of dimensions card(R) by card(R) where element $u_{i,j}$ indicates a value associated with a substitution of residue type R_j with residue type R_i. This distance metric is only used under the assumption that L is the identity operator (the elements of $\overline{g}$ is thus taken from the set R). The distance between two group vectors $\overline{g}_a$ and $\overline{g}_b$ is defined by

$$d_{a,b}^{(3)} = \delta^{(3)}(\overline{g}_a, \overline{g}_b) = \sum_{i=1}^{p} w_i h_i, \tag{4.23}$$

where

$$h_i = u_{\text{index}_R(g_{a,i}),\text{index}_R(g_{b,i})}, \tag{4.24}$$

and w_i is a weight as before. It should also be noted that metric 1 and metric 2 are special cases of metric 3.

4.6 CLASSIFICATION

Prediction of the secondary structure associated with residue $x_{i,j}^{test}$ proceeds with extracting the elements in the database that are somehow "near" $\overline{g}_{i,j}^{test}$

$$A^{i,j} = \{(\overline{g}_k, \overline{O}_k) \in A | \delta(\overline{g}_k, \overline{g}_{i,j}^{test}) \leq \epsilon, \epsilon \geq 0\}. \tag{4.25}$$

Let $\alpha^{i,j}$ be the number of elements in $A^{i,j}$, that is

$$\alpha^{i,j} = \mathrm{card}(A^{i,j}). \tag{4.26}$$

Let ϕ be a classification function that assigns a score vector $\overline{s}_{i,j}$ associated with amino acid $x_{i,j}^{test}$. $\overline{s}_{i,j}$ has a length equal to the cardinality of K, and

$$\overline{s}_{i,j} = \phi(A^{i,j}). \tag{4.27}$$

4.6.1 Classification Function 1

The classifier adds the counts for all qualifying group samples in the database, given by

$$\overline{s}_{i,j}^{(1)} = \phi^{(1)}(A^{i,j}) = \sum_{(\overline{g}_k, \overline{O}_k) \in A^{i,j}} \overline{O}_k. \tag{4.28}$$

The score vector could be normalized by dividing by $\sum_{i=1}^{\mathrm{card}(K)} s_{i,j,k}$.

4.6.2 Classification Function 2

The classifier assigns a score based on the the counts of all group samples in the database with minimum distance to the group in question (even a distance of 0), given by

$$\overline{s}_{i,j}^{(2)} = \phi^{(2)}(A^{i,j}) = \sum_{(\overline{g}_k, \overline{O}_k) \in A^{i,j}} z_k \overline{O}_k, \tag{4.29}$$

where

$$z_k = \begin{cases} 1 & \delta(\overline{g}_k, \overline{g}_{i,j}^{test}) \leq \delta(\overline{g}_x, \overline{g}_{i,j}^{test}) \forall \ (\overline{g}_x, \overline{O}_x) \in A^{i,j} \\ 0 & \text{otherwise} \end{cases} \tag{4.30}$$

4.6.3 Classification Function 3

The classifier assigns a weight w_k to each $\overline{O}_k$ in the database, given by

$$\overline{s}_{i,j}^{(3)} = \phi^{(3)}(A^{i,j}) = \sum_{(\overline{g}_k, \overline{O}_k) \in A^{i,j}} w_k \overline{O}_k. \tag{4.31}$$

The weight is a function of the distance between $\overline{g}_k$ and $\overline{g}_{i,j}^{test}$

$$w_k = \xi(\delta(\overline{g}_{i,j}^{test}, \overline{g}_k)). \tag{4.32}$$

Without loss of generality, ξ can be such that w_k is restricted to

$$w_k \in [0, 1]. \tag{4.33}$$

It should be noted that classification functions 1 and 2 can be derived from classification function 3 as special cases.

4.6.4 Classification Function 4

Let m be the distance of the nearest element in the database to $\overline{g}_{i,j}^{test}$, that is

$$m = \delta(\overline{g}_k, \overline{g}_{i,j}^{test}), \tag{4.34}$$

for some k, where

$$\delta(\overline{g}_k, \overline{g}_{i,j}^{test}) \leq \delta(\overline{g}_x, \overline{g}_{i,j}^{test}) \forall\ (\overline{g}_x, \overline{O}_x) \in A^{i,j}. \tag{4.35}$$

The classification function is given by

$$\overline{s}_{i,j}^{(4)} = \phi^{(4)}(A^{i,j}) = \sum_{(\overline{g}_k, \overline{O}_k) \in A^{i,j}} z_k \overline{O}_k, \tag{4.36}$$

where

$$z_k = \begin{cases} 1 & \delta(\overline{g}_k, \overline{g}_{i,j}^{test}) \leq m + d \\ 0 & \text{otherwise} \end{cases} \tag{4.37}$$

for a chosen value d.

4.6.5 Classification Function 5

Let m be the distance of the nearest element in the database to $\overline{g}_{i,j}^{test}$, as given by Equation 4.34.

The classification function is given by

$$\overline{s}_{i,j}^{(5)} = \phi^{(5)}(A^{i,j}) = \sum_{(\overline{g}_k,\overline{O}_k)\in A^{i,j}} z_k\overline{O}_k, \tag{4.38}$$

where

$$z_k = \begin{cases} 1 & \delta(\overline{g}_k, \overline{g}_{i,j}^{test}) \leq m \times c \\ 0 & \text{otherwise} \end{cases} \tag{4.39}$$

for a chosen value c.

4.7 ASSIGNMENT

Let $y_{i,j}^{',test} \in K$ be the predicted class label associated with amino acid $x_{i,j}^{test}$. Let ψ be an assignment function that maps a vector of score vectors $\overline{s}_{i,j}$ for protein with primary structure P_i to a set of predicted labels $y_{i,j}^{',test}$

$$y_{i,j}^{',test} = \psi(\{\overline{s}_{i,k}, k \in [1, n_i]\}). \tag{4.40}$$

Assignment of $y_{i,j}^{',test}$ is thus dependent on the score vectors over the whole protein P_i. This makes it possible to apply “smoothing” techniques. For instance, if a single alpha helix secondary structure is initially predicated for $y_{i,j}^{',test}$ using a maximum likelihood predication based on $\overline{s}_{i,j}$, but different adjacent secondary structures is predicted, the adjacent score vectors can be analysed to change the predication of $y_{i,j}^{',test}$.

A specific simplifying case is to let $y_{i,j}^{',test}$ be dependent on $\overline{s}_{i,j}$ only, that is

$$y_{i,j}^{',test} = \psi_{simp}(\overline{s}_{i,j}). \tag{4.41}$$

In this case, a suitable choice is

$$\psi_{simp}^{(1)}(\overline{s}_{i,j}) = arg_K(argmax(\overline{s}_{i,j})). \tag{4.42}$$

4.8 EVALUATION

The Q-score defined in Section 2.4.1.1 will be used for evaluation. The Q-score is redefined in this section in terms of the variables defined in this chapter. Define

$$z_i = \begin{cases} 1 & y_i = y_i^{'} \\ 0 & \text{otherwise} \end{cases} \tag{4.43}$$

Define the Q-score for a protein with secondary structure S and length n as

$$Q(S) = \frac{\sum_{i=1}^{n} z_i}{n}. \tag{4.44}$$

Define Q^* as

$$Q^*(S) = \frac{\sum_{i=1}^{n} z_i}{n - n^{'}}, \tag{4.45}$$

where $n^{'}$ is the number of secondary structures for which no prediction was made. In the case of $K = K_8$ we will refer to Q_8 and Q_8^* and in the case of $K = K_3$ we will refer to Q_3 and Q_3^*.

For a set of proteins with secondary structures $\overline{S}$, the Q-score is defined in terms of the total number of correctly identified secondary structures over all the proteins

$$Q(\overline{S}) = \frac{\sum_{i=1}^{X} \sum_{j=1}^{n_i} z_{ij}}{\sum_{i=1}^{X} n_i}. \tag{4.46}$$

Similarly,

$$Q^*(\overline{S}) = \frac{\sum_{i=1}^{X} \sum_{j=1}^{n_i} z_{ij}}{\sum_{i=1}^{X} (n_i - n_i^{'})}. \tag{4.47}$$

4.9 VARIABLES DEFINED

The variables defined in this chapter are summarized in Table 4.1.

4.10 OBJECTIVE

The objective is to find l, r, G, L, δ, ϵ, ϕ and ψ for that for a general P and associated S maximizes the value of Q_8(S).

Specifically, the three research questions addressed are:

- How to group amino acid residues? (l, r, G, L)
- How to measure the distance between group vectors? (δ, ϵ)
- How to classify and assign secondary structures based on distance metrics and score vectors? (ϕ, ψ)

Table 4.1: Summary of Variables Defined

Variable	**Description**
R	Set of residue labels
K	Set of class labels
K^8	Set of class labels (8 classes)
K^3	Set of class labels (3 classes)
x	Single amino acid residue
y	Single secondary structure
P or $\overline{x}$	Primary structure of a protein
S of $\overline{y}$	Secondary structure of a protein
n	Length of a protein
$\overline{P}$	Set of primary structures
$\overline{S}$	Set of secondary structures
X	Number of proteins in a set
l	Leftward extension of a window
r	Rightward extension of a window
$\overline{w}^{(l,r)}$	Window
N	Length of a window
ω	Window extraction function
G	Set of group labels
m	Number of group labels
g	Group label
$\overline{g}$	Group vector
p	Length of a group vector
L	Mapping between window and group vector
$\overline{O}$	Count vector
A	Database of group vectors with associated count vectors
$d_{a,b}$	Distance between two group vectors
δ	Distance metric
w	Weight
ϕ	Classification function
$\overline{s}$	Score vector
ξ	Weight function
ψ	Assignment function
Q	Q-score
Q^*	Q-star-score
$n^{'}$	Number of residues for which no prediction was made

Chapter 5

RESULTS

5.1 INTRODUCTION

This chapter presents the results obtained by the algorithm described in chapters 3 and 4 as well as other experiments that were conducted. The results are not necessarily described in the chronological order that they were executed, nor are all the experiments that were conducted described in this chapter. Rather, the most prominent experiments were selected and are described in such a way that it forms a "natural progression".

Section 5.2 describes the data that was used in the experiments. The rest of the sections in this chapter each describes a series of experiments that were conducted. The experiments can be divided into three main sets:

1. The first category of experiments deals with the general properties of the data that is being analysed. In the "prior probabilities" experiment (Section 5.3.1), the prior probabilities for the different amino acid residues and secondary structures as well as their joint probabilities are determined. It is shown that certain amino acids have an affinity for certain secondary structures, although this affinity is not strong. These results explain the limited success that was obtained using first and second generation methods, as summarised in Chapter 2. The residue prior probabilities are also compared to the probabilities as expected from the

genetic code. It is noted that in some cases there may be influences of natural selection acting on the probability with which certain amino acids form.

In the "structure lengths" experiment (Section 5.3.2), statistics about the lengths of different secondary structure elements are gathered and discussed. Alpha helices and beta sheets form the longest chains of consecutive sequences and can be seen as the main structural components of proteins. The other types of secondary structures are typically short in length.

In the "edge analysis" experiment (Section 5.3.3), an analysis is made of the amino acid residues and secondary structures that occur most regularly at the edges of a protein. The coil secondary structure is almost always found at the edges of a protein. There is also evidence to suggest that methionine occurs more regularly than expected at the start of protein sequences.

2. The second category of experiments deals more specifically with the properties of the mapping from the sequence of amino acid residue types to the sequence of secondary structures.

   In the "window structure" experiment (Section 5.4.1) and the subsequent "varying window size" experiment (Section 5.4.2) it is shown that larger window sizes should theoretically have more predictive power than smaller window sizes. This is practically limited by the amount of training data available, since an enormous amount of training data would be required to completely cover all the possible amino acid combinations that could be observed for the larger window sizes. A method thus needs to be devised by which to compare different amino acid sequences and to use "similar" sequences to make a prediction. All the subsequent experiments deal with multiple facets of this problem. These two experiments also show that the information about which secondary structure would form for a particular sequence of amino acids is distributed across the whole window, although there is a tendency for more central amino acids to contribute more to the secondary structure. The so called "transfer phenomenon" is observed and an attempt made at explaining it.

   How to combine different predictions based on the non-similar target and training sequences is investigated in the "classification function" experiment (Section 5.4.3). It is shown that indeed a large performance benefit can be achieved by using non-similar sequences and larger windows. However, it is uncertain how many such sequences should be allowed to contribute to a single prediction. This question is resumed in a later experiment after a refinement of the meaning of "similarity" between sequences has been made.

3. The third category of experiments aims to develop algorithms in which the mapping between a sequences of amino acids residues and the secondary structure can be studied in detail.

   Amino acid residue types that behave similarly are identified in the "grouping strategies" experiment (Section 5.5.1). The findings are consistent with findings that have been made in the literature, although the means by which the results are achieved are unique. Although the experiment indicates that different amino acids behave similarly, it does not show the degree to which they do so.

   In the "substitution matrix" experiment (Section 5.5.2), the degree to which different amino acids behave similarly is quantified. The experiment supports the findings made in the previous experiment, but does show that substitution between two amino acids is not totally commutative, i.e. if amino acid A can be substituted with amino acid B in a particular sequence, it does not necessarily imply that amino acid B can be substituted with amino acid A.

   The substitution matrix is used to develop a distance metric in the "distance metric - substitution" experiment (Section 5.5.3). The resulting performance is comparable to the best performance achieved in previous experiments; however much fewer similar sequences (under the new distance metric) are required to achieve this performance. Another distance metric based on the BLOSUM substitution matrix is developed in the "distance metric - BLOSUM" experiment (Section 5.5.4) and achieves similar performance.

   Given the new distance metric developed in the "distance metric - substitution" experiment, a new look is taken at the classification function in the "adaptive classification function" experiment (Section 5.5.5). It is found that the number of similar sequences that should contribute to the prediction of a particular target sequence depends on the distances of those sequences to the target sequence. A method that considers neighbours (a pattern recognition term that will be used to describe similar sequences) in a band of similarity values (dependent on the nearest neighbour to a particular target sequence) works well and achieves performance comparable to other methods found in the recent literature.

   An attempt is made at incorporating predicted secondary structure information in the prediction process in the "use of secondary structure information" experiment (Section 5.5.6). It is shown that the secondary structure information is predictive of other secondary structures, but that it is difficult to incorporate this information to achieve significantly better performance scores.

5.2 EXPERIMENTAL ENVIRONMENT

5.2.1 Data Used for Analysis

The data set used in this chapter is based on the data set used in the ground-breaking paper of Jones on position-specific scoring matrices [78]. The original data set consists of 2245 proteins, containing a total of 464122 amino acids.

Analysis of this data set revealed that some of these proteins contained regions with unknown amino acids. Some amino acids also had associated secondary structures which do not belong to one of the eight standard DSSP [52] classes. The proteins where such anomalies occurred were filtered out of the data set. This reduced data set contains 1873 proteins, with a total of 358307 amino acids.

This reduced data set was arbitrarily divided into a training set, containing 1494 proteins and 285320 amino acids, and test set, containing 379 proteins and 72987 amino acids.

5.2.2 Classification Scheme used for Analysis

The standard DSSP code was used as the classification scheme. The performance scores are expressed as Q_8 values unless otherwise noted. In some instances Q_3 scores are mentioned, typically for comparative purposes. These scores are computed by making a prediction using the eight class scheme, mapping it to three classes using Table 2.3 and calculating the score.

Some early experiments conducted (not described in this chapter) indicated that it is possible to first map the eight classes to three classes using Table 2.3, then to make a prediction and then calculate the Q_3 score. These scores are typically slightly higher than the Q_3 scores achieved in the previous paragraph. However, no such scores are presented in this chapter, since the main aim is prediction in the eight class problem.

5.2.3 Computer Programs

Computer programs were written in the C# language to process the protein data and to analyse the results. An object-oriented programming methodology was followed. In particular, computer programs were developed to gather information and classify secondary structures based on the algorithm explained in chapter 3 and mathematically described in chapter 4. Additional tests were also performed on the data.

Some of the experiments, but in particular the “grouping strategies”, “classification function” and “use of secondary structure information” experiments required considerable computing power (many computer weeks), due to the iterative nature or large number of tests that were conducted. The resulting information that was extracted can however be used to create fast and efficient algorithms that predict secondary structures relatively quickly. The actual computer programs and algorithms will not be further mentioned in the remainder of this chapter.

5.3 GENERAL PROPERTIES OF PROTEINS

5.3.1 Experiment: Prior Probabilities

5.3.1.1 Objective

The objective of this experiment is to determine the prior probabilities of the different amino acid residue types and secondary structures in order to gain some intuition about the problem.

5.3.1.2 Protocol

Computer programs were written to determine:
- The prior probabilities of the different amino acid residue types.

- The prior probabilities of the different secondary structures.
- The amino acid - secondary structure joint probabilities.

The computer programs were executed on the training set. The rationale behind this decision (not to include the testing set as well) was that if the statistics were to be used in classification algorithms, the classification algorithms would not be biased toward the test data.

5.3.1.3 Results and Discussion

Table 5.1 shows the prior and joint probabilities for the different amino acid residue types and secondary structures expressed as percentages. The most frequently occurring amino acid is Alanine in 8.19% of the samples and the least frequently occurring amino acid is Tryptophan in 1.53% of the samples.

The frequencies of occurrence of the amino acid residue types in the reduced Jones data set were compared to those of a similar study by Doolittle [112] containing a set of 1150 proteins. The correlation between the two data sets is 0.9786, indicating that these frequencies are fairly stable across different data sets.

The probability of occurrence of the DNA bases in nature are: Uracil - 22.0%, Adenine - 30.3%, Cytosine 21.7% and Guanine - 26.0% [123]. Based on these probabilities and the genetic code (refer to Section 2.1.4), filtering out the 3 codons mapping for stop sequences, the expected probabilities of occurrence of the different amino acids were also calculated. These probabilities, together with the frequencies of occurrence of the amino acid residue types in the Jones and Doolittle data sets are illustrated in Figure 5.1.

The correlation between the Jones data set and the probabilities of occurrence based on the genetic code is 0.6977 and is illustrated by the scatter diagram in Figure 5.2. The only real outlier is the amino acid Arginine, which occurs in only 4.50% of the amino acids in the Jones data set, whilst it is expected to occur in 10.66% of the samples when based on the genetic code. One explanation could be that the Arginine frequency is the product of natural selection acting on one or more of the codons coding for it. When the

Table 5.1: Prior and joint probabilities of amino acids and secondary structures in the reduced Jones data set

Structure →		3_{10} helix	α helix	π helix	β sheet	β strand	Turn	Bend	Coil	Total
Amino Acid ↓		G	H	I	E	B	T	S	C	
Alanine	A	0.3298	3.4740	0.0007	1.3830	0.0761	0.7998	0.6211	1.5074	8.1919
Arginine	R	0.1479	1.5972	0.0007	0.9000	0.0596	0.4760	0.4234	0.8909	4.4957
Asparagine	N	0.1945	0.9645	0.0011	0.6228	0.0662	0.9102	0.6119	1.2092	4.5805
Aspartic	D	0.3168	1.4254	0.0018	0.6855	0.0680	0.9095	0.7763	1.5880	5.7714
Cysteine	C	0.0690	0.4003	0.0014	0.6011	0.0473	0.1563	0.1661	0.5156	1.9571
Glutamine	Q	0.1360	1.3641	0.0000	0.7360	0.0382	0.4129	0.3582	0.7413	3.7866
Glutamic Acid	E	0.3021	2.3675	0.0014	0.9729	0.0424	0.6957	0.5674	0.9305	5.8801
Glycine	G	0.2408	1.0318	0.0011	1.1180	0.0747	2.1769	1.5169	1.7531	7.9132
Histidine	H	0.0985	0.5779	0.0007	0.4987	0.0365	0.2646	0.2366	0.4879	2.2014
Isoleucine	I	0.1104	1.6736	0.0004	2.0314	0.0960	0.2187	0.2916	0.8198	5.2418
Leucine	L	0.2751	3.1515	0.0021	2.1944	0.1272	0.5562	0.4858	1.3599	8.1523
Lysine	K	0.2229	1.9515	0.0007	1.1436	0.0796	0.7728	0.6162	1.1731	5.9603
Methionine	M	0.0617	0.7700	0.0007	0.4823	0.0235	0.1332	0.1405	0.3796	1.9914
Phenylalanine	F	0.1511	1.1156	0.0011	1.2628	0.0929	0.2979	0.2716	0.7167	3.9096
Proline	P	0.2450	0.5450	0.0011	0.4500	0.0470	0.9169	0.5422	2.0034	4.7505
Serine	S	0.3224	1.3346	0.0007	1.3781	0.0939	0.9547	0.8825	1.7352	6.7023
Threonine	T	0.1630	1.2298	0.0007	1.8232	0.1101	0.5811	0.6680	1.5775	6.1534
Tryptophan	W	0.0680	0.4409	0.0011	0.4966	0.0315	0.1241	0.1101	0.2583	1.5306
Tyrosine	Y	0.1465	0.9610	0.0007	1.2614	0.0841	0.3228	0.2888	0.6645	3.7298
Valine	V	0.1276	1.8782	0.0035	3.0089	0.1220	0.3277	0.4013	1.2309	7.1001
Total		3.7291	28.2546	0.0214	23.0510	1.4167	12.0079	9.9765	21.5428	100.0000

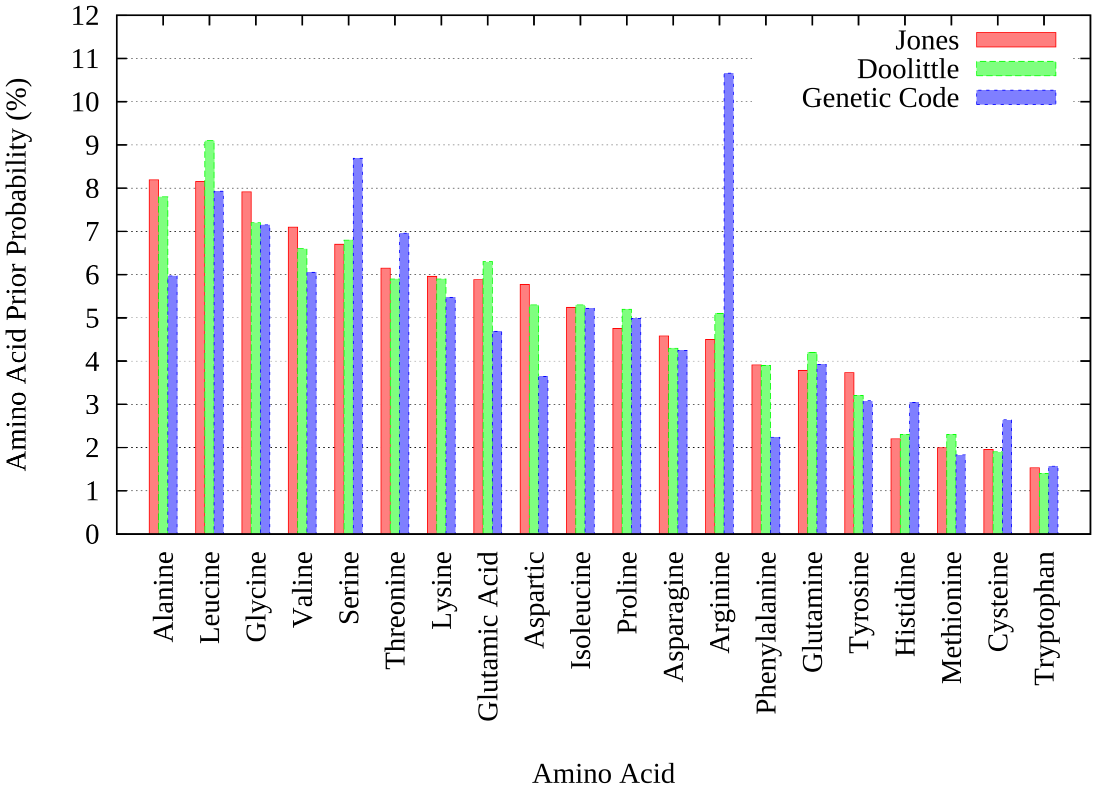


Figure 5.1: Frequency of occurrence of the different amino acid residue types in the Jones and Doolittle data sets and the probability as calculated based on the genetic code

Arginine frequency is excluded from the data set, the correlation coefficient is 0.8749. A reasonable conclusion may thus be that (except in the case of Arginine) the prior probabilities of amino acids are simply determined by the probability of occurrence of the codons coding for it.

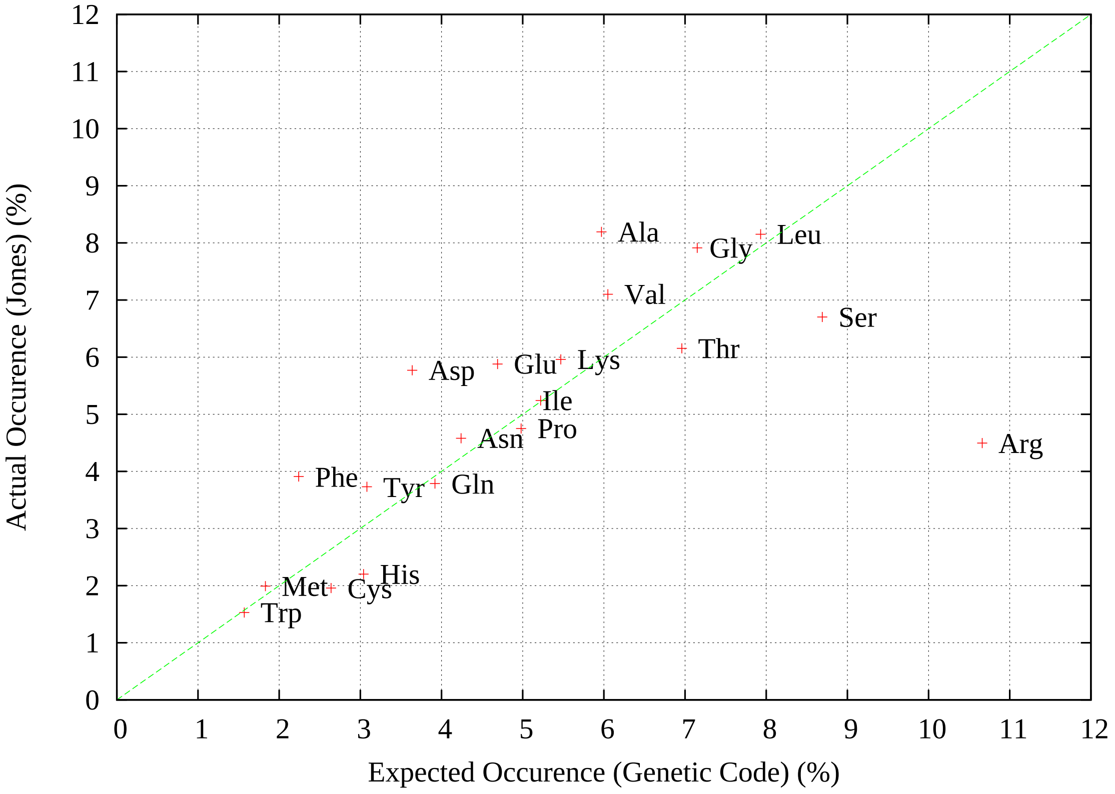


Figure 5.2: Scatter diagram of the expected and observed probability of occurrence of the amino acids in the Jones data set

Figure 5.3 illustrates a similar scatter diagram for the Doolittle data set. The scatter diagram takes a similar form to the one for the Jones data set. The correlation coefficient between the Doolittle data set and the probabilities of occurrence based on the genetic code is 0.7474. In the case that Arginine is left out of the correlation calculation, the correlation coefficient is 0.8880.

The α helix, β sheet and coil secondary structures are the most abundant at 28.25%, 23.05% and 21.54% respectively whilst the π helix occurs in only 0.02% of the samples, as is illustrated in Figure 5.4.

If the joint probabilities in Table 5.1 are carefully observed, it will be noticed that

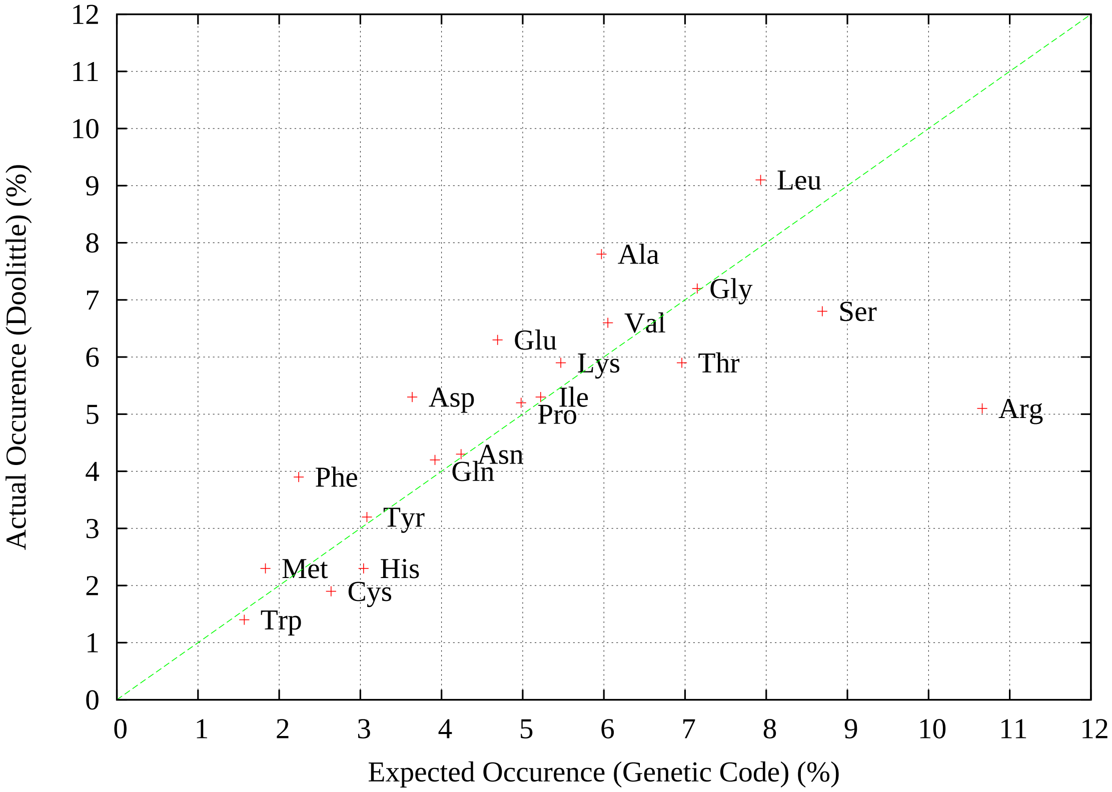


Figure 5.3: Scatter diagram of the expected and observed probability of occurrence of the amino acids in the Doolittle data set

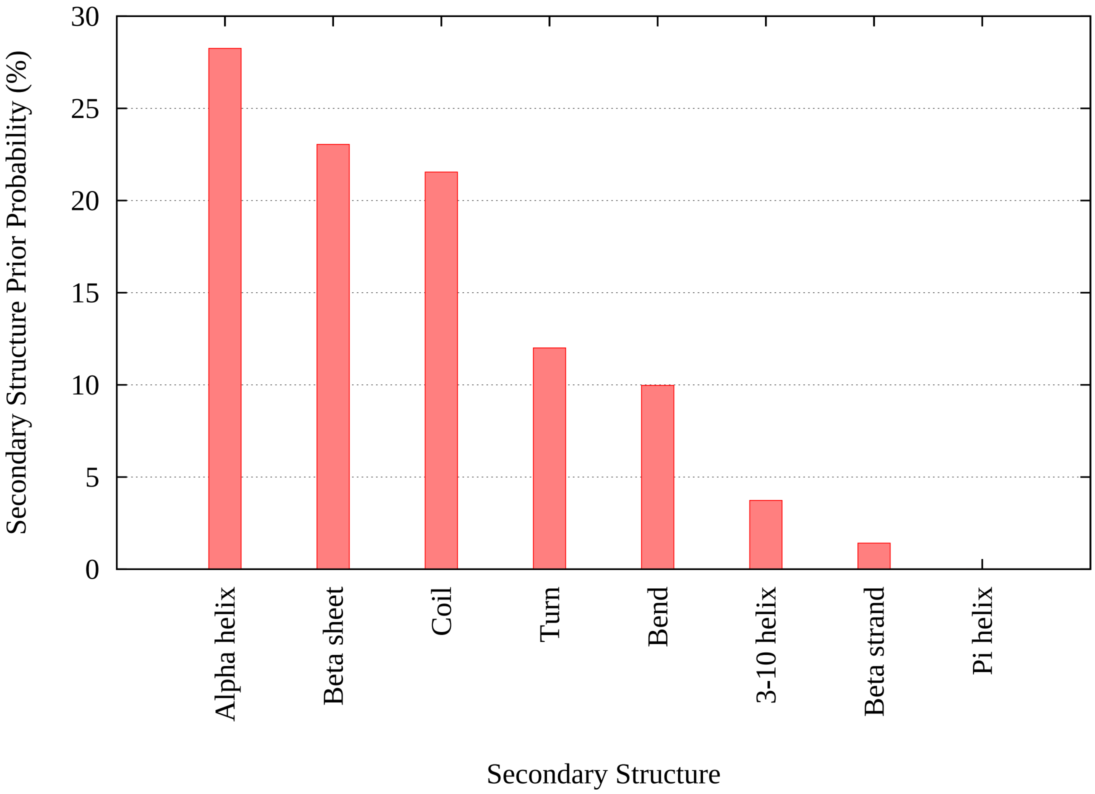


Figure 5.4: Secondary Structure Prior Probability

certain amino acids are more likely to form certain secondary structures than others (if only the joint probability is considered). This preference of amino acids to form certain secondary structures is shown in Table 5.2.

Table 5.2: Preference of Amino Acid Residues to form Secondary Structures

α **helix**	β **sheet**	**Coil**	**Turn**
Alanine	Cysteine	Asparagine	Glycine
Arginine	Isoleucine	Aspartic	
Glutamine	Phenylalanine	Proline	
Glutamic Acid	Threonine	Serine	
Histidine	Tryptophan		
Leucine	Tyrosine		
Lysine	Valine		
Methionine			

If the decision of which secondary structure $y_{i,j}$ to assign to amino acid j in protein i was based solely on the observation of $x_{i,j}$, the best possible classifier is the naive Bayesian classifier

$$y_{i,j} = argmax_K(P(K_k|x_{i,j})), \tag{5.1}$$

where

$$P(K_k|x_{i,j}) = \frac{P(x_{i,j}|K_k)P(K_k)}{P(x_{i,j})} = \frac{P(x_{i,j} \cap K_k)}{P(x_{i,j})}, \tag{5.2}$$

according to Bayes' theorem. The expected number of correctly classified secondary structures in this case is

$$\sum_{i=1}^{X}\sum_{j=1}^{n_i} max_K(P(K_k|x_{i,j}))P(x_{i,j}). \tag{5.3}$$

For the training data, the expected number of correctly classified secondary structures is calculated as 34.45%. Applying the naive Bayesian classifier to the testing data, it was found that 34.15% of the secondary structures were correctly assigned. This signifies a limited improvement over assigning the secondary structure with the highest prior probability of occurring (α-helix at 28.25%) (which in turn is significantly better than randomly assigning a secondary structure (12.5%)).

The conclusion from this result is that some information as to which secondary structure will form for an amino acid is contained within the residue type. However, a large portion of the information is not determined by the amino acid and is thus influenced by other processes or structures. The experiments that follow will investigate the extent to which small sequences of amino acids contribute to the formation of certain secondary structures.

5.3.1.4 Conclusion

The reduced Jones data set has similar attributes to other data used in the literature.

The probabilities of occurrence of the different amino acid residues (with the exception of Arginine) seem to be based fairly closely on the probability of occurrence of the different DNA bases and the codons coding for them. In the case of Arginine the frequency of occurrence may be the product of natural selection acting on one or more of the codons coding for it.

The frequencies with which different secondary structures occur vary considerably. The data seems to suggest that some of the knowledge of which secondary structure is associated with which amino acid is contained within the residue type.

5.3.2 Experiment: Structure Lengths

5.3.2.1 Objective

The objective of this experiment is to gather statistics about the length of proteins and the different secondary structures.

5.3.2.2 Protocol

Computer programs were written to determine:
- Statistics about the length of the different proteins in the training set.
- Statistics about the length of the different secondary structures in the training set.

5.3.2.3 Results and Discussion

Figure 5.5 shows a histogram of the lengths of the proteins in the training data set. The average length of the 1494 proteins in the set is 190.97 with a standard deviation of 142.41. The shortest protein in the set has a length of 20 and the longest a length of 907 amino acids. The median is at 150.5 with the 25% percentile mark at a length of 85 and the 75% percentile mark at a length of 256.

From Figure 5.5 there seems to be a slight anomaly at proteins with a length of about 220. It seems that the number of proteins with these lengths are more frequent than expected.

The 1494 proteins in the training set contain 285320 amino acid residues in which 92983 sequences of consecutive similar secondary structures occur. The average secondary structure length thus spans just over 3 amino acids. Figure 5.6 shows the lengths of the different secondary structures in the training set. Table 5.3 tabulates the corresponding statistics for the secondary structures.

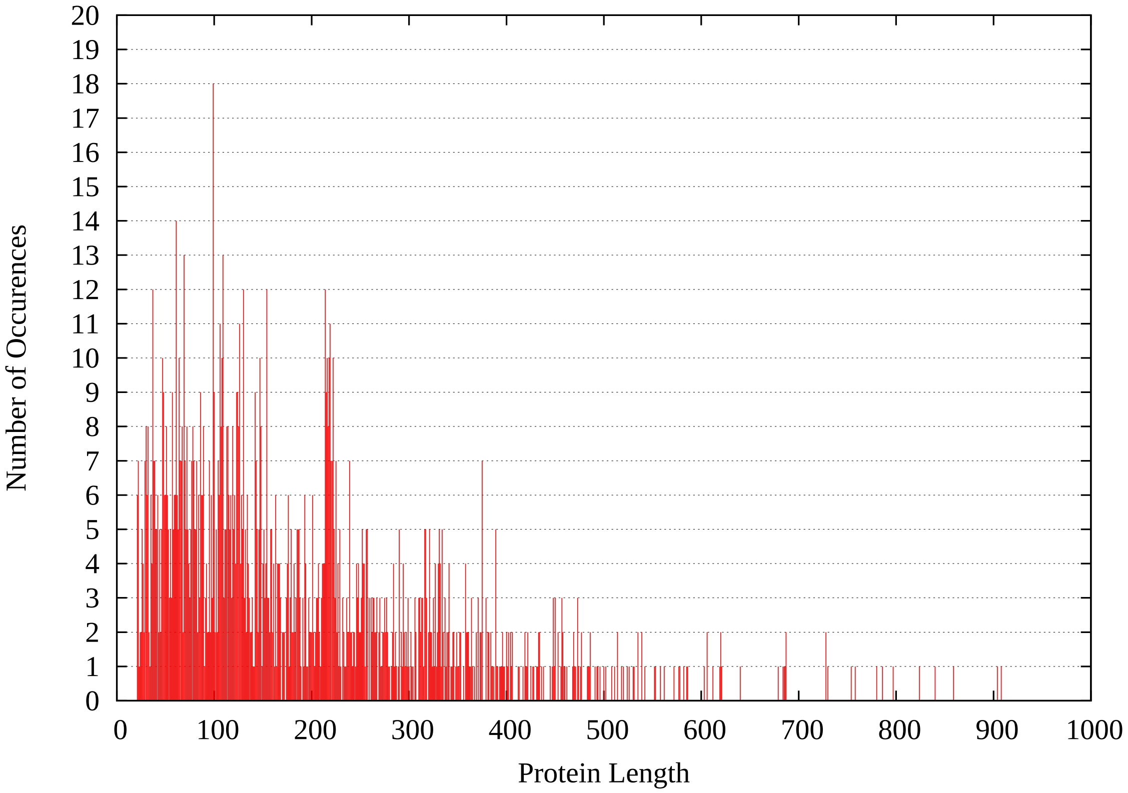


Figure 5.5: Protein lengths, as measured by the number of amino acids per protein

In the case of the 3_{10} helix, one of the samples had a length of a single amino acid. This is inconsistent with the definition of the 3_{10} helix, which requires a length of at least 3 amino acids. Similarly, there were two β sheets with length 1 (which should probably be classified as β strands or coils). For the purposes of the dissertation, it was decided not to filter the proteins containing these anomalies out of the training set.

Another observation is that there are two outliers in the case of the α helix (which is the secondary structure that forms the longest chains by far). These structures contain 109 and 107 amino acids respectively. For comparison, the third longest chain contains 67 amino acids.

Table 5.3: Structure lengths

	3_{10} helix	**α helix**	**π helix**	**β sheet**	**β strand**	**Turn**	**Bend**	**Coil**
Sequence count	3189	7359	12	12419	3958	16071	17635	32340
Minimum length	1	4	5	1	1	1	1	1
Maximum length	10	109	6	25	2	11	9	25
Average	3.3365	10.9547	5.0833	5.2958	1.0212	2.1319	1.6141	1.9006
Standard deviation	0.8385	6.0511	0.2764	2.7028	0.1441	0.8811	0.8986	1.3513

5.3.2.4 Conclusion

Apart from being the most abundant secondary structures, alpha helices and beta sheets also form the longest chains of consecutive sequences. These structures comprise the main structural elements of most proteins.

The other secondary structures tend not to form long sequences on average, but are instead rather compact. Most beta strands occur as a single secondary structure, turns typically have a length of two, whilst bends and coils usually have a length of one and sometimes two. 3_{10} helices typically have length three and π helices length five.

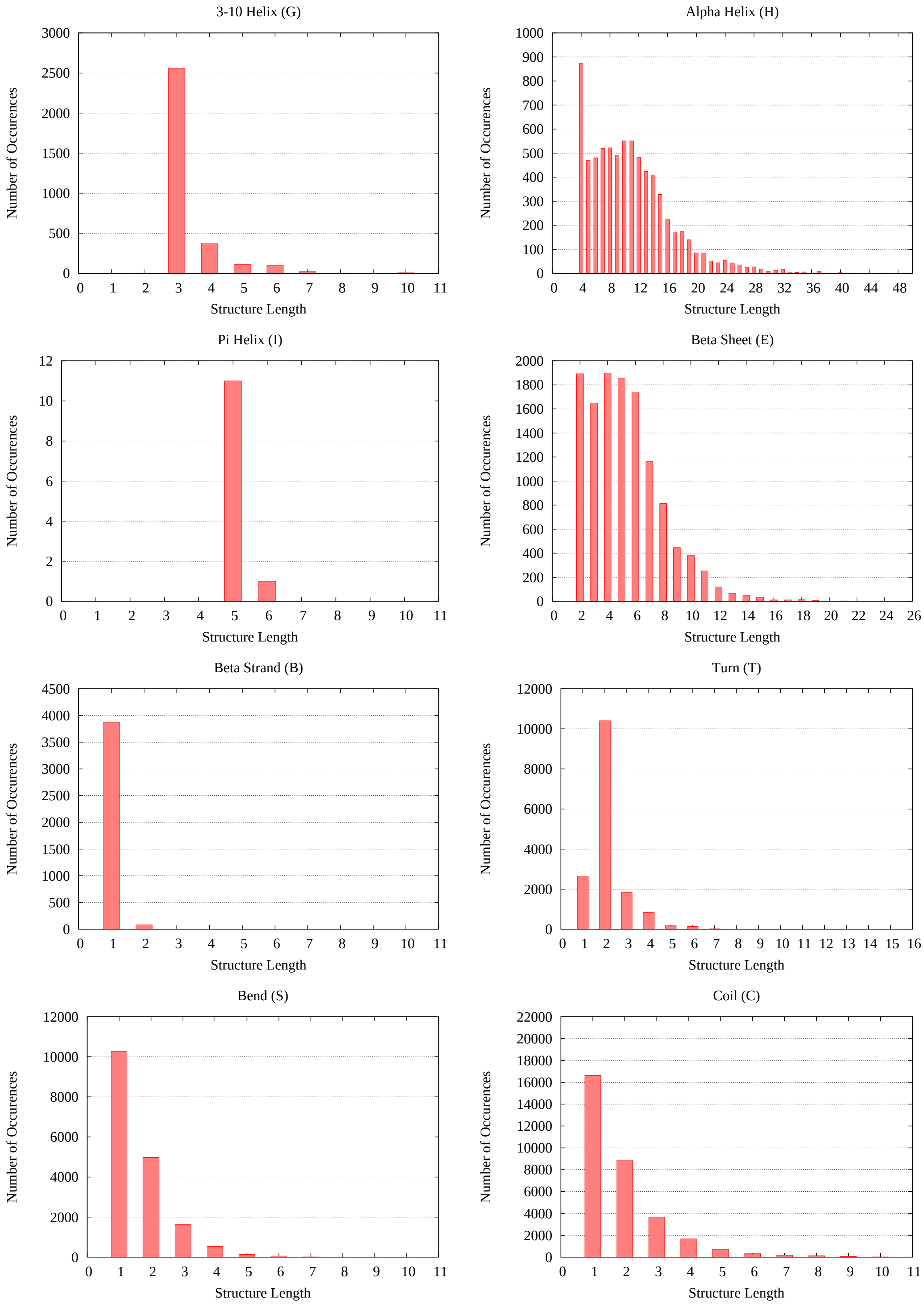


Figure 5.6: Secondary Structure Lengths

5.3.3 Experiment: Edge Analysis

5.3.3.1 Objective

The objective of this experiment is to determine whether certain secondary structures are more likely to form near the edges of a protein and to determine which amino acid residue types they are associated with.

5.3.3.2 Protocol

Computer programs were written to determine the probabilities with which different residue types are found in the different secondary structure conformations at the start and end of the proteins in the training set.

5.3.3.3 Results and Discussion

Tables 5.4 and 5.5 list the probabilities (expressed as a percentage) for each of the different amino acid residue types to be found in the different secondary structure conformations at the start and end of a protein sequence. The tables also show the percentage of occurrences with which each amino acid type was found at the start and end of the protein sequences (column 'Actual'), the expected percentage if it occurred randomly (as calculated in Table 5.1, shown in column 'Exp.') and the difference between the two (column 'Δ').

From the discussion on protein synthesis and the genetic code in Section 2.1.4, the expectation is that methionine would be the first amino acid in every protein sequence, since the start codon, AUG, codes for it. This is clearly not the case, since methionine appears at the start in only 13.65% of the protein sequences.

Meinnel [113] states that methionine is removed from most mature proteins after the translation process. This is achieved through enzymes acting on the proteins. This implies that the probabilities as given in 5.4 are the probabilities of finding the different

Table 5.4: Probability of different residue types to form different secondary structures at the start of a protein sequence

	3_{10} helix	**α helix**	**π helix**	**β sheet**	**β strand**	**Turn**	**Bend**	**Coil**	**Actual**	**Exp.**	**Δ**
Alanine	0.0000	0.4505	0.0000	0.0000	0.0000	0.0000	0.0000	99.5495	14.8594	8.1919	6.6676
Arginine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	3.0790	4.4957	-1.4167
Asparagine	0.0000	0.0000	0.0000	0.0000	0.0000	2.4390	0.0000	97.5610	2.7443	4.5805	-1.8362
Aspartic	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	5.6225	5.7714	-0.1489
Cysteine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	1.6734	1.9571	-0.2837
Glutamine	0.0000	0.0000	0.0000	1.7857	0.0000	0.0000	0.0000	98.2143	3.7483	3.7866	-0.0383
Glutamic Acid	0.0000	0.9524	0.0000	0.0000	0.0000	0.0000	0.0000	99.0476	7.0281	5.8801	1.1480
Glycine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.7042	99.2958	9.5047	7.9132	1.5915
Histidine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	0.6693	2.2014	-1.5320
Isoleucine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	2.9451	5.2418	-2.2967
Leucine	0.0000	0.0000	0.0000	1.6129	1.6129	0.0000	0.0000	96.7742	4.1499	8.1523	-4.0023
Lysine	0.0000	1.6129	0.0000	1.6129	0.0000	0.0000	0.0000	96.7742	4.1499	5.9603	-1.8104
Methionine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	13.6546	1.9914	11.6632
Phenylalanine	0.0000	0.0000	0.0000	8.3333	0.0000	0.0000	0.0000	91.6667	0.8032	3.9096	-3.1064
Proline	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.9231	98.0769	3.4806	4.7505	-1.2699
Serine	0.0000	0.0000	0.0000	1.4493	0.0000	0.0000	0.0000	98.5507	9.2369	6.7023	2.5346
Threonine	0.0000	0.0000	0.0000	1.1494	1.1494	0.0000	0.0000	97.7011	5.8233	6.1534	-0.3301
Tryptophan	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	0.5355	1.5306	-0.9951
Tyrosine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	0.9371	3.7298	-2.7928
Valine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	5.3548	7.1001	-1.7453

amino acid residue types at the start of a protein sequence due to the codons coding for it being found at the second codon position (in the case of methionine, both the first and second codon positions).

Another observation is that methionine constitutes 1.99% of the proteins in the training set (expected 1.83% as calculated by the genetic code). Even if methionine was universally removed from the start of all protein sequences, it is still expected that roughly a similar percentage of methionine amino acids would occur at the second position in a protein. However, the statistics indicate that it occurs in the first position of 13.65% of the proteins.

This implies that methionine is not universally removed from all proteins or that there is an above average expectation to find two codons coding for methionine at the start positions of a protein coding gene. One of two possible conclusions can be drawn. The first conclusion is that methionine has a special role to fulfill at the start position of some proteins apart from normal protein function. The second possible conclusion is that it may have no useful function at all and that it is simply not removed since it is not efficient to do so (this implies that it does not hamper the functioning of a protein).

Another observation is that alanine occurs at the start of 14.86% (even more than methionine) of protein sequences, 6.67% more than expected. Leucine occurs at the start of 4.15% of protein sequences, 4.00% less than expected. These results are interesting, since alanine and leucine are the most abundant amino acids in the training set. At the end of the protein chain, lysine and cysteine occur more often than expected (4.01% and 3.60% respectively).

The coil secondary structure is almost always found at the start and end of a protein sequence, as is evident from Table 5.4 and Table 5.5. It occurs at the start of 1479 and end of 1486 of the 1494 protein sequences in the training set. (One apparent exception: when Phenylalanine is the first amino acid in a protein sequence, the β sheet secondary structure appears at the start of 8.33% of such sequences. However, Phenylalanine occurs at the start of only 12 of the 1494 protein sequences and in only 1 of those cases the β sheet occurs. The apparent exception is therefore not significant.)

One explanation for finding the abundance of coils at the ends of a protein could

Table 5.5: Probability of different residue types to form different secondary structures at the end of a protein sequence

	3_{10} helix	**α helix**	**π helix**	**β sheet**	**β strand**	**Turn**	**Bend**	**Coil**	**Actual**	**Exp.**	**Δ**
Alanine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	8.2999	8.1919	0.1080
Arginine	0.0000	0.0000	0.0000	0.0000	0.0000	1.0101	0.0000	98.9899	6.6265	4.4957	2.1309
Asparagine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.1494	98.8506	5.8233	4.5805	1.2428
Aspartic	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	3.6145	5.7714	-2.1570
Cysteine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	5.5556	1.9571	3.5985
Glutamine	0.0000	1.4286	0.0000	0.0000	0.0000	0.0000	0.0000	98.5714	4.6854	3.7866	0.8988
Glutamic Acid	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.1765	98.8235	5.6894	5.8801	-0.1906
Glycine	0.0000	0.0000	0.0000	0.0000	0.0000	1.1111	0.0000	98.8889	6.0241	7.9132	-1.8891
Histidine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	2.4766	2.2014	0.2752
Isoleucine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	3.5475	5.2418	-1.6943
Leucine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	8.3668	8.1523	0.2146
Lysine	0.6711	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	99.3289	9.9732	5.9603	4.0129
Methionine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	1.8072	1.9914	-0.1842
Phenylalanine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	3.8153	3.9096	-0.0944
Proline	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	4.3507	4.7505	-0.3997
Serine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	6.6265	6.7023	-0.0758
Threonine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.9231	98.0769	3.4806	6.1534	-2.6729
Tryptophan	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	1.1379	1.5306	-0.3927
Tyrosine	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.0000	3.6145	3.7298	-0.1154
Valine	0.0000	0.0000	0.0000	0.0000	0.0000	1.4925	0.0000	98.5075	4.4846	7.1001	-2.6155

be that the ends are exposed to the surrounding environment and not buried toward the core of a protein like other structures, much like a tied shoelace, where the knot and loops represent structural components and the ends dangle freely. The coils at the ends thus have irregular structure because they do not form part of the main functional or structural units of the protein. This claim is somewhat supported by the second conjecture given above as to the high percentage of methionine found at the start of protein sequences, namely that it is not removed since it does not hamper the functioning of a protein but does not contribute to its functioning either. In fact, of the 1494 protein sequences, all 204 that started with methionine were found with a coil conformation.

5.3.3.4 Conclusion

The coil secondary structure is almost always found at the start and end of a protein.

Methionine is removed from the start of most proteins through post-translational processes, but the data seems to suggest that it is not removed in all cases. This can be attributed to the fact that it is either not necessary to do so, or that it serves a very specific purpose in the proteins in which it is not removed. It is clear however that there are very specific biological processes at work.

Alanine occurs more and leucine occurs less than expected at the start of proteins. At the end of a protein, lysine and cysteine occur somewhat more often than expected. Whether these observations are functionally significant remains to be determined.

5.4 PRIMARY TO SECONDARY STRUCTURE MAPPING

5.4.1 Experiment: Window Structure

5.4.1.1 Objective

The objective of this experiment is to determine whether different structural compositions of a window of amino acids around a central amino acid have any influence on the prediction accuracy of the algorithm explained in Chapters 3 and 4. In addition, the performance differences achieved between including and excluding the edges in the algorithm will be studied. The effect of forcing a prediction versus not forcing a prediction will be analysed.

5.4.1.2 Protocol

A series of experiments with window sizes ranging from 1 to 7 were executed ($N \in [1, 7]$). For each window size, the central amino acid was varied from the leftmost amino acid in the window ($l = 0, r = N - 1$) to the rightmost amino acid in the window ($l = N - 1, r = 0$). The set of group labels were the same as the set of residue labels, that is $G = R$, with L the identity function. $\delta^{(1)}$ was used as distance metric. The experiment was set up such that a prediction for a pattern in the test set will only be made if it is in the database, i.e. $\epsilon = 0$. $\phi^{(1)}$ was used as classification function and $\psi^{(1)}$ as assignment function.

The experiments were conducted for both the case where the edge effects are included (predictions are attempted for the whole length of the protein) and the case where edge effects are excluded (predictions are not attempted near the edges). For both these cases, an experiment was conducted where a prediction was forced over the region of interest (predict regardless of whether a similar pattern exists in the database) and the case where a prediction was not forced (no prediction is made if a pattern does not exist in the database). In the case where a prediction is forced when no patterns exist in the database, the secondary structure with highest prior probability for the observed amino acid is assigned. In the case where probabilities for multiple secondary structures are simultaneously higher than for other secondary structures, the decision is based on the secondary structure in that group that has the highest prior probability.

5.4.1.3 Results and Discussion

The results of the experiment are listed in Table 5.6 (analysis conducted taking into account edges) and Table 5.7 (edges not included). The tables present the results after different combinations of window structures were used in the prediction algorithm and applied to the test set. The tables list the percentage of correctly predicted secondary structures.

The first observation is that the training data contains all possible strings of length 1 and 2 that can be made from the different residue types. This was confirmed through independent testing, but can also be seen from the tables by observing that the results for the forced and unforced predictions are the same (which only implies that all the input patterns in the testing data are present in the training data, and not necessarily that all different types of input patterns are in the training data, which thus necessitated independent testing for the entire input space).

For window lengths of 3 and more, it is immediately obvious that all the input patterns in the testing data do not occur in the training data (due to the difference between the forced and unforced results). In the case of a window length of 3, the differences between the forced and unforced results are small, signifying that almost all the input patterns in the testing data are found in the training data. For window sizes of length 4 and more, the effect is more severe (and thus the benefit of forcing a prediction becomes more pronounced).

The reader may recall that there are 285320 different amino acid residues in the training set and thus at most the same number of different input patterns. The complete input space has 20^N distinct input patterns (or more, if edges are considered as well). For $N = 3$ this is 8000, for $N = 4$, 160000 and for $N = 5$, 3200000. Since many multiples of the same input pattern occur in the 285320 patterns in the training set, this reduces the number of distinct training samples to less than 20^N for $N = 3$ and $N = 4$. For $N = 5$ and larger, covering the complete input space is simply not possible, even if all

Table 5.6: Prediction Results for different Window Structures (with edges included)

			Forced	**Forced**	**Unforced**	**Unforced**
N	l	r	Q_8	Q_8^*	Q_8	Q_8^*
1	0	0	34.1485	34.1485	34.1485	34.1485
2	0	1	38.0383	38.0383	38.0383	38.0383
2	1	0	36.8422	36.8422	36.8422	36.8422
3	0	2	40.8648	40.8648	40.8593	40.8733
3	1	1	41.8116	41.8116	41.8074	41.8355
3	2	0	39.1056	39.1056	39.1001	39.1199
4	0	3	39.0001	39.0001	35.3912	41.0127
4	1	2	39.9345	39.9345	36.3393	42.2792
4	2	1	39.5207	39.5207	35.8968	41.7723
4	3	0	37.6999	37.6999	34.0650	39.4757
5	0	4	38.2150	38.2150	18.1594	61.6896
5	1	3	38.3137	38.3137	18.2882	63.0932
5	2	2	38.0369	38.0369	18.0416	62.9506
5	3	1	38.1808	38.1808	18.1416	62.5874
5	4	0	38.0561	38.0561	17.9580	60.9940
6	0	5	37.2231	37.2231	12.5625	83.6282
6	1	4	37.1217	37.1217	12.4324	85.4024
6	2	3	36.7449	36.7449	12.0830	85.5549
6	3	2	36.7641	36.7641	12.1282	85.7752
6	4	1	37.1025	37.1025	12.4748	85.6619
6	5	0	37.2436	37.2436	12.6543	84.1396
7	0	6	36.1585	36.1585	10.8649	86.7330
7	1	5	36.0516	36.0516	10.6937	88.7537
7	2	4	35.7173	35.7173	10.3525	89.2406
7	3	3	35.6228	35.6228	10.2772	89.2976
7	4	2	35.7242	35.7242	10.3909	89.4023
7	5	1	36.0544	36.0544	10.7457	89.0542
7	6	0	36.2585	36.2585	10.9869	87.5437

Table 5.7: Prediction Results for different Window Structures (without edges included)

			Forced	**Forced**	**Unforced**	**Unforced**
N	l	r	Q_8	Q_8^*	Q_8	Q_8^*
1	0	0	34.1485	34.1485	34.1485	34.1485
2	0	1	37.5245	37.7204	37.5245	37.7204
2	1	0	36.3270	36.5166	36.3270	36.5166
3	0	2	40.0811	40.5017	40.0770	40.5049
3	1	1	40.8347	41.2632	40.8306	41.2665
3	2	0	38.2685	38.6701	38.2657	38.6743
4	0	3	38.0821	38.6848	34.5418	40.5844
4	1	2	39.1453	39.7648	35.5803	41.8046
4	2	1	38.7398	39.3528	35.1391	41.2862
4	3	0	36.7463	37.3278	33.1456	38.9440
5	0	4	37.1162	37.9035	17.2661	61.2461
5	1	3	37.4957	38.2911	17.5935	62.4077
5	2	2	37.5245	38.3204	17.6031	62.4417
5	3	1	37.4026	38.1959	17.4278	61.8196
5	4	0	36.9956	37.7804	16.9907	60.2692
6	0	5	35.9256	36.8832	11.6349	84.7167
6	1	4	36.1434	37.1068	11.7213	85.3452
6	2	3	36.2229	37.1884	11.7336	85.4350
6	3	2	36.2763	37.2433	11.7569	85.6045
6	4	1	36.2393	37.2053	11.7377	85.4649
6	5	0	36.0557	37.0168	11.6391	84.7466
7	0	6	34.6527	35.7671	9.9072	88.4310
7	1	5	34.8939	36.0160	9.9689	88.9813
7	2	4	35.0542	36.1815	10.0059	89.3115
7	3	3	35.1405	36.2706	10.0114	89.3604
7	4	2	35.1652	36.2960	10.0127	89.3726
7	5	1	35.0788	36.2069	9.9785	89.0669
7	6	0	34.9446	36.0683	9.9346	88.6756

the different patterns in the training set were distinct.

The best forced result is obtained with a window length of $N = 3$, which achieves $Q_8 = 41.81\%$ (forcing prediction and including edges). The fact that this is the best result is not surprising, since no other mechanisms were put in place to match patterns in the training and test sets (elementary choice of functions, $G = R$, $\phi^{(1)}$, $\psi^{(1)}$, etc.). It is suspected that as more training data becomes available, better results would be achieved by larger window sizes (keeping the other variables the same), or more specifically, a window size that covers, or almost covers, the complete set of input patterns.

With a window size of $N = 1$ a performance of 34.15% was achieved. This is the same result as achieved in Section 5.3.1.3, since the application of the algorithm in this case assigns to each amino acid residue the secondary structure with highest probability of occurring according to its residue type. It is interesting that by forcing a prediction, even with a window length of $N = 7$ the performance is better than with $N = 1$.

The value $\frac{Q_8}{Q_8^*}$ indicates the fraction of secondary structures for which a prediction was made. For a window size of $N = 7$, a prediction attempt was made for only about 11-12% of the secondary structures when a prediction was not forced. It is surprising to find that by forcing a predication, a Q_8 score of 35-36% is achieved.

It is also observed that there is a performance benefit from including edges in the analysis. This may be attributed to the fact that patterns including edges are very likely to be associated with the coil structure as was illustrated in Section 5.3.3.3.

An interesting observation is that better performance is consistently achieved if the central amino acid is located toward the middle of the window in the case that edges are not included and for small windows in the case that edges are included. For larger windows where edges are included, better performance is achieved if the central amino acid is located towards the sides of the window. This latter effect can yet again be attributed to the fact that coil structures are almost certain to be found at the edges of a protein. In general however, windows where the central amino acid is closer to the middle of the window has a bigger performance benefit.

5.4.1.4 Conclusion

There is a performance benefit associated with larger window sizes. However this is practically limited by the amount of training data available. The indication is that a window size that covers all or most of the input pattern space will have the best performance (in this case $N = 3$ with a performance of 41.8%). Additional techniques are thus required to map "unknown" input patterns in the testing data to the available patterns in the training data, if larger window sizes are to be used.

The inclusion of edges and forcing a prediction does provide performance benefits and subsequent experiments will be conducted as such. The performance benefit established through the inclusion of edges is likely due to the fact that coil structures are almost certain to be found at the edges of a protein.

A performance benefit is also achieved if the central amino acid is located towards the middle of the window. This implies that the coupling between an amino acid and its associated secondary structure is influenced more by the amino acid and its immediate neighbors than by residues further removed from it. In the experiments that follow, this fact will be reflected by choosing $l = \lceil \frac{N}{2} \rceil - 1$ and $r = \lfloor \frac{N}{2} \rfloor$ for a given window size N.

5.4.2 Experiment: Varying Window Size

5.4.2.1 Objective

The previous experiment indicated that larger window sizes have more predictive power than smaller window sizes. Due to the limited amount of training data available, only window sizes of $N \in [1, 7]$ were considered. In this experiment, the objective is to quantify what is meant by "more predictive power". A method is also devised by which larger windows can contribute meaningfully to secondary structure predictions. The performance of this experiment will form the "baseline" against which subsequent experiments are compared.

5.4.2.2 Protocol

In this experiment, the set of group labels were the same as the set of residue labels, that is $G = R$, with L the identity function. $\delta^{(1)}$ was used as distance metric. $\phi^{(2)}$ was used as assignment function, with ϵ set to 0, such that only exact matches contribute toward classification. $\psi^{(1)}$ was used as assignment function.

An iterative approach is followed in predicting secondary structures, starting with a window size s. During each iteration, the sequences associated with unpredicted secondary structures in the test set are extracted. Using $\phi^{(2)}$ with $\epsilon = 0$, a check is made against the sequences in the training set for exact matches. If such sequence(s) are found, they are used to predict the secondary structure of the target sequence. If no such sequences are found, the next smaller window size is used. N thus ranges from s to 1. For odd values of N, $l = \frac{N-1}{2}$ and $r = \frac{N-1}{2}$ are used. For even values of N, $l = \lceil \frac{N}{2} \rceil - 1$ and $r = \lfloor \frac{N}{2} \rfloor$ are considered before $r = \lceil \frac{N}{2} \rceil - 1$ and $l = \lfloor \frac{N}{2} \rfloor$. s ranges from 1 to 15.

Using this method, the predictive power of larger sequences can be used, given that a match can be found between the target sequence and sequences in the training set.

In an adaption of the above method, if there is a split vote between two or more secondary structures for a given size of N, rather than forcing a prediction, the split vote is handled by a postponing prediction until a smaller value of N is reached where the original split vote can be settled uniquely.

5.4.2.3 Results and Discussion

The number of predicted secondary structures is shown in Table 5.8 and the percentage of correctly predicted secondary structures in Table 5.9. Each row in these tables indicates a complete experiment for a certain starting value of s. The cells in each row indicate the number of times a prediction for a certain window structure has been attempted (Table 5.8) and the percentage of times those predictions were correct (Table 5.9).

The first observation is that a prediction accuracy of 43.4% is achieved for s values of 6 and more. This is a 1.6% improvement over the best result achieved in the "window structure" experiment and forms the baseline accuracy against which subsequent experiments will be compared. More importantly, it can be seen that for window sizes of 6 and larger, 70% of the attempted predictions are correct. In fact, for a window size of 7, roughly 80% is achieved, whilst for window sizes of 8 and larger, roughly 90% is achieved. This clearly illustrates the benefit associated with larger window sizes. However, only 10308 (14%) of the 72987 secondary structures in the test set can be predicted using window sizes of 6 and larger.

An interesting observation is what can be described as the "transfer phenomenon", namely that secondary structures that can be predicted using sequences of length N and $N+1$ are considerably more accurate than secondary structures that can be predicted using sequences of length N but not length $N+1$, *even when* only sequences of length N are considered. This is very apparent when looking at the top entries in the columns marked "2 2" to "3 3" in Table 5.9. What is interesting is that the apparent benefit of being able to predict a secondary structure using a larger window size is somehow embedded in sequences of smaller size. This is reinforced by the observation that from $s = 5$ onwards, no significant performance benefit is achieved using larger window sizes. One possible explanation for this phenomenon is that where larger window sizes are matched, these are likely to have some biological function which is preserved over multiple sequences. The associated secondary structures, even for a smaller segment of these larger structures, are thus unlikely to change and are hinted at by these smaller segments.

As was explained in the protocol section, the method was adapted to handle split votes. This resulted in an improved accuracy of 44.05%.

5.4.2.4 Conclusion

It is clearly illustrated that larger window sizes have more predictive power than smaller window sizes. However as was also found in the previous experiment, this is practically limited by the amount of training data available, since an enormous amount of training data would be required to completely cover all the possible amino acid combinations that could be observed for larger window sizes.

Table 5.8: Number of secondary structures predicated per category

l	r	s	0 0	0 1	1 0	1 1	1 2	2 1	2 2	2 3	3 2	3 3	3 4
0	0	1	72987										
0	1	2	0	72987									
1	0	2	0	0	72987								
1	1	3	0	0	49	72938							
1	2	4	0	0	49	10205	62733						
2	1	4	0	0	49	3309	6908	62721					
2	2	5	0	0	49	3309	6908	41803	20918				
2	3	6	0	0	49	3309	6908	41803	10610	10308			
3	2	6	0	0	49	3309	6908	41803	8784	1814	10320		
3	3	7	0	0	49	3309	6908	41803	8784	1814	1920	8400	
3	4	8	0	0	49	3309	6908	41803	8784	1814	1920	923	7477
4	3	8	0	0	49	3309	6908	41803	8784	1814	1920	237	682
4	4	9	0	0	49	3309	6908	41803	8784	1814	1920	237	682
4	5	10	0	0	49	3309	6908	41803	8784	1814	1920	237	682
5	4	10	0	0	49	3309	6908	41803	8784	1814	1920	237	682
5	5	11	0	0	49	3309	6908	41803	8784	1814	1920	237	682
5	6	12	0	0	49	3309	6908	41803	8784	1814	1920	237	682
6	5	12	0	0	49	3309	6908	41803	8784	1814	1920	237	682
6	6	13	0	0	49	3309	6908	41803	8784	1814	1920	237	682
6	7	14	0	0	49	3309	6908	41803	8784	1814	1920	237	682
7	6	14	0	0	49	3309	6908	41803	8784	1814	1920	237	682
7	7	15	0	0	49	3309	6908	41803	8784	1814	1920	237	682

l	r	s	4 3	4 4	4 5	5 4	5 5	5 6	6 5	6 6	6 7	7 6	7 7
0	0	1											
0	1	2											
1	0	2											
1	1	3											
1	2	4											
2	1	4											
2	2	5											
2	3	6											
3	2	6											
3	3	7											
3	4	8											
4	3	8	7481										
4	4	9	705	6776									
4	5	10	705	579	6197								
5	4	10	705	73	497	6206							
5	5	11	705	73	497	520	5686						
5	6	12	705	73	497	520	441	5245					
6	5	12	705	73	497	520	68	369	5249				
6	6	13	705	73	497	520	68	369	375	4874			
6	7	14	705	73	497	520	68	369	375	320	4554		
7	6	14	705	73	497	520	68	369	375	45	274	4555	
7	7	15	705	73	497	520	68	369	375	45	274	281	4274

Table 5.9: Percentage of correctly predicated secondary structures per category

l	r	s	0 0	0 1	1 0	1 1	1 2	2 1	2 2	2 3	3 2	3 3	3 4	4 3
0	0	1	34.149											
0	1	2	-	38.038										
1	0	2	-	-	36.842									
1	1	3	-	-	61.224	41.922								
1	2	4	-	-	61.224	36.404	43.089							
2	1	4	-	-	61.224	32.638	35.741	42.675						
2	2	5	-	-	61.224	32.638	35.741	35.198	63.233					
2	3	6	-	-	61.224	32.638	35.741	35.198	43.205	85.574				
3	2	6	-	-	61.224	32.638	35.741	35.198	37.864	69.184	85.853			
3	3	7	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	89.250		
3	4	8	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	87.649	89.341	
4	3	8	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	81.857	89.883	89.413
4	4	9	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	81.857	89.883	89.929
4	5	10	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	81.857	89.883	89.929
5	4	10	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	81.857	89.883	89.929
5	5	11	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	81.857	89.883	89.929
5	6	12	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	81.857	89.883	89.929
6	5	12	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	81.857	89.883	89.929
6	6	13	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	81.857	89.883	89.929
6	7	14	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	81.857	89.883	89.929
7	6	14	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	81.857	89.883	89.929
7	7	15	-	-	61.224	32.638	35.741	35.198	37.864	69.184	70.052	81.857	89.883	89.929

l	r	s	4 4	4 5	5 4	5 5	5 6	6 5	6 6	6 7	7 6	7 7	Q_8
0	0	1											34.149
0	1	2											38.038
1	0	2											36.842
1	1	3											41.935
1	2	4											42.166
2	1	4											41.576
2	2	5											43.186
2	3	6											43.430
3	2	6											43.479
3	3	7											43.454
3	4	8											43.443
4	3	8											43.453
4	4	9	89.300										43.447
4	5	10	91.019	89.011									43.437
5	4	10	94.521	88.934	89.188								43.441
5	5	11	94.521	88.934	90.385	89.026							43.437
5	6	12	94.521	88.934	90.385	90.703	88.866						43.435
6	5	12	94.521	88.934	90.385	89.706	88.889	88.912					43.428
6	6	13	94.521	88.934	90.385	89.706	88.889	90.667	88.818				43.431
6	7	14	94.521	88.934	90.385	89.706	88.889	90.667	86.563	88.977			43.431
7	6	14	94.521	88.934	90.385	89.706	88.889	90.667	91.111	87.591	88.825		43.428
7	7	15	94.521	88.934	90.385	89.706	88.889	90.667	91.111	87.591	85.765	88.980	43.426

A method thus needs to be devised for comparing different amino acid sequences and to use "similar" sequences to make a prediction. The subsequent experiments deal with multiple facets of this problem.

5.4.3 Experiment: Classification Function

5.4.3.1 Objective

The previous two experiments indicated that better performance could be achieved if larger window sizes are used. This was practically limited by the amount of training data available, since the exact input patterns in the test data had to be present in the training data as well. The objective of this experiment is to determine how the performance of the algorithm for larger window sizes will be influenced if small differences between the patterns in the test and training data are allowed. These are controlled through the ϵ parameter in the algorithm. The effect of different classification functions (as defined in Section 4.6) will be studied.

5.4.3.2 Protocol

A series of experiments with window sizes ranging from 1 to 15 were executed ($N \in [1, 15]$) with $l = \lceil \frac{N}{2} \rceil - 1$ and $r = \lfloor \frac{N}{2} \rfloor$. For each window size, epsilon values of 0 to N were tested ($\epsilon \in [0, N]$). Distance metric $\delta^{(1)}$ was used. The set of group labels were the same as the set of residue labels, that is $G = R$, with L the identity function. The set of experiments were executed for classification functions $\phi^{(1)}$ and $\phi^{(2)}$. $\psi^{(1)}$ was used as the assignment function.

5.4.3.3 Results and Discussion

The results obtained with $\phi^{(1)}$ is shown in Table 5.10 and Figure 5.7 and with $\phi^{(2)}$ in Table 5.11 and Figure 5.8. In general, $\phi^{(2)}$ performs better than $\phi^{(1)}$. There are however specific "regions" (combinations of N and ϵ values) in which $\phi^{(1)}$ performs better.

The performance values for both $\phi^{(1)}$ and $\phi^{(2)}$ are the same for $\epsilon = 0$, since both classification functions use the same patterns in the training data for prediction. For a fixed value of N, the performance of the two classification functions differ significantly for different values of ϵ. The values for both $\phi^{(1)}$ and $\phi^{(2)}$ are nearly the same and increasing up to a certain value of ϵ. This point is usually where the performance of $\phi^{(1)}$ reaches a maximum. For larger values of ϵ the performance of $\phi^{(1)}$ start to decrease again, up to the value of 27.38% for $\epsilon = N$. This is to be expected, since more patterns that are further removed from the test pattern contribute to the prediction and as such "pollute" the result. At $\epsilon = N$, all the samples in the training data contribute to the prediction and thus the class with the highest prior probability of occurring is predicted (in this case the α-helix structure, which occurs in 27.38% of the test data). For $\phi^{(2)}$, increasing the value of ϵ further leads to a small increase in performance after which it saturates and stays constant. The saturation takes place at the ϵ value at which all the patterns in the test data are at most a distance ϵ from at least one of the patterns in the training data. When ϵ is increased further, the additional patterns in the training data are filtered out by $\phi^{(2)}$ with no additional performance benefit.

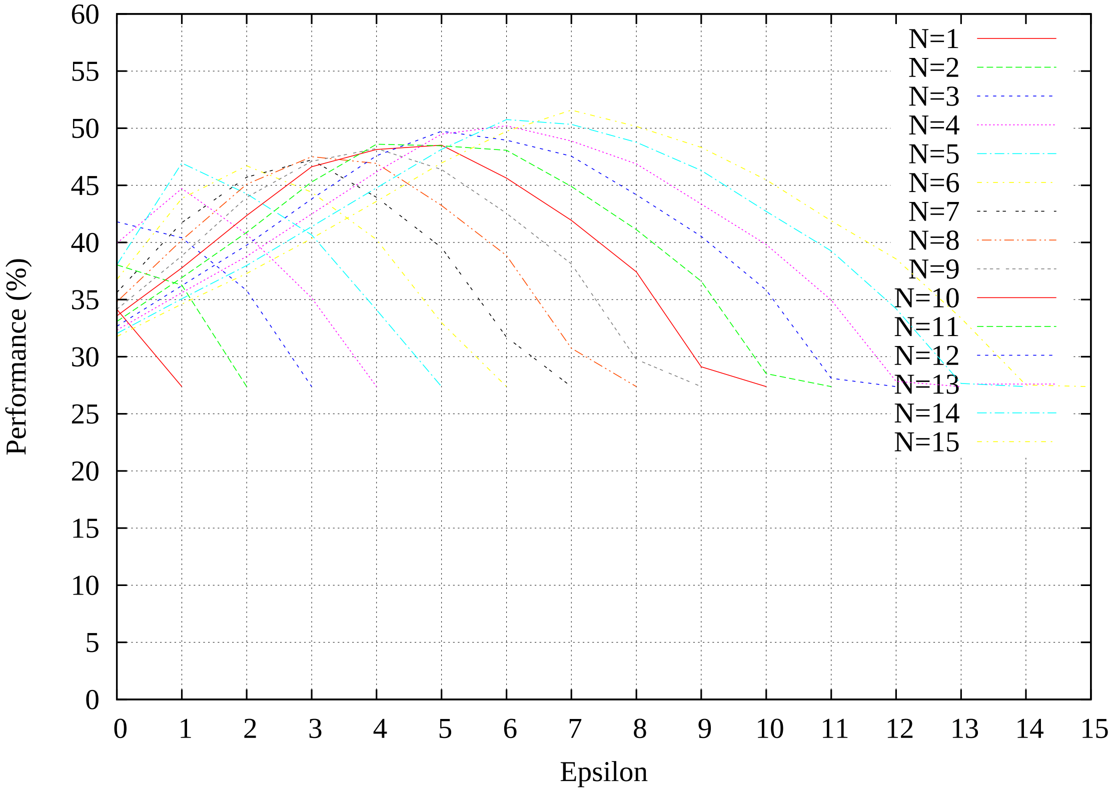


Figure 5.7: Classification Function 1

The performance increases with larger window sizes. In the case of $\phi^{(1)}$, the best

Table 5.10: Classification Function 1

$N \downarrow \epsilon \rightarrow$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Max.
1	34.1485	27.3788															34.1485
2	38.0383	36.2626	27.3788														38.0383
3	41.8116	40.4031	35.8050	27.3788													41.8116
4	39.9345	44.7134	40.7127	35.1446	27.3788												44.7134
5	38.0369	46.9426	44.2627	40.6579	34.1020	27.3788											46.9426
6	36.7449	43.8544	46.7097	44.3969	40.2469	32.9785	27.3788										46.7097
7	35.6228	41.7307	45.7150	47.2235	43.9558	39.5495	31.7070	27.3788									47.2235
8	34.8021	40.2113	45.0957	47.5002	46.9166	43.2392	38.8508	30.7438	27.3788								47.5002
9	34.1403	38.8275	43.9434	47.0947	48.2154	46.3822	42.5473	38.1260	29.7560	27.3788							48.2154
10	33.5758	37.7725	42.3500	46.6234	48.1305	48.5086	45.6342	41.9266	37.4176	29.1175	27.3788						48.5086
11	33.0840	36.9175	40.8840	45.2930	48.5991	48.4648	48.0688	44.9025	41.1155	36.6079	28.5092	27.3788					48.5991
12	32.6647	36.2188	39.7591	43.8160	47.5961	49.7404	48.9430	47.5455	44.1709	40.5346	35.8297	28.1119	27.3788				49.7404
13	32.3236	35.6036	38.7809	42.4870	46.1753	49.4855	50.2089	48.8745	46.8810	43.3598	39.8208	34.9583	27.8365	27.3788			50.2089
14	32.0331	35.0734	37.9876	41.3923	44.7765	48.1538	50.7542	50.3309	48.7607	46.3041	42.7145	39.2563	34.2006	27.6679	27.3788		50.7542
15	31.7673	34.6103	37.2834	40.3565	43.5927	46.9659	49.7280	51.5777	50.1596	48.3182	45.4697	41.8951	38.5507	33.3744	27.5227	27.3788	51.5777

Table 5.11: Classification Function 2

$N \downarrow \epsilon \rightarrow$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Max.
1	34.1485	34.1485															34.1485
2	38.0383	38.0383	38.0383														38.0383
3	41.8116	41.8499	41.8499	41.8499													41.8499
4	39.9345	41.9897	41.9897	41.9897	41.9897												41.9897
5	38.0369	46.7193	46.8275	46.8275	46.8275	46.8275											46.8275
6	36.7449	43.8147	48.3127	48.3141	48.3141	48.3141	48.3141										48.3141
7	35.6228	41.7198	48.0894	49.3170	49.3170	49.3170	49.3170	49.3170									49.3170
8	34.8021	40.2044	45.4026	50.8899	51.0927	51.0927	51.0927	51.0927	51.0927								51.0927
9	34.1403	38.8247	43.8969	48.3072	51.9558	51.9764	51.9764	51.9764	51.9764	51.9764							51.9764
10	33.5758	37.7739	42.2596	46.6042	50.6076	52.4299	52.4299	52.4299	52.4299	52.4299	52.4299						52.4299
11	33.0840	36.9203	40.8114	45.1793	48.7977	52.3518	52.9930	52.9930	52.9930	52.9930	52.9930	52.9930					52.9930
12	32.6647	36.2215	39.7016	43.7160	47.4290	50.5460	53.8712	54.0398	54.0398	54.0398	54.0398	54.0398	54.0398				54.0398
13	32.3236	35.6077	38.7288	42.3980	46.0315	49.3526	51.8860	54.4330	54.4727	54.4727	54.4727	54.4727	54.4727	54.4727			54.4727
14	32.0331	35.0788	37.9465	41.3224	44.6696	47.9565	50.8036	53.2341	54.9303	54.9399	54.9399	54.9399	54.9399	54.9399	54.9399		54.9399
15	31.7673	34.6116	37.2464	40.2976	43.4940	46.8111	49.5705	51.9408	54.3426	55.1920	55.1920	55.1920	55.1920	55.1920	55.1920	55.1920	55.1920

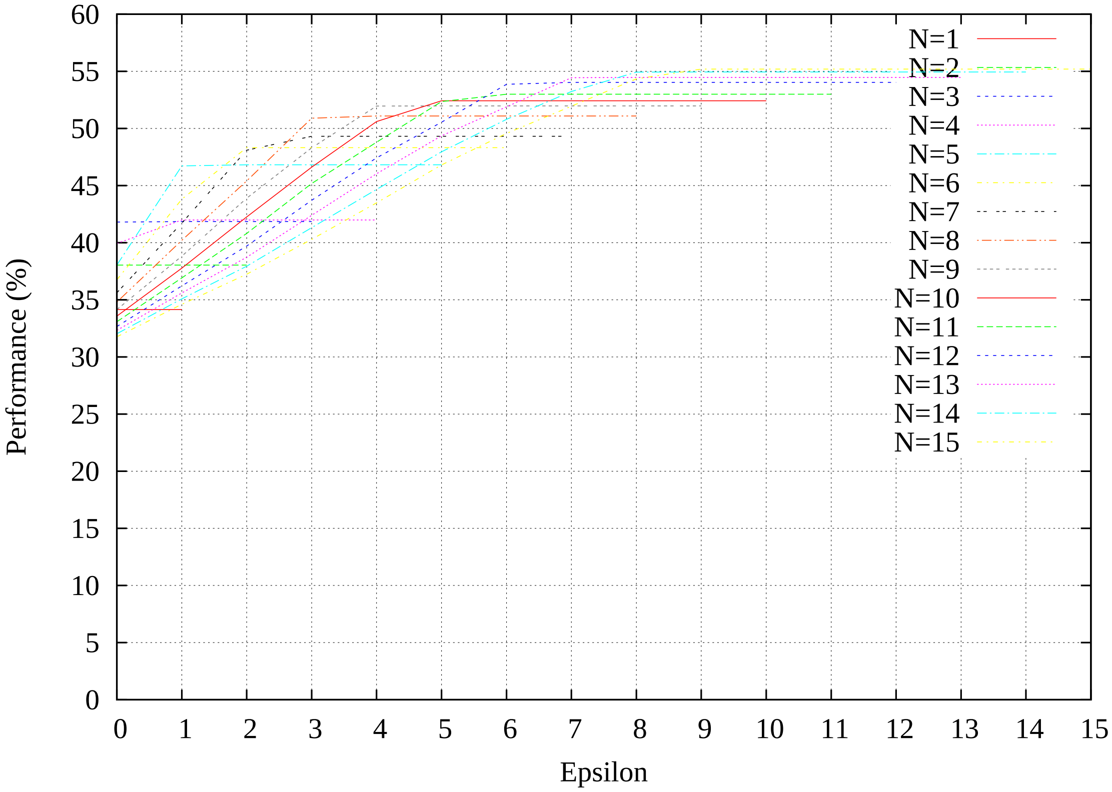


Figure 5.8: Classification Function 2

performance is 51.58% with $N = 15$ and $\epsilon = 7$. For $\phi^{(2)}$, the best performance is 55.19% with $N = 15$ and $\epsilon \geq 9$. This represents a significant performance increase over the performance of 41.81% ($N = 3, \epsilon = 0$) achieved in the "window structure" experiment or even the 44.05% achieved in the adapted "varying window size" experiment.

The question naturally arises whether larger window sizes will continue to add a performance benefit. Figure 5.9 plots the best performance (considered over the different ϵ values) of classification functions 1 and 2 for each window size. As can be seen in the figure, the performance increase from a window size of 1 to a window size of 8 are respectively 13% (for $\phi^{(1)}$) and 17% (for $\phi^{(2)}$). Comparatively, the performance increase from a window size of 8 to 15 is about 4% (for both $\phi^{(1)}$ and $\phi^{(2)}$). The rate of increase is declining as larger window sizes are considered.

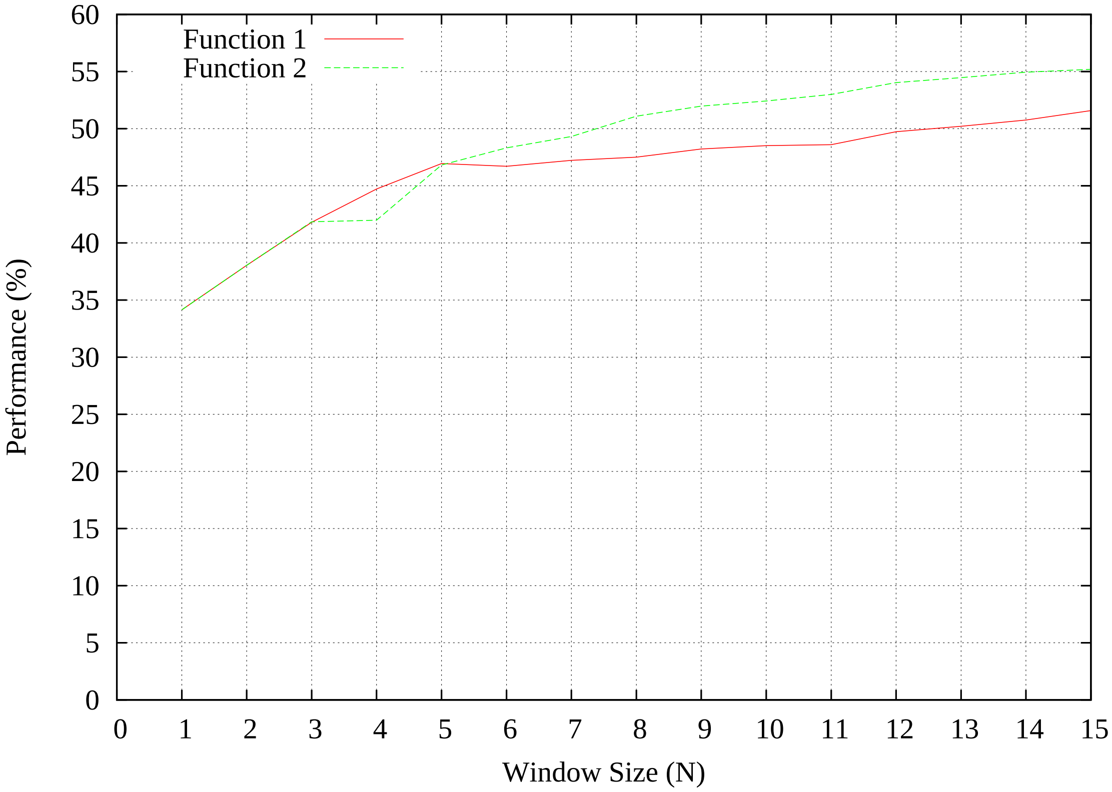


Figure 5.9: Performance of Classification Function 1 vs Classification Function 2

5.4.3.4 Conclusion

A performance benefit is achieved by increasing the window size and allowing patterns in the training and testing data that are not exactly the same but still similar to be matched. A performance of $Q_8 = 55.19\%$ is achieved for a window size of $N = 15$, with $\epsilon \geq 9$ for $\psi^{(2)}$. The performance increases as larger windows are considered, but the rate of increase declines.

The elementary distance metric $\delta^{(1)}$ was used in this experiment. Under this distance metric, the ϵ value indicates the number of positions at which secondary structures are different between two patterns. It assigns the same contribution to each position in the window and does not take into account which specific residue types are different. It is however quite possible that clusters of sequence patterns exist that are "close" to one another under the $\delta^{(1)}$ distance metric but which form different secondary structures. If this is indeed the case, this fact was not exploited in this experiment. Through proper design of the δ function and using larger window sizes, it may thus be possible to achieve even better results than achieved in this experiment.

5.5 DETAILED ANALYSIS

5.5.1 Experiment: Grouping Strategies

5.5.1.1 Objective

The objective of this experiment is to determine whether amino acid residues can be grouped together in a meaningful way. The procedure for mapping amino acid patterns to group vectors and its use in the construction of a database were explained in Section 4.3.

5.5.1.2 Protocol

A series of experiments with window sizes ranging from 1 to 15 were executed ($N \in [1, 15]$) with $l = \lceil \frac{N}{2} \rceil - 1$ and $r = \lfloor \frac{N}{2} \rfloor$. Distance metric $\delta^{(1)}$ was used with $\epsilon = 0$. $\phi^{(2)}$ was used as classification function and $\psi^{(1)}$ as assignment function.

The procedure to set up G starts by assigning $G = R$. Thus, initially there are 21 groups (one for each amino acid residue type and one for an edge). Unique pairs are selected from the different groups and combined (there are $\frac{m}{2}(m-1)$ such pairings, thus initially 210 tests need to be conducted using 20 groups each). The Q_8 score achieved by each of the pairings is noted. Once all the tests are completed, the pairing with the highest Q_8 score is retained and G adjusted accordingly. The process is then repeated, this time with the new G containing 20 groups. With 20 groups in G, 190 tests are conducted and G is reduced to 19 groups by the same process. The process is repeated until G consists of just 1 group (the trivial case) or until no performance gain is achieved.

It should be noted that 1540 tests need to be conducted to reduce G from 21 groups to 1 (these tests need to be conducted for each combination of other parameters, i.e. window size, classification and assignment functions etc.). There are nevertheless many more ways in which 21 groups can be segmented into fewer than 21 groups. The procedure described above is thus not guaranteed to find a configuration with optimum performance. Rather, the procedure is based on what is known as a “greedy” algorithm and it is hoped that the performance is near optimum.

In the discussion that follows, each group G_i will be designated by a group label using curly brackets $\{\}$. The amino acid residue types that belong to the group are listed between the brackets. The function L maps each amino acid in a window to a group label G_i based on the group to which it belongs.

5.5.1.3 Results and Discussion

The results achieved by the procedure described above are listed in Table 5.12. The table shows the performance without any groupings, the performance achieved by the optimum grouping (optimum in the sense of the algorithm previously discussed), the gain achieved by using the grouping scheme, the grouping that resulted in optimum performance and the number of different groups (m) that achieve optimum performance.

As can be seen from the results, a performance gain can be achieved irrespective of the window size. The best performance is 44.09% for a window size of $N = 5$. This represents an increase of 6.05% over the case where no grouping is used.

Table 5.12: Application of Grouping Strategy (Forcing Prediction, $\epsilon = 0$)

N	No Grouping	Grouping	Gain	Optimum Grouping	m
1	34.1485	34.1828	0.0343	{ARQELKM#}{NHDPS}{CIFTWYV}{G}	4
2	38.0383	38.1493	0.1110	{AE}{RK}{NS}{D}{C}{Q}{G}{H}{IW}{L}{M}{F}{P#}{T}{Y}{V}	16
3	41.8116	41.9623	0.1507	{A}{R}{N}{D}{CIW}{Q}{E}{G}{HK}{L}{MF}{P}{S}{T}{Y}{V}{#}	17
4	39.9345	43.6626	3.7281	{A}{REKQ}{NSD}{CWYIVF}{G}{HT}{LM}{P}{#}	9
5	38.0369	44.0928	6.0559	{ARKQE}{NSTHD}{CW}{G}{IVLFYM}{P}{#}	7
6	36.7449	43.6735	6.9286	{A}{RNSTDKQEH}{CIVLFYMW}{G}{P}{#}	6
7	35.6228	41.7102	6.0874	{ARKEQH}{NGD}{CIVLSTFYMW}{P#}	4
8	34.8021	40.3072	5.5051	{ARKQHE}{NDGP#}{CIVLSTFYWM}	3
9	34.1403	39.0165	4.8762	{AIVLSTNFYCHMW}{RKQED}{G#}{P}	4
10	33.5758	37.5766	4.0008	{ASTIVLNGFYHMWC}{RKQE}{D#}{P}	4
11	33.0840	37.0765	3.9925	{ASTIVLNGFYCH}{RK}{D#}{QE}{M}{P}{W}	7
12	32.6647	36.9230	4.2583	{ASTIVLNGPC}{RK}{DE}{Q}{H}{M}{FY#}{W}	8
13	32.3236	36.5997	4.2761	{ASTIVLNGPD}{RKH#}{C}{QE}{M}{FY}{W}	7
14	32.0331	36.3544	4.3213	{ASTIVLRKNPQ}{DE}{C}{G}{H}{M#}{FY}{W}	8
15	31.7673	36.0503	4.2830	{ASTIVLRKNGP}{DE}{C}{Q}{H}{M#}{FY}{W}	8

In Table 5.13 the experiments were repeated, but this time the predictions were not forced as in Table 5.12. Interestingly, the performance values are about the same (but on average slightly worse). Large performance gains are achieved for larger window sizes (relative to the unforced case with $G = R$). This effect can be explained by the fact that the number of patterns that need to be stored in the database to be representative of the entire input space is reduced from 21^N to m^N. Although m^N is still large for larger window sizes, it is probably the case that only a fraction of those patterns are required to be representative of the actual proteins in the data sets. In fact, the objective of the grouping function is to reduce the complexity in that way. It is also observed that different groups form for window sizes of $N \geq 4$ in the case of unforced prediction. The group size (m) is also smaller for larger window sizes.

Table 5.13: Application of Grouping Strategy (Not Forcing Prediction, $\epsilon = 0$)

N	No Grouping	Grouping	Gain	Optimum Grouping	m
1	34.1485	34.1828	0.0343	{ARQELKM#}{NHDPS}{CIFTWYV}{G}	4
2	38.0383	38.1493	0.1110	{AE}{RK}{NS}{D}{C}{Q}{G}{H}{IW}{L}{M}{F}{P#}{T}{Y}{V}	16
3	41.8074	41.9623	0.1549	{A}{R}{N}{D}{CIW}{Q}{E}{G}{HK}{L}{MF}{P}{S}{T}{Y}{V}{#}	17
4	36.3393	43.8955	7.5562	{A}{REKQ}{ND}{CIVFYW}{G}{HT}{LM}{P}{S}{#}	10
5	18.0416	41.6417	23.6001	{ALVIFYRMCW}{ND}{QEKH}{G}{P}{ST}{#}	7
6	12.0830	39.9331	27.8501	{AIVLEKRTQFYMHW}{NDS}{C}{G}{P}{#}	6
7	10.2772	39.0494	28.7722	{AIVLEKTSRQNDHM}{CFYW}{G}{P}{#}	5
8	9.1619	38.9960	29.8341	{AIVLKETSRQNDYMH}{C}{G}{FW}{P}{#}	6
9	8.2960	38.9974	30.7014	{ASTIVLEKRDNQYFHMW}{C#}{G}{P}	4
10	7.5630	36.2900	28.7270	{ASTIVLGKERDNQYMH}{C}{FW}{P#}	4
11	6.9314	36.2832	29.3518	{ASTIVLGKEDNRQYMHFW}{C}{P}{#}	4
12	6.3820	36.4202	30.0382	{ASTIVLGKEDNRQYMHFW}{C}{P}{#}	4
13	5.9312	36.2804	30.3492	{ASTIVLGNKEDRQYFHMW}{C}{P}{#}	4
14	5.5489	36.2708	30.7219	{ASTIVLRKEGDNQYFMHW}{C}{P}{#}	4
15	5.2009	36.2434	31.0425	{ASTIVLRKEGDNQYFMHW}{C#}{P}	3

There thus seems to be merit in grouping different amino acids together. The question is whether there is a gain to be achieved by combining a grouping strategy with other parameters in the algorithm.

The best performance achieved was 44.09% for a window size of $N = 5$. In the experiment where the different classification functions were considered, a performance score of 46.94% was achieved using $\phi^{(1)}$ ($\epsilon = 1$) and 46.71% using $\phi^{(2)}$ ($\epsilon = 1$) with $N = 5$. To make a fair comparison, the experiments were repeated using $\epsilon = 1$ for window sizes $N \in [3, 7]$ (the large amount of computational power required to execute the experiments limited the range of cases that could be tested). The results of these experiments are shown in Table 5.14 for the forced case and Table 5.15 for the unforced case.

The best performance achieved was $Q_8 = 46.88\%$ ($Q_3 = 61.05\%$). This is hardly an improvement over the case where no groupings are used. If the actual groupings that are formed are observed, it will be noted that the only groupings were M with W and F with Y. The tendency for the other window sizes is to form more groups as well (preserving the unique attributes of the different amino acid residue types).

It is the opinion of the author that no significant performance gain (relative to other parameters in the algorithm) will be achieved using larger window sizes and ϵ values using the current grouping strategy. It is suspected that the current grouping strategy will eventually (with larger N and ϵ values) reach a state where $G = R$ is the optimum grouping, and the performance will thus be the same as that achieved in the experiment on classification functions (Section 5.4.3).

It is extremely important to note that the current grouping strategy could have been implemented as a more advanced distance metric. Such a distance metric would assign a score of 0 to amino acids that are in the same group and a score of 1 to amino acids that are not in the same group. This would have the same effect as applying the grouping strategy initially and then applying $\delta^{(1)}$ on the resulting patterns. From this it can be concluded that work should rather be conducted in developing a better distance metric, as was the conclusion in Section 5.4.3.4.

Figureau et al [110] found that the grouping {CFWY}{IV}{LM}{HQR}{EK}{DN}{SP}{A}{G}{T} led to good results in the classification of pentapeptides. They achieved Q_3 scores in the order of 65% using this grouping, although the technique and application they use are different. To compare results, their grouping was used in the algorithm designed in this dissertation (adding an edge as an additional grouping). A Q_8 score of 41.57% is achieved (with $\psi^{(2)}$ and $\epsilon \geq 1$). The corresponding Q_3 score is 56.97%. This is about 4-5% lower than the best results achieved using the current grouping strategy.

An interesting aspect to consider is the actual amino acid residue types that were grouped together using the grouping strategy developed in this experiment. Tables 5.12 and 5.13 list only the optimal groupings. From this it is difficult to find specific prominent groupings. A better approach is to study how different groups are combined during the optimization process. This is conveniently illustrated in the dendograms in Figures 5.10 to 5.22. The dendograms in the figures are associated with window sizes 3 to 15 in Table 5.12.

There are many diverse patterns that form, depending on the window size; however some are more readily identifiable than others. The most distinct grouping is I with V. IV is also often associated with L. S and T are found together, often combined with A and/or with IVL. R and K are found together, as are Q and E. R, K, Q and E are also found together in various combinations. H and A is sometimes found in combination with R, K, Q and E. F, Y, C and W are sometimes found in combination. The other

Table 5.14: Application of Grouping Strategy (Forcing Prediction, $\epsilon = 1$)

N	No Grouping	Grouping	Performance Gain	Optimum Grouping	m
3	41.8499	41.9732	0.1233	{A}{R}{N}{D}{CIW}{Q}{E}{G}{HK}{L}{MF}{P}{S}{T}{Y}{V}{#}	17
4	41.9897	43.9832	1.9935	{A}{REQK}{ND}{CIVFYW}{G}{HT}{LM}{P}{S}{#}	10
5	46.7193	46.8837	0.1644	{A}{R}{N}{D}{C}{Q}{E}{G}{H}{I}{L}{K}{MW}{FY}{P}{S}{T}{V}{#}	19
6	43.8147	46.6809	2.8662	{A}{R}{N}{D}{CW}{QE}{G}{HM}{IVL}{K}{FY}{P}{ST}{#}	14
7	41.7198	46.6453	4.9255	{A}{RK}{NST}{D}{CIVLFM}{QE}{G}{H}{P#}{WY}	10

Table 5.15: Application of Grouping Strategy (Not Forcing Prediction, $\epsilon = 1$)

N	No Grouping	Grouping	Performance Gain	Optimum Grouping	m
3	41.8499	41.9732	0.1233	{A}{R}{N}{D}{CIW}{Q}{E}{G}{HK}{L}{MF}{P}{S}{T}{Y}{V}{#}	17
4	41.9897	43.9832	1.9935	{A}{REQK}{ND}{CIVFYW}{G}{HT}{LM}{P}{S}{#}	10
5	46.6179	46.8577	0.2398	{A}{R}{N}{D}{C}{Q}{E}{G}{H}{I}{L}{K}{MW}{FY}{P}{S}{T}{V}{#}	19
6	34.8857	46.2562	11.3705	{A}{RK}{ND}{CW}{QE}{G}{H}{IVLFM}{P}{S}{T}{Y}{#}	13
7	20.4927	44.6134	24.1207	{ALVIFYM}{RQEK}{ND}{C}{G}{H}{P}{ST}{W}{#}	10

types (N, D, M, G, P and #) do not seem to form regular combinations.

It is interesting to compare the above findings with the grouping used by Figureau. Both seem to suggest that I and V as well as C, F, W and Y could be clustered together. Figureau clusters H, Q and R as well as E and K, which can be supported with the findings above. The clustering of D and N is also somewhat suggested by the findings above.

It is also interesting to consider the chemical properties of the residue: sulfhydryl (C), small hydrophilic (S, T, P, A, G), acid amide and hydrophilic (N, D, E, Q), basic (H, R, K), small hydrophobic (M, I, L, V), and aromatic (F, Y, W) ([53], p. 82). The clustering results found above seem to be somewhat correlated by the chemical properties of the side chain: ILV, ST, ND, EQ, HRK and FYW share similar chemical characteristics. The implication of this is important: substitution of amino acids with similar chemical properties may preserve the formation of secondary structures.

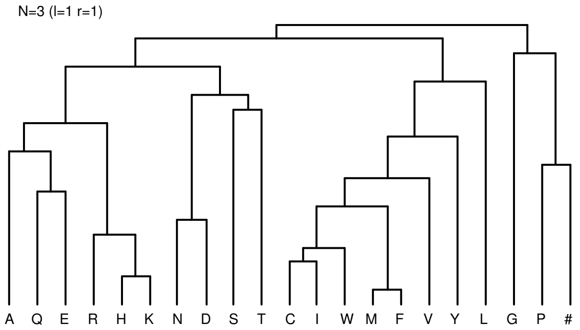


Figure 5.10: Dendrogram indicating clusterings for $N = 3$

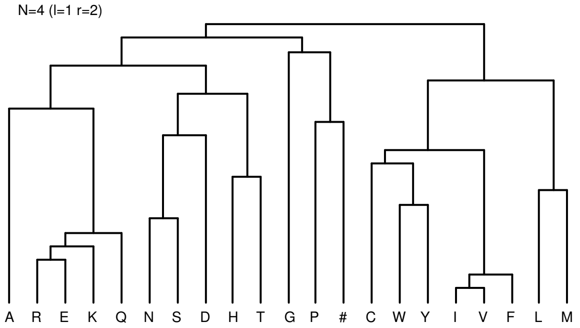


Figure 5.11: Dendrogram indicating clusterings for $N = 4$

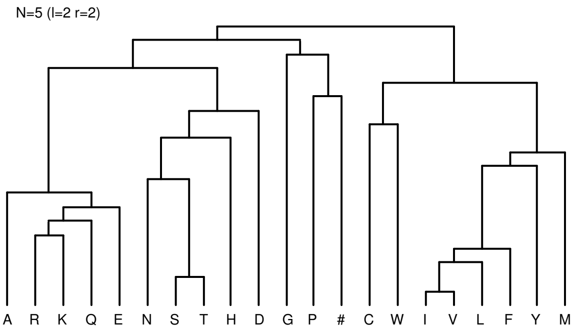


Figure 5.12: Dendrogram indicating clusterings for $N = 5$

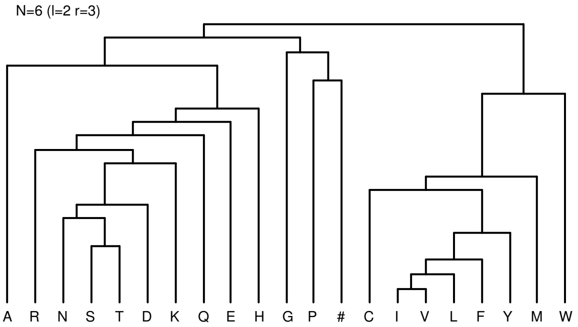


Figure 5.13: Dendrogram indicating clusterings for $N = 6$

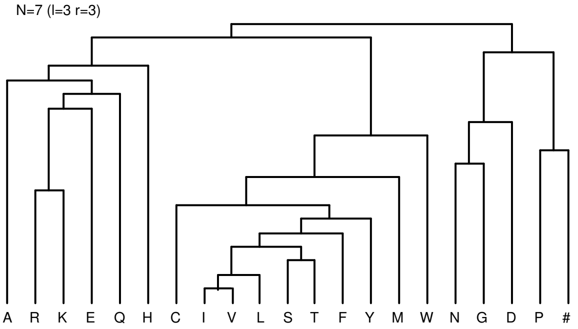


Figure 5.14: Dendrogram indicating clusterings for $N = 7$

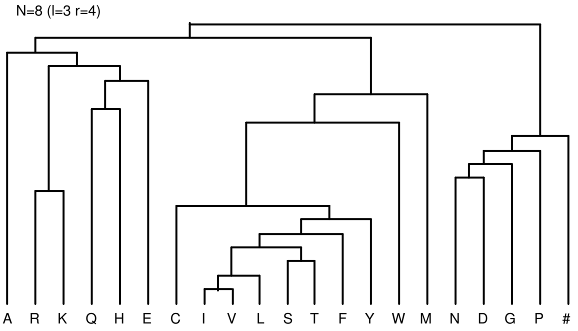


Figure 5.15: Dendrogram indicating clusterings for $N = 8$

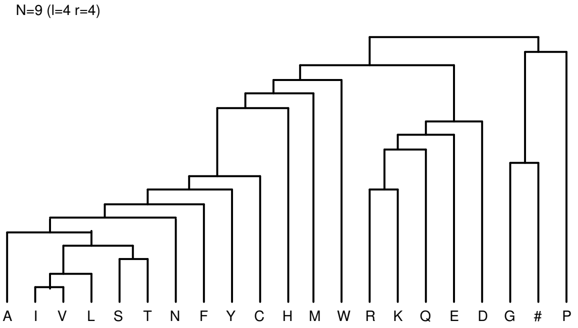


Figure 5.16: Dendrogram indicating clusterings for $N = 9$

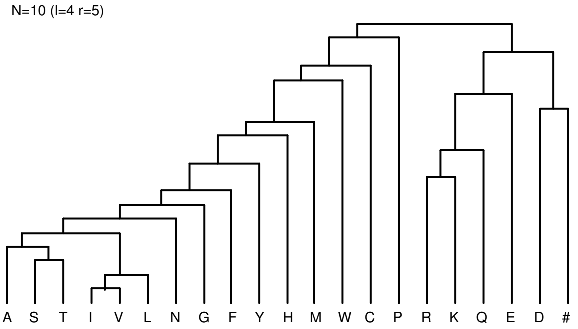


Figure 5.17: Dendrogram indicating clusterings for $N = 10$

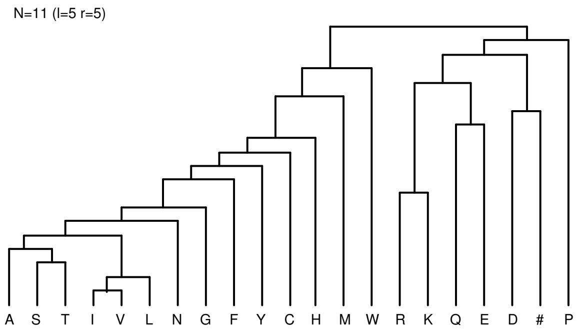


Figure 5.18: Dendrogram indicating clusterings for $N = 11$

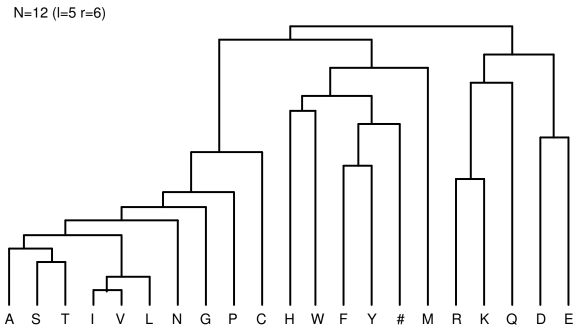


Figure 5.19: Dendrogram indicating clusterings for $N = 12$

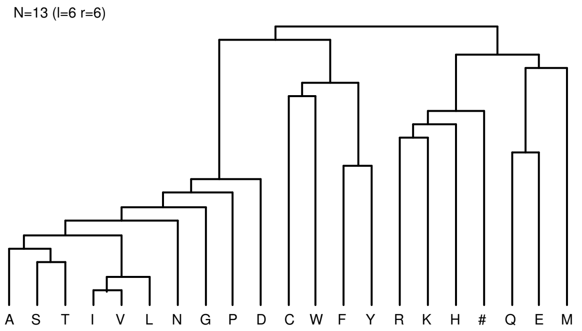


Figure 5.20: Dendrogram indicating clusterings for $N = 13$

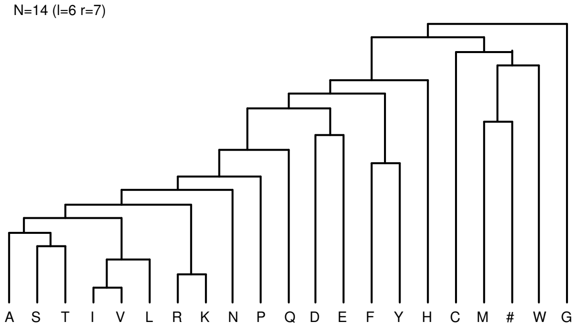


Figure 5.21: Dendrogram indicating clusterings for $N = 14$

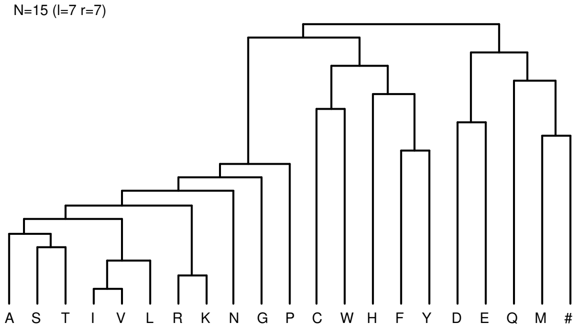


Figure 5.22: Dendrogram indicating clusterings for $N = 15$

5.5.1.4 Conclusion

A small but consistent gain can be achieved by grouping different amino acids together. It was found that (IV)(L), ST, (RK)(QE), FYCW and DN are good groupings. H and A are sometimes found in combination with R, K, Q and E. The data seems to suggest that in some cases, substitution of amino acids with similar chemical properties may preserve the formation of secondary structures.

If more leniency (larger ϵ values) is allowed in the similarity of patterns, the tendency is for the optimum grouping to consist of more groups, i.e. the unique attributes of the amino acids are preserved. This seems to suggest that although a gain can be achieved by grouping amino acids together, it is not effective when used in conjunction with other parameters that can be controlled in the algorithm.

The grouping strategy could be implemented as a more advanced distance metric. Together with the conclusion reached in the previous experiment, this seems to suggest that the distance metric is a large contributing factor to the performance of the algorithm.

5.5.2 Experiment: Substitution Matrix

5.5.2.1 Objective

Previous experiments indicated the need to be able to determine the similarity between two different group vectors. From this experiment onwards it is assumed that the mapping function L is the identity mapping. The aim now is to measure the similarity between two different sequences of amino acids in more sophisticated ways.

The “grouping strategies” experiment (see Section 5.5.1) illustrated that there are certain amino acids that behave similarly in general. However, it did not quantify the similarity between the different amino acids. The objective of this experiment is to create a substitution matrix - a matrix quantifying the similarity between different amino acids. This quantification can then be used to create a better distance metric (as is done in a the subsequent "distance metric - substitution matrix" experiment (see Section 5.5.3)).

5.5.2.2 Protocol

For this experiment, the training data was divided into a new training and validation data set, in order not to bias subsequent experiments that are reliant on the substitution matrix. The division was roughly 80%/20%, with the new training data set containing 1174 proteins (225019 amino acid residues) and the validation set containing 320 proteins (60301 amino acid residues). The algorithm used the new training set for training purposes, and used the validation set to extract values for the substitution matrix.

A window size of 15 ($l = 7, r = 7$) was used, with distance metric $\delta^{(1)}$. The set of group labels were the same as the set of residue labels, that is $G = R$, with L the identity function. $\phi^{(2)}$ was used as assignment function, with a large ϵ value such that all sequences in the training data are considered. $\psi^{(1)}$ was used as assignment function.

Under this experimental setup, the "nearest neighbour(s)" to each target sequence in the validation data set were determined under the distance metric $\delta^{(1)}$. For the 60301 target sequences, 247949 such neighbours were found.

Consider now making a prediction for a single target sequence using a single neighbour. The target sequence and neighbour will have similar amino acid residue types in some positions and different residue types in others. Given a residue type A in position k of the target sequence and a residue type B in position k of the neighbour, it is said that a substitution of A with B has been made (even where A and B are equal).

Let C^k and I^k be matrices of dimension 21 x 21, where $k \in [1, 15]$ is an index associated with the k^{th} position in the window. Let $C^k_{m,n}$ indicate the number of times that a residue type n has been substituted with residue type m in position k over all target sequences and their neighbours, such that the neighbour correctly predicted the secondary structure associated with the target sequence. Similarly, let $I^k_{m,n}$ indicate the number of times that a residue type n has been substituted with residue type m

in position k over all target sequences and their neighbours, such that the neighbour incorrectly predicted the secondary structure associated with the target sequence.

Let

$$C = \sum_{k=1}^{15} C^k, \tag{5.4}$$

and

$$I = \sum_{k=1}^{15} I^k. \tag{5.5}$$

The matrices C and I thus indicate the total number of times that different substitutions were made in all positions of a window for correctly and incorrectly predicted secondary structures respectively.

Let P be a matrix where element $p_{m,n}$ of P is defined according to elements $c_{m,n}$ and $i_{m,n}$ of C and I by

$$p_{m,n} = \frac{c_{m,n}}{c_{m,n} + i_{m,n}}. \tag{5.6}$$

Element $p_{m,n}$ thus indicates the fraction of times that a substitution of residue type n with residue type m was observed in a correctly predicted secondary structure.

Let S be a substitution matrix where element $s_{m,n}$ is defined by

$$s_{m,n} = \frac{p_{m,n}}{\max_{l=1}^{15} p_{l,n}}. \tag{5.7}$$

The elements of S are thus normalised similarity values between different residue types.

5.5.2.3 Results and Discussion

Table 5.16 shows the calculated substitution matrix. With the exception of R, D, E, I and L, all diagonal entries have values equal to 1, as should be expected. It is also evident that the matrix is not symmetrical, implying that substitution between two amino acids is not commutative.

Substitutions with a similarity value between 0.8 and 1 are shown in Table 5.17. The table illustrates that I and V are similar as was found in the previous experiment. It also suggests that L and M are similar (and that I, V, L and M are alike in general), a result which was indicated by Figureau et al [110], but was not duplicated in the previous experiment. I, V, L and M are all small hydrophobic amino acids.

The results also show that R and K are similar, as are E and Q, and these four residues are in general very alike, a result that was also found in the previous experiment. It also indicates that A is somewhat alike to elements in this group. Although not shown in the table, there is some evidence that H shares some similarity with these residue types.

F and Y are similar. However, no evidence was found to show that C and W are similar as was found in the previous experiment. In fact, C, W and P are the only residue types that seem not to have any good substitutions. Interestingly, C, W and P seem to be most alike to the edge type.

S and T are alike, as are D and N, both results having been suggested by the previous experiment. There is also some new evidence that N and K are alike, as are E and D.

There are a surprising number of residue types that can be exchanged with an edge type. This may be due to the regularity with which edge types in a window are predictive of the coil secondary structure. No readily discernable patterns could be detected from the other high scoring substitutions.

Table 5.16: Substitution Matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	#
A	1.000	0.839	0.744	0.836	0.557	0.780	0.811	0.783	0.666	0.760	0.766	0.839	0.723	0.694	0.631	0.741	0.764	0.561	0.660	0.784	0.623
R	0.814	0.991	0.770	0.799	0.510	0.827	0.782	0.703	0.717	0.697	0.779	0.955	0.690	0.647	0.559	0.670	0.690	0.471	0.637	0.691	0.569
N	0.738	0.795	1.000	0.957	0.514	0.748	0.741	0.775	0.701	0.652	0.672	0.829	0.644	0.545	0.582	0.749	0.769	0.456	0.647	0.609	0.591
D	0.775	0.793	0.822	0.995	0.463	0.725	0.839	0.715	0.597	0.641	0.621	0.701	0.641	0.589	0.560	0.697	0.681	0.496	0.565	0.615	0.610
C	0.722	0.647	0.576	0.664	1.000	0.626	0.633	0.677	0.641	0.690	0.658	0.602	0.665	0.643	0.494	0.646	0.587	0.437	0.572	0.734	0.919
Q	0.780	0.938	0.767	0.894	0.527	1.000	1.000	0.673	0.743	0.758	0.798	0.935	0.690	0.650	0.556	0.707	0.655	0.503	0.571	0.649	0.807
E	0.864	0.848	0.806	1.000	0.524	0.861	0.952	0.735	0.703	0.712	0.715	0.849	0.730	0.684	0.581	0.683	0.690	0.470	0.607	0.659	0.790
G	0.746	0.687	0.765	0.783	0.479	0.619	0.701	1.000	0.597	0.575	0.605	0.677	0.596	0.598	0.563	0.669	0.601	0.480	0.543	0.619	0.568
H	0.696	0.814	0.884	0.700	0.544	0.785	0.656	0.607	1.000	0.705	0.660	0.726	0.624	0.670	0.537	0.682	0.621	0.468	0.661	0.601	0.727
I	0.753	0.718	0.659	0.689	0.527	0.657	0.651	0.615	0.609	0.885	0.913	0.689	0.769	0.770	0.526	0.603	0.727	0.522	0.600	0.945	0.519
L	0.783	0.746	0.677	0.704	0.499	0.707	0.744	0.640	0.608	0.897	0.985	0.717	0.865	0.786	0.517	0.578	0.627	0.568	0.644	0.823	0.524
K	0.850	1.000	0.842	0.787	0.504	0.859	0.832	0.709	0.695	0.721	0.718	1.000	0.683	0.651	0.593	0.699	0.764	0.525	0.607	0.667	0.635
M	0.871	0.797	0.695	0.727	0.642	0.751	0.687	0.707	0.608	0.959	1.000	0.832	1.000	0.903	0.568	0.614	0.744	0.569	0.687	0.757	0.687
F	0.700	0.718	0.685	0.635	0.484	0.632	0.665	0.618	0.621	0.763	0.789	0.653	0.689	1.000	0.522	0.586	0.641	0.603	0.845	0.717	0.617
P	0.728	0.697	0.687	0.734	0.432	0.637	0.632	0.649	0.548	0.543	0.554	0.668	0.527	0.563	1.000	0.687	0.642	0.463	0.499	0.581	0.714
S	0.841	0.759	0.799	0.901	0.497	0.697	0.730	0.771	0.577	0.634	0.631	0.746	0.577	0.657	0.634	1.000	0.873	0.471	0.580	0.628	0.759
T	0.778	0.748	0.774	0.770	0.476	0.645	0.721	0.662	0.641	0.755	0.633	0.763	0.590	0.611	0.561	0.860	1.000	0.469	0.623	0.735	0.531
W	0.726	0.783	0.650	0.633	0.402	0.614	0.624	0.600	0.613	0.642	0.793	0.628	0.610	0.744	0.500	0.514	0.569	1.000	0.779	0.605	0.475
Y	0.737	0.753	0.714	0.709	0.497	0.621	0.681	0.649	0.727	0.670	0.689	0.614	0.637	0.916	0.548	0.647	0.615	0.575	1.000	0.708	0.334
V	0.807	0.746	0.630	0.638	0.464	0.628	0.693	0.654	0.589	1.000	0.837	0.713	0.691	0.697	0.513	0.582	0.679	0.493	0.619	1.000	0.708
#	0.857	0.945	0.869	0.881	0.615	0.792	0.852	0.936	0.703	0.749	0.651	0.900	0.718	0.718	0.902	0.794	0.843	0.655	0.525	0.660	1.000

5.5.2.4 Conclusion

The experiment reinforces the findings of the "grouping strategies" experiment, namely that (IV)(LM), (RK)(QE), ST, DN and FY are similarity groups, but shows no evidence that C and W are similar to one another or to the FY group.

More importantly, the experiment quantifies the similarity between different residue types, which makes it possible to develop a better distance metric. It is also noted that substitution between two residue types is not commutative.

5.5.3 Experiment: Distance Metric - Substitution Matrix

5.5.3.1 Objective

The objective of this experiment is to determine whether the substitution matrix developed in the previous experiment can be used to develop a distance metric that has better success than the $\delta^{(1)}$ distance metric that was used in previous experiments.

Table 5.17: Substitutions with similarity values between 0.8 and 1

I	V	1.000	R	K	1.000	A	E	0.864	R	#	0.945	D	S	0.901
L	M	1.000	E	Q	1.000	A	K	0.850	G	#	0.936	D	Q	0.894
I	M	0.959	K	R	0.955	R	A	0.839	#	C	0.919	N	H	0.884
V	I	0.945	R	Q	0.938	K	A	0.839	P	#	0.902	A	M	0.871
L	I	0.913	K	Q	0.935	A	R	0.814	K	#	0.900	A	S	0.841
I	L	0.897	Q	E	0.861	E	A	0.811	D	#	0.881	D	A	0.836
M	L	0.865	Q	K	0.859				N	#	0.869	K	M	0.832
L	V	0.837	K	E	0.849				A	#	0.857	R	H	0.814
V	L	0.823	R	E	0.848				E	#	0.852	A	V	0.807
			E	K	0.832				T	#	0.843	N	E	0.806
			Q	R	0.827				#	Q	0.807			
F	Y	0.916	T	S	0.873	D	N	0.957	N	K	0.842	D	E	1.000
Y	F	0.845	S	T	0.860	N	D	0.822	K	N	0.829	E	D	0.839

5.5.3.2 Protocol

A window size of 15 ($l = 7, r = 7$) was used. The set of group labels were the same as the set of residue labels, that is $G = R$, with L the identity function. $\psi^{(1)}$ was used as assignment function.

The distance metric $\delta^{(3)}$ (see Section 4.5.3) is designed with elements $u_{i,j}$ of matrix U defined by

$$u_{i,j} = 1 - s_{i,j}, \tag{5.8}$$

where the $s_{i,j}$ (defined by Equation 5.7) are elements of the substitution matrix S created in the "substitution matrix" experiment, and the weights w_i associated with positions in the window are all set to 1.

The performance of this distance metric is compared to the performance of distance metric $\delta^{(1)}$ used in previous experiments. To make the comparison fair, classification function $\phi^{(2)}$ was used but adapted in such a way that exactly the k closest neighbours in the training set contribute to the prediction. For each target sequence the number of contributing neighbours is thus equal under both $\delta^{(3)}$ and $\delta^{(1)}$ (more precisely, equal to k), where there would otherwise be different numbers contributing. k was tested in the range [1, 10].

5.5.3.3 Results and Discussion

Table 5.18 shows the performance of $\delta^{(3)}$ and $\delta^{(1)}$ under the experimental setup using exactly k contributing neighbours.

For both $\delta^{(3)}$ and $\delta^{(1)}$, best performance is achieved using $k = 1$, with performance values of 55.37% and 52.42% respectively. $\delta^{(3)}$ thus performs roughly 3% better than $\delta^{(1)}$, indicating that there is a significant benefit in the new way in which sequences are compared.

An interesting observation is that for the Q_3 performance, there is a local maximum for both $\delta^{(3)}$ and $\delta^{(1)}$ at $k = 5$. For $\delta^{(3)}$, the Q_8 values are already in a declining phase, but still a maximum is achieved for Q_3. This might be indicative that certain amino acid sequences form "similar" secondary structures in the eight class problem, in the sense that secondary structures are similar if they are mapped to the same class in the three class problem.

Table 5.18: Comparison between the performance of $\delta^{(3)}$ and $\delta^{(1)}$ using exactly k neighbours

	$\delta^{(3)}$	$\delta^{(3)}$	$\delta^{(3)}$	$\delta^{(3)}$	$\delta^{(1)}$	$\delta^{(1)}$	$\delta^{(1)}$	$\delta^{(1)}$
k	$\#Q_8$	Q_8 (%)	$\#Q_3$	Q_3 (%)	$\#Q_8$	Q_8(%)	$\#Q_3$	Q_3 (%)
1	40411	55.367	48876	66.965	38260	52.420	46544	63.770
2	39061	53.518	46203	63.303	36801	50.421	43147	59.116
3	38673	52.986	46001	63.026	36637	50.197	42737	58.554
4	38754	53.097	46706	63.992	36563	50.095	43320	59.353
5	38545	52.811	46834	64.168	36594	50.138	43492	59.589
6	38302	52.478	46599	63.846	36585	50.125	43341	59.382
7	38044	52.124	46387	63.555	36340	49.790	43001	58.916
8	37952	51.998	46507	63.720	36271	49.695	42950	58.846
9	37781	51.764	46443	63.632	36292	49.724	43058	58.994
10	37545	51.441	46251	63.369	36223	49.629	42986	58.895

It is also useful to understand what happens if $\phi^{(2)}$ is not limited to exactly k neighbours, but is used as originally defined, i.e. that all nearest neighbours to a particular target sequence contribute to classification.

Under this condition, $\delta^{(3)}$ correctly predicts 55.59% of the secondary structures, a marginal improvement over the 55.37% achieved using $k = 1$. In doing so, 94588 neighbours were used, an average of 1.29 neighbours per sequence.

$\delta^{(1)}$ correctly predicts 55.82% of the secondary structures, but uses 294076 neighbours in doing so (an average of 4.02 per sequence). At first glance it may appear that this result should be more or less equal to the one obtained using $k = 4$. It should however be kept in mind that with $k = 4$, every sequence has *exactly* 4 neighbours whilst here the *average* is roughly 4 (thus some sequences have fewer and some more neighbours of equal minimum distance).

$\delta^{(3)}$ and $\delta^{(1)}$ thus have similar performance under $\phi^{(2)}$, however $\delta^{(3)}$ requires much fewer neighbours to achieve this performance than $\delta^{(1)}$. This should be expected, since the similarity values between different amino acid residue types are now much more diverse than under the hard 1/0 function, resulting in a more measurable difference between different sequences. In terms of a pattern recognition problem, this means that the decision boundary used under the $\delta^{(3)}$ metric is "less fuzzy" than under the $\delta^{(1)}$ metric.

Another interesting observation is that of the 294076 neighbours found under $\delta^{(1)}$, only 117270 (39.88%) correctly predict secondary structures when viewed in isolation, yet when neighbours of equal minimum distance are combined per target sequence, it manages to correctly predict 55.82% of the structures. An investigation into the nature of these neighbours (results not listed here) showed that there are more neighbours for sequences that are further from the target sequences. This implies a relationship between the number of qualifying neighbours and the distance of these neighbours from the target sequence. This relationship is further analysed in the "adaptive classification function" experiment (Section 5.5.5).

5.5.3.4 Conclusion

The distance metric based on the substitution matrix created in the previous experiment is an improvement on the distance metric used up to now, in the sense that fewer training samples are required to achieve similar performance. This reinforces the findings about specific amino acids that were found to be similar in the previous experiment.

There is also evidence to suggest that there is a relationship between the number of qualifying neighbours and the distance of those neighbours to the target sequence.

5.5.4 Experiment: Distance Metric - BLOSUM

5.5.4.1 Objective

As was the case in the previous experiment, the objective of this experiment is to design a distance metric based on a substitution matrix. This time, an existing substitution matrix, namely the BLOSUM matrix (refer to Table 2.4) is used. If the algorithm performs well using this metric, it implies that the matrix is indicative of good amino acid substitutions.

5.5.4.2 Protocol

A metric was designed based on the BLOSUM matrix (refer to Table 2.4). The notation $\delta^{(B)}$ will be used to indicate this metric. The metric is defined by

$$d_{a,b}^{(B)} = \delta^{(B)}(\overline{g}_a, \overline{g}_b) = -\sum_{i=1}^{p} s_i, \tag{5.9}$$

where s_i is the entry in the BLOSUM matrix for substituting $g_{a,i}$ with $g_{b,i}$. Note that $G = R$ is used, with L the identity mapping. This ensures that the group labels are simply the residue types, which makes it possible to use the matrix. The matrix does not define substitution values for edges. A value of $s = 12$ was used for substitution of an edge with another edge and a value of $s = -3$ for substitution of an edge with an amino acid or vice versa.

Note the minus sign in the distance metric. Positive values in the BLOSUM matrix indicate likely substitutions and negative values unlikely substitutions. The minus sign is used to ensure smaller distance values for patterns that are more alike to one another. Note that under this metric, distances of less than 0 are possible. The restriction that $d_{a,b} \geq 0$ is relaxed in this case, since it does not influence the execution of the algorithm and the results achieved with it. By adding a constant value of $17N$ (17 being the largest value for any substitution) to the distance calculation, the metric can easily be guaranteed to evaluate to a value greater or equal to zero.

Experiments were executed for $N = 15$ ($l = 7, r = 7$). Different ϵ values in the range [-45,10] were examined. $\phi^{(1)}$ and $\phi^{(2)}$ were tested as classification functions and $\psi^{(1)}$ as assignment function.

5.5.4.3 Results and Discussion

The results obtained are shown in Table 5.19 and are illustrated in Figure 5.23. An increase from 51.57% to 53.75% for $\phi^{(1)}$ and from 55.19% to 56.18% for $\phi^{(2)}$ is achieved using $\delta^{(B)}$ instead of $\delta^{(1)}$.

Table 5.19: Performance achieved by using BLOSUM distance metric

ϵ	$Q_8(\phi^{(1)})$	$Q_8(\phi^{(2)})$	ϵ	$Q_8(\phi^{(1)})$	$Q_8(\phi^{(2)})$	ϵ	$Q_8(\phi^{(1)})$	$Q_8(\phi^{(2)})$
-45	50.2350	50.2377	-25	51.6133	56.1799	-5	43.4735	56.1744
-44	50.6487	50.6556	-24	51.6284	56.1730	-4	42.7473	56.1744
-43	51.0461	51.0666	-23	51.3941	56.1744	-3	42.0253	56.1744
-42	51.3982	51.4366	-22	51.3023	56.1744	-2	41.1936	56.1744
-41	51.7339	51.8106	-21	51.0420	56.1744	-1	40.4387	56.1744
-40	52.0641	52.1942	-20	50.8090	56.1744	0	39.6947	56.1744
-39	52.3710	52.5696	-19	50.6433	56.1744	1	38.8110	56.1744
-38	52.7053	52.9642	-18	50.2692	56.1744	2	38.1232	56.1744
-37	53.0642	53.4218	-17	49.9034	56.1744	3	37.2998	56.1744
-36	53.3848	53.8863	-16	49.5842	56.1744	4	36.4983	56.1744
-35	53.5863	54.3206	-15	49.2197	56.1744	5	35.7173	56.1744
-34	53.7575	54.8002	-14	48.5826	56.1744	6	34.8843	56.1744
-33	53.6657	55.1701	-13	48.1360	56.1744	7	34.0280	56.1744
-32	53.4876	55.5784	-12	47.5852	56.1744	8	33.2730	56.1744
-31	53.1574	55.7866	-11	47.0440	56.1744	9	32.4948	56.1744
-30	52.7080	55.9579	-10	46.5069	56.1744	10	31.8783	56.1744
-29	52.2723	56.0305	-9	45.9767	56.1744			
-28	52.0449	56.1470	-8	45.3807	56.1744			
-27	51.9312	56.1785	-7	44.7847	56.1744			
-26	51.8476	56.1867	-6	44.1969	56.1744			

The good performance under this metric indicates that the substitution values in the matrix are good indicators of the similarity between different amino acid residue types. It is thus a good idea to further investigate the values in the matrix.

The first observation is that the matrix is symmetrical, i.e. substitutions between different residue types are commutative. Both C and W, and to a lesser extent P, are not well substituted with any other other residue type, as was found in the previous experiment. G now joins the ranks of amino acids that are not well substituted by other amino acids. F and Y are a good substitution, as are I and V, M and L and all four these with each other as was found in the previous experiment. R and K are good substitutes as are E and Q. However, unlike in the previous experiment, there is no strong correspondences between R and K with E and Q. H and A can be substituted with elements from the R, K, E, Q group but there is no strong correspondence. The similarity between N and D and between D and E is confirmed using this matrix, and to a lesser extent the similarities between S and T and between N and K.

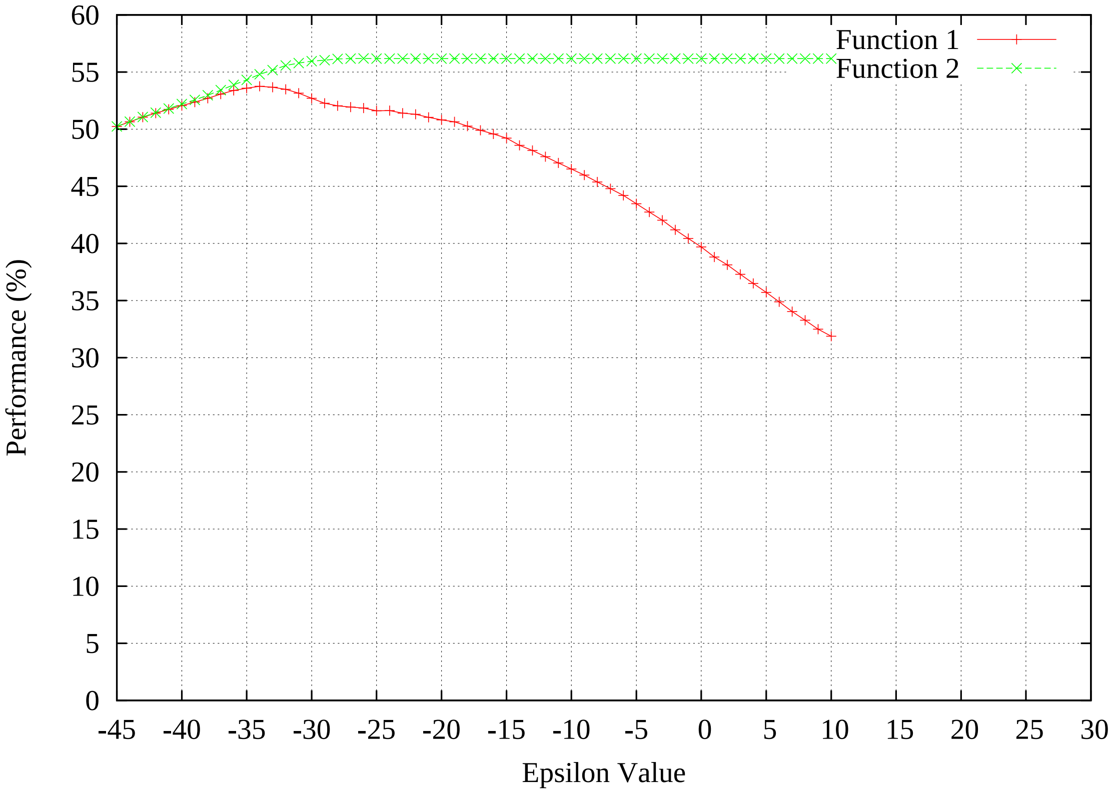


Figure 5.23: Performance achieved by using BLOSUM distance metric

5.5.4.4 Conclusion

The experiment confirmed that there are similarities between certain types of amino acid residues. Although there are exceptions, most of these similarities are the same as those found in the "grouping strategies" and "substitution matrix" experiments.

5.5.5 Experiment: Adaptive Classification Function

5.5.5.1 Objective

In the "distance metric - substitution matrix" experiment (Section 5.5.3), it was shown that better performance is achieved if the number of similar sequences that are used in the prediction of the secondary structure associated with a particular target sequence is not fixed, but rather depends on the target sequence itself. The objective of this experiment is to see whether a more intelligent choice can be made in the classification function, and in doing so, how the dependency between the number of similar sequences and their distance from the target sequence is quantified.

5.5.5.2 Protocol

A window size of 15 ($l = 7, r = 7$) was used. The set of group labels were the same as the set of residue labels, that is $G = R$, with L the identity function. $\psi^{(1)}$ was used as assignment function.

The distance metric $\delta^{(3)}$ was used, with the matrix U as defined by Equation 5.8. Classification functions $\phi^{(4)}$ and $\phi^{(5)}$ were tested (refer to Sections 4.6.4 and 4.6.5 respectively). For $\phi^{(4)}$, d values in the range [0, 1.5] were tested and for $\phi^{(5)}$, c values in the range [1, 1.5]. Given that the distance from the target sequence to its nearest neighbour in the training set is given by m, $\phi^{(4)}$ simply states that all sequences in the training set that are as close as $m + d$ should take part in the classification process. Likewise, $\phi^{(5)}$ simply states that all sequences in the training set that are as close as mc should take part in the classification process.

5.5.5.3 Results and Discussion

The resulting performance using $\phi^{(4)}$ is shown in Table 5.20 and Figure 5.24. The resulting performance using $\phi^{(5)}$ is shown in Table 5.21 and Figure 5.25.

Both classification functions achieve a best performance of about 59.2%, a substantial improvement on the 55.59% achieved using $\phi^{(2)}$ under similar test conditions (which is by design the value achieved using $d = 0$ and $c = 1$). This performance is achieved using $d = 0.35$ and $c = 1.18$. The average number of qualifying neighbours used to achieve this performance are 14.06 and 15.52 respectively.

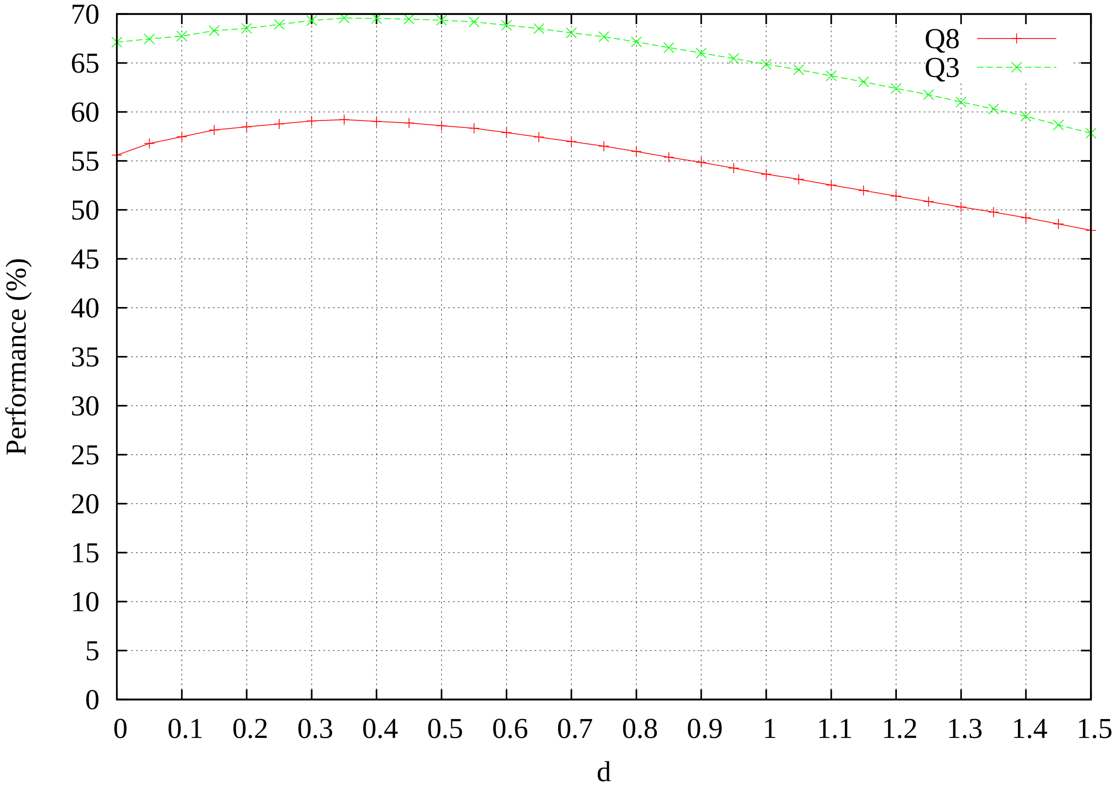


Figure 5.24: Performance using $\phi^{(4)}$

5.5.5.4 Conclusion

The number of contributing neighbours used for classification of a particular sequence should not be a fixed number but should be dependent on properties of the sequence itself. In this experiment, it was found that better prediction results are achieved if all

Table 5.20: Performance using $\phi^{(4)}$

d	$\#Q_8$	Q_8 (%)	$\#Q_3$	Q_3 (%)	total neighbours	neighbours per sequence
0.00	40578	55.596	48986	67.116	94588	1.296
0.05	41444	56.783	49233	67.454	125384	1.718
0.10	41942	57.465	49440	67.738	170793	2.340
0.15	42446	58.156	49856	68.308	241928	3.315
0.20	42687	58.486	50019	68.531	343268	4.703
0.25	42893	58.768	50311	68.931	493645	6.763
0.30	43123	59.083	50596	69.322	711592	9.750
0.35	43215	59.209	50789	69.586	1026242	14.061
0.40	43083	59.028	50759	69.545	1480537	20.285
0.45	42971	58.875	50710	69.478	2128628	29.164
0.50	42768	58.597	50624	69.360	3059005	41.912
0.55	42578	58.336	50509	69.203	4349743	59.596
0.60	42250	57.887	50256	68.856	6145464	84.199
0.65	41914	57.427	50009	68.518	8625779	118.182
0.70	41581	56.970	49685	68.074	12008961	164.536
0.75	41242	56.506	49397	67.679	16659348	228.251
0.80	40849	55.968	49016	67.157	22827053	312.755
0.85	40420	55.380	48592	66.576	31034220	425.202
0.90	40034	54.851	48177	66.008	41871486	573.684
0.95	39607	54.266	47771	65.451	55999729	767.256
1.00	39152	53.642	47348	64.872	74555133	1021.485
1.05	38767	53.115	46937	64.309	98074815	1343.730
1.10	38342	52.533	46484	63.688	127959621	1753.184
1.15	37941	51.983	46034	63.072	165675807	2269.936
1.20	37521	51.408	45551	62.410	212712873	2914.394
1.25	37115	50.852	45083	61.769	271778896	3723.662
1.30	36709	50.295	44531	61.012	343354249	4704.321
1.35	36330	49.776	44014	60.304	430388303	5896.780
1.40	35913	49.205	43466	59.553	535486618	7336.740
1.45	35445	48.563	42828	58.679	660840087	9054.216
1.50	34962	47.902	42202	57.821	811391789	11116.936

Table 5.21: Performance using $\phi^{(5)}$

c	$\#Q_8$	Q_8 (%)	$\#Q_3$	Q_3 (%)	total neighbours	neighbours per sequence
1.00	40578	55.596	48986	67.116	94588	1.296
1.01	40953	56.110	49058	67.215	104470	1.431
1.02	41236	56.498	49107	67.282	116502	1.596
1.03	41500	56.859	49237	67.460	130900	1.793
1.04	41702	57.136	49316	67.568	147892	2.026
1.05	41822	57.301	49371	67.644	168131	2.304
1.06	42065	57.634	49567	67.912	191613	2.625
1.07	42196	57.813	49645	68.019	219636	3.009
1.08	42400	58.093	49839	68.285	252742	3.463
1.09	42554	58.304	49975	68.471	291863	3.999
1.10	42639	58.420	50017	68.529	337654	4.626
1.11	42743	58.562	50164	68.730	391414	5.363
1.12	42851	58.710	50323	68.948	454026	6.221
1.13	42834	58.687	50355	68.992	528269	7.238
1.14	42991	58.902	50478	69.160	614324	8.417
1.15	43088	59.035	50584	69.305	715767	9.807
1.16	43110	59.065	50617	69.351	834108	11.428
1.17	43174	59.153	50728	69.503	972358	13.322
1.18	43230	59.230	50800	69.601	1133385	15.529
1.19	43172	59.150	50819	69.627	1319442	18.078
1.20	43123	59.083	50798	69.599	1535899	21.043
1.21	43203	59.193	50917	69.762	1787768	24.494
1.22	43171	59.149	50917	69.762	2079701	28.494
1.23	43105	59.058	50864	69.689	2417576	33.123
1.24	43009	58.927	50799	69.600	2806913	38.458
1.25	43027	58.952	50834	69.648	3256182	44.613
1.26	42915	58.798	50721	69.493	3774825	51.719
1.27	42857	58.719	50736	69.514	4368207	59.849
1.28	42823	58.672	50689	69.449	5049444	69.183
1.29	42771	58.601	50654	69.401	5830512	79.884
1.30	42735	58.552	50656	69.404	6723672	92.122
1.31	42677	58.472	50630	69.369	7744682	106.110
1.32	42577	58.335	50505	69.197	8910715	122.086
1.33	42474	58.194	50390	69.040	10237340	140.263
1.34	42343	58.014	50313	68.934	11742912	160.890
1.35	42220	57.846	50192	68.768	13452230	184.310
1.36	42156	57.758	50155	68.718	15390154	210.862
1.37	42052	57.616	50087	68.625	17584299	240.924
1.38	41932	57.451	49967	68.460	20061561	274.865
1.39	41821	57.299	49828	68.270	22857820	313.177
1.40	41674	57.098	49674	68.059	25998688	356.210
1.41	41613	57.014	49614	67.976	29534671	404.657
1.42	41502	56.862	49498	67.818	33495635	458.926
1.43	41395	56.716	49412	67.700	37934486	519.743
1.44	41290	56.572	49299	67.545	42897131	587.737
1.45	41222	56.479	49216	67.431	48437402	663.644
1.46	41149	56.379	49158	67.352	54609948	748.215
1.47	41029	56.214	49023	67.167	61475988	842.287
1.48	40902	56.040	48850	66.930	69094126	946.663
1.49	40790	55.887	48707	66.734	77547428	1062.483
1.50	40719	55.789	48604	66.593	86852705	1189.975

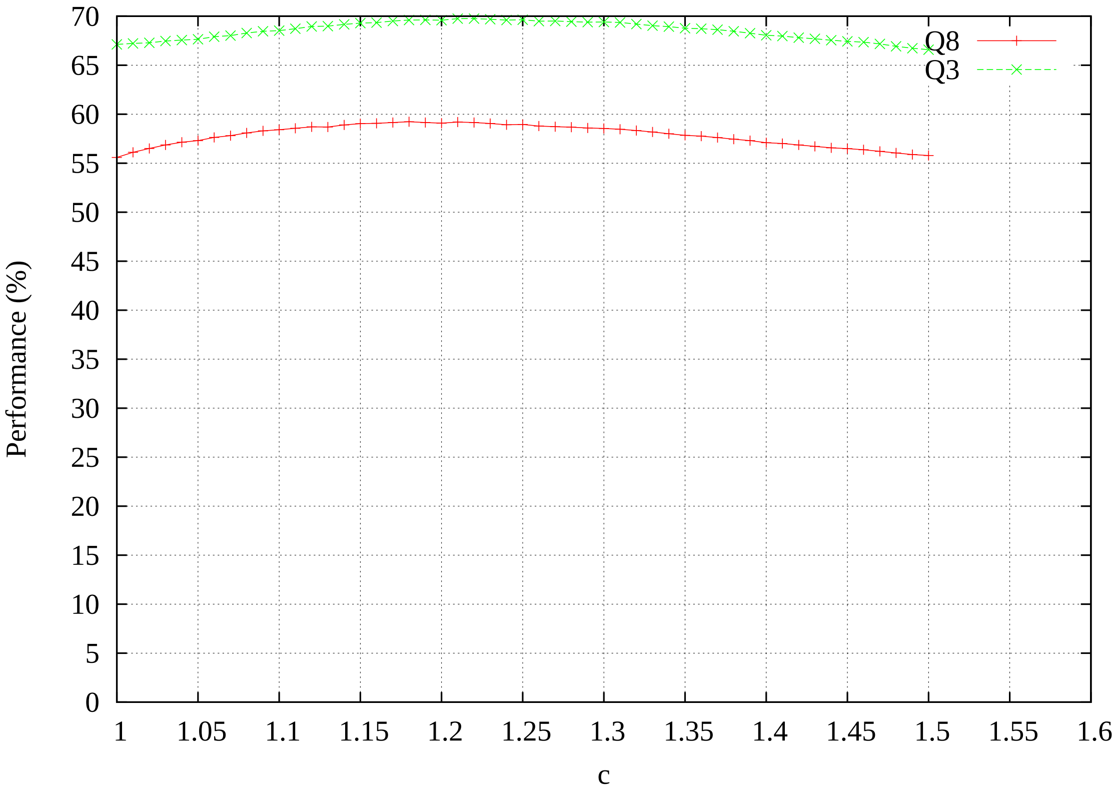


Figure 5.25: Performance using $\phi^{(5)}$

neighbours within a band of the nearest neighbour to the target sequence contribute to the classification. This size of this band can either be a small fixed value (0.35 under $\delta^{(3)}$ with $N = 15$) or can depend on the distance of the nearest neighbour (in this case a width of 0.18 times the distance of the nearest neighbour was found to be effective under $\delta^{(3)}$). The latter method seems slightly more preferable, since it is invariant with respect to the size of the window.

A Q_8 score of 59.2% was achieved and a Q_3 score of 69.76%. It should be noted that in recent results published by Martin et al [96], Q_3 scores of 67.9% and 66.8% for the OSS-HMM and PSIPRED predictions on single sequences were achieved. The current method thus compares well with some of the best existing methods.

5.5.6 Experiment: Use of Secondary Structure Information

5.5.6.1 Objective

In all previous experiments, a secondary structure is predicted by comparing the sequence of amino acids associated with that secondary structure to other sequences in the training set. The prediction of the secondary structure is based solely on the secondary structures associated with similar sequences. It is however known that there is a strong correspondence between neighbouring secondary structures [96]. For instance, given that a number of consecutive alpha helix structures have been observed, there is a strong preference for the next secondary structure to be a helix as well. The objective of this experiment is investigate whether predicted secondary structure information can be fruitfully incorporated in the prediction process.

5.5.6.2 Protocol

A window size of 15 ($l = 7, r = 7$) was used. The set of group labels were the same as the set of residue labels, that is $G = R$, with L the identity function. $\phi^{(2)}$ was used as classification function and $\psi^{(1)}$ was used as assignment function.

The idea in this experiment is that already predicted secondary structures should be incorporated in the prediction process to predict neighbouring secondary structures. Initially however, there will be no such predicted secondary structures to begin with. It should also be noted that there is an uncertainty in any predicted secondary structure: thus, good predictions are required to start the sequence off.

Target proteins are considered one at a time. The process followed is an iterative one. In each iteration, one or more secondary structures are predicted at different positions in the protein. In following iterations, it is assumed that already predicted secondary structures were correctly predicted, and subsequent predictions are based on this assumption. It is thus entirely possible that an incorrectly predicted secondary structure could steer the whole process in a wrong direction. For this reason, at each iteration, the only secondary structures predicted are the ones with the highest confidence of being correct.

To illustrate the idea further, Figure 5.26 shows an example of a prediction that was done for a protein in the test set. The line marked "-P" is the primary structure of the protein and the line marked "-S" the secondary structure of the protein. The lines from "01" to "27" indicate that 27 iterations were necessary to predict all 55 secondary structures in the protein and each corresponding line shows the secondary structures that was predicted up to that iteration. In the final line, a star (*) indicates which secondary structures were correctly assigned. The four columns to the side of each iteration indicate respectively the cumulative number of predicted secondary structures at that iteration, the cumulative number of correctly predicted secondary structures at that iteration, the Q_8 value at that iteration and a similarity value used in that iteration; a concept that will be explained below.

In the first iteration, 11 secondary structures were predicted. These predictions were based solely on the primary structure. Furthermore, the algorithm determined that these 11 predictions are the most likely (and equally likely) candidates in all the positions of the protein. In the second iteration, 4 additional predictions were made. This time however, the primary structure information was used and it was assumed that the 11 predicted secondary structures in the previous iteration were correctly predicted. Of course, of the 11 predicted structures only 6 were correctly predicted. The impact it had can be observed by considering the sequence of six secondary structures CCCCST in the first iteration. Of these CCCC were correctly predicted but ST were incorrectly

Figure 5.26: Example of iteratively incorporating secondary structure information in the prediction process

```
-P AYVINEACISCGACEPECPVDAISQGGGSRYVIDADTCIDCGACAGVCPVDAPVQA
-S CEEECTTCCCCCTTGGGCTTCCEECCSSSCEECTTTCCCCCHHHHTCTTCCEEEC
01 CC----------------------------------CCCCST----------CCC 11 6 54.54 39
02 CC--------------------------------TTCCCCSTTT--------CCC 15 8 53.33 42
03 CC-------------------------------TTTCCCCSTTTTTC-----CCC 19 11 57.89 43
04 CC-------------------------------TTTCCCCSTTTTTC--C--CCC 20 12 60.00 44
05 CC-------------------------------TTTCCCCSTTTTTCT-CC-CCC 22 14 63.63 44
06 CC-------------------------------TTTCCCCSTTTTTCTTCCBCCC 24 15 62.50 45
07 CC------------------------------CTTTCCCCSTTTTTCTTCCBCCC 25 16 64.00 43
08 CC-----------------------------ECTTTCCCCSTTTTTCTTCCBCCC 26 17 65.38 43
09 CC----------------------------EECTTTCCCCSTTTTTCTTCCBCCC 27 18 66.66 43
10 CC---------------------------EEECTTTCCCCSTTTTTCTTCCBCCC 28 18 64.28 43
11 CC--------------------------SEEECTTTCCCCSTTTTTCTTCCBCCC 29 19 65.51 43
12 CC-------------------------SSEEECTTTCCCCSTTTTTCTTCCBCCC 30 20 66.66 40
13 CC------------------------SSSEEECTTTCCCCSTTTTTCTTCCBCCC 31 21 67.74 40
14 CC-----------------------CSSSEEECTTTCCCCSTTTTTCTTCCBCCC 32 22 68.75 40
15 CC----------------------ECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 33 22 66.66 40
16 CC---------------------EECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 34 23 67.64 40
17 CC----------------T----EECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 35 24 68.57 39
18 CC---------------ST-C--EECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 37 25 67.56 39
19 CC--------------TSTTC-EEECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 40 27 67.50 40
20 CC--------------TSTTCCEEECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 41 28 68.29 42
21 CC-------------TTSTTCCEEECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 42 28 66.66 40
22 CC------------TTTSTTCCEEECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 43 28 65.11 40
23 CCEECTT-------TTTSTTCCEEECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 48 33 68.75 38
24 CCEECTTC------TTTSTTCCEEECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 49 34 69.38 40
25 CCEECTTCC-----TTTSTTCCEEECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 50 35 70.00 40
26 CCEECTTCCC--TTTTTSTTCCEEECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 53 38 71.69 40
27 CCEECTTCCCCCTTTTTSTTCCEEECSSSEEECTTTCCCCSTTTTTCTTCCBCCC 55 40 72.72 42
-- *.************....******.****.**********.....******...*
```

predicted. In the second iteration, this had a likely influence on predicting the two TT structures to the left and right of CCCCST, of which TT structure to the left of the correctly CCCC structure is correctly predicted but the TT structure to the right of the incorrectly predicted ST is also incorrect.

The question now becomes how the algorithm decides which secondary structure to predict next and how already predicted secondary structures are incorporated in the prediction. The solution presented in the algorithm is to adapt the distance metric. During each iteration, all unpredicted secondary structures are considered for prediction. For each of these, a window of length 15 is created in the target protein and both the primary structure and partially predicted secondary structure is noted. Thus, for every such window, there are exactly 15 amino acid residues and between 0 and 14 partially predicted secondary structures. This sequence of amino acids and secondary structures is then compared to similarly construed structures in the training set.

Comparison of amino acids is straightforward, and can be done using any of the already created distance metrics. The algorithm was however slightly adapted such that a score of w is assigned to two matching residue types, and a score of 0 is assigned to two non-matching residue types. A similarity value is then calculated as the sum of all these values over 15 residues. Comparison of secondary structures is slightly more complicated. If a partially predicted secondary structure matches a secondary structure in the training set in the same position, a value of 1 is assigned. If a partially predicted secondary structure does not match the secondary structure in the training set in the same position, or if no prediction has been made, a value of 0 is assigned. A value is then calculated as the sum of all these values over the 14 secondary structures. The combined residue and secondary structure score is then used as a similarity value. The algorithm was tested for $w \in [1, 4]$.

During each iteration, all the similarity values are calculated for all unpredicted secondary structures. All structures with the highest similarity values are retained and a prediction of secondary structure is then made using a process akin to that used with $\phi^{(2)}$.

5.5.6.3 Results and Discussion

The results of the experiment are shown in Table 5.22. The best result is obtained using $w = 3$, with a Q_8 score of 56.87% (a comparative Q_3 score of 67.09% was achieved). $w = 2$ and $w = 4$ perform similarly, but $w = 1$ performs significantly worse. This is to be expected, since with $w = 1$ each predicted secondary structure contributes as much to the similarity value as each amino acid in the primary structure. A number of consecutive incorrect predictions can thus more easily lead the process astray. With a larger value of w, it is easier for the algorithm to "resynchronise".

Table 5.22: Performance achieved using different methods incorporating predicted secondary structure information

Method	$\#Q_8$	Q_8 (%)
$w = 1$	39351	53.915
$w = 2$	41195	56.442
$w = 3$	41513	56.877
$w = 4$	41431	56.765
$w = 3$ (no edges)	40862	55.985
$w = 3$ (no coils)	41301	56.587

A test was conducted to see the effect that edges have on a prediction. In the "edge analysis" experiment (Section 5.3.3), it was demonstrated that coils are very likely to form near the edge of the protein. This behaviour was readily observed in analysis of the order in which secondary structures are predicted. Consider Figure 5.26 as an example, where the coil structures towards the edges of a protein are predicted first, and other structures are then predicted working inwards. This behaviour could possibly bias structures toward the center of the protein, which are more likely to contain biological function. To counter this effect, the function calculating similarity was changed in such a way that edge types in the primary structure do not contribute to the calculated similarity values. The forming behaviour changed such that structures toward the center of the protein are predicted first. However, the achieved performance dropped to 55.985%. Since no improvement was made in the performance (and actually an inferior result was achieved), it might be concluded that it is useful to include edge information in the prediction process.

A test was also conducted where predicted coil secondary structures do not contribute to the similarity score. The idea was that since coils do not form regular structures, their predictive power may be limited. In this scenario, the performance achieved reduced slightly to 56.59%. Since no improvement was made in the performance, it is not harmful to include coils as predicted secondary structures.

5.5.6.4 Conclusion

The performance of 56.87% achieved by including predicted secondary structures in the prediction process is better than the 55.82% achieved in a prior experiment under similar circumstances. It can thus be concluded that secondary structures are predictive of other secondary structures, but that it is difficult to incorporate this information to achieve significantly better performance scores.

This is made especially difficult in some cases where there is difficulty in making good predictions initially. For such cases, inclusion of predicted secondary structures in the prediction process may lead it astray rather than improving it.

A good feature about this method is that it can be descriptive of some theories regarding the actual forming process. In the nucleation and directed folding models (Section 2.2.3) local stable folded conformations form, from which the eventual structure of the protein is determined. Similar behaviour is observed using this iterative method. First, local structures that are the most likely to form at certain positions in a sequence are predicted. The process continues by filling in "gaps" and/or predicting other local structures, propagating from the already formed structures.

Chapter 6

CONCLUSION

6.1 KEY FINDINGS

The best performance achieved using the method developed in this dissertation for secondary structure prediction is $Q_8 = 59.2\%$. The comparative Q_3 score is 69.76%. In a recent study (2006), Martin et al [96] reported Q_3 scores of 67.9% and 66.8% for OSS-HMM and PSIPRED, two of the leading techniques for prediction of secondary structure. These results are achieved when predictions are made on single sequences, as is done in this dissertation. It is difficult to compare the results directly, since different datasets are used. It is safe to say that the new method compares well with the leading existing methods. It should be noted however that OSS-HMM achieves a score of 75.5% [96] and PSIPRED a score of 76% [65] when multiple sequence alignments are used. Multiple sequences alignments have not been considered in this dissertation.

A number of key findings have been made. Of these, the main ones are discussed below.

- Good predictions can be made when sections of the primary sequence in a target protein can be mapped to similar sequences in a training set, especially for larger stretches of matching sequences, i.e. longer sequences have more predictive power then smaller sequences. This is however practically limited for larger sequences by the amount of training data available, since not all possible target sequences would be covered in the training data. It is thus necessary to have some method by which the similarity of different sequences can be compared.

- Information about which secondary structure would form for a particular sequence of amino acids is distributed across the whole window. However, there is a tendency for more central amino acids to contribute more to secondary structure.

- The similarity of sequences can be expressed as a measure of the similarity of amino acids in matching positions. This similarity can be quantified through the creation of a similarity matrix. One observation from the similarity matrix is that substitution between two residue types is not totally commutative. Specific groups of similar amino acids residues that have been found through different experiments are: (IV)(LM), (RK)(EQ)(H)(A), ST, FY, DN, NK and ED. C, W and P are not well substituted by any other residue type.

- An interesting effect, named the "transfer phenomenon" is observed, namely that secondary structures that can be predicted using sequences of both lengths N and $N+1$, are considerably more accurate than secondary structures that can be predicted using sequences of length N but not $N+1$, even when sequences of only length N are considered. This occurs for N from about 3 to 7 and where an exact match is required to make a prediction.

- It is advantageous to use a number of sequences similar to a target sequence when a secondary structure is predicted. The number of such similar sequences that should be used is not fixed but rather is dependent on the distance of those sequences to the target sequence. Good performance is achieved when all neighbours that contribute to the prediction lie within a certain band of the distance of the nearest neighbour. The size of the band can either be a small fixed value or a small multiple of the distance of the nearest neighbour.

- Secondary structures are predictive of other secondary structures. In order to incorporate this fact into a prediction scheme requires use of the already predicted secondary structures. This implies making good predictions initially. Due to the inherent uncertainty in the predictions, it is difficult to incorporate relationships between secondary structures in the prediction process in order to achieve better results.

6.2 FUTURE WORK

A number of suggestions for future work are discussed below. Each of these is believed to add valuable insight in understanding the formation of secondary structures and can be used to further enhance the method developed in this dissertation.

6.2.1 Iterative Adaptation of Substitution Matrix

The substitution matrix developed in Section 5.5.2 was created using the hard $\delta^{(1)}$ distance metric. Using this substitution matrix, a new distance metric $\delta(3)$ was developed in the "distance metric - substitution matrix" experiment (Section 5.5.3).

One idea is that this process can be iteratively repeated, i.e. the new distance metric can be used instead of the old distance metric, to create a new substitution matrix. The new substitution matrix is then used to create a new distance metric, and this process is then repeated until values in the substitution matrix settle.

Although it is suspected that the values in the final substitution matrix will not differ much from the ones in the current matrix, it will be a more truthful expression of the similarity between different residue types. It may also lead to better classification performance.

6.2.2 Position Specific Substitution Matrices

The matrices C^k and I^k (defined by Equations 5.4 and 5.5 respectively), can be used to define position specific substitution matrices. These matrices indicate the similarity of amino acids in specific positions in a window.

By studying these matrices, it may be possible to determine whether there are position specific substitutions that influence the formation of secondary structures at the central amino acid.

These matrices can also be iteratively adapted. By incorporating these matrices into a new distance metric, it may be possible to increase the performance score.

6.2.3 Weight Assignment

Experiments such as "window structure" (Section 5.4.1) and "varying window size" (Section 5.4.2) indicated that central amino acids influence the formation of local secondary structure more than amino acids toward the edges of a window.

This influence has not been quantified and could perhaps be used with success in distance metrics such as $\delta^{(2)}$ and $\delta^{(3)}$, where equal weight assignments have been made in current experiments. The influence could perhaps be quantified by studying the C^k and I^k matrices or using a brute force search for appropriate values.

6.2.4 Secondary Structure Similarity

It has been suggested in the experiments that some secondary structures may be more alike than others. This is an assumption that is often implicit in secondary structure research, where the eight classes in the DSSP code are mapped to three, implying similarity between classes that map to the same structure.

This similarity has not been quantified in the experiments conducted, and it may be interesting to determine how alike different secondary structures are. It may be possible to use an approach similar to that used in the creating of the substitution matrix to create a secondary structure similarity matrix.

6.2.5 Use of Predicted Secondary Structure in Other Predictions

The "Use of Secondary Structure Information" experiment (Section 5.5.6) indicated that secondary structures are predictive of other secondary structures, but that it is difficult to incorporate this information to achieve significantly better results using the suggested algorithm.

Perhaps other methods which incorporate predicted secondary structures in the prediction process could be created, or the current method extended. One way to extend the current method is to determine whether there are small sequences (window size of seven and smaller) that are reliably indicative of secondary structures. These good predictions can then be used (together with larger matching sequences) to start off the prediction process.

Another way could be to incorporate a secondary structure similarity matrix as discussed in Section 6.2.4 as well as the substitution matrix to create a better distance metric for matching structures.

It may also be possible to use a probabilistic approach when assigning secondary structures. Thus, instead of assigning a specific secondary structure to a specific position (and thereafter assuming that it was correctly predicted), it may be possible to assign probabilities of observing the different secondary structures to each such position. These probabilities are then used in subsequent iterations. It may even be possible to adapt the method such that the probabilities can change in subsequent iterations.

The findings of the “adaptive classification function” experiment (Section 5.5.5) also need to be included, which will further improve performance results. Finally, there is good reason to suspect that the substitution matrices should themselves actually depend on the surrounding secondary structure.

6.2.6 Multiple sequence alignment

The current method is applicable to the prediction of single sequences. This method may be extended such that multiple sequence information is taken into account.

It is suspected that this will further increase performance of the algorithm, and will make it possible to compare this method more reliably with others found in literature.

Bibliography

Discovery of proteins, amino acids and the peptide bond

[1] FJ Mulder, "Über die Zusammensetzung einiger thierischen Substanzen (on the composition of some animal substances)," *Journal für praktische Chemie*, Vol. 16, p. 129, 1839.

[2] H Ritthausen, *The Proteins of the Cereals, Legumes and Oil Seeds*, Bonn, 1872.

[3] F Hofmeister, "Über Bau und Gruppierung der Eiweisskorper (On the Structure and Grouping of the Protein Bodies)," *Ergebnisse der Physiologie*, Vol. 1, pp. 759-802, 1902.

[4] HE Fischer, "Über die Hydrolyse der Proteinstoffe (On the Hydrolysis of Proteins)," *Chemiker-Zeitung*, Vol. 26, pp. 939-940, 1902.

[5] HB Vickery, "Evidence from Organic Chemistry Regarding the Composition of Protein Molecules," *Annals of the New York Academy of Sciences*, Vol. 41, p. 87, 1941.

[6] AJP Martin and RLM Synge, "Analytical chemistry of the proteins," *Advances in Protein Chemistry*, Vol. 2, pp. 1-83, 1945.

[7] RLM Synge, "Partial Hydrolysis Products Derived from Proteins and Their Significance for Protein Structure,“ *Chemical Reviews*, Vol. 32, p. 135-172, 1943.

[8] F Sanger, "The arrangement of amino acids in proteins," *Advances in Protein Chemistry* Vol. 7, pp. 1-67, 1952.

[9] MF Perutz, MG Rossmann, AF Cullis, G Muirhead, G Will and AT North, "Structure of haemoglobin: A three-dimensional Fourier synthesis at 5.5Åresolution, obtained by X-ray analysis," *Nature*, Vol. 185, pp. 416-422, 1960.

[10] JC Kendrew, RE Dickerson, BE Strandberg, RJ Hart, DR Davies and DC Philips, "Structure of myoglobin: A three-dimensional Fourier synthesis at 2Åresolution," *Nature*, Vol. 185, pp. 422-427, 1960.

[11] JD Watson and FH Crick, "Molecular Structure of Nucleic Acids," *Nature*, Vol. 171, pp. 737-738, 1953.

[12] G Gamow, "Possible relation between deoxyribonucleic acid and protein structures," *Nature*, Vol. 173, p. 318, 1954.

[13] JH Matthaei and MW Nirenberg, "Characteristics and Stabilization of DNA ase-Sensitive Protein Synthesis in E. coli Extracts," *Proceedings of the Natural Academy of Sciences*, Vol. 47, pp. 1580-1588, 1961.

[14] MW Nirenberg and JH Matthaei, "The Dependence of Cell-Free Protein Synthesis in E. coli upon Naturally Occurring or Synthetic Polyribonucleotides," *Proceedings of the Natural Academy of Sciences*, Vol. 47, pp. 1588-1602, 1961.

[15] HG Khorana, "Polynucleotide synthesis and the genetic code," *Cold Spring Harbor Symposia on Quantitative Biology*, Vol. 31, pp. 39-49, 1966.

[16] GN Ramachandran et al, "Stereochemistry of polypeptide chain configurations," *Journal of Molecular Biology*, Vol. 7, pp. 95-99, 1963.

[17] RD Fleischmann et al, "Whole genome random sequencing and assembly of haemophilus influenzae," *Science* Vol. 269, pp. 496-512, 1995.

[18] F Sanger, S Nicklen and AR Coulson, "DNA sequencing with chain-terminating inhibitors," *Proceedings of the Natural Academy of Sciences*, Vol. 74, pp. 5463-5467, 1977.

[19] F Sanger et al, "The nucleotide sequence of bacteriophage phi-X714," *Journal of Molecular Biology*, Vol. 125, pp. 225-246, 1977.

[20] F Sanger et al, "Nucleotide sequence of bacteriophage phi-X714," *Nature*, Vol. 165, pp. 687-695, 1977.

[21] JD Watson, "The Human Genome Project, past, present and future," *Science*, Vol. 248, pp. 44-49, 1990.

[22] International Human Genome Sequencing Consortium, "Initial sequencing and analysis of the human genome," *Nature*, Vol. 409, pp. 860-921, 2001.

[23] International Human Genome Sequencing Consortium, “Finishing the euchromatic sequence of the human genome,” *Nature*, Vol. 431, pp. 931-945, 2004.

[24] SG Gregory et al, “The DNA sequence and biological annotation of human chromosome 1,” *Nature*, Vol. 441, pp. 315-321, 2006.

[25] LD Stein, “Human genome: End of the beginning”, *Nature*, Vol. 431, pp. 915-916, 2004.

[26] *Collier’s Encyclopedia Volume 19*, Macmillan Educational Corporation, New York, pp. 426-430, 1979.

Protein Folding

[27] CB Anfinsen, E Haber, M Sela and FW White, “Kinetics of Formation of Native Ribonuclease during Oxidation of the Reduced Polypeptide Chain,” *Proceedings of the Natural Academy of Sciences*, Vol. 47, pp. 1309-1314, 1961.

[28] G Taubes, “Misfolding the way to disease,” *Science*, Vol. 271, pp. 1493-1495, 1996.

[29] PJ Thomas, B Qu and PL Pedersen, “Defective Protein Folding as a Basis of Human Disease,” *Trends in Biochemical Sciences*, Vol. 20, pp. 456-459, 1995.

[30] C Hooper, “An exciting ‘if’ in Alzheimer’s,” *The Journal of NIH Research*, Vol. 3, pp. 65-70, 1991.

[31] E Haber and CB Anfinsen, “Side-chain interactions governing the pairing of half-cystine residues in ribonuclease,” *Journal of Biological Chemistry*, Vol. 237, pp. 1839-1844, 1962.

[32] CJ Levinthal, “Are there pathways for protein folding?,” *Journal of Chemical Physics*, Vol. 65, pp. 44-45, 1968.

[33] CJ Epstein, RF Goldberger and CB Anfinsen, “The genetic control of tertiary protein structure: studies with model systems,” *Cold Spring Harbor Symposia on Quantitative Biology*, Vol. 28, pp. 439444, 1963.

[34] DB Wetlaufer, “Nucleation, rapid folding, and globular intrachain regions in proteins,” *Proceedings of the Natural Academy of Sciences*, Vol. 70, pp. 697-701, 1973.

[35] DB Wetlaufer and S Ristow, “Acquisition of three-dimensional structure of proteins,” *Annual Review of Biochemistry*, Vol. 42, pp. 135-158, 1973.

[36] A Ikai and C Tanford, "Kinetic evidence for incorrectly folded intermediate states in the refolding of denatured proteins," *Nature*, Vol. 230, pp. 100-102, 1971.

[37] TY Tsong, RL Baldwin and EL Elson, "The Sequential Unfolding of Ribonuclease A: Detection of a Fast Initial Phase in the Kinetics of Unfolding," *Proceedings of the Natural Academy of Sciences*, Vol. 78, pp. 2712-2715, 1971.

[38] OB Ptitsyn, "Sequential mechanism of protein folding," *Doklady Akademii Nauk SSSR*, Vol. 210, pp 1213-1215, 1973.

[39] PS Kim and RL Baldwin, "Intermediates in the folding reactions of small proteins," *Annual Review of Biochemistry*, Vol. 59, pp. 631-660, 1990.

[40] , HJ Dyson and PE Wright, "Peptide conformation and protein folding," *Current Opinion in Structural Biology*, Vol. 3, pp. 60-65, 1993.

[41] DA Dolgikh et al, "Alpha-Lactalbumin: compact state with fluctuating tertiary structure?," *FEBS Letters*, Vol. 136, pp. 311-315, 1981.

[42] KA Dill, S Bromberg, KZ Yue, KM Fiebig, DP Yee, PD Thomas and HS Chan, "Principles of protein folding - A perspective from simple exact models," *Protein Science*, Vol. 4, pp. 561-602, 1985.

[43] OB Ptitsyn, "How molten is the molten globule?," *Nature Structural Biology*, Vol. 3, pp. 488-490, 1996.

[44] VI Abkevich, AM Gutin and EI Shakhnovich, "Specific nucleus as the transition-state for protein-folding - Evidence from the lattice model," *Biochemistry*, Vol. 33, pp. 10026-10036, 1994.

[45] TE Creighton, "Experimental studies of protein folding and unfolding," *Progress in Biophysics and Molecular Biology*, Vol. 33, pp. 231-297, 1978.

[46] PE Leopold and JN Onuchic, "Protein folding funnels - A kinetic approach to the sequence structure relationship," *Proceedings of the Natural Academy of Sciences*, Vol. 89, pp. 8721-8725, 1992.

[47] JD Bryngelson, JN Onuchic, ND Socci and PG Wolynes, "Funnels, pathways, and the energy landscape of protein folding - A systhesis," *Proteins: Structure, Function and Genetics*, Vol. 21, pp. 167-195, 1995.

[48] R Srinivasan and GD Rose, "LINUS: a hierarchic procedure to predict the fold of a protein," *Proteins*, Vol. 22, pp. 81-99, 1995.

Protein Secondary Structures

[49] L Pauling, RB Corey and HR Branson, "The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain," *Proceedings of the Natural Academy of Sciences*, Vol. 37, pp. 205-234, 1951.

[50] L Pauling and RB Corey, "Configurations of polypeptide chains with favored orientations around single bonds: Two new pleated sheets," *Proceedings of the Natural Academy of Sciences*, Vol. 37, pp. 729-740, 1951.

[51] JS Richardson, "The Anatomy and Taxonomy of Protein Structure," *Advances in Protein Chemistry*, Vol. 34, pp. 167-339, 1981.

[52] W Kabsch and C Sander, "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, Vol. 22, pp. 2577-2637, 1983.

[53] DW Mount, *Bioinformatics - Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press, New York, 2001.

[54] M Levitt and J Greer, "Automatic identification of secondary structure in globular proteins," *Journal of Molecular Biology*, Vol. 114, pp. 181-239, 1977.

[55] M Schiffer and AB Edmundson, "Use of helical wheels to represent the structures of proteins and to identify segments with helical potential," *Biophysics Journal* Vol. 7, pp. 121-135, 1967.

[56] C Chothia, "Conformation of twisted beta-pleated sheets in proteins," *Journal of Molecular Biology*, Vol. 75, pp. 295-302, 1973.

[57] CM Venkatachalam, "Stereochemical criteria for polypeptides and proteins. V. Conformation of a system of three linked peptide units," *Biopolymers* Vol. 6, pp.1425-1436, 1968.

[58] PN Lewis, FA Momany, HA Scheraga, "Chain reversals in proteins," *Biochimica et Biophysica Acta*, Vol. 303, pp. 211-229, 1973.

[59] ID Kuntz, "Protein folding," *Journal of the American Chemical Society*, Vol. 94, pp. 4009-4012, 1972.

[60] GD Rose, "Prediction of chain turns in globular proteins on a hydrophobic basis," *Nature*, Vol. 272, pp. 586-590, 1978.

Measures of Performance

[61] BW Matthews, “Comparison of the predicted and observed secondary structure of T4 phage lysozyme,” *Biochimica et Biophysica Acta*, Vol. 405, pp. 442-451, 1975.

[62] B Rost, C Sander and R Schneider, “Redefining the goal of protein secondary structure prediction,” *Journal of Molecular Biology*, Vol. 235, pp. 13-26, 1994.

[63] A Zemla, C Venclovas, K Fidelis and B Rost, “A modified definition of SOV, a segment-based measure for protein secondary structure prediction assignment,” *Proteins*, Vol. 34, pp. 220-223, 1999.

Review

[64] GJ Barton, “Protein secondary structure prediction,” *Current Opinion in Structural Biology*, Vol. 5, pp. 372-376, 1995.

[65] B Rost, “Review: Protein secondary structure prediction continues to rise,” *Journal of Structural Biology*, Vol. 134, pp. 204-218, May 2001.

Early Secondary Structure Prediction Methods

[66] AG Szent-Györgyi and C Cohen, “Role of proline in polypeptide chain configuration of proteins,” *Science*, Vol. 126, p. 697, 1957.

[67] PY Chou and GD Fasman, “Prediction of secondary structure of proteins from their amino acid sequence,” *Advances in Enzymology and Related Areas of Molecular Biology*, Vol. 47, pp. 45-147, 1978.

[68] J Garnier, DJ Osguthorpe and B Robson, “Analysis of the accuracy and implication of simple methods for predicting the secondary structure of globular proteins,” *Journal of Molecular Biology*, Vol. 120, pp. 97-120, 1978.

[69] J Garnier, JF Gibrat and B Robson, “GOR method for predicting protein secondary structure from amino acid sequence,” *Methods in Enzymology*, Vol. 266, pp. 540-553, 1996.

[70] LBM Ellis and R Milius, “Valid an invalid implementations of GOR secondary sructure predictions,” *Computer Applications in the Biosciences*, Vol. 10, pp. 341-348, 1994.

Neural Networks

[71] N Qian and TJ Sejnowski, "Predicting the secondary structure of globular proteins using neural network models," *Journal of Molecular Biology*, Vol. 202, pp. 865-884, 1988.

[72] S Muggleton, RD King and MJ Sternberg, "Protein secondary structure prediction using logic-based machine learning," *Protein Engineering*, Vol. 5, pp. 647-657, 1992.

[73] P Stolorz, A Lapedes and Y Xia, "Predicting protein secondary structure using neural net and statistical methods," *Journal of Molecular Biology*, Vol. 225, pp. 363-377, 1992.

[74] B Rost and C Sander, "Prediction of protein secondary structure at better than 70% accuracy," *Journal of Molecular Biology*, Vol. 232, pp. 584-599, 1993.

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

[76] LH Holley and M Karplus, "Neural networks for protein structure prediction," *Methods in Enzymology*, Vol. 202, pp. 204-224, 1991.

[77] JD Hirst and MJ Sternberg, "Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural networks," *Biochemistry*, Vol. 31, pp. 7211-7218, 1992.

[78] DT Jones, "Protein secondary structure prediction based on position-specific scoring matrices," *Journal of Molecular Biology*, Vol. 292, Issue 2, pp. 195-202, 17 September 1999.

[79] GPS Raghava, "APSSP2: Protein secondary structure prediction using nearest neighbor and neural network approach," *CASP4*, pp. 7576, 2000.

[80] CM Bishop, *Neural Networks for Pattern Recognition*, Oxford Univeristy Press, Oxford, UK, 1995.

[81] G Pollastri, D Przybylski, B Rost and P Baldi, "Improving the Prediction of Protein Secondary Structure in Three and Eight Classes using Recurrent Neural Networks and Profiles," *Proteins* Vol. 47, pp. 228235, 2002.

[82] JA Cuff and GJ Barton, "Application of multiple sequence alignment profiles to improve protein secondary structure prediction," *Proteins*, Vol. 40, pp. 502511, 2000.

[83] J Meiler, M Mueller, A Zeidler and F Schmaeschke, "Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks," *Journal of Molecular Modeling*, Vol. 7, pp. 360369, 2001.

[84] TN Petersen et al, "Prediction of Protein Secondary Structure at 80% Accuracy," *Proteins*, Vol. 41, pp. 1720, 2000.

[85] SF Altschul, W Gish, W Millers, EW Myers and DJ Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, Vol. 215, pp. 403-410, 1990.

[86] SF Altschul, TL Madden, et al, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Research*, Vol. 25, pp. 3389-3402, 1997.

[87] S Henikoff and JG Henikoff, "Amino acid substitution matrices from protein blocks," *Proceedings of the Natural Academy of Sciences*, Vol. 89, pp. 10915-10919, 1992.

Nearest-neighbor Methods

[88] JM Levin, B Robson and J Garnier, "An algorithm for secondary structure determination in proteins based on sequence similarity," *FEBS Letters*, Vol. 205, pp. 303-308, 1986.

[89] S Salzberg and S Cost, "Predicting protein secondary structure with a nearest-neighbor algorithm," *Journal of Molecular Biology*, Vol. 227, pp. 371-374, 1992.

[90] X Zhang, JP Merisov and DL Waltz, "Hybrid system for protein secondary structure prediction," *Journal of Molecular Biology*, Vol. 225, pp. 1049-1063, 1992.

[91] TM Yi and ES Lander, "Protein secondary structure prediction using nearest-neighbor methods," *Journal of Molecular Biology*, Vol. 232, pp. 1117-1129, 1993.

[92] AA Salamov and VV Solovyev, "Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments," *Journal of Molecular Biology*, Vol. 247, pp. 11-15, 1995.

[93] AA Salamov and VV Solovyev, "Protein secondary structure prediction using local alignments," *Journal of Molecular Biology*, Vol. 268, pp. 31-36, 1997.

[94] D Frishman and P Argos, "Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence," *Protein Engineering*, Vol. 9, pp. 133-142, 1996.

[95] D Frishman and P Argos, "Seventy-five percent accuracy in protein secondary structure prediction," *Proteins*, Vol. 27, pp. 329-335, 1997.

Hidden Markov Models

[96] J Martin, JF Gibrat and F Rodolphe, "Analysis of an optimal hidden Markov model for secondary structure prediction," *BMC Structural Biology*, Vol. 6, pp. 25-44, 2006.

[97] CM Stultz, JV White and TF Smith, "Structural analysis based on state-space modelling," *Protein Science*, Vol. 2, pp. 305-314, 1993.

[98] JV White, CM Stultz and TF Smith, "Protein classification by stochastic modeling and optimal filtering of amino-acid sequences," *Mathematical Biosciences*, Vol. 119, pp. 35-75, 1994.

[99] TJ Hubbard and J Park, "Fold recognition and ab initio structure predictions using hidden Markov models and β-strand pair potentials," *Proteins*, Vol. 23, pp. 398-402, 1995.

[100] V Di Francesco, J Garnier and PJ Munson, "Protein topology recognition from secondary structure sequences: application of the hidden Markov models to the alpha class proteins," *Journal of Molecular Biology*, Vol. 267, pp. 446-463, 1997.

[101] K Asai, S Hayamizu, KI Handa, "Prediction of Protein Secondary Structure by the Hidden Markov Model," *Computer Applications in Biosciences*, Vol. 9, pp. 141-146, 1999.

[102] SC Schmidler, JS Liu and DL Brutlag, "Bayesian Segmentation of Protein Secondary Structure," *Journal of Computational Biology*, Vol. 7, pp. 233248, 2000.

[103] C Bystroff, V Thorsson and D Baker, "HMMSTR: a Hidden Markov Model for Local Sequence Structure Correlations in Proteins," *Journal of Molecular Biology* Vol. 301, pp. 173190, 2000.

Support Vector Machines

[104] S Hua and Z Sun, "A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach," *Journal of Molecular Biology*, Vol. 308, pp. 397-407, 2001.

[105] JJ Ward, LJ McGuffin, BF Buxton and DT Jones, "Secondary structure prediction with support vector machines," *Bioinformatics*, Vol. 19, Issue 13, pp. 1650-1655, 2003.

[106] H Kim and H Park, "Protein secondary structure based on an improved support vector machines approach," *Protein Engineering*, Vol. 16, pp. 553560, 2003.

[107] MN Nguyen and JC Rajapakse, "Two-stage support vector machines for protein secondary structure prediction," *Neural, Parallel and Scientific Computations*, Vol. 11, pp. 1-18, 2003.

[108] MN Nguyen and JC Rajapakse, "Multi-Class Support Vector Machines for Protein Secondary Structure Prediction," *Genome Informatics*, Vol. 14, pp. 218227, 2003.

[109] J Guo, H Chen, Z Sun and Y Lin, "A Novel Method for Protein Secondary Structure Prediction using Dual-Layer SVM and Profiles," *Proteins* Vol. 54, pp. 738743, 2004.

New Secondary Structure Prediction Methods

[110] A Figureau, MA Soto and J Tohá, "Secondary Structure of Proteins and Three-dimensional Pattern Recognition," *Journal of Theoretical Biology*, Vol. 201, Issue 2, pp. 103-111, November 1999.

[111] Y Liu, J Carbonell, J Klein-Seetharaman and V Gopalakrishnan, "Comparison of probabilistic methods for protein secondary structure prediction," *Bioinformatics*, Vol. 20, Issue 17, pp. 3099-3107, 2004.

General

[112] RF Doolittle, "Redundancies in protein sequences," *Prediction of Protein Structures and the Principles of Protein Conformation (GD Fasman, ed.)*, pp. 599-623, Plenum Press, New York, 1989.

[113] T Meinnel, Y Mechulam and S Blanquet, "Methionine as translation start signal: a review of the enzymes of the pathway in Escherichia coli," *Biochimie*, Vol. 75, Issue 12, pp. 1061-1075, 1993.

Online Resources

[114] "Protein Data Bank," www.pdb.org. Last accessed on 31 July 2006.

[115] "PHD prediction server," http://cubic.bioc.columbia.edu/predictprotein. Last accessed on 31 July 2006.

[116] "PSIPRED prediction server," http://bioinf.cs.ucl.ac.uk/psipred. Last accessed on 31 July 2006.

[117] "PREDATOR prediction server," http://www.embl-heidelberg.de/cgi/predator_serv.pl. Last accessed on 31 July 2006.

[118] "NNSSP prediction server," http://dot.imgen.bcm.tmc.edu:9331/pssprediction /pssp.html. Last accessed on 31 July 2006.

[119] "Folding@Home," http://folding.stanford.edu/. Last accessed on 31 July 2006.

[120] "Predictor@Home," http://predictor.scripps.edu/. Last accessed on 31 July 2006.

[121] "Rosetta@Home," http://boinc.bakerlab.org/rosetta/. Last accessed on 31 July 2006.

[122] "BlueGene Supercomputer," http://www.research.ibm.com/bluegen. Last accessed on 31 July 2006.

[123] "Amino Acid Frequency", http://www.tiem.utk.edu/ gross/bioed/webmodules /aminoacid.htm. Last accessed on 6 February 2007.

Appendix A

LIST OF PROTEINS

Table A.1 lists the proteins that were used in the training set for the results obtained in chapter 5.

Table A.1: Proteins in the Training Set

No.	No.	No.	No.	No.	No.	No.	No.
119l00	1a0aA0	1a0b00	1a34A0	1aab00	1aaf00	1ab300	1aboA0
1abrA0	1abv00	1abz00	1ac000	1ac500	1aca00	1acf00	1acp00
1ad0A0	1ad0B0	1ad200	1ad3A0	1ad9H0	1ad9L0	1adeA0	1adjA0
1adn00	1adoA0	1adr00	1ads00	1adwA0	1adx00	1ae6H0	1ae700
1aeiA0	1aep00	1aew00	1af700	1af800	1afi00	1afoA0	1afp00
1afvH0	1ag200	1ag8A0	1ag9A0	1agdA0	1agg00	1agi00	1agjA0
1agnA0	1agrE0	1agt00	1agx00	1ah600	1ah700	1ah900	1ahdP0
1ahl00	1aho00	1ahpA0	1ahq00	1ahsA0	1ahtL0	1ai1H0	1aie00
1aihA0	1aijL0	1aijM0	1aikC0	1aikN0	1aim00	1aipC0	1air00
1aisB0	1ajj00	1ajsA0	1ajyA0	1ak000	1ak200	1ak4C0	1ak600
1akz00	1al010	1al0B0	1al300	1ala00	1alo00	1alvA0	1aly00
1am300	1amb00	1amf00	1amk00	1amm00	1amp00	1amw00	1amy00
1an2A0	1an4A0	1an9A0	1ang00	1ann00	1ans00	1anu00	1anwA0
1ao7D0	1aocA0	1aoeA0	1aogA0	1aohB0	1aokB0	1aoo00	1aorA0
1aotF0	1aoy00	1aozA0	1ap6A0	1ap800	1apa00	1apf00	1apq00
1aps00	1apxA0	1apyB0	1aq0A0	1aq5A0	1aq6A0	1aqb00	1aqdA0

Continued on next page. . .

Table A.1 – Continued

No.	No.	No.	No.	No.	No.	No.	No.
1aqdB0	1aqkH0	1aqt00	1ar1A0	1ar1B0	1ar1C0	1ar1D0	1arb00
1ard00	1ark00	1arn00	1ars00	1aru00	1as4A0	1as8A0	1ash00
1ass00	1atiA0	1atlA0	1atu00	1aty00	1au7A0	1auiA0	1auiB0
1aun00	1autC0	1auuA0	1auwA0	1auyA0	1avk00	1avmA0	1avoA0
1avoB0	1avpA0	1avqA0	1avsA0	1avyA0	1aw2A0	1awcA0	1awcB0
1awd00	1awe00	1awj00	1axh00	1axj00	1axsH0	1axsL0	1ayaA0
1ayj00	1aym10	1aym20	1azcA0	1azsA0	1azsC0	1azvA0	1azzA0
1b5m00	1babA0	1babB0	1bafH0	1bak00	1bal00	1bba00	1bbjL0
1bbpA0	1bbt10	1bbt20	1bbt30	1bbt40	1bcpC0	1bcpD0	1bcpF0
1bdo00	1bds00	1bec00	1beo00	1bet00	1bfd00	1bfg00	1bfi00
1bfmA0	1bfs00	1bftA0	1bgf00	1bgk00	1bgp00	1bhgA0	1bhp00
1bi6H0	1bif00	1binA0	1ble00	1blf00	1blj00	1blu00	1bme00
1bmfG0	1bmg00	1bmtA0	1bmv10	1bmv20	1bnb00	1bndB0	—
1bomA0	1bor00	1bovA0	1bp100	1bpyA0	1bquB0	1breA0	1brnL0
1bryY0	1bsrA0	1btl00	1btmA0	1btn00	1btq00	1bts00	1bucA0
1bunA0	1burS0	1bv100	1bvd00	1bvp10	1bw300	1c2rA0	1cauA0
1cauB0	1cb100	1cb2A0	1cbg00	1cbh00	1cbn00	1cbs00	1cc500
1ccd00	1cdg00	1cdkI0	1cdlG0	1cdq00	1cdtA0	1cdy00	1ceaA0
1cei00	1cem00	1cewI0	1cex00	1cfaA0	1cfb00	1cfe00	1cfh00
1cfr00	1cfvH0	1cfvL0	1cfyA0	1cghA0	1cgmE0	1cgt00	1chc00
1chkA0	1chl00	1cid00	1cii00	1ciu00	1ciy00	1ckaA0	1cksA0
1clc00	1cleA0	1clf00	1clh00	1cll00	1cloL0	1clpA0	1clxA0
1clzH0	1cmr00	1cod00	1coi00	1colA0	1coo00	1cosA0	1cov20
1cov30	1cpcA0	1cpo00	1cpq00	1cpy00	1crb00	1cre00	1crkA0
1cry00	1cseI0	1csh00	1csn00	1csp00	1csyA0	1ctaA0	1ctf00
1ctn00	1cto00	1ctt00	1cwpA0	1cxc00	1cydA0	1cynA0	1cyo00
1cyx00	1d66A0	1daaA0	1dad00	1danH0	1dapA0	1dbbH0	1dcoA0
1dctA0	1ddf00	1deaA0	1-Dec-00	1def00	1dehA0	1dem00	1dfbH0
1dfnA0	1dhmA0	1dhpA0	1difA0	1dipA0	1div00	1djxA0	1dkzA0
1dlc00	1dmb00	1dmc00	1dme00	1dmr00	1dnpA0	1dokA0	1dorA0
1dpe00	1dpgA0	1dpo00	1dro00	1drs00	1drw00	1dtc00	1dubA0
1dupA0	1dutA0	1dvfC0	1dxgA0	1dxy00	1dynA0	1dyr00	1eaf00
1eal00	1eapB0	1ebdA0	1eca00	1ecfA0	1eciA0	1eciB0	1ecmA0
1ecrA0	1ede00	1edg00	1edhA0	1edi00	1edmB0	1edn00	1edt00
1efnB0	1eft00	1efuB0	1efvA0	1efvB0	1eg1A0	1ego00	1ehs00

Continued on next page...

Table A.1 – Continued

No.	No.	No.	No.	No.	No.	No.	No.
1eit00	1elg00	1elpA0	1elt00	1emn00	1emy00	1enh00	1enp00
1eny00	1epmE0	1eps00	1erd00	1eriA0	1erk00	1erp00	1erv00
1esc00	1esl00	1etfB0	1etpA0	1eur00	1exg00	1exp00	1extA0
1ezm00	1f3z00	1faiH0	1fas00	1fbaA0	1fbiH0	1fbr00	1fcdA0
1fdhG0	1fdx00	1fgjA0	1fgjA2	1fgnH0	1fgp00	1fgvL0	1figH0
1fipA0	1fjlA0	1fjmA0	1fkf00	1fleI0	1fliA0	1flp00	1fmb00
1fmcA0	1fmd10	1fmd30	——	1fna00	1fonA0	1fosF0	1fptH0
1frd00	1fre00	1froA0	1frrA0	1frsA0	1frvA0	1frvB0	1fsd00
1ft1A0	1ft1B0	1ftn00	1ftpA0	1ftt00	1fua00	1fujA0	1furA0
1fvcB0	1fvkA0	1fvl00	1fwcB0	1fwp00	1fxd00	1fxiA0	1fxrA0
1fyc00	1fzbA0	1gadO0	1gafH0	1gafL0	1gai00	1gal00	1ganA0
1gbqA0	1gcb00	1gcmA0	1gcn00	1gd1O0	1gdhA0	1gecE0	1gen00
1gesA0	1gfc00	1ggaO0	1ggiH0	1ggiL0	1ghc00	1ghfH0	1ghj00
1gia00	1gifA0	1gigH0	1gks00	1gky00	1gln00	1glqA0	1gnd00
1gnhA0	1gnwA0	1gof00	1gotB0	1gotG0	1gp2G0	1gpb00	1gpc00
1gpl00	1gpmA0	1gpoH0	1gps00	1gpt00	1gsa00	1gseA0	1gsuA0
1gta00	1gtqA0	1guaA0	1guaB0	1gur00	1gvp00	1gypA0	1hae00
1havA0	1hbg00	1hbhA0	1hcc00	1hcd00	1hcgB0	1hcnA0	1hcrA0
1hcv00	1hcz00	1hdaA0	1hdaB0	1hdcA0	1hdgO0	1hdp00	1hdsB0
1hev00	1hfc00	1hfi00	1hfs00	1hfyA0	1hiaI0	1hilA0	1hip00
1hiwA0	1hjrA0	1hks00	1hleA0	1hlm00	1hloA0	1hme00	1hml00
1hmpA0	1hmt00	1hnf00	1hnr00	1hocA0	1hoe00	1hp800	1hpgA0
1hplA0	1hpm00	1hpt00	1hqi00	1hrc00	1hrdA0	1hrjA0	1hrm00
1hroA0	1hrtI0	1hryA0	1hsbA0	1hsbB0	1hslA0	1hsq00	1htiA0
1htn00	1htrB0	1hucA0	1hueA0	1huiB0	1hulA0	1humA0	1hup00
1hurA0	1huw00	1hxn00	1hymA0	1hymB0	1hyxH0	1iab00	1iag00
1iaiI0	1iaiM0	1iba00	1ibeA0	1ibeB0	1ibgH0	1ibgL0	1ica00
1iceA0	1iceB0	1idaA0	1idk00	1idsA0	1idy00	1ieaA0	1ieaB0
1if1A0	1ifc00	1ife00	1ifi00	1ift00	1igcL0	1igd00	1igfH0
1igl00	1igmH0	1igmL0	1igtB0	1ihfA0	1ihfB0	1ihvA0	1iibA0
1il600	1iml00	1indH0	1inp00	1ioaA0	1iob00	1iow00	1iphA0
1ipsA0	1irsA0	1iscA0	1iskA0	1isuA0	1itf00	1ithA0	1iuz00
1iva00	1ivd00	1ivyA0	1ixh00	1jacA0	1jafA0	1jbc00	1jcv00
1jdc00	1jdw00	1jer00	1jetA0	1jhgA0	1jhlL0	1jli00	1jlyA0
1joi00	1jrhI0	1jsg00	1jsuC0	1jswA0	1jtb00	1jug00	1-Jul-00

Continued on next page. . .

Table A.1 – Continued

No.	No.	No.	No.	No.	No.	No.	No.
1junA0	1kal00	1kao00	1kapP0	1kaz00	1kbaA0	1kbcA0	1kdu00
1kevA0	1kit00	1klo00	1kmmA0	1knb00	1koa00	1kptA0	1krn00
1krs00	1ksr00	1kst00	1ktx00	1kuh00	1kvdA0	1kveA0	1kvoA0
1kxu00	1kzuA0	1kzuB0	1lam00	1latA0	1lba00	1lbd00	1lbeA0
1lbu00	1lcl00	1lct00	1ldg00	1ldl00	1ldnA0	1ldr00	1lea00
1lefA0	1lehA0	1lenB0	1lfaA0	1lghA0	1lghB0	1lht00	1liaA0
1liaB0	1lid00	1lilA0	1lis00	1lkkA0	1lldA0	1llp00	1lmb30
1lmkA0	1lmq00	1lmwB0	1loeB0	1loi00	1lopA0	1lpbB0	1lpfA0
1lpp00	1lpt00	1lqh00	1lre00	1lrv00	1lsi00	1lt5D0	1lte00
1ltsA0	1ltsC0	1lucA0	1lve00	1lvl00	1lxa00	1lxdA0	1lybB0
1lyp00	1lzr00	1maj00	1mamH0	1mat00	1maz00	1mba00	1mbe00
1mbs00	1mcpH0	1mctA0	1mctI0	1mdaH0	1mdl00	1mdyA0	1mea00
1meeA0	1mek00	1melA0	1memA0	1meyC0	1mgsA0	1mh100	1mhcA0
1mhlA0	1mhyB0	1mhyD0	1mimH0	1mimL0	1mioA0	1mioB0	1mjc00
1mkaA0	1mla00	1mlbB0	1mldA0	1mmc00	1mml00	1mn100	1mnmA0
1mnmC0	1mntA0	1mof00	1molA0	1mpp00	1mrg00	1mrj00	1mrk00
1msc00	1msi00	1msk00	1mspA0	1mtx00	1mtyB0	1mtyG0	1mugA0
1mup00	1mvi00	1mvj00	1mwe00	1mzm00	1nah00	1nal10	1nar00
1nawA0	1nbaA0	1nbvH0	1ncbH0	1ncbL0	1nciA0	1ncs00	1nct00
1ncvA0	1nea00	1nfa00	1nfdA0	1nfdE0	1nfdF0	1nfp00	1ngr00
1nhkL0	1nhp00	1nif00	1nin00	1nipA0	1nirA0	1nkl00	1nloC0
1nmbH0	1nnc00	1nnt00	1noa00	1nor00	1novA0	1novD0	1nox00
1noyA0	1np400	1npc00	1npk00	1npoA0	1nqbA0	1nra00	1nscA0
1nsgB0	1nsj00	1nsyA0	1ntn00	1ntr00	1ntx00	1nueA0	1nxb00
1nzyA0	1obpA0	1obr00	1obwA0	1obwB0	1oef00	1oeg00	1ofgA0
1ofv00	1ojt00	1omn00	1onrA0	1opbA0	1opc00	1opgH0	1opr00
1osa00	1ospH0	1ospL0	1ospO0	1otfA0	1otgA0	1ounA0	1outA0
1ovwA0	1oxa00	1oyc00	1p3800	1pamA0	1pax00	1paz00	1pbk00
1pbn00	1pbwA0	1pce00	1pcfA0	1pch00	1pcs00	1pdc00	1pdo00
1pdr00	1pdz00	1pea00	1peh00	1pei00	1pex00	1pfc00	1pfiA0
1pfkA0	1pft00	1pfxC0	1pgb00	1pgs00	1pgtA0	1phb00	1phk00
1phnA0	1pho00	1php00	1phr00	1pht00	1pidA0	1pidB0	1pk400
1pkm00	1pkp00	1pla00	1plc00	1plfA0	1plgH0	1plp00	1plq00
1pls00	1pmaA0	1pmaB0	1pmc00	1pmlA0	1pmpA0	1pmy00	1pnbB0
1pnh00	1pnkA0	1pnkB0	1poa00	1poc00	1poiA0	1poiB0	1ponB0

Continued on next page. . .

Table A.1 – Continued

No.	No.	No.	No.	No.	No.	No.	No.
1pot00	1poxA0	1pp2R0	1ppa00	1ppeI0	1ppfE0	1ppo00	1pprM0
1ppt00	1prn00	1pru00	1ps200	1psdA0	1pse00	1pskL0	1psm00
1psoE0	1ptf00	1pth00	1ptq00	1pty00	1puc00	1pud00	1pueE0
1put00	1pvc20	1pvc30	1pyaA0	1pyc00	1pyiA0	1pysA0	1pysB0
1pytA0	1pytD0	1qapA0	1qba00	1qdp00	1qli00	1qnf00	1qoaA0
1qorA0	1que00	1qyp00	1r0910	1r0920	1r1a10	1r1a20	1r1a30
1r6900	1ra900	1raiA0	1raiB0	1rblA0	1rblM0	1rcb00	1rcf00
1rcy00	1rdg00	1rdo10	1rds00	1reqA0	1reqB0	1res00	1rfs00
1rhi20	1rhi30	1rhpA0	1rie00	1ril00	1ris00	1rlw00	1rmd00
1rmfH0	1rmvA0	1rodA0	1roe00	1rom00	1roo00	1rot00	1rpa00
1rpb00	1rpmA0	1rpo00	1rro00	1rsy00	1rvaA0	1sacA0	1sap00
1sat00	1sba00	1sbp00	1schA0	1scmA0	1sco00	1sctA0	1sctB0
1scuB0	1scy00	1se400	1semA0	1sesA0	1sfe00	1sgpE0	1sgpI0
1sh100	1shaA0	1shfA0	1sis00	1sju00	1sltA0	1sly00	1smd00
1smeA0	1smpI0	1smrA0	1smtA0	1snb00	1sol00	1sp100	1sp200
1spf00	1spgA0	1sphA0	1spiA0	1sqc00	1srdA0	1sro00	1srrA0
1srsA0	1stfI0	1stmA0	1stu00	1sup00	1sva10	1svb00	1svn00
1svpA0	1svq00	1sxm00	1tafA0	1tap00	1tbd00	1tbrR0	1tc3C0
1tca00	1tdtA0	1tehA0	1ten00	1ter00	1tf3A0	1tf4A0	1tfe00
1tfi00	1tfpA0	1tfs00	1tfxC0	1tgsI0	1tgxA0	1theA0	1thm00
1thv00	1thx00	1tib00	1tih00	1tiiC0	1tiiD0	1tis00	1tiv00
1tlfA0	1tme10	1tme20	1tmy00	1tnrA0	1tns00	1tocR0	1tof00
1tph10	1trkA0	1trlA0	1trnA0	1try00	1tsg00	1tsk00	1tsy00
1ttbA0	1tuc00	1tud00	1tul00	1tupA0	1tvdA0	1tvs00	1tvxA0
1txa00	1txm00	1tys00	1tzeE0	1uae00	1ubdC0	1ubi00	1ubsB0
1uby00	1ucbH0	1ucbL0	1uch00	1ucyH0	1ucyJ0	1udc00	1udg00
1udh00	1udiI0	1ukrA0	1ukz00	1ula00	1unkA0	1urnA0	1utg00
1uxc00	1uxy00	1vapA0	1vcaA0	1vdc00	1vdfA0	1vdrA0	1vfaA0
1vfaB0	1vgeH0	1vgeL0	1vhh00	1vhiA0	1vhp00	1vhrA0	1vid00
1vii00	1vin00	1vip00	1vktA0	1vls00	1vlxA0	1vnc00	1vnd00
1vokA0	1volA0	1vpi00	1vpsA0	1vpu00	1vsd00	1vsgA0	1vtmP0
1vtx00	1vvc00	1wab00	1wad00	1waj00	1wapA0	1wba00	1wdcA0
1wdcB0	1wdcC0	1wer00	1wfbA0	1wgjA0	1whi00	1who00	1whtA0
1wtuA0	1xaa00	1xbl00	1xbrA0	1xdtR0	1xgsA0	1xib00	1xikA0
1ximA0	1xlaA0	1xnb00	1xsm00	1xtcA0	1xtcC0	1xxbA0	1xyn00

Continued on next page. . .

Table A.1 – Continued

No.	No.	No.	No.	No.	No.	No.	No.
1xyzA0	1yaiA0	1yasA0	1yat00	1ycqA0	1ycrA0	1ycsB0	1ydvA0
1yecL0	1yedH0	1yge00	1ykfA0	1yna00	1ypcI0	1yprA0	1yrnA0
1yrnB0	1ytbA0	1ytfB0	1ytfC0	1ytiA0	1ytw00	1yua00	1yub00
1yuf00	1yuhH0	1yuiA0	1yveI0	1zaq00	1zda00	1zec00	1zfd00
1zfo00	1zia00	1zin00	1ztn00	1zxq00	256bA0	2aaa00	2aaiB0
2aak00	2abk00	2abxA0	2acg00	2act00	2afgA0	2ak3A0	2-Apr-00
2asr00	2atcB0	2baa00	2bbkH0	2bbkL0	2bbmB0	2bltA0	2bnh00
2bopA0	2bpa10	2bpa20	2bpa30	2btfA0	2cba00	2ccyA0	2cdx00
2cgpC0	2cgrH0	2chbD0	2chr00	2chsA0	2cnd00	2cro00	2cstA0
2ctx00	2cy300	2cyp00	2dgcA0	2dldA0	2drpA0	2dtr00	2ech00
2eql00	2eti00	2ezdA0	2ezh00	2fb4L0	2fbjH0	2fx200	2fxb00
2gdm00	2gf100	2gliA0	2gmfA0	2gsq00	2gsrA0	2h1pH0	2hipA0
2hmqA0	2hpdA0	2hppP0	2hpqP0	2hrpH0	2hrpL0	2hvm00	2ifo00
2ilk00	2imn00	2jxrA0	2ldx00	2leu00	2lhb00	2liv00	2ltnA0
2masA0	2mcm00	2mev10	2mev20	2mev30	2mhr00	2mhu00	2mrb00
2mtaC0	2nacA0	2nllA0	2ohxA0	2omf00	2pelA0	2pgd00	2pghA0
2pghB0	2phy00	2pia00	2pii00	2pkaA0	2pkaB0	2plc00	2pldA0
2plt00	2polA0	2por00	2prd00	2pspA0	2ptd00	2ptl00	2rbiA0
2rhe00	2rmcA0	2rn200	2sas00	2scpA0	2sfa00	2sga00	2sicI0
2sil00	2sn300	2sns00	2spcA0	2sttA0	2stv00	2tbs00	2tbvA0
2tgi00	2tmdA0	2tmvP0	2trxA0	2tysB0	2u1a00	2ucz00	2vaaA0
2vik00	2vpfB0	2wbc00	351c00	3adk00	3btoA0	3c2c00	3chy00
3cla00	3cyr00	3dfr00	3gar00	3gpdR0	3grs00	3il800	3ladA0
3ldh00	3lip00	3lzt00	3mddA0	3ovo00	3p2pA0	3pchA0	3pfk00
3pmgA0	3pte00	3rnt00	3rp2A0	3rubS0	3sdhA0	3sdpA0	3sicI0
3tgl00	3tss00	4aahA0	4cpv00	4fxc00	4gatA0	4gpd10	4kbpA0
4mdhA0	4pgaA0	4pgmA0	4rhn00	4sbvA0	4sgbI0	5cytR0	5hpgA0
5icb00	5ldh00	5nul00	5p2100	5pal00	5pti00	5znf00	6cel00
6fabH0	6fd100	6gsvA0	6rlxB0	6rxn00	6taa00	7aatA0	7ahlA0
7pcy00	7rsa00	8abp00	8acn00	8dfr00	8fabA0	8i1b00	8rucI0
8rucK0	8rxnA0	8tlnE0	9ldtA0	9pcy00	9rnt00		

Table A.2 lists the proteins that were used in the testing set for the results obtained in chapter 5.

Table A.2: Proteins in the Testing Set

No.	No.	No.	No.	No.	No.	No.	No.
1aa200	1aa7A0	1abrB0	1ac6A0	1aci00	1acw00	1ae6L0	1af6A0
1afrA0	1afsA0	1afwB0	1aijH0	1ail00	1aisA0	1aj300	1aj8A0
1aje00	1ajz00	1ak100	1akeA0	1ako00	1akp00	1aky00	1alkA0
1allA0	1allB0	1aonO0	1aoqA0	1ap2B0	1apj00	1apo00	1apyA0
1aqkL0	1as4B0	1aszA0	1ata00	1atx00	1atzA0	1aua00	1auoA0
1avdA0	1aw000	1axn00	1aym30	1bbhA0	1bbi00	1bbrL0	1bcpB0
1bebA0	1bed00	1bkf00	1bmfA0	1bmfD0	1bmp00	1bndA0	1bno00
1broA0	1bunB0	1caa00	1cbiA0	1cby00	1cch00	1ccr00	1cd1A0
1cd800	1cdcB0	1cdkA0	1cdoA0	1cdwA0	1cerO0	1cfg00	1cfpA0
1cg2A0	1chd00	1chmA0	1cis00	1ckmA0	1cld00	1cmbA0	1cnpA0
1cnv00	1cot00	1cov10	1cpcB0	1cpn00	1cseE0	1ctj00	1cx2A0
1cyg00	1cyj00	1cyu00	1dhkA0	1dhr00	1dhx00	1dja00	1dktA0
1doi00	1dot00	1dox00	1drf00	1dsuA0	1dtk00	1dvh00	1eapA0
1ebdC0	1ebpA0	1eceA0	1ecpA0	1egdA0	1egf00	1ethA0	1fca00
1fcdC0	1fct00	1fecA0	1fgnL0	1fmd20	1fnc00	1forH0	1fosE0
1fsb00	1ftz00	1fvcA0	1fvpA0	1fwcA0	1fzbB0	1gab00	1gatA0
1gbg00	1gca00	1gclA0	1gdoA0	1gff10	1gff20	1gggA0	1ghsA0
1gluA0	1gowA0	1gpoL0	1gpr00	1gtmA0	1guqA0	1gzi00	1hbhB0
1hcb00	1hcnB0	1hcqA0	1hdj00	1hdsA0	1hfx00	1hgeA0	1hlb00
1hlcA0	1hleB0	1hlpA0	1hma00	1hmy00	1hna00	1hph00	1hpi00
1hra00	1hstA0	1htmB0	1htp00	1htrP0	1hucB0	1hyp00	1hyxL0
1iaiH0	1ido00	1igjB0	1igtA0	1ikfH0	1ilr10	1imp00	1irf00
1iro00	1itbB0	1iyu00	1jfo00	1jhlH0	1jmcA0	1jpc00	1jud00
1jvr00	1jxpA0	1kelH0	1kid00	1knyA0	1kpf00	1ksaA0	1kte00
1kveB0	1kxiA0	1lab00	1lccA0	1lgyA0	1lit00	1lki00	1lktA0
1lpbA0	1lucB0	1lybA0	1mai00	1mbg00	1mblA0	1mdaL0	1mhcB0
1mhlC0	1mhyG0	1mil00	1mmq00	1mpaH0	1mpgA0	1mrp00	1mtyD0
1mut00	1mvmA0	1myjA0	1mylA0	1myn00	1nbcA0	1ndh00	1neq00
1nfdB0	1nldH0	1nls00	1nsa00	1nsqA0	1nwpA0	1oatA0	1ocp00
1octC0	1oneA0	1orc00	1ordA0	1outB0	1ovaA0	1paa00	1pafA0
1pal00	1pbe00	1pca00	1pdnC0	1pfsA0	1pi200	1pii00	1pmi00

Continued on next page. . .

Table A.2 – Continued

No.	No.	No.	No.	No.	No.	No.	No.
1pnbA0	1pne00	1pov10	1pov30	1ppn00	1prr00	1pscA0	1psj00
1psv00	1pvaA0	1pvc10	1pyaB0	1qpg00	1qrdA0	1r0930	1r0940
1rcd00	1regX0	1reiA0	1rfbA0	1rgeA0	1rgs00	1rhi10	1rip00
1rkd00	1rlaA0	1rmg00	1ron00	1rtp10	1ryt00	1scuA0	1sdf00
1seiA0	1sftA0	1sgt00	1shcA0	1shg00	1shp00	1skyE0	1skz00
1smnA0	1smvA0	1spbP0	1sra00	1srb00	1sso00	1std00	1sxcA0
1sxl00	1tabI0	1tadA0	1tcrA0	1tgj00	1thjA0	1tif00	1tig00
1tit00	1tlk00	1tme40	1tml00	1tnfA0	1tpfA0	1tpg00	1tx4A0
1ulo00	1vcc00	1vcpA0	1vie00	1vig00	1vmoA0	1vtp00	1vwlB0
1whtB0	1wit00	1wjdB0	1wtlA0	1xsoA0	1xvaA0	1ybvA0	1ycc00
1yecH0	1ytc00	1zer00	1zncA0	1znf00	1zto00	1zymA0	2acy00
2arcA0	2atjA0	2ayh00	2bb200	2bbvA0	2cmd00	2ctc00	2dkb00
2dri00	2ebn00	2end00	2erl00	2ezk00	2fb4H0	2fbjL0	2fcr00
2fha00	2gsaA0	2hlcA0	2knt00	2lbp00	2mltA0	2msbA0	2myr00
2ncm00	2nllB0	2pkc00	2plh00	2pna00	2pta00	2pth00	2ran00
2sak00	2uce00	3minB0	3mra00	4aahB0	4hb100	4mt200	6fabL0
6ldh00	7catA0	7timA0					