Improving Protein Structure Prediction through Data Purification

Fan Zhang (Mr.)
Submit in partial fulfilment for the Degree of Doctor of Philosophy From the University of Surrey

UniS

Department of Computing;
School of Electronics and Physical Sciences;
Division of Chemistry;
School of Biomedical and Molecular Science;
University of Surrey
Guildford, Surrey GU2 5XH, UK

March 2007

© Fan Zhang (Mr.) 2007
Abstract

Achieving the ability to predict the three dimensional structure of a protein solely from its sequence is widely acknowledged to be essential in tackling most genetic diseases. Deoxyribonucleic acids (DNA) constitute chemical blueprints for protein construction, but once these blueprints are damaged, the correct structure and function of proteins, which are constructed according to these blueprints, can no longer be guaranteed. Most genetic diseases are caused by irregularly constructed proteins, and structural abnormalities of proteins may lead to irregular biological behaviour within the human body, such as uncontrolled cell division. Genetic disease is one of the major causes of human death, from a very early age, and for centuries mankind has tackled these diseases using a variety of approaches. Within the research world this has become a wide spread and international wide activity, and the advancements in communication techniques and computational hardware make this research more exciting and effective than it has ever been.

It is well known that proteins are the foundation of human life, and their biological functions catalyze and control almost all the chemical reactions within the human body. These biological functions rely completely on the shape of the proteins in three-dimensional space and therefore a knowledge of protein structure is required in advance to cure diseases caused by protein malfunctions.

The problem of protein folding remains critical and as yet unsolved in the literature. Owing to the close correlation between a protein’s structure and its biological functions, understanding three dimensional protein structures becomes the front line of the war against protein disorder.

Almost all proteins automatically fold into their unique shapes spontaneously when they are put into a moist environment. The folding process is too fast and is impossible to be monitored at the moment. Conversely, scientists determine the sequences and structures of proteins sequentially, and explore the correlation between the sequences and the structures. These discovered correlations lead to the possibility of conducting a successful
prediction of protein structure from sequence alone. The discovery of a successful methodology of protein structure prediction not only enormously cuts down the expense of determining protein structure but also dramatically increases the efficiency. In general, if successfully implemented, it will offer drastic improvements in the state of the art.

There were outstanding successes towards understanding protein structures in the past few decades. X-Ray crystallography is one of the favourite techniques used in determining protein structure (Abola et al. 2000; Smyth & Martin 2000; Yaffe 2005), and nuclear magnetic resonance (NMR) spectroscopy is another widely used approach in screening structure of proteins (Hawkes et al. 1980; Markley 1989; Van, V, Bellon, & Lanens 1988; Wuthrich 1989). Using these facilities to determine protein structure used to be both time consuming and resource intensive, and sometimes can go wrong if they were not implemented correctly (Nabuurs et al. 2006). The invention of high throughput screening technologies granted the users the ability to screen over hundreds of samples at the same time and therefore enormously reduced the time compared to the time cost of sequential screening (Capelle, Gurny, & Arvinte 2006), but unfortunately the process is still expensive. An alternative approach to protein structure determination uses computational algorithms and methodologies to predict the structure of a protein, based on applying machine learning schemes to a set of protein samples, whose structures are already resolved.

The ultimate target is to achieve the ability to predict protein structure based solely on its sequence. In order to fulfill this eventual purpose, the prediction of protein secondary structure is the first strategic step; a good secondary structure prediction methodology provides a good starting point toward tertiary structure prediction. Christian Anfinsen’s seminal discovery, in 1973 of the denaturing process of ribonuclease (Anfinsen 1973) laid the foundation of modern in silico research into protein structural determination from sequence information alone. In the past thirty years, various choices of machine learning schemes have been applied to the applications of protein structure prediction, however the reported prediction accuracy remains less than 80%.

In this thesis, the author pursues the target of improving accuracy of protein structural prediction through the procedure of data purification. A Protein Attributes Microtuning System (PAMS) is developed to prepare a variety of new datasets as and when required.
Furthermore, a Protein Structural Accuracy Reckoner (PSAR) framework is used to recommend procedures that might lead to high prediction accuracy. By using the PSAR, it is shown that using a refined dataset generated by the PAMS, and implementing an appropriate window mechanism considerably improves the accuracy of protein structure prediction by 12%, giving a best accuracy of 90.97%. On average, almost all classifiers that are applied in the experiments result in accuracy increases of 10%-15%. A list of classifiers is categorized according to their prediction performances and classification efficiencies. A few refined datasets are proposed as benchmark datasets.

Apart from the aforementioned achievements, examination of a total of 3,135,393 predictions tasks, which carried out by the PSAR framework, yielded 139 ‘best’ and 73 ‘worst’ combinations of amino acid features descriptors. In this analysis, the ‘best’ prediction gave 82.34%, and the ‘worst’ prediction gave 73.65%.

To achieve a greater computational capacity the PSAR infrastructure is hosted on the Condor (Thain, Tannenbaum, & Livny 2005) platform in the Department of Computing, University of Surrey (Grid Computing project, Department of Computing, University of Surrey).

Key words: Protein Secondary Structure Prediction; Data Purification; High Throughput Computing; Machine Learning.
Acknowledgments

First of all I want to dedicate my thankful heart to my parents, who never lose their faith in me. To be honest, doing research in England is never as easy as I imagined before I came to here. During the past four years, whenever I had troubles they comforted me; whenever I was lonely they showed their generous love to me, encouraged me to place footsteps firmly toward the target of finishing this research.

Professor David Povey (Professor of Computer Aided Chemistry) is my hero. He helped me in obtaining an opportunity to work in this fascinating research area. He provided me easy access to equipments and placed me in Cecil Davies Laboratory, Division of Chemistry. During my time been there, I acquired the necessary knowledge from a chemical viewpoint to form this thesis. Professor Povey, as my secondary supervisor, inspired me to develop the infrastructure and kept encouraging me to delve deeper in to this research area. Obviously the saying “he is the hero of the Chemistry Division” is very true.

I also offer my thanks to Professor Paul Krause (Professor of Software Engineering) for offering me an extraordinary chance after I had difficulties. As my principal supervisor, his unbounded knowledge in software engineering inspired me in various aspects to form the PAMS/PSAR framework. His kindness and generosity is so important for the exciting and relaxing work environment in this project and brought lots of fun.

Thanks to Dr Ian Hamerton for providing excellent proof reading of the composition of this thesis. The time that I spent in the roots café in the past three years with Dr Brendan Howlin, Dr Ian Hamerton, and Dr Gabriel Cavalli-Petraglia was very pleasant. The friendly environment that we created was delightful and kept the project going well.

Mr Chris Bradshaw, as the IT support manager in School of Biomedical and Molecular Sciences in University of Surrey provided lots of facilities for this project. Odin.chem.surrey.ac.uk, as the major server for this project was fully supported by Chris.
I would like also thank my friends Victor Zhou (Electronic Engineering@Surrey), Jin Wu (Computing@Leeds), Paul Tapper (Computing@Bath), and John Elliott for keeping me company while I stayed in Guildford.
Contents

Abstract .........................................................................................................................................................ii

Acknowledgments ......................................................................................................................................v

Contents .....................................................................................................................................................vii

List of Figures ............................................................................................................................................xi

List of Tables ..........................................................................................................................................xiii

List of Terminology ...............................................................................................................................xiv

1 Introduction ...............................................................................................................................................1
  1.1 Medication in Human Civilization .................................................................................................1
  1.2 Proteins and Modern Structural Biology .........................................................................................2
  1.3 Structural Bioinformatics ..................................................................................................................3
  1.4 Statement of Hypothesis ....................................................................................................................4
  1.5 Structure of the Thesis .........................................................................................................................5

2 Protein and Protein Structure .............................................................................................................. 6
  2.1 Protein is Life .................................................................................................................................... 6
  2.2 The Physical Composition of Proteins ............................................................................................7
  2.3 The Process of Protein Folding .......................................................................................................10
  2.4 Protein Composition and Structures ..............................................................................................11
    2.4.1 Amino Acids as Building Blocks of Protein .........................................................................12
    2.4.2 Primary Structure as Sequence Indicator .............................................................................13
    2.4.3 Secondary Structure as Pattern of General 3D Form ............................................................14
    2.4.4 Super-Secondary Structure ....................................................................................................17
    2.4.5 Tertiary Structure ..................................................................................................................17
    2.4.6 Quaternary Structure .............................................................................................................19
  2.5 Essentiality of Protein Structure ....................................................................................................19
  2.6 Conclusions .......................................................................................................................................19

3 Protein Secondary Structure Prediction ........................................................................................21
  3.1 Overview of General Protein Structure Prediction ........................................................................21
  3.2 Historical View of Protein Secondary Structure Prediction .........................................................23
5 Protein Structural Accuracy Reckoner ...........................................................................................65
  5.1 Motivation behind the PSAR Development ...............................................................................65
  5.2 Characteristics of the PSAR Infrastructure ................................................................................66
  5.3 Functional Modules of the PSAR Infrastructure .......................................................................68
    5.3.1 BioData-Oriented Modules ................................................................................................69
      5.3.1.1 The Database ..............................................................................................................69
      5.3.1.2 The PAMS ..................................................................................................................70
      5.3.1.3 The Classifiers ..........................................................................................................72
    5.3.2 Configuration-Oriented Modules .........................................................................................72
      5.3.2.1 Feature Sampling Module .........................................................................................72
      5.3.2.2 Protein Sampling Module .........................................................................................73
      5.3.2.3 Window Sampling Module .......................................................................................74
      5.3.2.4 Combination Sampling Module ...............................................................................75
    5.3.3 Result-Oriented Modules ....................................................................................................75
      5.3.3.1 Database module ........................................................................................................75
      5.3.3.2 Results Capturer Module .........................................................................................77
      5.3.3.3 Results Reviewing Module .......................................................................................77
  5.4 Safety Evaluation of the PAMS/PSAR Framework ...................................................................77
    5.4.1 System Security of the Hosts with the PAMS/PSAR Installed ........................................78
    5.4.2 Reliability of the Procedures Defined by PAMS/PSAR ...................................................80
    5.4.3 Validity of the Biological Dataset .....................................................................................80
  5.5 Performances Evaluation of PAMS/PSAR ...............................................................................80
  5.6 Conclusions .............................................................................................................................81

6 Implementation of the PSAR Framework .....................................................................................82
  6.1 Sampling the Feature Descriptors ...........................................................................................82
  6.2 Determining the Numbers of Applied Feature Descriptors ...................................................86
  6.3 Ranking the 3080 Protein Samples .........................................................................................88
  6.4 Selecting Window Length ......................................................................................................92
  6.5 Selecting Linear or Exponential ............................................................................................102
  6.6 Selecting Combinations of Amino Acid Feature Descriptors ..............................................106
  6.7 Conclusions ............................................................................................................................109

7 Future Work ...............................................................................................................................110
  7.1 Technical Improvements for PSAR .......................................................................................110
  7.2 Scientific Improvements for PSAR .......................................................................................111
  7.3 From Secondary Structure to Tertiary Structure ....................................................................112
Contents

7.4 General Conclusion .................................................................................................................. 113
List of Figures

FIGURE 2-1: DNA – RNA – PROTEIN (© THE NOBEL FOUNDATION).......................................................... 7
FIGURE 2-2: UNIVERSAL GENE CODE (© UTAH UNIVERSITY).............................................................. 7
FIGURE 2-3: COMPARISONS BETWEEN DNA AND SINGLE STRANDED RNA. (GNU LICENSED)............... 9
FIGURE 2-4: STRUCTURE OF AMINO ACID (SHOWING ZWITTERIONIC STRUCTURE)........................... 12
FIGURE 2-5: STRUCTURE OF ALANINE (SHOWING ZWITTERIONIC STRUCTURE)................................ 12
FIGURE 2-6: STRUCTURE OF GLYCINE (SHOWING ZWITTERIONIC STRUCTURE).................................. 12
FIGURE 2-7: A TRIPLEPEPTIDE WITH TWO PEPTIDE BONDS (SHOWING ZWITTERIONIC STRUCTURE)............... 13
FIGURE 2-8: PROTEIN PRIMARY STRUCTURE. (GNU LICENSED)............................................................ 14
FIGURE 2-9: THREE DIMENSIONAL STRUCTURE OF THE MYOGLOBIN PROTEIN. (GNU LICENSED)........ 16
FIGURE 2-10: STICK REPRESENTATION OF OVERALL STRUCTURE OF INSULIN................................. 18
FIGURE 3-1: PROTEIN STRUCTURE PREDICTION FLOWCHART (BY ROB RUSSELL, EMBL)..................... 21
FIGURE 3-2: ARCHITECTURE OF BAYESPROT (CHINNASAMY, SUNG, & MITTAL 2004)......................... 31
FIGURE 3-3: TAN BAYESIAN CLASSIFIER (CHINNASAMY, SUNG, & MITTAL 2004).............................. 31
FIGURE 4-1: STRUCTURE OF THE PAMS.............................................................................................. 37
FIGURE 4-2: THE SAMPLE OF PROTEIN 2:1.B WAS TAKEN FROM THE RS126 DATASET............................ 45
FIGURE 4-3: THE SAMPLE OF PROTEIN 1:1C WAS TAKEN FROM THE CB396 DATASET............................ 45
FIGURE 4-4: ARFF HEADER INFORMATION SECTION, PART ONE. (PROFESSOR DAVID POVEY NOW)........ 48
FIGURE 4-5: ARFF HEADER INFORMATION SECTION, PART TWO....................................................... 48
FIGURE 4-6: ARFF DATA SECTION........................................................................................................ 49
FIGURE 4-7: THE PAMS, DATA SET SELECTION PAGE............................................................................ 50
FIGURE 4-8: THE PAMS, ATTRIBUTES SELECTION I.............................................................................. 51
FIGURE 4-9: THE PAMS, ATTRIBUTES SELECTION, II.......................................................................... 52
FIGURE 4-10: THE PAMS, CONFIGURING THE WINDOW MECHANISM.................................................. 54
FIGURE 4-11: THE PAMS, MYSQL DATABASE CONNECTION.............................................................. 55
FIGURE 4-12: THE PAMS, GENERATING THE TARGET DATASETS......................................................... 56
FIGURE 4-13: A WINDOW FRAME, WITH THE LENGTH OF 11............................................................... 57
FIGURE 4-14: SLIDING THE WINDOW ONE POSITION TO THE RIGHT................................................... 58
FIGURE 4-15: A WINDOW FRAME WITH WEIGHTS ASSIGNED............................................................. 59
FIGURE 4-16: LINEAR WEIGHT VARIATION METHOD WHEN N = 11.................................................... 60
FIGURE 4-17: EXPONENTIAL WEIGHT VARIATION METHOD WHEN N = 11............................................ 61
FIGURE 4-18: THE DIVERSE WEIGHT ASSIGNING METHOD................................................................... 62
FIGURE 4-19: WEIGHTED WINDOW FRAME......................................................................................... 63
FIGURE 5-1: DISTRIBUTED PSAR INFRASTRUCTURE.............................................................................. 67
FIGURE 5-2: INFRASTRUCTURE OF THE PSAR.................................................................................... 68
FIGURE 5-3: Nessus client for security vulnerabilities assessment......................................................... 78
FIGURE 5-4: NMAP REPORTING THE SECURITY ASSESSMENTS............................................................ 79

xi
Figure 6-1: PSAR terminal output for attribute sampling........................................83
Figure 6-2: PSAR terminal output for experiment of numbers of applied feature descriptors. 87
Figure 6-3: Prediction varies along with numbers of applied feature descriptors........88
Figure 6-4: PSAR terminal output of protein sampling............................................89
Figure 6-5: The quality of protein chains.................................................................91
Figure 6-6: Terminal outputs of selecting window length with FAN + exponential model; ......93
Figure 6-7: Selecting window length by FAN + Exp................................................94
Figure 6-8: PSAR terminal output and plot showing selecting window length (IBK + Linear). 95
Figure 6-9: PSAR terminal output and plot showing selecting window length (Kstar + Exp). ..96
Figure 6-10: PSAR terminal output and plot showing selecting window length (LMT + Exp). ...97
Figure 6-11: PSAR terminal output and plot showing selecting window length (SMO + Exp). ....98
Figure 6-12: PSAR terminal output and plot showing selecting window length (Jrip + Exp). .....99
Figure 6-13: PSAR terminal output and plot showing selecting window length (PART + Exp). 100
Figure 6-14: Window length influences to prediction result with different classifiers.......101
Figure 6-15: Results that obtained when classifiers worked with linear weight variation model.........................................................................................................................................................103
Figure 6-16: Results that obtained when classifiers worked with exponential weight variation model........................................................................................................................................................................................................104
Figure 6-17: Performances of classifiers with linear variation model.................................105
Figure 6-18: Performances of classifiers with exponential variation model......................105
Figure 6-19: Combinations of feature descriptors produced prediction accuracy greater than 84%..............................................................................................................................................................................................................107
Figure 6-20: Occurrences of features with prediction accuracies >84%...............................107
Figure 6-21: Combinations of feature descriptors produced prediction accuracy less than 73%. ..............................................................................................................................................................................................................108
Figure 6-22: Occurrences of attributes with prediction accuracies <73%.............................108
Figure 7-1: PSAR interface design.................................................................................111
List of Tables

Table 2-1: List of Amino Acids (GNU Licensed) ................................................................. 9
Table 4-1: The weight values that assigned by linear weight variation model .................. 60
Table 4-2: The weight values that assigned by exponential weight variation model .......... 61
Table 5-1: The configuration of a particular dataset .......................................................... 71
Table 5-2: Configuration of feature sampling module ...................................................... 73
Table 5-3: Configuration for protein sampling module ..................................................... 74
Table 5-4: Configuration for window sampling module ................................................... 74
Table 5-5: Section definition in the table, which records results ...................................... 76
Table 6-1: Top ten ranked attributes in experiment 3.3.1 by using Bayes.TAN ................. 84
Table 6-2: Top ten ranked attributes in experiment 3.3.1, by using IB1 ............................ 85
Table 6-3: Classification of top 168 ranked attributes for Bayes.TAN ............................. 85
Table 6-4: The pattern of self testing ranks ...................................................................... 90
Table 6-5: The pattern of peptide chain length ............................................................... 90
Table 6-6: Performances of learning schemes with weight variation model ................. 102
List of Terminology

Protein:

Proteins are large organic compounds made by combinations of the twenty natural occurring amino acids. Proteins are essential for living life and are a major composition of living cells.

Amino Acid:

Amino acids are any molecules that contain both amine group and carboxyl functional groups. All amino acids share a common structure with a central $\alpha$-carbon. In the thesis amino acids constructs blocks of proteins.

Deoxyribonucleic Acid (DNA):

DNA is a nucleic acid that contains the genetic instructions for biological development of a cellular form of life or a virus. DNA contains two complementary strands, which twist together and form a double helical shape. Two complements of helices in the process of DNA replication, before the cell holding it divides to form two daughter cells.

Ribonucleic Acid (RNA):

RNA is a nucleic acid that contains ribose rings and uracil, unlike DNA. It is transcribed from DNA by enzymes called RNA polymerases and further processed by other enzymes. RNA serves as the template for translation of genes into proteins, transferring amino acids to the ribosome to form proteins.

Protein Attributes Microtuning System (PAMS):

The PAMS is the software that was developed in this research to generate training and testing dataset automatically from unformatted biological data resources. The invention of the PAMS allowed the procedures of data pre-processing, data cleansing, data discretization etc, to be automated.
Protein Structural Accuracies Reckoner (PSAR):

The PSAR infrastructure was developed in order to perform a designated series of prediction tasks automatically according to the specifications submitted by the users. The PSAR infrastructure is a distributed computational platform that requires minimum human intervention.
1 Introduction

1.1 Medication in Human Civilization

Modern archaeological research has established that the earliest human civilization dates back to some 7,000 years (i.e. 5000 B.C.). The knowledge of medication has passed through many generations, despite the rise or fall of a particular dynasty. Early treatments mostly focused on people's beliefs and originated from a variety of sources such as myth, religion and experiences in using herbs (Berenzon & Saavedra 2002; Cowley, Young, & Raffin 1992; Gross 1992; Karpozilos & Pavlidis 2004; Parryjones & Parryjones 1991; Peponi 2002; Ramoutsaki et al. 2001). For example the traditional Chinese herbal medicine came from the myth of a brave doctor who tasted all types of traditional herbal medicines and wrote the "book of St Nong". This precious book is still heavily used in modern traditional herbal medicine research to this day. These types of medication records are not only excavated in China, but also in other ancient civilizations, such as Egypt and ancient Babylon (Andersen 1997; Manyam 1992; Martin-Araguz et al. 2002; Ocklitz 1997; Trevisanato 2005). The papyri dug out in Egypt, known as the "Edwin Smith" and "George Ebers" scrolls, reported complicated treatments for cancer. The Smith papyrus describes surgery, while the Eber papyrus outlines pharmacological, mechanical, and magical treatments. Humankind never gives up trying to defeat diseases and attempting to extend life.

In modern science, along with the advances in the understanding of human anatomy, some of the mysterious diseases have been examined using modern theorems and technologies. For instance, the use of the modern microscope provides opportunities to study diseases and their functions such as the HIV virus, and cancer cells, while cellular pathology provides the scientific basis for the modern pathologic study of these diseases. The word "cancer" came from the Latin word expression "carcinoma", defined by the great doctor Hippocrates (Craik 2000). The word itself reflects the observation that the dissected surface of a solid malignant tumour looked like a crab, which in Greek is denoted as "carcinos". Apart from cancers there are more than 20 diseases that have been...
identified as being caused by protein misfolding, therefore they are called “protein misfolding diseases” (Ellisdon & Bottomley 2004).

1.2 Proteins and Modern Structural Biology

Proteins carry out essential biological functions in every cell of living organisms. To carry out these functions correctly, proteins have to fold themselves into their unique favoured three dimensional shapes in a moist environment. The process of protein folding is remarkably quick, however sometimes it can go wrong. Only in the 1990s did it become clear that wrongly folded proteins are involved in the development of many diseases, such as amyloidoses (Buxbaum & Tagoe 2000; Cohen 1994; Falk & Skinner 2000), Alzheimer’s disease (Lage 2006), prion diseases (Ferrer 2001; Kogan et al. 2002), and some cancers. Improvements in the understanding of protein structures dramatically changed the world of new drug discovery and became the focus of pharmaceutical research and biotech companies; the future of genetic therapy is very promising.

In the past few decades, techniques for identifying protein sequences from the corresponding gene blueprints have become increasingly mature and efficient. However, even with the success of high throughput screening applications, the process of determining protein structure experimentally is still expensive and time consuming. Thus, structural determination tends to outpace sequence determination leading to an increasing gap between them. Computational biologists pursue the target of determining protein structure according to sequence alone, trying to avoid the enormous expense of determining protein structures experimentally, by using X-ray crystallography or nuclear magnetic Resonance (NMR) spectroscopy.

Although the achievement of experimental determination of the structure of biological macromolecules is outstanding, it still cannot meet the current demands, which grow every day. Computational biologists are developing theorems and methodologies to predict the protein structure in silico, and in the past few decades, have achieved remarkable progress.
1.3 Structural Bioinformatics

In the early 1960s Christian Anfinsen and his colleagues investigated the protein ribonuclease, and carried out the experiment whose results were considered to be the cornerstone of modern proteomic science (Anfinsen 1973). The ribonuclease contains 124 amino acids, and folds into a $\beta$-sheet and three $\alpha$-helices. The structure of ribonuclease is stabilized by several disulphide bridges and may be collapsed by adding certain chemicals, or simply by the application of heat. The process of breaking the stable structure of protein is called "denaturing", and by breaking the disulphide bonds the active structure of ribonuclease turns into a useless ball. The process of denaturing ribonuclease can be reversed by the restoration of the environment, more precisely by removal of the previously added chemicals, or by lowering the temperature to the origin status. The ribonuclease then folds back into its natural functional state on its own. From the experiment Anfinsen et al., concluded that the amino acid sequence determined the shape of the protein, a result which brought Anfinsen the Nobel Prize in Chemistry in 1972.

Anfinsen’s discovery shows that it is possible that protein structure can be deduced by applying certain folding functions in its sequence. The ‘trial and error’ method was proved to be infeasible by Cyrus Levinthal in 1966, and was named as the “Levinthal Paradox” (Levinthal 1968). Levinthal stated that if one considered a small protein, perhaps comprising 100 amino acid residues, and allocated each flexible residue to adopt only two different spatial conformations. Then theoretically it would still take 100 billion years to try all possibilities. There are several other approaches toward bypassing the Levinthal Paradox, stated in other publications (Finkel'shtein & Badretdinov 1997; Honig 1999; Karplus 1997).

There are a few methodologies available in the literature to predict protein tertiary structure computationally (Chiu & Goldstein 2000; Fetrow & Bryant 1993). To predict the tertiary structure of a protein, it is good to have a good accurate secondary structure prediction method in hand to provide assistance (Eyrich et al. 1999). Despite producing novel machine learning algorithms, this thesis discusses the possibility of improving protein secondary structure prediction from proposing procedures that purify biological data.
Chapter 1: Introduction

Currently in the literature, comparative modelling is the most commonly studied method of protein secondary structure prediction. It has achieved good records in predicting the secondary structures of proteins, which share sequence identities, but is not necessarily good in predicting proteins, with less sequence similarity (less than 25% sequence identity). Protein threading techniques were developed to cope with the situation that proteins share no significant sequence identity, but share a certain degree of conformational similarities. The thesis makes no use of homology information, yet still achieves accurate results. The Protein Attributes Microtuning System (PAMS) software package presented in this thesis provides a convenient way to prepare the training dataset and testing dataset; and the produced datasets are evaluated within the Protein Structural Accuracy Reckoner System (PSAR) infrastructure. The methodology proposed in this experiment is a statistical, template-free approach to the problem of protein secondary structure prediction. The PAMS/PSAR framework can perform various tasks aimed at either obtaining good prediction accuracy or for discovering patterns in biological data.

In general, the PAMS/PSAR infrastructure proposed in this thesis fulfils several intended tasks. First of all, the hypothesis of this thesis is verified; secondly, certain machine learning classifiers are discovered to predict protein secondary structure more accurately than others; at the same time a few others are discovered to perform the prediction efficiently. The FD232 dataset is proposed as a good protein candidates set for protein secondary structure prediction. In general, the PAMS/PSAR infrastructure offers an insight into the correlations revealed by the purity of biological samples and the accuracy of protein structural prediction.

1.4 Statement of Hypothesis

"As the purity of dataset is refined and the "quality" is improved, the classification of dataset becomes better and therefore the prediction accuracy will improve regardless of which classifier is being used."
1.5 Structure of the Thesis

The structure of this thesis is assembled as following: Chapter One introduces the fundamental concepts and gives a historical review of the state of the art; it also contains the hypothesis. The second chapter reviews protein and protein structure from a biological viewpoint, and indicates the concepts of primary, secondary, tertiary, and quaternary protein structure. Chapter Three reviews the state of the art of protein secondary structure prediction, and the three generations of protein secondary structure prediction methodologies are identified and studied. Chapter Four describes the development of the PAMS. Chapter Five presents the development of the PSAR infrastructure, along with the evaluation of it. In Chapter Six the experimental results obtained by implementing the PSAR infrastructure are illustrated. Finally in Chapter Seven, further work and possible extensions of the current software packages and software framework are described.
2 Protein and Protein Structure

2.1 Protein is Life

The most appropriate phrase to describe the concept of protein from the biological viewpoint is "protein is life". Protein is essential to the structures and functionalities of living cells, and it is suggested that protein forms nearly 75% of the dry weight in the human body. Proteins are one of the foundations of carbon-based life forms living on the Earth, they are complicated, high-molecular-mass organic compounds, made up from 20 naturally occurring amino acids joined together by peptide bonds. With the unlimited combinations of these twenty building blocks, millions of proteins are formed in nature, and there are more or less 100,000 proteins within the human body, carrying out various functions.

Proteins such as enzymes catalyze chemical reactions within the human body, such as hydration of a carbon dioxide (Plhar 1965); or modification of small organic molecules. Enzymes are involved in the process of reading genetic information from the DNA, the first procedures of proteins synthesis. Proteins are functioning as transporters within the human body, carrying nutrients and other relevant molecules around all over the body throughout the cerebral circulation lymphatic system (Beaven & McEwan 1969; HALLGREN 1954). Proteins such as collagen contain three polypeptide strains, each of which is a left-handed α-helix. The three half helix strands twist together by numerous hydrogen bonds, to form a strong stable physical structure. These bundles of proteins are the major component of connective tissue in living animals, also giving external cellular structure (Bunyaratavej & Wang 2001; Jones 1976; Lachman et al. 1992; Lowther 1978). Transmembrane proteins form pores in cellular membranes allows ions to pass through (Fleishman, Unger, & Ben-Tal 2006). The transportation of proteins relies on some other proteins as well. Proteins are critical in the process of the immune system fighting against the intrusion of harmful bacteria and viruses. The effectiveness of the immune system highly depends on the production of antibodies which are proteins capable of binding to specific foreign particles such as bacteria and viruses, disabling the
harmful effects (CHERCHENKO 1963; Kosmas, Linardou, & Epenetos 1993; Scallon et al. 2006; Tuormaa 1988). In short proteins are the foundation of life.

2.2 The Physical Composition of Proteins

Proteins are constructed by various types of RNAs within cells. The DNA carries the information of inheritance like a giant book of blueprints, which instructs the construction of biological molecules within living organisms. Any mistakes occurred in the gene code have the potentials to cause disastrous consequences - genetic disorders (Cournoyer et al. 1990; Kingston 1989; Mayeux 2005; Ponz de 1994).
As represented in the above diagram Figure 2-1, DNA records the building blueprint of a particular protein. The messenger RNA (mRNA) transports a piece of gene code out from the cell nucleus, and transport RNA (tRNA) brings building blocks - amino acids to form the protein polypeptide chains. All genetic code is presented by a set of four, e.g. A, D, T and G. Each amino acid is encoded by three codons, as presented in Figure 2-2. A DNA gene code “GCTTGCAGATGAA” may present a sequence of amino acids as “ACDE”. Apart from the code of amino acids, several codons also indicate extra information bits such as the start indicator and end point of a protein. In the follow table a list of acronyms of amino acids is presented:

<table>
<thead>
<tr>
<th>Amino Acid</th>
<th>3-Letter</th>
<th>1-Letter</th>
<th>Side chain polarity</th>
<th>Side chain acidity or basicity</th>
<th>Hydropathy index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alanine</td>
<td>Ala</td>
<td>A</td>
<td>nonpolar</td>
<td>neutral</td>
<td>1.8</td>
</tr>
<tr>
<td>Arginine</td>
<td>Arg</td>
<td>R</td>
<td>polar</td>
<td>strongly basic</td>
<td>-4.5</td>
</tr>
<tr>
<td>Asparagine</td>
<td>Asn</td>
<td>N</td>
<td>polar</td>
<td>neutral</td>
<td>-3.5</td>
</tr>
<tr>
<td>Aspartic acid</td>
<td>Asp</td>
<td>D</td>
<td>polar</td>
<td>acidic</td>
<td>-3.5</td>
</tr>
<tr>
<td>Cysteine</td>
<td>Cys</td>
<td>C</td>
<td>polar</td>
<td>neutral</td>
<td>2.5</td>
</tr>
<tr>
<td>Glutamic acid</td>
<td>Glu</td>
<td>E</td>
<td>polar</td>
<td>acidic</td>
<td>-3.5</td>
</tr>
<tr>
<td>Glutamine</td>
<td>Gln</td>
<td>Q</td>
<td>polar</td>
<td>neutral</td>
<td>-3.5</td>
</tr>
<tr>
<td>Glycine</td>
<td>Gly</td>
<td>G</td>
<td>nonpolar</td>
<td>neutral</td>
<td>-0.4</td>
</tr>
<tr>
<td>Histidine</td>
<td>His</td>
<td>H</td>
<td>polar</td>
<td>weakly basic</td>
<td>-3.2</td>
</tr>
<tr>
<td>Isoleucine</td>
<td>Ile</td>
<td>I</td>
<td>nonpolar</td>
<td>neutral</td>
<td>4.5</td>
</tr>
<tr>
<td>Leucine</td>
<td>Leu</td>
<td>L</td>
<td>nonpolar</td>
<td>neutral</td>
<td>3.8</td>
</tr>
<tr>
<td>Lysine</td>
<td>Lys</td>
<td>K</td>
<td>polar</td>
<td>basic</td>
<td>-3.9</td>
</tr>
<tr>
<td>Methionine</td>
<td>Met</td>
<td>M</td>
<td>nonpolar</td>
<td>neutral</td>
<td>1.9</td>
</tr>
<tr>
<td>Phenylalanine</td>
<td>Phe</td>
<td>F</td>
<td>nonpolar</td>
<td>neutral</td>
<td>2.8</td>
</tr>
<tr>
<td>Proline</td>
<td>Pro</td>
<td>P</td>
<td>nonpolar</td>
<td>neutral</td>
<td></td>
</tr>
<tr>
<td>Serine</td>
<td>Ser</td>
<td>S</td>
<td>polar</td>
<td>neutral</td>
<td>-0.8</td>
</tr>
</tbody>
</table>
Chapter 2: Protein and Protein Structure

<table>
<thead>
<tr>
<th>Amino Acid</th>
<th>Code</th>
<th>Charge</th>
<th>Polarity</th>
<th>pKa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threonine</td>
<td>Thr</td>
<td>T</td>
<td>polar</td>
<td>neutral</td>
</tr>
<tr>
<td>Tryptophan</td>
<td>Trp</td>
<td>W</td>
<td>nonpolar</td>
<td>neutral</td>
</tr>
<tr>
<td>Tyrosine</td>
<td>Tyr</td>
<td>Y</td>
<td>polar</td>
<td>neutral</td>
</tr>
<tr>
<td>Valine</td>
<td>Val</td>
<td>V</td>
<td>nonpolar</td>
<td>neutral</td>
</tr>
</tbody>
</table>

Table 2-1: List of Amino Acids (GNU Licensed)

The amino acids are bonded together by dehydrating a water molecule from the amino group and carboxyl group to form a peptide bond which is quite strong. Once the bond has been formed, it requires large amount of energy (vary from 100kJmol$^{-1}$ to 300kJmol$^{-1}$, depends on the types of amino acids to form the peptide bond.) to be broken. The following diagram Figure 2-3 represents the comparison of the DNA structures on a double helix and the single stranded RNA structure:

Figure 2-3: Comparisons between DNA and single stranded RNA. (GNU Licensed)
The proteins perform their designated functions after they are formed within living cells. The cell can regulate the presence of a certain protein by marking unwanted proteins with a label consisting of the polypeptide ubiquitin. Labelled proteins are then broken down, or degraded rapidly in cellular “waste disposers” called proteasomes (Ciechanover 2005; Ciechanover 2006). The released amino acids can be reused again to form another protein.

### 2.3 The Process of Protein Folding

Proteins tend to fold into their favourite three dimensional shapes in a moist environment. The process of folding is driven by intrinsic physical forces, which appears unclear in the literature. There are several statistical analyses to this problem, producing matrices, that model inter-residue contact energy between different types of amino acids (Oliveberg et al. 1998; Wang & Lee 2000). The time cost of protein folding varies from microseconds to nanoseconds, depending on the size and complexity of it (Baldwin 1994; Chan 1998; Kubelka, Hofrichter, & Eaton 2004; McCammon 1996). However, observing the process of protein folding in real time is not a possible task at the moment, even with the most accurate microscope and most sophisticated camera. The folded protein is floating in a liquid environment and looks like a tiny crystal submarine under water.

Proteins denature if the stable “peaceful” environment goes mad – in a denaturing condition\(^1\). Biologically say, the denatured proteins are useless mass balls, not capable to carry out their biological functions. The process of heating the solvent damages the protein: as the temperature increases the protein within gains energy and literally shakes apart the bonds such as disulphide bridges and H-bonds between segments of amino acid strings, causing the protein to unfold. When the temperature goes higher, the protein gains more energy, to form new, stronger bonds with the neighbouring molecules. An example would be to boil an egg, the liquid egg whites turns hard and opaque. Certain chemicals cause the Soya milk to become solid, and make the famous Chinese food – “TOFU”. In some case the proteins can refold into their native states if the solvent is restored back to origin status, e.g. the temperature is lowered and certain chemicals are removed. In other

---

\(^1\) The denaturing condition includes: urea, guanidine hydrochloride, guanidine thiocyanate, organic solvent or elevated temperature.
cases, the proteins cannot fold back, such as it is not possible to return a boiled egg to its raw state, and make the tofu back into soya milk.

Protein misfolding occurs as a result of protein composition damaging. For instance removal of a number of amino acids from the peptide chain can cause the collapse of the protein structure and lead to arbitrary useless spherical conformations.

Empirically the protein structure could be determined by X-ray crystallography or NMR spectroscopy. In silico the protein structure could be predicted by a learning network, which is trained previously. In the following paragraph a series of protein structural concepts is illustrated.

2.4 Protein Composition and Structures
Proteins are formed from sequences of amino acids, which are bonded together by covalent bonds in a ‘head to tail’ configuration. Some proteins contain only a single peptide chain; some of the more complicated proteins contain more than one peptide chain and form complicated quaternary structures. Peptide chains fold into a stable structure by inter-molecular or intra-molecular forces, such as van der Waals forces\(^2\), hydrophobic forces\(^3\), H-bonds\(^4\) etc. The protein structure is dominated by a mixture of effects of these forces. In general, protein folding is an amazing phenomena that occurs right here on Earth.

---

2 In chemistry, the term van der Waals force refers to a particular class of intermolecular forces, which describes electromagnetic forces that acts between molecules or between widely separated regions of a macromolecule.

3 In chemistry, the term hydrophobic refers to the physical property of a molecule that repelled from a mass of water molecule.

4 In chemistry, the term hydrogen bond describes a special type of attractive interaction that exists between certain chemical groups of opposite polarity. The hydrogen is stronger than van der Waals forces, however much weaker than both the ionic bond and the covalent bond.
2.4.1 Amino Acids as Building Blocks of Protein

Amino acids are the building blocks to construct proteins. There are twenty naturally occurring amino acids, which are bonded sequentially in a “head-to-tail” configuration, forming millions of different proteins. Each amino acid shares a common structure of a central tetrahedral α carbon (Cα), which is covalently linked to both an amino group (NH₃) and a carboxyl group (COOH).

![Structure of amino acid](image)

Figure 2-4: Structure of amino acid (showing zwitterionic structure)

As illustrated in the above diagram (Figure 2-4), amino acids are differentiated from each other by the ‘R’ residue – the side chain. For instance, alanine has a methyl group attached as the ‘R’ residue; glycine has a single hydrogen atom attached as its ‘R’ residue. The ‘R’ residue defines the characteristics of the amino acid, *i.e.* whether it is a weak acid or weak base, a hydrophobic residue or hydrophilic residue, a polar residue or a non-polar residue *etc.* It is believed that the characteristics of the ‘R’ residue are critical for protein structures and functions (Hill & DeGrado 2000; Schueler-Furman & Baker 2003). As the 26 letters in the English language form the endless literature of poetry, opera, masterpieces *etc*, the 20 amino acids in nature, form the book of life.

![Structure of alanine](image) ![Structure of glycine](image)

Figure 2-5: Structure of alanine (showing zwitterionic structure) Figure 2-6: Structure of glycine (showing zwitterionic structure)
Amino acids join together via peptide bonds. The amino group and the carboxyl group of each amino acid may react with each other in a ‘head-to-tail’ fashion by dehydrating one water molecule, and forming a covalent linkage. Certainly, at both ends of the protein, there remains a free amino group, and a carboxyl group, which are referred to as the N and C termini. The C-terminus of proteins, which includes the single terminal alpha-carboxyl group and preceding residues, is uniquely positioned to serve as a recognition signature for a variety of cell biological processes, such as protein targeting, subcellular anchoring and static and dynamic formation of macromolecular complexes (Chung et al. 2002). However the study of termini is not included in this thesis. Two linked amino acids are called dipeptides, three amino acids are named tripeptides, and four amino acids are called tetrapeptides. Long chains of amino acids also are called oligopeptide or polypeptides, or proteins. An example of a tripeptide is represented in the following Figure 2-7:

As the dehydration reaction causes the linkage of peptide bonds, the amide hydrolysis reaction (the addition of water molecular) breaks the peptide bonds. The process of debonding is extremely slow in living organisms, and is usually catalyzed by enzymes. Further reading about peptide bonds can be found in (Pauling 1931;Pauling 1932;Pauling 1985;Pauling & Sherman 1933b;Pauling & Sherman 1933a;Pauling & Wheland 1933).

2.4.2 Primary Structure as Sequence Indicator

The primary structure of a protein is represented by the amino acid sequence within the polypeptide chain(s). In the primary structure of a particular protein, only the identities of amino acids are declared. There is no structural information carried within the primary
structure of a protein, unlike concepts such as the secondary structure, or the tertiary structure of a protein, mainly focus in presenting three dimensional conformational structures of it.

In the following diagram Figure 2-8 a snapshot of the protein primary structure is presented:

![Figure 2-8: Protein primary structure. (GNU Licensed)](image)

The term “primary structure” was firstly coined by Kaj Ulrik Linderstrom-Lang in his 1951 Lane Medical Lecture: “Proteins and Enzymes”.

### 2.4.3 Secondary Structure as Pattern of General 3D Form

The secondary structure is used to describe the captured general patterns of three dimensional forms of a local segment in the polypeptide chain. The formal definition of the secondary structure relies on the patterns of hydrogen bonds between backbone amide groups. The commonly used secondary structure types of the Dictionary of Protein Secondary Structure (DSSP) are defined according to above definition (Hooft et al. 1996; Kabsch & Sander 1983b):

- G = 3-turn helix (3_10 helix). Minimum length 3 residues.
Chapter 2: Protein and Protein Structure

- \( H \) = 4-turn helix (\( \alpha \) helix). Minimum length 4 residues.
- \( I \) = 5-turn helix (\( \beta \) helix). Minimum length 5 residues.
- \( T \) = hydrogen bonded turn (3, 4 or 5 turn).
- \( E \) = \( \beta \) sheet in parallel and/or anti-parallel sheet conformation (extended strand). Minimum length 2 residues.
- \( B \) = residue in isolated \( \beta \)-bridge (single pair \( \beta \)-sheet hydrogen bond formation).
- \( S \) = bend (the only non-hydrogen-bond based assignment).

The remaining bits are classified as coil, stated as “random coil”. In secondary structure prediction, there are three secondary structural units, which named ‘\( \alpha \) helix’, ‘\( \beta \) strand’, and ‘turn’, widely used in the literature. At the moment, helix, strand and coil are the three secondary structure units that applied in most of protein secondary structure prediction applications. Apart from the eight secondary structures defined in the DSSP programme (Kabsch & Sander 1983a), there are several other secondary structure clarification methodologies introduced in the literature, such as DEFINE (Richards & Kundrot 1988) and STRIDE (Frishman & Argos 1995), which are all heavily cited in the literature. There are reviews in the literature describing the applications uses of these secondary structure alignments (Frishman & Argos 1997a; Frishman & Argos 1997b; Frishman & Argos 1996b).

To simplify the problem, usually only three of secondary structure units are applied in the methodologies to predict protein secondary structure: helix, strand, and coil (H, E, and C). There are several eight to three reduction methods to be applied to reduce the complexity:

1. \( H, G \) and \( I \) to \( H \); \( E \) to \( E \); the rest to \( C \).
2. \( H, G \) to \( H \); \( E, B \) to \( E \); the rest to \( C \).
3. \( H, G \) to \( H \); \( E \) to \( E \); the rest to \( C \).
4. \( H \) to \( H \); \( E, B \) to \( E \); the rest to \( C \).
5. \( H \) to \( H \); \( E \) to \( E \); the rest to \( C \).
It is obvious that the method 5 increases the accuracy (Cuff & Barton 2000; Cuff & Barton 1999). The PAMS applied method 1 to provide a strict measurement to the prediction accuracy.

In the following Figure 2-9, a three dimensional structure of the myoglobin protein is presented. The coloured amino acids segments present $\alpha$ helices, and the coils are presented in white. There are no $\beta$ strands in this protein. This protein was the first to have its structure solved by X-ray crystallography by Max Perutz and Sir John Cowdery Kendrew in 1958 (Kendrew & Perutz 1957). This discovery brought them the Nobel Prize in Chemistry in 1962.

![Figure 2-9: Three dimensional structure of the myoglobin protein. (GNU Licensed)](image)

The research into protein secondary structure prediction is widely published in the literature. It is considered to be the first step towards the ultimate goal of protein structure determination (Baldi et al. 1999; Frishman & Argos 1997b). This thesis presents an effective framework to process the biological samples, in order to achieve better protein secondary structure prediction accuracy regardless of the machine learning schemes used.
2.4.4 Super-Secondary Structure

In 1973 the discovery was made that a few α-helices and β-strands were frequently repeated within structures, named “super-secondary structure” (Rao & Rossmann 1973). It was suggested that these structural motifs were associated with particular biological functions, while others were part of a big structure and formed overall biological functions. According to the classification (Levitt & Chothia 1976), there are four major groups:

- proteins containing mostly α-helices;
- proteins containing mostly β-sheets;
- proteins that contain α-helices and β-sheets in an irregular sequence;
- α/β proteins with alternate segments of α-helices and β-sheets.

The simplest motif is two helices joined by a loop, which is responsible for DNA binding, and calcium binding, thereby regulating cellular activities (Procyshyn & Reid 1994a; Procyshyn & Reid 1994b; Reid & Procyshyn 1995).

2.4.5 Tertiary Structure

The tertiary structure of a protein defines its actual shape. It contains the conformational information of all atoms within it without considering its relationships with other molecules. Knowing the tertiary structure of a protein is essential in the study of understanding a protein’s biological functions. Currently, the determination of tertiary protein structure is highly dependent on the participation of X-ray crystallography technique or multi-dimension NMR spectroscopy. However, sometimes the screening methodologies are not accurate and even could not determine some bits of the protein structure. e.g. X ray diffraction gives the atoms in position in space, but they have to be in a crystal. However some bits of the protein cannot be crystallised such as those responsible for connection bits in the protein of HNF4 (alpha) or Hepatocyte Nuclear Factor-4a, which are constantly changing and thus not be seen properly (Aggelidou et al. 2006; Dhe-Paganon et al. 2002; Duda, Chi, & Shoelson 2004). The prediction methods will
help in this kind of situation. Obtaining high prediction accuracy for the secondary structure is the first step towards predicting tertiary structure.

The tertiary structure of a protein may contain more than one peptide chain. In the following Figure 2-10, an insulin sample with two chains is presented. Chain A contains 21 amino acids and chain B contains 30 amino acids. These two chains are bonded together with two disulphide bonds to establish a reasonable stable structure for the insulin.

![Figure 2-10: Stick representation of overall structure of insulin.](image)

The tertiary structure of this protein is maintained by the protein core, which is formed by the hydrophobic residues oriented inside, and hydrophilic residues staying outside (Esipova & Tumanyan 1972). The disulfide bridges, formed between two cysteine residues, are another major force in the preservation of the tertiary structure.
2.4.6 Quaternary Structure

The quaternary structure is an even more complicated protein structure, formed by polypeptide chains in space and reliant on non-covalent interactions. The proteins that could have quaternary structures must have no chains which are covalently associated. A typical example is haemoglobin that contains four polypeptide chains held together in a specific conformation as required for its function (Kilmartin, Hewitt, & Wootton 1975). Both tertiary structure and quaternary structure are mentioned by structural biologists quite often in the literature. A better understanding of tertiary structure and quaternary structure brings new insight for the discovery of new drugs to attack diseases, which are currently incurable, such as cancer.

2.5 Essentiality of Protein Structure

The research of protein structure has become the centre of pharmaceutical research because of the promising future. Between December 1995 and March 1996, the Food and Drug Administration (FDA) of United States of America approved the first three HIV protease inhibitors - Hoffman-La Roche’s Invirase™ (saquinavir), Abbott’s Norvir™ (ritonavir), and Merck and Co., Inc.’s Crixivan® (indinavir). These medicines have drastically cut the numbers of AIDS deaths in developed countries in the last few years, and although these HIV protease inhibitors did not become the miracle cure they expected to be, they also represent a triumph of developing new drugs.

The ultimate goal of curing genetic diseases and developing individual-oriented therapies requires a fast, economic instrument to determine the structure and movements of the corresponding faulty protein. By making the malfunctioned protein work, the disease may be cured.

2.6 Conclusions

Protein is life. The phrase has been acknowledged for over one hundred years, but the understanding of protein structure is insufficient for industrial demands. However a new wave of breakthroughs in this area is being formed at the moment, and a large number of
reviews suggest that the golden age of protein is coming (Bai & Englander 1996; Baldi et al. 1999; Chen et al. 2005; Chopra 1998; Duncan, Ringsdorf, & Satchi-Fainaro 2006; Geisow 1992; Johnson-Leger et al. 2006; Katzen, Chang, & Kudlicki 2005; Koehl & Levitt 1999; Middaugh 1994; Myles 2006; Rosenberg & Goldblum 2006; Shehadi, Yang, & Ondrechen 2002; Tramontano 2004; Tsunasawa 1994; Waterlow 1995). Progress in understanding both structural and functional characteristics of the protein is essential and must be achieved.

The next chapter introduces the state of art of protein structure prediction, more precisely, secondary structure prediction.
3 Protein Secondary Structure Prediction

Since Levinthal observed that there was insufficient time to search the entire conformational space exhaustively for appropriate protein structure candidates, people realized that the prediction of protein structure has to be approached in an alternative way. The techniques applied in predicting protein structures are multi-disciplinarily diverse, such as evolution theory, data mining, and machine learning. In the past four decades the study of protein secondary structure has received a great many attentions in the literature. Currently, the highest prediction accuracy is approximately 78% (Rost & Sander 2000).

3.1 Overview of General Protein Structure Prediction

The process of protein 3D structure prediction at the atomic level could be defined by following the following flowchart in (Figure 3-1):

![Figure 3-1: Protein structure prediction flowchart (by Rob Russell, EMBL)]
As described in the above flowchart, once the protein sequence is identified from the experimental data, it can be applied by multiple sequence alignment (MSA) straightforwardly. The MSA method results whether the input sequences have an evolutionary relationship by which they share a lineage and are descended from a common ancestor (Bacon & Anderson 1986). The MSA methods are commonly used to assess sequence conservation of protein domains, protein secondary structure, and protein tertiary structure. More and more sophisticated algorithms have been implemented in improving MSA accuracy such as the implementation of Fuzzy Hidden Markov Models, Bayesian segmental models etc (Chu et al. 2006; Collyda et al. 2006; Morgenstern et al. 2006; Pei & Grishin 2006; Roshan & Livesay 2006; Yamada, Gotoh, & Yamana 2006).

Similar to the protein secondary structure prediction, various machine learning models have been applied. Once the sequence is identified to be homologous to a family of proteins with known structures comparative modelling techniques (Centeno, Planas-Iglesias, & Oliva 2005; Contreras-Moreira, Fitzjohn, & Bates 2002) can be used to produce a three dimensional protein model. While the sequence shares no obvious homologous characteristic with proteins with known structure, the three dimensional protein models have to be predicted by following an alternative route. The first step toward that route is to predict the secondary structure of the sequence. This prediction provides the location of α helices and β sheets. Once a secondary structure of protein is predicted both the sequence and secondary structure are fed into a fold recognition program. The theoretical foundation of this procedure is that some of the proteins may not share significant sequence similarity, but they still share common structures (Godzik 2003; Koretke et al. 1999; Marchler-Bauer & Bryant 1999; Ota 2002). The fold recognition programs tend to detect similarities between protein 3D structures, but require extra structural information generated by the first step. If there are no obvious folds recognized, the protein has to go to ab initio structure prediction server (Bradley et al. 2005; Bystroff & Baker 1997; Bystroff & Baker 1998; Rohl et al. 2004), and end up having a structure predicted over there. If the result is positive that the protein adopts a particular fold within the database, a procedure of analyzing the fold family is required to both validating the fold results and perform an alignment of the secondary structure elements to the protein and the found proteins that adopt a similar fold. The SCOP, CATH, FSSP protein structure classification database could be used in this procedure (Greene et al. 2006; Hadley & Jones 1999; Holm & Sander 1996; Reedy & Bourne 2003). The suitable
fold via fold recognition can be fed into the homology modelling techniques to be used to deduce the tertiary structure of the given protein. The homology modelling techniques includes WHAT IF, MODELLER, etc (Fiser & Sali 2003; Vriend 1990).

In order to predict the tertiary structure from sequence alone, the prediction of secondary structure is the initial step. Without a prediction of secondary structure the pure \textit{ab initio} prediction of tertiary structure from sequence alone is very computationally expensive. In the following paragraph a literature review of protein secondary structure prediction is presented.

### 3.2 Historical View of Protein Secondary Structure Prediction

Historically speaking the first attempt to determine the secondary structure of a protein from its sequence was performed in 1957, trying to correlate the proline with the formation of \( \alpha \) helix (Szent-Gyorgyi & Cohen 1957), shortly after Pauling and Corey suggested the existence of the \( \alpha \) helix and \( \beta \) sheet (Pauling & Corey 1951a; Pauling & Corey 1951b; Pauling & Corey 1951c; Pauling, Corey, & Branson 1951). In general protein structure prediction methods have been categorized into three generations, according to their characteristics and performance.

#### 3.2.1 Methods of First Generation

The most notable characteristic of methods in first generation is that the prediction is based on statistical review of propensities that each single amino acid residue adopt secondary structure units (Rost & Sander 2000). Later on, several physical attributes were taken into account to bridge the correlations between amino acids and their structural conformations. One of the obvious disadvantages of first generation methods is that the influence of neighbouring amino acids is ignored. Another disadvantage is that the number of proteins with known sequences and structures was very limited during the period of 1970s – 1980s. There were not many data provided to computational chemists to analyse. The analytical process itself relied on heavy manual work and was therefore
not at all efficient. The most notable and cited methods have been published is a number of studies (Chou & Fasman 1974b; Chou & Fasman 1978; Chou & Fasman 1979; Garnier, Gibrat, & Robson 1996; Lim 1974a; Lim 1974b). Some of the methods are reported to consequently influence later on prediction methods (Kabsch & Sander 1983c).

3.2.2 Methods of Second Generation

The protein secondary structure prediction methods improve along with the growth of protein data from crystal structures, and the advances in computer hardware. The achievements in computer hardware increases computational capacity, and the growth of proteins with structure crystallised provide more samples for investigation. The spread of Internet allows research institutes and individuals to publish their protein samples online, for whoever is interested to collect and investigate. The advances of communication also allow projects around world to be unified, provide data in a common data standard, to avoid incompatibility. A good example is the birth of Uniprot project (Apweiler et al. 2004; Bairoch et al. 2005; Leinonen et al. 2004; Wu et al. 2006a).

The progress in machine learning has also initiated a wave of developments in protein secondary structure prediction methodologies. Various machine learning schemes have been applied to the problem to explore the correlations of protein sequences and structures. The first method that applied neural network into protein secondary structure prediction was performed by Qian and Sejnowski in 1988 (Qian & Sejnowski 1988b); the method also applied a window mechanism along the sequence, to predict secondary structure of a particular amino acid according to a segment of amino acids. The implementation of the window mechanism is an epoch-making improvement, indicating the acknowledgement of taking neighbouring amino acids into account while performing the prediction. Since then, techniques such as graphic theory (Canutescu, Shelenkov, & Dunbrack, Jr. 2003), expert system (Robson et al. 1990), decision tree (He et al. 2006; Selbig, Mevissen, & Lengauer 1999), support vector machine (Byvatov & Schneider 2003; Cai et al. 2003; Chen et al. 2006a; Chen et al. 2006b; Hu et al. 2004; Hua & Sun 2001b; Lewis, Jebra, & Noble 2006; Pham, Satou, & Ho 2003; Wang et al. 2004b; Wang, Xue, & Xu 2006; Zhang, Yoon, & Welsh 2005), Bayesian theorem and
Bayesian network classifiers (Chinnasamy, Sung, & Mittal 2005; Chinnasamy, Sung, & Mittal 2004; Chu et al. 2006; Raval, Ghahramani, & Wild 2002; Robles et al. 2004; Schmidler, Liu, & Brutlag 2000), and other applications applied neural network (Cai & Zhou 2000; Cai et al. 2002; Cai, Liu, & Chou 2003; Cai, Liu, & Chou 2001; Choy, Sanctuary, & Zhu 1997; Dalmas, Hunter, & Bannister 1994; Dombi & Lawrence 1994; Holbrook, Dubchak, & Kim 1993; Holley & Karplus 1989; Huang et al. 2005; Imoto 2006; Kneller, Cohen, & Langridge 1990; Muskal & Kim 1992; Sasagawa & Tajima 1993; Sreerama & Woody 1994; Vieth & Kolinski 1991; Yu & Head-Gordon 1995). The support vector machine was also applied to classify quaternary structure of proteins (Zhang et al. 2003). There is also a perceptron network application in predicting protein super-secondary structure (Sun et al. 1997). Applying machine learning schemes in predicting protein structure has become a fashion in the literature, widely studied and progressed rapidly. In general, the literature of secondary structure prediction is a blossoming research field, and these available methods presented unlimited imagination and creative spirit, encouraging scientists around the world keep attacking the protein folding problem.

3.2.3 Methods of Third Generation

The main features of third generation methods are the involvement of evolutionary information, and revolutionary multiple levels of computation during the process of prediction (Rost 2003). The PHD method (Rost 1996a; Rost, Sander, & Schneider 1994a) is an outstanding prediction method that presents the computational characteristics of third generation. The prediction accuracies of third generation methods approach 80%.

The prediction of secondary structure has kept progressing during the past four decades, targeting the ultimate goal of achieving a prediction accuracy of protein secondary structure of 100%. However, Rost suggested that in fact the goal was unachievable and also the prediction of protein secondary structure may not necessarily require being 100% accurate (Rost, Sander, & Schneider 1994c).
3.3 Review of Prediction Methods

A list of reviews to various methods covering first generation, second generation, and third generation is introduced. The review is focused on the implemented mathematical models and the preparation of the dataset.

3.3.1 Chou-Fasman Method

The propensities of amino acids to tend to fold into particular secondary structure units were very early studied. The propensities revealed interesting conformational properties of amino acids, just as Chou and Fasman discovered in their research (Chou & Fasman 1974a). The Chou-Fasman method attempted to analyze statistically the secondary structure propensities for amino acids with known structures. Each amino acid was assigned a likelihood to three different secondary structure units:

\[
\begin{align*}
\frac{\Pr[i \mid \beta - sheets]}{\Pr[i]} & \quad \frac{\Pr[i \mid \alpha - helices]}{\Pr[i]} & \quad \frac{\Pr[i \mid Others]}{\Pr[i]}
\end{align*}
\]

The probability of amino acid \(i\) in each secondary structure is determined by the above three likelihoods. For example, there are a total of 20,000 amino acids, and 2,000 of them are serine, and there are 5,000 amino acids in helices structure, 500 of them are serine.

The serine propensity to helix in this case is \(\frac{500}{5000} = 1\), which means all serines tend to fold into helices, gives a strong argument that serine tend to fold into helix.

Once the propensities of amino acids are calculated, according to their propensities values, amino acids are categorized as helix-former, helix-breaker, and helix-indifferent. The helix-former amino acids are favoured to fold into helices, and helix-breakers are likely to stop the helices continuing. Helix-indifferent amino acids have in-between tendencies to fold to helices. In the same way sheet-former, sheet-breaker, and sheet-indifferent were defined. Each amino acid was classified as one of helix-related definitions or one of sheet-relevant definitions, \(i.e.\) glycine and proline are found to be helix-breaker.
The results of Chou-Fasman method were used to find out the nucleation sites. The average three states accuracies are reported to be 54.4%. The Chou-Fasman method was extended by applying a bigger database (Argos, Hanei, & Garavito 1978), and implemented in predicting conformations of other cases (Nishikawa 1983; Rawal & Raval 1990; Venkaiah & Kumar 1991). However, the result of Chou-Fasman method was also criticized according to reliability (Kyngas & Valjakka 1998).

3.3.2 GOR Method

The initial idea of the GOR method was to provide a more rigorous method to calculate secondary structure likelihood than the Chou-Fasman method did (Garnier, Osguthorpe, & Robson 1978). The GOR method is based on information theory, which it uses to analyze the correlations between amino acids and their secondary structure. The GOR method reported 55-60% prediction accuracy, approximately 7% increases in prediction accuracy. The GOR method has been continually updated since it was firstly introduced, and in the latest GOR V version, it declares that the usage of both information and Bayesian statistic yielded a full-jackknifing prediction accuracy in Q₃ reached 73.5% (Kloczkowski et al. 2002b; Sen et al. 2005; Sen et al. 2006).

3.3.3 Qian-Sejnowski Method

The Qian and Sejnowski method (Qian & Sejnowski 1988a) made the first attempt to predict protein secondary structure by using a feed-forward neural network model. The method applied a window mechanism into the training data, took a segment of amino acids as the input while predicting the secondary structure of the amino acid in the central position of that window framework. There are three target classes, helix, strand, or coil, defined by curtailing the eight structural classes into three, following the rules:

\[ H, G \rightarrow H; \ E \rightarrow E; \ B, I, S, T, \rightarrow C. \]
The Qian-Sejnowski method took 106 proteins with known structures from the Brookhaven protein data bank as candidates for both training and testing. These 106 proteins contain no homologous information, comprising training patterns of an equivalent number of contained amino acid residues.

The network is constructed with an input of 17 nodes, 5 hidden nodes in hidden layer and 3 outputs. Initially, these nodes are fully connected and randomly weighted, then trained by using random presentations of the overall dataset. The weights are adjusted after the training process finished. Predictions are made based on the winner-takes-all basis. The strongest network output determines the secondary structure of that particular amino acid.

The overall Qian and Sejnowski method reported a prediction accuracy of percentage 64.3%. The method increased the prediction accuracy for nearly 10% than the representative methods of first generation prediction method. The authors draw the conclusions that although the prediction increases, however the methods suffer from a list of drawbacks: 1) the prediction is based on a limited local content (window size), and non-local factors are not been taken into account. Yet there is considerable evidence to suggest that long-range interactions constrain the formation of protein secondary structure; 2) The prediction is based on limited amount of protein samples and the physico-chemical attributes of amino acids are not presented in the network; 3) the network model involved no available knowledge about principles underlying protein structure; 4) The prediction is uncorrelated. The prediction of amino acid in position $i$ has no influence on the prediction of amino acid in position $i+1$. 5) The prediction relies strongly on the performance of that single network. There are quite a few of publications describing work dealing with these drawbacks (Chandonia & Karplus 1995; Chandonia & Karplus 1996; Holley & Karplus 1989; Riis & Krogh 1996; Rost & Sander 1993a; Rost & Sander 1993c; Rost & Sander 1994), following the publishing of Qian and Sejnowski's work.

### 3.3.4 Applications Using Support Vector Machine

Support Vector Machines have become more and more popular in recent years. There are various methods applying support vector machines with different encoding schemes to
amino acids, which achieved reasonable good prediction accuracies (Hu et al. 2004; Hua & Sun 2001a; Wang et al. 2004a). The support vector machine represents a new learning scheme of supervised pattern discovery, which has been widely applied to discover patterns, including object recognition, speaker identification, gene function prediction with microarray expression profiles etc. The assessment of a support vector machine's performance in protein secondary structure prediction is also positive. The method of Hu introduced a new tertiary classifier (which is different from tertiary structure classifier!), which combine the results of one-versus-one binary classifiers, and compare the results with other tertiary classifiers. The method reported a prediction accuracy of the classifier, reached 78.8%. The method of Hua implemented similar techniques on a dataset of 513 non-homologous protein entries – CB513 (Cuff & Barton 1999), achieved a segment overlap accuracy of 76.2% through sevenfold cross validation, and per-residue three state prediction accuracy of 73.5%. Wang and his colleagues in 2004 implied the support vector machine learning scheme in a multiple dimension vectors representative dataset for amino acids in the CB513 dataset, achieved a three state per-residue prediction accuracy of 78.44%. Apart from constructing a multi-layered support vector machine classifier, the method also included physico-chemical properties and structural propensities of twenty natural occurring amino acids. The method conducted three experiments on the RS126 dataset (Rost & Sander 1993b) and concluded the optimal window length in their experiments was a number of 15. The authors declared that the accuracy could be slightly increased by taking more physico-chemical attributes into account and using the latest data set. The disadvantage of this method was the number of classifiers used would increase exponentially along with the dimension of the vector and therefore slow down the process of the classification.

Another SVM approach to protein secondary structure prediction was made by Ward and his colleagues (Ward et al. 2003). The method deployed three 'one-versus-one' binary support vector machine classifiers with the quadratic kernel for sequence-to-structure classification. The space and time complexities also were discussed in the paper. It presented that using 'one-versus-one' required less memory than 'one-versus-rest' class assignments.
The support vector machine learning schemes are considered as promising methods in the task of predicting protein secondary structure. The major disadvantage is the space complexity of the programme increases exponentially along with the dimension numbers of vector increases.

3.3.5 Methods Using Bayesian Network Classifiers

The Bayesian network classifiers are more and more widely implemented in different aspects of literature due to their strong arguments in mathematical basis (Friedman, Geiger, & Goldszmidt 1997). Early in 1995, the Bayesian network classifiers were considered to have a promising future in biomedicine and biomedical research (Erb 1995).

In this paragraph, the BAYESPROT project is selected to be introduced as a representative of applications that uses Bayesian network classifiers. The BAYESPROT applied a Tree-Augmented naïve Bayesian (TAN) classifier to carry out the task. The method first extracted attributes and transformed them into features, namely composition, secondary structure propensities, hydrophobicity, polarity, polarizability, and van der Waals volume. These features are treated separately, because some of them are believed to be more important than others. Two TAN classifiers were then constructed for composition and secondary structure propensities respectively, and all the rest of the features were input into the third TAN classifier with the same weights assigned in. This characteristic is considered as an advantage of this method. The method adopted two datasets to validate its performance and efficiency; Dataset I (Dubchak et al. 1999) and Dataset II (Markowetz, Edler, & Vingron 2003). The overall infrastructure of the processing training set and predicting the testing set is presented graphically in the following diagram (Figure 3-2):
Chapter 3: Protein Secondary Structure Prediction

There is a total of three copies of data which are fed into three Tree-augmented naïve Bayesian network classifiers, and the outputs of these classifiers are processed by Mean Probability Voting module, and averaged to produce the final result.

The internal correlation between attributes and class is presented as a typical Tree-Augmented Naïve Bayesian network classifier in Figure 3-3:

![Figure 3-3: TAN Bayesian Classifier (Chinnasamy, Sung, & Mittal 2004).](image)

The BAYESPROT method reported their prediction accuracies of protein secondary structure were at least as good as other methods. It employed fewer classifiers than methods using support vector machine classifiers, but reported similar prediction accuracy. The process of classification is much faster than other methods, which have similar performances.
3.4 The Measurement of Prediction Accuracy

In order to be able to validate the performance of prediction methods, a common standard of measurement of prediction accuracies has to be proposed. At the moment, there are various measurements, which have been applied in the literature, and so far it is not clear which measurement is the best. The most popular measurement is per-residue three-state accuracy, and per-segment overlap accuracy. In the following paragraph, not only these two commonly used prediction accuracies are introduced, but also the cross validation and accuracy matrix that have been brought into this thesis are introduced. The latter measurement of prediction is a building in component of the WEKA package, evaluating the prediction performances performed by the WEKA classifiers. WEKA provides a set of various choices of machine learning schemes that are adopted in this thesis to perform the task of learning and testing. In recent years the WEKA package more and more often used in the bioinformatics area (Frank et al. 2004).

3.4.1 Per-residue Three-State Accuracy

The per-residue three-state accuracy is very straightforward, simple, and effective. In a prediction, there is a calculation of prediction accuracy for each single conformational state, in order to indicate the correctness of prediction in that particular state $i$:

$$Q_i = \frac{\text{number of residues correctly predicted in state } i}{\text{number of residues observed in state } i} \times 100,$$

A $Q_{\text{helix}}$ indicated the percentage of correctly predicted amino acids in helix. The three-state per-residue prediction accuracy describes the overall percentage of correctly predicted amino acid residues along the polypeptide chain:

$$Q_3 = \frac{\text{number of residues correctly predicted}}{\text{number of residues observed}} \times 100$$
number of all residues

There is a general list of $Q_3$ scores presented in (Jenny & Benner 1994) and they are evaluated. Criticize to this measurement said that the results that $Q_3$ may be misleading because they do not reflect the nature of three-dimensional protein structure very well (Zemla et al. 1999). In order to evaluate the prediction results in a more structurally meaningful sense, the Segment Overlap Measurement (SOV) is developed.

### 3.4.2 Per-Segment Overlap Accuracy (SOV)

The SOV measurement was first proposed in (Rost, Sander, & Schneider 1994b). The SOV measurement is calculated based on secondary structure segments rather than individual residues. The paper indicated that the ultimate goal was to predict reliably the tertiary structure of a protein, not 100% single residue accuracy for secondary structure; it is sufficient to predict the approximate location of helix, strand, turns and loop segments.

The SOV measurement is developed to be insensitive to small variations in secondary structure assignments, and therefore provided an estimate about the segment structural information, which are more valuable for later alignments in searching folds or searching remotely homologous candidates. The overlap is defined as:

\[
SOV = \frac{1}{N_i} \sum_{S_i} \frac{\text{MINOV}(S_i; S_2) + \text{DELTA}(S_i; S_2)}{\text{MAXOV}(S_i; S_2)}
\]  

With the following definition:

- $S_1$ and $S_2$ are the observed and predicted secondary structure segments;
- \(\text{MINOV}(S, T)\) is the length of actual overlap of $S$ and $T$, i.e. the extent to which both segments have residues in state $i$;
- \(\text{MAXOV}(S, T)\) is the length of the total extending for which either of the segments $S$ and $T$ has a residue in state $i$;
- \(\text{DELTA}(S, T)\) is the integer value defined as being equal to the following:
\[
\text{DELTA}(S, T) = \min \begin{cases} 
\frac{\text{MAXOV}(S, T) - \text{MINOV}(S, T)}{\text{MINOV}(S, T)} \\
\text{INT}(0.5, \text{LEN}(S)) \\
\text{INT}(0.5, \text{LEN}(T))
\end{cases}
\]

3-2

\( N(i) \) is the number of residues in state \( i \) defined as follows:

\[
N_i = \sum_{S(i)} \text{LEN}(S) + \sum_{S'(i)} \text{LEN}(S)
\]

3-3

\( S(i) \) is the number of all the pairs of segments (Abola, Kuhn, Earnest, & Stevens 2000) where \( S \) and \( T \) have at least one residue in state \( i \) in common.

\( S'(i) \) is the number of segments \( S \) that do not produce any segment pair.

The all – three – states quality measure is:

\[
SOV = \frac{1}{N} \times \sum_{S(i)} \frac{\text{MINOV}(S_i; S_j) + \text{DELTA}(S_i; S_j)}{\text{MAXOV}(S_i; S_j)} \times \text{LEN}(S_i)
\]

3-4

With:

\[
N = \sum_i N_i
\]

Various scoring functions can be selected to measure the accuracy.

### 3.5 Cross-Validation and Confusion Matrix

The prediction accuracies in this thesis are evaluated by the embedded functionality of WEKA package. In applying WEKA functions, enable cross validation for an experiment is simple but remarkably effective. The data are divided into \( N \) partitions (\( N \) is determined by the users’ request, for instance in 10-folds cross validation, \( N = 10 \)), and for each turn one partition of the data is selected to be testing dataset, the rests are used as the training set. For a \( N \)-division dataset, the experiments are carried out \( N \) times. The prediction accuracies are averaged and the result is reported as \( N \)-cross validation prediction accuracy. The cross validation is able to present the general quality of the dataset. In the
Chapter 3: Protein Secondary Structure Prediction

extreme situation, one record is taken as the testing dataset, and all the rest are used as training set. This is called jack-knifing cross validation.

The advantages of the cross validation techniques are significant. By applying the cross validation techniques, all data are effectively treated as testing dataset. The sizes of training and testing dataset are configurable. Also, it is believed that applying cross validation is beneficial while evaluating the quality of dataset.

The disadvantages of the cross validation are also obvious. First of all, the computational costs are raised along with the increase of number N. Along with the increases of number N, the computational cost increases enormously. A commonly used N value is the number of 10, considering both prediction accuracies and the effectiveness of the program. In this research all the experiments are examined with the employment of 10-folds cross validation.

The confusion matrix is very important for validating classifiers, as they display not only the correctly predicted instances but also display the incorrectly predicted instances. To evaluate the performances of two classifiers, not only are the prediction accuracies examined, but also false-negative scenarios.

3.6 Conclusion

Protein secondary structure prediction is a highly active and fascinating research field in protein science. What could be done in this chapter is to capture a small portion of the state of the arts, people have various approaches to either increasing prediction accuracies, or make the available results more meaningful. The advances of protein structural prediction can bring us promising future.

In this thesis, the author develops a distributed computational environment, to provide better prediction accuracies, and also make the results meaningful. The next chapter introduces the software package – Protein Attributes Microtuning System (PAMS), which is able to produce training and testing datasets as users demanded.
Chapter 4. Protein Attributes Microtuning System

4 Protein Attributes Microtuning System

The preparation of the training and testing datasets is one of the essential procedures of a computational methodology for predicting protein secondary structure. To evaluate the prediction accuracies of different experiments, the training and testing datasets have to be well designed and prepared in a common standard, to avoid variation of prediction accuracy brought about by noise. In a supervised learning process the training datasets include various sets of numerical attributes which represent characteristics of the amino acid, and its corresponding secondary structure unit. The characteristics are believed to have an impact on the secondary structure unit. The testing sets contain the amino acid’s identity, and the numerical representations of corresponding characteristics. The learning network is then able to predict the secondary structure type of the amino acid according to the input testing set. The prediction accuracy relies not only on the performance of the machine learning classifiers used, and the learning architecture of the classifiers, but also on the quality of provided training datasets. In this thesis, the author has improved the prediction accuracies by providing better qualified training datasets.

The phrase of “providing better qualified training datasets” is a general term. In detail it presents behaviours such as “providing appropriate protein samples”, “providing appropriate physico-chemical attributes”, “capturing correlations between these attributes and secondary structure types” etc. These improvements will be described in subsequent chapters; in this chapter a software package, which is able to deliver datasets in special requests, is proposed.

This chapter comprises the discussions of general introduction, followed by a brief illustration of the software architecture in the first section; the section two describes the characteristics of the PAMS, and the benefits that can be obtained from it; Section three contains the system analysis and design of the software package, also includes the introduction to the development environment; the section also describes the evaluation of the PAMS. The author’s conclusions are drawn in section five.
4.1 General Introduction of the PAMS

The initial design purpose of the PAMS is to make it capable of generating the datasets from bits and pieces of information that it has gathered from the users. This information includes the sets of proteins that the users want to apply as the training data; the sets of amino acid feature descriptors that the users believe to influence the secondary structure, etc. The PAMS software package can run as independent software with a graphic user interface under X environment, or as a software function. While the PAMS is activated as individual software, the configuration of datasets⁵ are input from the interface; while the PAMS is used in another programme the configurations are transmitted through parameters. The following diagram (Figure 4-1) indicates the basic data flow within the PAMS after it has received the configurations:

![Diagram of PAMS structure](image)

**Figure 4-1: Structure of the PAMS**

---

⁵ In this thesis, the configuration of datasets represents the users choices of generating a dataset, includes the selection of protein samples, the selection of feature descriptors, the length of window frame, the weight variation model, whether the weights require to be normalized, etc.
The database storage warehouses various biological samples, including the samples of protein sequences and structures, the samples of amino acid feature descriptors, the protein identities set of certain characteristic, such as non-homologous etc. the jBNC+WEKA package provides various choices of machine learning classifiers to be applied in the generated training and testing datasets. Obviously if the datasets produced do indeed satisfy the users and also are correct (exactly as the users wanted), then users will be happy and have smiles on their faces.

The four modules framed by the dashed line are the main functioning modules of the PAMS. The DB handler retrieves the requested biological samples from the data storage device according to the specification listed in the configuration, and sends them into the window frame module. The module places the sequence into a frame, assigns each position a weight. The weights represent the impact to the amino acid in the central position. The greater the value the heavier is the impact. The win handler then passes the frame to the Feature calculus, the feature calculus sums up the numerical value from each individual feature value times its weight value. The PAMS output each training and testing set in ARFF\(^6\) format, for the convenience that the generated datasets could be processed by WEKA classifier directly.

### 4.2 Characteristics of the PAMS

The characteristics of the PAMS represent the outcome of the design process. On the one hand, the PAMS is designed to process the biological data samples according to a user’s will, therefore it is required to be 100% accurate in processing those biological data. Also the PAMS is required to carry out a user’s demands precisely, there must have no alternation to the user’s demands. On the other hand, the PAMS outputs must be acceptable by the WEKA software package. In general, the characteristics of the PAMS could be summarised as:

\(^6\) ARFF file format is designed to provide as much information as possible to the WEKA classifiers package; therefore a valid classification could be performed.
4.2.1 Reliability of the PAMS

The reliability of the PAMS is defined in a hierarchical architecture and this spirit is applied in the whole process of development. The reliability of the biological dataset is the top priority. The PAMS is allowed to stop generating outputs, but not allowed to generate error outputs. In order to do so, on one hand, the exception capturing mechanism has been applied maximally; on the other hand the process of calling a functioning module must follow after the process of parameter type checking. The purpose of type checking is to make sure that all biological data samples are successfully retrieved from the database, and they are in the correct position. Apart from the type checking mechanism, the PAMS also applies a privilege system, which defines that only those functional modules that are allowed to modify the database of biological samples, when required. The access to the generated outputs is also limited to the corresponding modules. The implementation of this privilege system ensures the reliability of outputs, even while the PAMS is required to be extended in taking more functionalities.

There may be other exceptions, for example the failure of database modules, or if one of the modules is killed by the host root. In order to avoid error data generation, there is a test-alive mechanism within the PAMS. Once one of the required modules is absent, the PAMS records its current status and may give reasons for shutting down, the current position if the PAMS is running in an unsupervised mode, and other relevant information; then the PAMS shuts itself down.

The running of the PAMS requires a large amount of system memory, and the robustness of the programme is ensured by the Java program itself. It is trivial to study the minimum requirement of the usage of the system memory, for convenience all launches of the PAMS software package with the graphic user interface enabled takes 256M memory at the start, ensure that the programme does not crash due to a lack of memory.
4.2.2 Compatibility of the PAMS

The PAMS is written in the Java\textsuperscript{7} language. Applications written in Java could be executed in any platforms which have a Java virtual machine (JVM) built in. The graphic user interface requires the X library and therefore needs the X library to be supported. If the PAMS is called as a Java class in another programme, then the X library is not required because the configurations are transmitted to the PAMS by parameters.

4.2.3 Usability of the PAMS

For a programmer, there is a list of documentation to support the usage of the PAMS. The parameters are introduced in format requirements and their content. As long as the parameters are sent in correctly, the PAMS class is able to output the ARFF file correctly.

The PAMS may be launched from the command prompt. The programme will check itself to see whether the required components such as the connection to the database, the creating of matrices in buffer, the privilege to write to local file system, etc. Once everything is ready, the user is able to use the interface to generate the datasets that he wants.

4.3 System Analysis and Design of the PAMS

In this section, the author reviews the process of system analysis and design in the past, and records the development procedures of the PAMS. The PAMS was developed by applying several modern implementation techniques in IT, hence although it is complicated, it did not take a long time to finish. In this section, the technical background is introduced, especially the implementations that were adopted in the PAMS. Furthermore the available biological data resources are introduced; the WEKA acceptable data format is also introduced in a later section.

\textsuperscript{7}Java is an object-oriented programming language. It provides a large amount of classes to be used to fulfil the programmer's designing purpose. It is also considered as effective by applying the multi-threading programming.
4.3.1 The Development Platform of PAMS

The PAMS was developed in the latest Fedora Core 4 Linux platform (current release version is Fedora Core 6). Fedora Core, as the central Fedora project, is sponsored by Red Hat and guided by the Fedora foundation. Since 1998 there have been more than 200 applications developed in the Linux and Linux clusters platforms to support biomedical or biomedicine research activities, due to its nature in distributing computing, parallel computing, network service providing, etc. Furthermore the Linux operating system is particular favourable to the bioinformaticians (Dewar et al. 2005; Kostin & Mokhov 2005; Landsteiner, Olson, & Rutherford 2005; Lau et al. 2006; Lehtovuori & Nyronen 2006; Moll et al. 2005; Smith, Chandonia, & Brenner 2006; Tovchigrechko & Vakser 2006). By applying Linux clusters the research institute is able to achieve huge computational capacities for relatively small cost. The condor project proposed a software system that allows users to share computing resources between different computer platforms, this has also been increasingly applied in the bioinformatics research area in recent years (Dowsey, Dunn, & Yang 2004; Fritschy et al. 2005; McGuffin et al. 2006; Reddy et al. 2003; Sulakhe et al. 2005; Swain et al. 2005).

Aside from this, the author particularly favours the use of the Fedora Core operating system for the following reasons:

**It is an economic choice**

It has been guaranteed that the Fedora Core operation system is will always free for anyone to use, modify, and distribute, now and forever, thus the budget can be saved to do more things.

**It is supported well, and naturally supports open source software**

The Fedora Core operating system is well supported by unlimited fans, scientists, and technicians. There are many technical forums available to discuss problems that have occurred in the process of administration and development, etc. These forums cover
almost all aspects needed by the author in the development process. At the same time, the Fedora Core operating system supports open source software well, and recently the open source software community has been catching more and more people’s attention. A considerable amount of open source software has been implemented in the bioinformatics research (Arshinoff et al. 2006; Cathelin, Lopez, & Klopp 2006; Kerrien et al. 2006; Pavy et al. 2006; Ptitsyn, Zvonic, & Gimble 2006; Schlueter et al. 2006; Sucurovic 2006; Toyoda et al. 2006; Xu, Li, & Kong 2005).

It is a secure computational platform

The Fedora Core operating system is considered to be an active leader in implementing the most up to date security initiatives. The Fedora Core operating system is itself an ideal choice as a network service provider without being vulnerable to the cyberspace intruder. Apart from the strength of security that is managed by the operating system, the software is updated by the RPM and yum utilities in a comfortable way to overcome newly discovered vulnerabilities. These facts strengthen the author’s confidence in using the Fedora Core operating system to execute the development and furthermore carry out experiments on it.

It is a reliable computational platform

The Fedora Core operating system is a reliable developing platform. Without fatal hardware conflicts the system usually does not crash. The Fedora Core system is able to work continuously for quite a long time (167 days in odin.chem.surrey.ac.uk).

---

8 RPM, Redhat Package Management tool, is a good way to manage the installation of various software, and avoiding the conflicts of libraries.

9 Yum is an automatic updater and package installer/remover for rpm systems. It automatically computes the dependencies and figure out what things should occur to install packages. The official website of the yum package is available in http://linux.duke.edu/projects/yum/.
4.3.2 The Modern Techniques that PAMS has Implemented

The PAMS was written in Java, and used the MySQL database as data storage device. In this section, the author reveals why these choices were made.

The PAMS was developed by Java

Java is now possibly the most popular programming language around the world. What makes the Java so popular? First of all, Java is able to overcome the problem of incompatibility. Applications written in Java can be executed anywhere and have JVM support. At the same time, there is a wide range of application samples available online to which users may refer. Secondly, the common problem of buffer overflow, which exists in the C code, does not appear in the Java code. The abandonment of pointers in Java prevents the risk of the code being exploited. The compiled java code is always executed in Java virtual machine, which is limited to a certain area in memory. The memory recycling mechanism of Java relieves coders from the burden of memory manipulation, such as set free the allocated memory. A potential concern about the memory recycling mechanism is that it may affect program performance, for the extra resources used to remove objects occupy CPU times. Fortunately, good memory recycling mechanism has been developed in the past. In most cases, the garbage collector runs in a transparent thread, which is not noticed during the application execution. Further reading about Java’s memory recycling mechanism could be obtained from the famous book “Thinking in Java”.

The JFC/Swing library is used to create the graphical user interface in the X window environment, and this may be researched in the online tutorial (Swing, 2006).

The MySQL database is used as data storage device

The MySQL database was used in this research as a general data storage device for only the biological data resources, but also the later experiments results. The MySQL database, as one of the biggest accepted open source multi relational database systems, is being increasingly accepted both in academia and industry, according to its remarkable performance and reliability. It is implemented often in the bioinformatics research area.
because there are large amounts of data to be processed and managed. The latest applications to have applied the MySQL database in the literature can be found in (Johnson et al. 2006; Monge et al. 2006; Nahum et al. 2006; Titulaer et al. 2006; Wu et al. 2006b); obviously there are many more. In this application, available databases such as PDB_SELECT, AAIndex were pre-parsed and stored in the MySQL database. The MySQL Connector/J provided the connections between for Java code and the MySQL database.

4.3.3 The Biological Resources That the PAMS Has Adopted

The PAMS applied several protein sequence data resources and structural data resources to perform its job. The PDB_SELECT database (Hobohm et al. 1992; Hobohm & Sander 1994; Westbrook et al. 2002b) defines a list of representative protein entries from the protein data bank (PDB) database (Westbrook et al. 2002a). The PDB_FIND2 database defines a summary of PDB, DSSP and HSSP information with some added information, provides an easier and better organized source (Hooft et al. 1996). The PAMS cleaned these resources (missing values; chain breaks etc), and deposited them into the database. Apart from that, selections of protein identities such as the RS126 dataset, CB396 dataset (Cuff & Barton 1999), and CB513 dataset defined were also deposited into the database. In the following section these databases are introduced.

Protein identities selection

There are several reputable datasets available in the literature, such as the RS126 dataset. The latter contains 126 non-redundant polypeptide chains, of which no pairs of proteins share more than 25% of sequence similarity over a length of more than 80 residues. The downloaded dataset contains protein identities, protein sequences, DSSP secondary structure definitions, and other relevant prediction results from methods such as PHD (Rost 1996b), JPred (Cuff et al. 1998) etc. In the PAMS implementation, the following two lines of sequence and DSSP entries were used:
Figure 4-2: The sample of protein: 2ilb was taken from the RS126 dataset.

While processing this dataset, the "-" is replaced by the secondary structure type "C".

Another well recognized dataset in use is CB396 which was originally created by Cuff and Barton at the European Bioinformatics Institute (EBI) in 1999. Within this dataset there are 396 non-homologous protein chains. The CB series datasets include another family of 406 protein chains and a further bigger dataset with 513 protein chains. Different from the above RS126 dataset, no eight to three state reduction method has been applied to the data. The dataset looks like:

Figure 4-3: The sample of protein: 1bnc was taken from the CB396 dataset.

While this dataset was used an eight to three DSSP simplification was applied to all the DSSP entries to convert "_" to "C", and "B" to "H", "S" to "C", etc.
Combination of the PDB\_FINDER2 and PDB\_SELECT25

As of December, 2006 the latest PDB\_SELECT25 contains 3080 polypeptide chains with 459963 amino acid residues. Proteins within it share less than 25% identity similarity. Combining PDB\_FINDER2 and PDB\_SELECT25 yielded 3080 protein chains deposited in the database with their sequences and DSSP secondary structures. This set was considered to be the supreme collection of protein sequences and secondary structures that were used in this research. Along with the updates of PDB\_FINDER2 and PDB\_SELECT25, this supreme collection also updates.

The above two define the index and content of protein secondary structures and sequences. By selecting different indices the PAMS was able to make use of different sets of protein entries to generate the demanded training and testing datasets.

AAIndex database provides feature descriptors for amino acids

The AAIndex (Kawashima, Ogata, & Kanehisa 1999) is published by the Genome Net project in Japan, contains 516 amino acid indices at the moment. Each amino acid index defines a set of twenty numerical values, describing a particular feature of these amino acids. The AAIndex database has kept growing in the past decade, and all these indices are collected from the literature. This database was pre-parsed and deposited in the database. Along with the update of AAIndex, the database that PAMS used also will be updated. By adopting this database, the PAMS is able to choose amino acid feature descriptors from a collection collected from diverse research area.

4.3.4 The Machine Learning Classifiers Package that the PAMS has Applied

The machine learning classifiers package that the PAMS used in this research is a combination of the jBNC (Sacha, Goodenday, & Cios 2002) and the WEKA practical machine learning system.

In 1999, Sacha developed a new synthesis of Bayesian Network Classifiers and Cardiac SPECT Image Interpretation. Along with the publication of this thesis, a Java toolkit was
developed for implementing Bayesian network classifiers, including Naïve Bayes, TAN (Tree-Augmented Naïve Bayesian network), FAN (Forest-Augmented Naïve Bayesian network), STAN (Selective Tree-Augmented Naïve Bayesian network), STAND (Selective Tree-Augmented Naïve Bayesian network with Node-Discarding), SFAN (Selective Forest-Augmented Naïve Bayesian network), and SFAND (Selective Forest-Augmented Naïve Bayesian network). The jBNC provided a quick access for this thesis to implement these Bayesian network classifiers.

WEKA is a collection of machine learning schemes that are coded in Java. The implementation includes diverse choices of machine learning classifiers from the simplest one-level decision tree oneR to Bayesian network classifiers, Support Vector machine Classifiers etc. The encapsulated algorithms can either be launched in the command line as an alone stand program, or be called directly in users' java code. The jBNC is especially good at providing Bayesian network classifiers, and the WEKA is providing the rests, therefore using the jBNC_WEKA package, which is proposed by the jBNC project, to bind the jBNC with the WEKA package, hence obtained a general set of machine learning classifiers basically cover all options. Apart from providing diverse choices of machine learning algorithms the WEKA package also provides procedures for data processing and visualization. For example, the process of data discretization is applied to all the experiments in this research. The process of discretizing turns the numerical attributes into nominal attributes and hence increases the classification efficiency enormously.

The ARFF format

WEKA acceptable data have to be formatted into a specified ARFF format before being fed in. The ARFF format is a specified data format, designed to provide enough information to the classifiers for designated classification tasks. The format of ARFF file has two distinct sections: the Header information zone and the Data information zone.

The following diagram (Figure 4-4) contains the first part of required information of the dataset: the name of the dataset and the comments, which provides the user further more information about the dataset:
Chapter 4. Protein Attributes Microtuning System

Figure 4-4: ARFF header information section, part one. (Professor David Povey now)

Any line starts with "%" is a comment. The @RELATION declaration declares the name of this data file and, by ignoring the comments; it is the first line in the file.

The second part of the header information contains a list of @ATTRIBUTE statements, declaring the name of the attributes and their data type. The data type of the attributes can be any of the following four types, which currently supported by the WEKA package: Numeric; Nominal; String; Date. The keywords of numeric, nominal, string, date are case insensitive, as represented in following diagram (Figure 4-5).

If an attribute is declared as numeric, then the WEKA classifiers will be able to recognize the attribute in real type or integer type. The WEKA will report no mistakes unless the attribute declared as numeric is in fact not numeric. The nominal attributes are defined with a list of declared possible values {'nom-value1', 'nom-value2', 'nom-value3'...} following the attributes name of "aa1D" or "class". In the above diagram the "class" is a typical nominal attribute, with possible values {C, H, E}. The String attributes allow users to create arbitrary textural values. The Date attribute allows the WEKA classifiers
to recognize and parsing date values. The default format takes the ISO-8601 combined
data and time format: "yyyy-MM-dd'T'HH:mm:ss".

As in the above case, there are two types of the attributes listed:

- the “aalD” and the last “class” are nominal data types, with the following “{ }”
  section declaring possible values of the attribute;
- the attributes of BUNA790101, FASG760104, FAUJ880113 are taken from the
  AAIndex database according to the amino acid’s identity, and all three data types are
  numeric.

The @DATA section of the ARFF file looks like the following diagram (Figure 4-6):

```
@DATA
H, 106.687374, 125.59448, 54.771004, E
F, 106.033005, 125.584, 54.873497, E
V, 105.89465, 105.06399, 55.031742, E
R, 104.731705, 125.479004, 55.072002, E
C, 104.08006, 125.47625, 55.016003, E
```

Figure 4-6: ARFF data section.

The “@DATA” is a single line denoting the start of the data section. Each line in the
following part represents an instance with a list of attribute values, and the last entry of
the line denotes the class. The carriage returns denote the end of the instance and attribute
values for each instance are delimited by commas. Taking the first line after the
“@DATA” declaration in the above diagram as an example, the first column declares that
the amino acid’s identity area “aalD” is valued “H”; the BUNA790101 attribute value is
106.687374; the FASG760104 attribute value is 125.59448; and the FAUJ880113
attribute is 54.771004. The final column of the line denoting the class is the “E”, which

10 These identities are defined by the authors of AAINDEX database.
means “randomly coil”. By generating an ARFF file format directly from the PAMS, the efficiency of training and testing is improved.

4.4 The Graphical User Interface of the PAMS

The GUI of the PAMS was developed within an X environment in the Fedora Core operating system that was implemented in this research. The GUI contains five tabs for users to make their decisions, and in the last tab generating the designated dataset files in ARFF format. The programme started with a few resources check, such as figure out whether the database device was activated, whether the corresponding biological database were alive and so on. The PAMS reported error message if any of these required resources are not activated, and showed the following interface while there were no mistakes occurring (Figure 4-7):

![Figure 4-7: The PAMS, data set selection page.](image-url)
In the above diagram, the users are allowed to select the collections of protein identities, e.g. RS126, CB396, CB406, and CB513, or the pre-selected dataset from the collection of PDB_SELECT25. The “additional dataset three” option is an empty slot, allows more datasets to be plugged into the system. When the users click the option of the dataset, a brief description of the current dataset is displayed in the right text box. The left frame can be extended and scrolled up and down to contain more data set options if more data sets are inserted into the PAMS system.

The second diagram (Figure 4-8) displays the options for user to select the supported attributes. There is a total of 506 attributes warehoused in the database device (although 516 attributes available, 10 of them are omitted because they contain undetermined values), for the PAMS programme to access. For the consideration of efficiency, the PAMS retrieved these data at the start of the programme launch, and then imaged these data into a multiple dimensional matrix in the memory. Hence the subsequent demands of these data will be directed to the memory, instead of accessing the database again.

![Figure 4-8: The PAMS, attributes selection I.](image)
In the right hand column the blank text area is designed for displaying the information relevant to the particular feature descriptor, which is selected in the left hand side column. In the left hand frame, there are total 6 categories of feature descriptors, which are displayed. The categorizations of these feature descriptors are defined by the AAIndex database. These descriptors describe the propensities of the amino acids to form α-helices, turns, or to fold into β-sheets. The remaining "composition", "hydrophobicities", "physicochemical properties" and "other properties" categories contain the feature descriptors presenting composition characteristics, hydrophobicity, physicochemical properties and other relevant properties respectively. The classification of these categories is based on the literature review that has been performed by the people that proposed the AAIndex database. The users can select multiple feature descriptors by clicking the button in the bottom of right hand side of column sequentially. Multiple clicks to one single feature descriptors will only be recorded once. The selection operation was performed in following diagram (Figure 4-9):
If the user clicks five feature descriptors in left hand side of the column, then there will have five feature descriptors applied in the process of generating the ARFF format training and testing datasets. The spirit of using this GUI of the PAMS is to provide a convenient and intuitive method for chemists to place their hypothesis of “whether this feature descriptor has impacts to the accuracies of protein secondary structure prediction? Whether the current protein structural data can validate these impacts?” in this software package, then validate these impacts in later on learning and predicting processes.

There are four options listed in the tab of configuring window mechanism in that particular process of dataset generation. The selected length of the window size tells the PAMS software that this time the author want to incorporate that certain number of amino acids are taken into account to calculate propensities to secondary structure type. In this thesis there is a global optimal window length of 21 (according to our experiments, the 21 is optimal for almost all classifiers). The second option is used to determine the relative weight of the positions in window edges compared to the window central position. In this research the default weights of the positions in two window edges are set to 0.1. The weight of central position is set to 1 by default. By selecting either linear or exponential weight variation models in the third option, either the linear or exponential weight variation model is applied to the window frame to calculate the weight distribution over it. Currently the linear and exponential weight variation models are supported, and a short time after this thesis was published, the Gaussian weight variation model was also adopted in this system. For the purpose of comparing different types of weight variation models, the normalization of the weight scores is also proposed. The overall GUI is looks like the follow diagram of (Figure 4-10):
The PAMS is developed to interact with a database device to exchange data. The MySQL connection tab is designed for users to allow connection to another database. There are a few schemes embedded in this tab, such as the building up of the database, checking the table, optimize the table, etc. The PAMS is portable and can be installed into a laptop with Fedora Core Linux operating system. Then the users can carry the laptop around to performing the tasks of generating training and testing datasets anywhere (s)he wants. The interface of the tab is displayed as the follow diagram (Figure 4-11):
Continually using the PAMS package will generate vast amounts of training and testing datasets for protein structure prediction. It is acceptable for these files to be recorded in the local file system if there are not too many. But along with the time the manipulation of these training and testing datasets can be a headache. The PAMS also allows the users to deposit the files into the MySQL dataset, along with the date, purposes, and configurations of those datasets. Using the MySQL is one of the advantages of this research owing to its powerful facilities for managing data.

The final tab of the PAMS package (Figure 4-12) shows the menu for displaying configurations and generating the designated datasets. In the bottom there are two buttons: left “refresh” and right “Ready? Go!”. The left button refresh the above page, print out the configurations for users to review. If the user feels unhappy with a particular feature descriptor, (s)he could go back and deselect that particular feature descriptor, then come back to press the “Ready? Go!” button. The Go button is clicked to start the generation process.
Chapter 4. Protein Attributes Microtuning System

4.5 The Introduction to Modules’ Functionality

The PAMS was developed by defining modules and later on combining these modules together. Each module is a class, which encapsulates functionalities, properties, within it. By create an object inherit this class; the object is able to perform the functions that defined in the class. In the PAMS, there are two handlers manipulate certain resources, and three functional modules.

Figure 4-12: The PAMS, generating the target datasets.

The implementation of the GUI of the PAMS package brought convenience for the author to conduct further experiments. The process of preparing the datasets used to take large amount of time and labours, but now can be finished in seconds. In the following paragraph, the functionalities of the modules in the PAMS are illustrated.
4.5.1 The DB Handler Manages the Database

The DB Handler defines a list of SQL (Standard Query Language) functions that manipulate the relevant biological data resources such as retrieve protein sequences and secondary structure, retrieve corresponding feature descriptor for a particular amino acid, etc. A MySQL Connector/J was applied, to convert the JDBC (Java Database Connectivity) calls into the network protocol used by the MySQL database. The DB handler receives the user’s configurations and transfers them into corresponding SQL enquiries. The sequences that retrieved from the database are handed to the window frame functioning module, and assigned weights.

The DB handler works in two directions. Not only can the data can be retrieved from the database, but the generated datasets can also be deposited into the database.

4.5.2 The Window Frame Module Turns Sequences into Segments

The data were sent by the DB handler to the window frame module. This module performs the functionalities of mapping segments of sequence into the window frame and assign weights to each position of this frame. The implementation of the window mechanism improves the accuracies of protein secondary structure predictions because not only the characteristics of the corresponding amino acid is considered, but also the characteristics of the neighbouring amino acids within the frame are also taken account in with different priorities. A graphic illustration of the window mechanism is presented in following diagram Figure 4-13):

```
... Q C T L R D S Q Q K S L N ...
```

Figure 4-13: A window frame, with the length of 11.
In the above diagram, the coloured segment represents a selected window section of 11 amino acids. The central amino acid of “S” in red is the key amino acid, and the neighbouring amino acids are believed to contribute to the secondary structure type of the “S”.

When the operation to the amino acid “S” is finished, the window slides one position to the right direction and starts to apply the same operation, taking the amino acid “Q” as in the central position, as in the following diagram (Figure 4-14):

![Figure 4-14: Sliding the window one position to the right.](image)

After calculating the value of the amino acid “Q” the window frame slides further to the right head direction to map the next amino acid “Q”. By repetition, the window will map all the amino acids along in the polypeptide chains. The window scheme helped in improving the prediction accuracy of protein secondary structure. There are two factors needed to be determined to determine a window scheme: the decision of the optimal window length and the determination of the weight variation model.

### 4.5.2.1 Deciding an Optimal Window Length

In the PAMS application with a GUI the window length is allowed to vary from 5 to 25. Subsequent research suggests that the optimal window length for this research is 21. A detailed description of the experiments show that 21 is the optimal window length is presented in the chapter 6. In the PAMS application without a GUI the window length is allowed to vary from 3 to 51, and the window length is transmitted by parameters.
4.5.2.2 Determining the Window Weight Variation Model

The amino acids within the window frame need to be weighted before sending to the calculus. At the moment the PAMS package adopts two weight variation models to assigning these weight values according to the position to central position. Usually, the closer the position is to the central position, the heavier the value is weighted. Assuming that there is a window frame with the length 11, then to each position a weight is assigned, as represented in the following diagram (Figure 4-15):

Assuming that the character within a series of boxes represents different amino acid residues along with a polypeptide chain, then the value of \{W_1, W_2, \ldots, W_N\} is the weight value assigned to them, according to the distances between the amino acids to the central amino acid “S”. In this window frame there are N=11 amino acids. \(W_{(N-1)/2}\) represents the weight value of the central amino acid, usually valued as 1; \(W_{((N-1)/2)\pm1}\) represents the weight values of the two amino acids next to the centre amino acid; \(W_{((N-1)/2)\pm2}\) values are assigned to represent the weight values of the two amino acids in the second position on both sides. The value of \(W_1, W_N\) represents the weight values of amino acids in both edges of the window frame. Two weight variation methods are principally used to determine the variation of weight value: the linear weight variation method and the exponential weight variation method. Then the linear weight variation function is presented as following:

\[
W_i = W_{(N-1)/2} - \frac{2 \times (N-1) / 2 - i \times (W_{(N-1)/2} - W_1)}{N-1}, \quad \text{where } i \text{ is the position of amino acid.}
\]
Chapter 4. Protein Attributes Microtuning System

The mathematical form of this function is plotted in (Figure 4-16):

![Figure 4-16: Linear weight variation method when N = 11.](image)

The weight values of this model are assigned as following (Table 4-1):

<table>
<thead>
<tr>
<th>W_1, W_{11}</th>
<th>W_2, W_{10}</th>
<th>W_3, W_9</th>
<th>W_4, W_8</th>
<th>W_5, W_7</th>
<th>W_6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.28</td>
<td>0.46</td>
<td>0.64</td>
<td>0.82</td>
<td>1</td>
</tr>
</tbody>
</table>

Apart from the linear weight variation method, the exponential weight variation method is an alternative choice to determine the weight value. The function is presented as:

\[ W_i = e^{-\frac{(\frac{N-1}{2}) \cdot \log^i} {N-1}} \]

where \( i \) is the position of amino acid.
The diagram of the exponential function is plotted as following with the N value equal to 11 Figure 4-17:

![Figure 4-17: Exponential weight variation method when N = 11](image)

Similarly, the weight values are assigned as showing in the following table (Table 4-2):

<table>
<thead>
<tr>
<th>$W_1$</th>
<th>$W_{11}$</th>
<th>$W_2$</th>
<th>$W_{10}$</th>
<th>$W_3$</th>
<th>$W_9$</th>
<th>$W_4$</th>
<th>$W_8$</th>
<th>$W_5$</th>
<th>$W_7$</th>
<th>$W_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0.16</td>
<td>0.25</td>
<td>0.40</td>
<td>0.63</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In nature, the weight variation model is trying to capture the essence of the neighbour amino acids' impact on the central amino acid. It has not been determined which weight variation model naturally make more sense, but an initiative taste can be trying out all the variation models within the area quote of by the blue lines in diagram (Figure 4-18). Apart from that, the Gaussian variation model also can be incorporated into the PAMS framework. In order to compare the performance of different weight variation models, these assigned weight values must be normalized. That is also the reason why the PAMS provides the options to normalize the weight data.
For simplicity, the PAMS package implements the linear weight variation model and the exponential variation model and does not contain the other variation models. The adoption of other weight variation models is scheduled for post doctoral research.

4.5.3 The Feature Calculus Computes the Attribute Value for the Central

After the weight assignments are finished, the selected feature descriptors were map into the sequence frame, and labelled as \( \{V_1, ..., V_{11}\} \), according to the identity of the corresponding amino acids. The feature representative of the centre “S” was calculated by summing up amino acids' attributes multiplied by their corresponding weight values. Taking the above sequence as the example in (Figure 4-19):
The value of $V_6$ is calculated as:

$$V_6' = \sum_{i=1}^{11} W_i \times V_i$$

Using this summed value $V_6'$ is considered to make more sense naturally. Although it is believed that by applying the window mechanism is still not sufficiently good to capture the long distance interaction in the protein, but it has its basis in empirical observation that the application of a window frame improved the prediction accuracy.

### 4.5.4 The ARFF Writer Outputs the Datasets in ARFF Format

The calculated datasets are outputted in the ARFF format. The PAMS package runs a few multiple threads so that once the file is generating, the user can make another selection. The new selected configuration will be put into the queue; therefore the writer can output files one by one, without disturbing users for details in file writing.

There is additional information recorded along with the generation of this file, for example the configuration for generating this dataset, the date, the host name, etc. These messages are written in the comments section of the generated dataset files.
4.6 Dealing with the Missing Value

The problems of missing or undetermined values are unavoidable in any data mining task and so it is here. There are two major data missing scenarios in this research, and in this section the treatment to these problems is discussed.

First of all, there are undetermined values existing in the AAIndex database. At the moment there are 516 amino acid indices available in the AAIndex database and 10 indices contain missing values. There are a few techniques to deal with these missing values, such as filling them with zero, filling them with averaged values, etc. In this thesis, these values were simply omitted because they show no significant correlations to the protein secondary structure.

There are other missing values from the secondary structure of proteins. All the broken chains are removed, and the undetermined structures are replaced by the “coil”. The undetermined structures share the pattern of:

\[ \text{CCCCC?CCCCC} \]

It is quite intuitive that the “?” is replaced by the type of “C”, naturally means random coil.

4.7 Conclusion

The PAMS software package has been developed to generate training and testing datasets according to the user’s will. During the development process, the usability, robustness, reliability, and effectiveness of a software package must be considered. The PAMS software is a fundamental component for the later on Protein Structural Accuracy Reckoner (PSAR) infrastructure, which is introduced in Chapter Five.
5 Protein Structural Accuracy Reckoner

The Protein Structure Accuracy Reckoner (PSAR) is an automated framework to process users’ configurations, generate corresponding training and testing datasets and record the prediction results. In this chapter the author presents an overview to the PSAR, which has been developed to verify the hypothesis of this thesis.

5.1 Motivation behind the PSAR Development

The development of the PSAR is motivated by several reasons. First of all, the data preparation process in the past was trivial and time consuming. Also, there were not many choices in selecting feature descriptors of amino acids. The initial design purpose is to produce an automated infrastructure to verify the hypothesis of “protein secondary structure prediction increases along with the process of data purification regardless of the classifiers used”, and certain the automation saved a lot of time. The concept of automation becomes increasingly important along with the increasing of biological database volume (Chen & Murphy 2006; Friedberg 2006; Mooij et al. 2006; Takeda-Shitaka et al. 2006; Ventzki et al. 2006; Wang & Samudrala 2006). Processing this large amount of data manually becomes more and more inappropriate; this scenario also happened in the protein secondary structure prediction area. The PSAR is able to generate designated training and testing according to the user’s configuration, perform the prediction, and then record the results in the MySQL database.

The second motivation for developing the PSAR infrastructure was to find a better quality dataset, which can improve the prediction accuracies of protein secondary structure; the simplest method is to try all the datasets out.

The third motivation of developing the PSAR framework was to evaluate the performance of different machine learning schemes. There are various types of machine learning classifiers and different encoding schemes available in the literature. As they were implemented by different people, using different protein samples and different feature
descriptors of amino acids, it was difficult to compare the outputs of two algorithms in detail. The comparison of two machine learning algorithms requires for the datasets to be identical, which was not possible in the past. The PSAR provides the most up to date biological samples for users to select.

Another motivation for developing the PSAR framework was to be able to appoint the most appropriate window mechanism. The characteristics of the window frame heavily influence the prediction results. The length of the window length determines how many amino acid residues are taken into account, and the window variation model shows how much influence a neighbouring amino acid residue has on the amino acid in the central position of the window frame.

There were also other motivations such using the PSAR to verify the quality of single protein, to measure the correlation between a particular feature descriptor of amino acids and the structural characteristics of the amino acid.

5.2 Characteristics of the PSAR Infrastructure

The PSAR Infrastructure was developed with the intention of making it an automated, distributed computational platform. The PSAR is currently hosted in the Linux clusters of the Department of Computing in the University of Surrey.

The characteristics of the PSAR infrastructure are listed:

- the software is fully automated. The users must specify a set of configurations, which are used to generate training and testing datasets, or specify a variety of spatial configurations; the user then checks the results. The process of data generation, learning and prediction requires minimum human intervention;

- the PSAR infrastructure is distributed. In order to gain maximum computational capacity, the infrastructure was developed into a main job distributor, and several computational agents. The communication between job distributor and computational agents is based on network transmission. The job distributor can be deployed on a
particular machine, and the rest of the computational agents (handling the producing of training and testing datasets and the learning and predicting) can be deployed into other computational platforms such as Linux clusters. The results generated are written into the MySQL database, and by so doing the PSAR installed in the Department of Computing can have access to over 150 CPUs;

- the PSAR package is effective, as computation agents among this framework are programmed to be multiple threading. The agents will use as much resource as can be deployed in the computational system. For example, in a multiple-processors computer with eight CPUs and also hypo-threading enabled, the computation agents will spawn 14 threads to perform the calculation.

The diagram to show the distributed PSAR is shown in the following (Figure 5-1):

![Figure 5-1: Distributed PSAR infrastructure.](image-url)
5.3 Functional Modules of the PSAR Infrastructure

Three types of functional modules exist in the PSAR infrastructure, defined by the types of data they manipulate. The results-oriented modules contain the results reviewer, database, and the results capturer. In the following diagram (Figure 5-2) they are coloured red. The BioData-oriented modules contain the PAMS and the Classifiers, performing the preparation of the training and testing dataset, and the learning and predicting activity to the generated datasets. In the following diagram, these modules are coloured blue. The configuration-oriented modules include the attribute sampling module, the proteins sampling module, the window selecting module, the combination sampling modules; these modules are coloured cyan. These modules are combined to form the infrastructure of the PSAR.

The communication between modules was converted into network transmissions and by so doing the modules may be located in different computers. The user submits his/her configurations into the PSAR, and the PSAR will handle everything. (S)he only interacts
with the Structural Accuracy Reckoner module, and reviews the results from the results
reviewer. The PSAR sets a priority check for all the modules and only the BioData-
oriented modules are allowed to manipulate the data; the purpose being to avoid the
contamination of the biological data resources. Similarly, only the results capturer and
results reviewer are allowed to manipulate the results. The installation priority checking
mechanism avoids accidental damage to the data.

5.3.1 BioData-Oriented Modules

The biological data oriented modules are particular designed to manipulate biological data,
e.g. warehousing data, generating datasets, performing the prediction. Moreover, they are
emphasised by providing an extra mechanism to check the correctness and integrity of the
biological data samples.

5.3.1.1 The Database

The main part of this module is a MySQL database with a few pre-defined schemes to
perform certain tasks, such as building up the tables, optimizing the tables, analysing the
tables, etc. The Java Connector/J provides connection methods for the Java code to
interact with the MySQL database effectively. Within the PSAR infrastructure, a few
databases were defined with the installation of the system:

- AA_INDEX: this database contains 506 sets of quantitative values. Each set of
  numerical values are presents either a particular physico-chemical attribute for the 20
  amino acids, or statistical summary of amino acids propensities to secondary
  structure types. These data are updated along with the official release of the
  AAIndex@GeNome Project;

- Prot3080: the PDB_FINDER2 contains the same number of protein chains as the
  PDB databank does. The protein samples within the PDB record the three
dimensional coordinates of each atom in the protein, and the PDB_FINDER2 records
  simplified information such as amino acid sequence, DSSP, HSSP, etc. It is more
  convenient to consult the PDB_FINDER2 database instead of extracting secondary

69
structure from the PDB three dimensional coordinates of atoms. The PDB_SELECT25 contains 3080 identities of protein chains. Each pair of protein chains shares less than 25% sequence identities. The sequence and structural information of these protein chains are retrieved and deposited in the Prot3080 database;

- ProtCOL: in this database are stored the collections of protein identities, which are believed to share particular characteristics. For example, the identities in RS126 dataset, CB396 dataset, CB513 dataset, etc;

- RESULTS: all experiment results are recorded in this database, not only are the accuracy values recorded, but also the configurations of dataset, used classifiers, cost time, implementation hostname, starting time, and ending time. The latter information is recorded to adjust the infrastructure of the PSAR in order to achieve better efficiency.

The MySQL module was like a giant data warehouse, depositing and dispatching many items. Although using the MySQL database is less efficient than saving the results record in local file system, the benefits are significant. By using the MySQL database, all the results are stored in a central database system, not distributed in many computers over the network. Resource conflicts are well handled in the MySQL system hence a large number of computational agents can write to the MySQL database spontaneously.

The management of the MySQL database is straightforward, and could be performed by the implementation of PhPMyAmin project. This web-based managing tool of MySQL database turns the management of the database into be an easy task.

5.3.1.2 The PAMS

The PAMS software package in this chapter is a function which could be called the computational agents within the execution. The configuration for generating the dataset is defined by parameters. The task of the PAMS is to generate the designated training and testing dataset for the classifiers to process.
Chapter 5: Protein Structural Accuracy Reckoner

The PAMS takes the corresponding biological data from the database, therefore it is configured to have the access to the database. The configurations that are required to form datasets are listed in the following table (Table 5-1):

Table 5-1: The configuration of a particular dataset.

<table>
<thead>
<tr>
<th>Configuration member</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protein List (PL)</td>
<td>List of identities of protein candidates.</td>
</tr>
<tr>
<td>Window Length (WL)</td>
<td>The length of window frame.</td>
</tr>
<tr>
<td>Weight Variation Model (WVM)</td>
<td>Either linear or exponential.</td>
</tr>
<tr>
<td>Attributes List (AL)</td>
<td>List of feature candidates.</td>
</tr>
<tr>
<td>Machine Learning Classifier (MLC)</td>
<td>Implemented machine learning classifier.</td>
</tr>
<tr>
<td>Cross Validation (CV)</td>
<td>Specify training set and testing set respectively, or use cross-validation.</td>
</tr>
<tr>
<td>Normalization (NOR)</td>
<td>Whether the weights are normalized.</td>
</tr>
</tbody>
</table>

The variable PL indicates the list of protein chains identities that are used in a particular experiment. The WL specifies a numerical value, which defines the length of mapping window; the WVM defines either linear or exponential variation model, for assigning weights to amino acids in particular positions within the window frame; the variable AL points out a list of amino acid numerical characteristic attributes; the MLC variable shows that the type of machine learning classifier used in this experiment; the CV variable is a sign of whether use cross-validation to the whole dataset, or manually split the dataset into training set and testing set. In general, cross-validation is used in most of experiments and as a result, its output is more general and representative. The variable NOR indicates that whether the weighting values require normalization. This option is essential if the users wanted to compare the outputs of two different weight variation models. In subsequent experimental descriptions, the table of configurations will be used to present the users purpose and his definitions of variation.
5.3.1.3 The Classifiers
The classifiers module contains the machine learning classifiers defined in jBNC and WEKA package. The input of the classifier module is the generated datasets, and the output of the module is the prediction accuracy with confusion matrix. The prediction accuracies are validated by 10-folds cross validation. The WEKA package has been continually updated in past years. By updating the embedded WEKA package, the PSAR obtains the ability to implemented new machine learning classifiers.

5.3.2 Configuration-Oriented Modules
The configuration-oriented modules are designed to manipulate the configurations, in other words to interpret the user’s experimental purpose. These modules are also required to optimize the amount of configurations if the number is too big. To speed up the experiment these modules also provide suggestion based rules that are mined from pre-performed experimental results. These configurations are generated in the job distributor and then sent to allow the compute agents to be performed. To avoid the heavy traffic between the job distributor and the compute agents, the configurations are transmitted in blocks, e.g. transmitting 1000 configurations into one compute agent at a time; the results are sent back to the database directly. For some of the jobs, which have enormous configuration spaces such as the combination sampling tasks, the job distributor needs to keep a record of which configurations have already been explored and which have not.

5.3.2.1 Feature Sampling Module
The feature descriptor defined in this thesis, is the numerical value, which can be used to describe either a physico-chemical property of an amino acid, or a structural propensity of an amino acid. There are 506 feature descriptors recorded in the database and to apply all of them in each experiment is significantly time consuming. The purpose of setting up this feature sampling model is to allow the user to sample all these feature descriptors and find out which one is the good for predicting protein secondary structure. Some of the feature descriptors, which are not suitable for being used in the protein secondary structure prediction and according to the results of this module they will be omitted. The performed experiments uses the following configuration (Table 5-2):
Table 5-2: Configuration of feature sampling module.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PL</td>
<td>3080.</td>
</tr>
<tr>
<td>WL</td>
<td>21.</td>
</tr>
<tr>
<td>WVM</td>
<td>Exponential.</td>
</tr>
<tr>
<td>AL</td>
<td>Optional.</td>
</tr>
<tr>
<td>MLC</td>
<td>Bayes.TAN.</td>
</tr>
<tr>
<td>CV</td>
<td>Yes.</td>
</tr>
</tbody>
</table>

After the PSAR has performed these experiments then these feature descriptors are ranked by their corresponding performances. The module that search for the good combinations of these feature descriptors is presented in the later paragraph.

### 5.3.2.2 Protein Sampling Module

The PSAR routinely warehouses 3080 protein entries, which contain a total of nearly half a million amino acid residues. Applying all the 3080 protein samples in each test produces a dataset file about 700MB size and the application of learning and predicting process to this giant dataset is also extremely slow. The protein sampling module proposed in this thesis ranked all the 3080 protein polypeptide chains according to their qualities and length, thus producing a representative dataset, which is particularly good for protein secondary structure prediction. These protein entries represent a subset of non-homologous protein entries of the PDB data bank.

The experiment that was proposed by this module contained 3080 learning and predicting activities for each protein chain. The top ranked 33 attributes are adopted into this prediction. Among our 3080 protein chains, the minimum peptide chain length is 33, therefore a window length of 21 is taken, for the reason that both 21 is the global optimal window length, and for a peptide chain of 33 amino acid residues there are still 12 instances for training and testing, which satisfy the minimum requirement of 10-fold cross validation. This experiment applied the Tree Augmented Naïve Bayesian network classifier (TAN), because it is the fastest classifier with good prediction accuracy. The
other relevant members of configuration are: listed in below (Table 5-3):

| Table 5-3: Configuration for protein sampling module. |
|-------------|----------------|
| PL          | Optional.     |
| WL          | 21.           |
| WVM         | Exponential.  |
| AL          | 33 attributes.|
| MLC         | Bayes.TAN.    |
| CV          | Yes.          |

The results of this module are illustrated in Chapter Six.

5.3.2.3 Window Sampling Module

The major task of window sampling module was set to determine the optimal window length. Apart from this, it also compared the performances and efficiencies of different classifiers when applied with a certain weight variation model. The configuration of this module is defined in below (Table 5-4):

| Table 5-4: Configuration for window sampling module. |
|-------------|----------------|
| PL          | FD232.         |
| WL          | Optional.     |
| WVM         | Optional.     |
| AL          | 33 attributes.|
| MLC         | Optional.     |
| CV          | Yes.          |

The experiments defined by this module demonstrated that the instance-based classifiers produced highest accuracies, but they are very slow; the Bayesian network classifiers produced both good prediction accuracy and were very quick. The selection of classifiers and the corresponding weight variation models are discussed in Chapter Six.
5.3.2.4 Combination Sampling Module

This module was designed to have the maximum degree of freedom in varying the configurations. The performing of this module is very time consuming, and usually is set to perform in the background, not to waste the computational resources. That is also the reason why these modules are assigned with a priority. While the users do not require the PSAR to work, the PSAR will resume the job in this module, and record all the results from this module. The user can specify the range of the variation in configurations in this module, to perform a task, which is not urgent, and very time consuming.

5.3.3 Result-Oriented Modules

The prediction accuracies are amongst the most important datasets, which are manipulated within the PSAR infrastructure. Along with implementation of the PSAR infrastructure, the produced results are accumulating. At the time of writing, the PSAR infrastructure obtained 5,386,481 results and they were deposited in the database. Therefore, the first result-oriented module in this section is the database system.

5.3.3.1 Database module

The database contains a table comprising sections that store prediction results, configurations, runtime information of that particular experiment. The purpose of recording these relevant messages is to allow user to repeat this experiment. In the current literature, it is difficult to repeat the experiments unless the results reporters themselves wanted to do it.

Additional information, such as consumed time, is also recorded to compare the performance of predictions in different computational platforms. The table is constructed in overleaf (Table 5-5):
Table 5-5: Section definition in the table, which records results

<table>
<thead>
<tr>
<th>Sections</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResDescriptor</td>
<td>A short keyword defines the purpose of this particular experiment.</td>
</tr>
<tr>
<td>WindowMec</td>
<td>The definition of window variation model.</td>
</tr>
<tr>
<td>WindowSize</td>
<td>The length of the window frame.</td>
</tr>
<tr>
<td>Normalize</td>
<td>Whether the weight values are normalized</td>
</tr>
<tr>
<td>ResAccuracy</td>
<td>The prediction accuracy of this particular experiment.</td>
</tr>
<tr>
<td>Classifier</td>
<td>The applied classifier for this experiment.</td>
</tr>
<tr>
<td>Discrete</td>
<td>Whether the dataset was discretized before being fed into the classifier.</td>
</tr>
<tr>
<td>FeatureList</td>
<td>The applied feature descriptors of the amino acids in that particular experiment.</td>
</tr>
<tr>
<td>SampleList</td>
<td>The applied protein samples in that particular experiment.</td>
</tr>
<tr>
<td>TestTime</td>
<td>The time when the experiment was performed.</td>
</tr>
<tr>
<td>RunMachina</td>
<td>The host name of where the experiment was performed.</td>
</tr>
<tr>
<td>ConsumeTime</td>
<td>The time cost of that particular experiment to finish.</td>
</tr>
<tr>
<td>TestDate</td>
<td>The date when the experiment was performed.</td>
</tr>
</tbody>
</table>

The results were directly sent to the database by results capturing module after the learning and predicting was finished.
5.3.3.2 Results Capturer Module

The output that the WEKA package produced was in a particular format, which contained a lot of information; some of which was not required in the PSAR. The results capturer module captures the outputs from the WEKA package, and retrieves important information such as prediction accuracy and confusion matrix, then deposits them into the database.

5.3.3.3 Results Reviewing Module

At the time of writing, the results are reviewed by the build-in functions of the PHPMyAmin project. An individual web-based results reviewer is planned in the schedule, for chemists to check the results, validate the prediction results with their chemical expertise. The results reviewer is important because it is the only way to extract useful information and convert it into knowledge. After the set up of the PSAR infrastructure it has performed over 5.6 million experiments. The verification and validation of these results becomes increasingly important for this project.

5.4 Safety Evaluation of the PAMS/PSAR Framework

The safety evaluation of the PAMS/PSAR framework contains several aspects. First of all, the PAMS/PSAR is established in a fully-networked environment, and the host of the system is open for anyone to access with authorization. By doing so, no matter where the author is (s)he still is capable of performing the tasks remotely by access from SSH or the web; the security of the system must be ensured. Secondly, the reliability of the procedures that prepares datasets; performs prediction; and captures results must be absolutely correct. Any mistakes occurred during the procedures defined above will produce dirty data, which may lead to misunderstanding. Finally, the warehoused biological data need to be protected to avoid contamination. Corrupted data in biological samples certainly produce undetermined data, therefore leading to misunderstanding as well.
5.4.1 System Security of the Hosts with the PAMS/PSAR Installed

Security is increasingly important in a connected world, for the PSAR system it is also a big issue. To ensure the security of the PSAR infrastructure, the security of the system must be ensured first. When cyber hackers successful in obtaining the privileges to access the system, they could easily destroy everything. The PSAR is dealing with biological data samples, in future it may deal with confidential data, and therefore the security issue is very important.

The PAMS/PSAR was hosted in a Fedora Core operating system with an X established over it. Although the Fedora Core is considered to be a secure system but also along with time possible vulnerabilities may be discovered. A good way to keep the system up to date is to perform scheduled security check on the host in which PAMS/PSAR sit. At the moment in this project the author implement the Nessus vulnerabilities assessment tool to do this job. The Nessus is able to discover over eleven thousand security vulnerabilities, which may bring deadly consequences to the host. The follow diagram shows the graphic user interface of this tool (Figure 5-3):

![Nessus client for security vulnerabilities assessment](image)

Figure 5-3: Nessus client for security vulnerabilities assessment
In the preceding diagram, the host, which was used as the implementation platform contains a security hole and a security warning. Patching the system according to the provided suggestions keeps the system secure.

Another commonly used security assessment tool is the build-in NMAP application. The screen shot of NMAP reporting security assessments to this host is presented in below (Figure 5-4):

![Figure 5-4: NMAP reporting the security assessments.](image)

It shows that, after patching the system, there is no significant security vulnerability existing in the host.
5.4.2 Reliability of the Procedures Defined by PAMS/PSAR

Although the implementation of PAMS/PSAR has already adopted procedures such as setting up priorities, restricted data resources to be accessed only by certain modules, each procedure of processing data also is double checked in types and range before being used. The implementation of window mechanism and the application of linear weight variation model and exponential weight variation model are widely proved in the literature (Baldi et al. 1999; Delcher et al. 1993; Frishman & Argos 1996a; Hu et al. 2004; Karypis 2006; Kloczkowski et al. 2002a; Liu et al. 2004; Ouali & King 2000; Sadeghi et al. 2005; Yi & Lander 1993). The concept of processing biological data by configurations was first inspired by the web-based primary sequence analysis tool protscale@ ExPASy (Wilkins et al. 1999).

5.4.3 Validity of the Biological Dataset

These biological datasets are essential for this research. To ensure that they are correct, the data which are transmitted between the PAMS/PSAR and the database, or between different functioning modules, must be type checked before they are accepted. By doing so the scenarios that the parameters are filled by “null” due to the absence of resources are avoided. Besides that the MySQL database also checks every piece of data in data storage types. While the PAMS/PSAR was processing the amino acid feature descriptors, an upper bound and bottom bound were defined to make sure that all feature descriptors were transmitted correctly.

5.5 Performances Evaluation of PAMS/PSAR

The PAMS/PSAR is able to make use of main parts of computational power, and achieves an average on output prediction accuracy of one per 4.5 second. The performance of each individual experiment is determined by the number of applied protein samples, the applied machine learning classifiers, the applied amino acid feature descriptors, and the window variation model. Each experiment result was recorded along with a ConsumeTime key in the database, indicating the used time to finish. The PSAR is believed to occupy a potentially major portion of the computational resources available in
the workstation to perform the data preparation and prediction tasks, considered to be effective.

5.6 Conclusions

The PAMS/PSAR infrastructure is designed for the purpose of validating the hypothesis in this research. It is also intended to help chemists to validate their own hypotheses, based on the results and relevant recorded information. During the process of developing the PAMS/PSAR framework, the usability, effectiveness, robustness, security and safety, correctness of generated datasets are all concerned. The communication between modules based on network is encrypted. Passwords are set to authorized different modules have different priorities. In the next chapter, the implementation of PAMS/PSAR infrastructure is illustrated.
6 Implementation of the PSAR Framework

The PSAR infrastructure is currently occupying 96 processors that are running Linux. The job distributor is hosted on a Pentium 4 machine, which is running the Fedora Core operating system in the Chemistry Division at the University of Surrey. In this chapter the author represents the implementation of the PSAR infrastructure, and also shows a few experimental results, which were obtained by applying the PSAR to a certain question. The following sections represent the experiments, which have been performed to obtain optimal ranges of configurations, and the experiments that have been performed to get good prediction accuracy in a case study.

6.1 Sampling the Feature Descriptors

506 amino acid feature descriptors, which were taken from the AAIndex database, were ranked in this experiment, according to their performance to the overall 3080 protein chains. In this experiment, each amino acid feature descriptor was applied to over half a million amino acids respectively. The prediction accuracies of the experiment were taken as the measurements of the influences that the corresponding amino acid feature descriptors to the secondary structure. 506 tasks were performed by the PSAR in this experiment, and the generated prediction accuracies are used to rank these feature descriptors. The outputted results in this experiment were used frequently in later on experiments.

The test results were captured by the results capturer module, and the terminal output is shown in overleaf (Figure 6-1). This experiment employs the Tree-augmented Bayesian network classifier to perform the tasks of learning and predicting. The first line indicates the identity of the feature descriptor used in that particular test; the second line clarifies the result of that designated test. The two lines indicate one test to the quality of an amino acid feature descriptor, which performed by the PSAR in one turn. After the experiment had been performed, there was a total of 506 ranks of the amino acid feature descriptors.
stored in the database.

![Figure 6-1: PSAR terminal output for attribute sampling.](image)

From the preceding diagram it is apparent that the use of a different feature descriptor to describe the amino acid residue’s property leads to different ranking results. For example, the feature descriptor of the “CHAM830101” is ranked with its self testing accuracy,
54.45%, and for the feature descriptor of the “BEGF750101” the self testing accuracy is 53.97%. At the same time there are feature descriptors which reported low prediction accuracies, e.g. the feature descriptor of the “BIOV880101” reported a self test rank of 48.92%. In general, all the attributes are tested by being implemented in the super protein dataset, which contains 3080 protein chains with half million amino acid residues. The self testing ranks are considered as numerical representative of the quality of data being used in protein secondary structure prediction. Currently, the PSAR infrastructure picked a top 33 feature descriptors. These 33 feature descriptors are considered to be well qualified in the process of protein secondary structure prediction. A top ten ranked attributes are listed below (Table 6-1):

<table>
<thead>
<tr>
<th>Attribute ID</th>
<th>Self-Testing Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTIO830101</td>
<td>55.40%</td>
</tr>
<tr>
<td>MUNV940102</td>
<td>55.32%</td>
</tr>
<tr>
<td>ROBB760103</td>
<td>55.22%</td>
</tr>
<tr>
<td>AURR980113</td>
<td>55.16%</td>
</tr>
<tr>
<td>SUEM840101</td>
<td>55.00%</td>
</tr>
<tr>
<td>KANM800103</td>
<td>54.93%</td>
</tr>
<tr>
<td>AURR980109</td>
<td>54.90%</td>
</tr>
<tr>
<td>QIAN880107</td>
<td>54.86%</td>
</tr>
<tr>
<td>MUNV940101</td>
<td>54.83%</td>
</tr>
<tr>
<td>RACS820108</td>
<td>54.81%</td>
</tr>
</tbody>
</table>

These labels are defined by the AAINDEX project. This table shows several significant patterns. First of all, the ranks of these feature descriptors do not vary greatly from each other. The rank of the feature descriptor of the “PTIO830101” is only 0.08% higher than the second highest ranked feature descriptor of the “MUNV940102”. Also it seems that all the members of QIANXXXXX family of feature descriptors reported good rank order, and they are all statistical secondary structure propensities of amino acids in certain positions. It is intuitive to accept that these feature descriptors received good self testing ranks.
In order to find out whether this scenario happened in all classifiers, there were another series of experiments, performed by using KStar, IB1, J48 classifiers, with same configurations. The result was that the self testing ranks by using other classifiers were similar to the results that have been achieved by using TAN classifier. In other words if a particular feature descriptor is considered to be good in one particular classifier, it is also good when implemented in another classifier. The following table (Table 6-2) shows the top ten ranked feature descriptors by using the IB1 classifier. The feature descriptors are similar to the feature descriptors listed in Table 6-1.

Table 6-2: Top ten ranked attributes in experiment 3.3.1, by using IB1.

<table>
<thead>
<tr>
<th>Attribute ID</th>
<th>Self-Testing Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTI0830101</td>
<td>56.36%</td>
</tr>
<tr>
<td>SUEM840101</td>
<td>56.32%</td>
</tr>
<tr>
<td>QIAN880107</td>
<td>56.14%</td>
</tr>
<tr>
<td>AURR980113</td>
<td>56.09%</td>
</tr>
<tr>
<td>MUNV940102</td>
<td>56.05%</td>
</tr>
<tr>
<td>ONEK900101</td>
<td>55.98%</td>
</tr>
<tr>
<td>AURR980109</td>
<td>55.94%</td>
</tr>
<tr>
<td>ROBB760103</td>
<td>55.75%</td>
</tr>
<tr>
<td>ISOY800101</td>
<td>55.71%</td>
</tr>
<tr>
<td>RACS820108</td>
<td>55.66%</td>
</tr>
</tbody>
</table>

Because the Bayesian network classifier performed the job accurately and quickly, in the following experiments, it becomes one of the main characters on the stage. Table 6-3 presents the distribution of the top 168 feature descriptors which all have self testing ranks better than 53%.

Table 6-3: Classification of top 168 ranked attributes for Bayes.TAN.

<table>
<thead>
<tr>
<th>Accuracy range</th>
<th>Numbers of Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;= 55.0%</td>
<td>5</td>
</tr>
<tr>
<td>&gt;= 54.0% &amp; &amp; &lt;= 55.0%</td>
<td>28</td>
</tr>
</tbody>
</table>
In most of subsequent experiments, the PSAR use at most the top 33 feature descriptors; producing good prediction accuracy, and a fairly quick prediction. Using more than 33 feature descriptors slows down the prediction enormously, especially applied to the classifiers such as instance based classifiers.

### 6.2 Determining the Numbers of Applied Feature Descriptors

In order to find out the minimum number of feature descriptors required in an experiment, and how the prediction accuracies vary along with the numbers of applied feature descriptors varied, this experiment put the numbers of applied feature descriptors to the test. It is believed that applying more feature descriptors certainly slows down the process of prediction, but does not necessarily improve the prediction accuracy. Carrying out this experiment helps to find a balance point, which ensures the learning and predicting of biological datasets is both accurate enough, and quick enough.

The following experiment was performed to find how the numbers of attributes correlated with the prediction accuracy. The top 33 ranked feature descriptors were applied in this experiment, picked from top to bottom. In the first experiment, the top three feature descriptors were selected to be applied; in the second experiment, the top four feature descriptors were picked to be applied; and so on; the experiment stopped while taking the top 33 feature descriptors. The experiment took the super set of 3080 protein chains as the protein samples, and implemented the TAN classifier. The window length was set to 21, and the exponential weight variation model was selected in the experiment. All training and testing datasets were discretized before they were classified. The terminal output from the PSAR was presented as following diagram (Figure 6-2):
<table>
<thead>
<tr>
<th>Testing attributes number: 3</th>
<th>Correctly Classified Instances: 5734</th>
<th>Error: 81.345</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing attributes number: 4</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 5</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 6</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 7</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 8</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 9</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 10</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 11</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 12</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 13</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 14</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 15</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 16</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 17</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 18</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 19</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 20</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
<tr>
<td>Testing attributes number: 21</td>
<td>Correctly Classified Instances: 5734</td>
<td>Error: 81.345</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 6-2: PSAR terminal output for experiment of numbers of applied feature descriptors.

In the preceding diagram tests from using 3 to 21 feature descriptors are represented. In fact the experiment explored the numbers of applied feature descriptors in the range of [3, 33]. From the terminal outputs it is apparent that these prediction accuracies are quite similar to one other. It does not make a great deal of difference when seven or seventeen feature descriptors in the experiment. This may not be true for each experiment taking different datasets, implementing different amino acid feature descriptors, but the scenario
of prediction accuracy varying little is commonly occurring. A graphic presentation for this experiment is shown in the following diagram (Figure 6-3):

![Diagram showing prediction accuracy varying with numbers of used attributes](image)

Figure 6-3: Prediction varies along with numbers of applied feature descriptors.

In the above diagram the X axis presents the numbers of applied feature descriptors in the test, and it varies from the range [3, 33]. The Y axis indicates the corresponding prediction accuracy.

### 6.3 Ranking the 3080 Protein Samples

The selection of 3080 protein samples represents the non-homologous proteins within the PDB databank, but does not necessarily offer a good mean of predicting protein secondary structure. At the same time, using all of these 3080 protein samples in each experiment is not economic. In this experiment, the 3080 protein chains were examined, and a subset of these was selected to be used in further investigation. This subset is considered to be both better qualified and representative.
Chapter 6: Protein Structural Accuracy Recommender

This experiment applied the TAN classifier to perform the learning and predicting task. The exponential weight variation model was applied to calculate the weight distribution along the window frame and the window length was set to 21; all data were discretized before fed into the machine learning classifier. The top ranked 33 feature descriptors were taken to describe the characteristics of each amino acid. The terminal output of the PSAR infrastructure is presented as following diagram (Figure 6-4):

![Figure 6-4: PSAR terminal output of protein sampling.](image_url)
Chapter 6: Protein Structural Accuracy Recommender

Obviously there is only a small portion of protein chains presented in the terminal output. There are 3080 records currently recorded in the database and there are significant patterns available in the experiment results as well. In order to be able to display these results more intuitively, after the experiment, each protein chain was assigned a two dimensional vector by the patterns defined in following tables (Table 6-4) and (Table 6-5):

**Table 6-4: The pattern of self testing ranks.**

<table>
<thead>
<tr>
<th>Self-Testing Quality Measurement</th>
<th>Quality Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
<td>&gt;= 85%</td>
</tr>
<tr>
<td>Medium</td>
<td>&gt;= 75% &amp;&amp; &lt;= 85%</td>
</tr>
<tr>
<td>Bad</td>
<td>&lt;= 75%</td>
</tr>
</tbody>
</table>

In the above table, any protein chain that has self testing rank greater than 85%, is considered as a good candidate for secondary structure prediction; if the self testing rank of a particular protein chain is less than 75%, it is considered as a bad candidate, and not appropriate for participating in protein secondary structure prediction. The protein chains in the middle are considered as medium in self testing ranks.

Apart from the self testing ranks the author found out that the length of the protein chain is another important measurement of the protein's quality. The pattern is defined as:

**Table 6-5: The pattern of peptide chain length.**

<table>
<thead>
<tr>
<th>Protein Length</th>
<th>Length Number</th>
<th>Number of Proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long length globular protein</td>
<td>&gt;= 501 &amp;&amp; &lt;= 1264</td>
<td>35</td>
</tr>
<tr>
<td>Medium length globular protein</td>
<td>&gt;= 300 &amp;&amp; &lt;= 501</td>
<td>248</td>
</tr>
<tr>
<td>Short length globular protein</td>
<td>&lt;=300</td>
<td>2797</td>
</tr>
</tbody>
</table>

The 3080 peptide chains were plotted in a 2-dimensional space with the X axis
representing the length of peptide chain, and Y axis denoting the self-testing quality. Each protein was plotted as a blue dot in the two dimensional space, with the coordinate set to \langle\text{length}, \text{accuracy}\rangle. The maximum length in this experiment was 1294, and the minimum length was 33. The maximum accuracy reached 100%, and the bottom reached 0. The diagram is presented as (Figure 6-5):

![Figure 6-5: The quality of protein chains.](image)

There are several interesting results revealed from this plot. Firstly there is one protein chain that was predicted with an accuracy of 0%. There were also several protein chains, which were predicted as 100%. These protein chains are good candidates to form the training set for protein secondary structure prediction. The selection of protein candidates for protein secondary structure has to satisfy both the requirement of self testing ranks and the coverage of protein length. A selection was carries out by the author and was named FD232 dataset; it contains 232 protein chains, which cover the best self testing ranks and protein length. A list of protein identities of this dataset is presented in the appendix.
6.4 Selecting Window Length

The length of window frame influences the prediction result. The setting of a window frame with a particular amino acid residue in the centre presents the belief that a certain number of amino acid residues are contributing to the secondary structure of the amino acid in the centre. Although the implementation of window frame omits the long distance interaction it still improves the prediction accuracy by 20%. In the literature it has been suggested that if the window length is too short, then maybe some of the structural information will be miscaptured by the machine learning classifier and, if the window length is too long, noises may be involved to influence the prediction accuracy. In this experiment by using the PSAR the author was trying to find a global optimal window length for protein secondary structure prediction.

In this experiment, as usual, the Forest-Augmented Naïve Bayesian network classifier was applied as the classifier, performing the tasks of learning and predicting. The FD232 protein samples are implemented to provide protein candidates for this experiment. Thirty-three top ranked amino acid feature descriptors have been applied in this experiment, to describe the characteristics of amino acids. The PSAR infrastructure is built based on the one of the fundamental assumptions, which is: “The 3-dimensional structure of an amino acid is determined not only by the characteristics of the residue itself, but also the influences of neighbouring amino acids.” In the following diagram, the terminal output of the PSAR infrastructure is showed. Each two lines together, represent one experiment. The first line indicates the length of the window frame applied in that particular experiment, and the second line indicate the prediction accuracy achieved in that experiment:
<table>
<thead>
<tr>
<th>Testing winlength</th>
<th>Correctly Classified Instances</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5196</td>
<td>78.1247%</td>
</tr>
<tr>
<td>7</td>
<td>5056</td>
<td>79.1284%</td>
</tr>
<tr>
<td>9</td>
<td>5005</td>
<td>79.1174%</td>
</tr>
<tr>
<td>11</td>
<td>4995</td>
<td>79.1724%</td>
</tr>
<tr>
<td>13</td>
<td>4965</td>
<td>79.2024%</td>
</tr>
<tr>
<td>15</td>
<td>4935</td>
<td>79.2124%</td>
</tr>
<tr>
<td>17</td>
<td>4905</td>
<td>79.2024%</td>
</tr>
<tr>
<td>19</td>
<td>4875</td>
<td>79.2024%</td>
</tr>
<tr>
<td>21</td>
<td>4845</td>
<td>79.2024%</td>
</tr>
<tr>
<td>23</td>
<td>4815</td>
<td>79.2024%</td>
</tr>
<tr>
<td>25</td>
<td>4785</td>
<td>79.2024%</td>
</tr>
<tr>
<td>27</td>
<td>4755</td>
<td>79.2024%</td>
</tr>
<tr>
<td>29</td>
<td>4725</td>
<td>79.2024%</td>
</tr>
<tr>
<td>31</td>
<td>4695</td>
<td>79.2024%</td>
</tr>
<tr>
<td>33</td>
<td>4665</td>
<td>79.2024%</td>
</tr>
<tr>
<td>35</td>
<td>4635</td>
<td>79.2024%</td>
</tr>
<tr>
<td>37</td>
<td>4605</td>
<td>79.2024%</td>
</tr>
<tr>
<td>39</td>
<td>4575</td>
<td>79.2024%</td>
</tr>
<tr>
<td>41</td>
<td>4545</td>
<td>79.2024%</td>
</tr>
<tr>
<td>43</td>
<td>4515</td>
<td>79.2024%</td>
</tr>
<tr>
<td>45</td>
<td>4485</td>
<td>79.2024%</td>
</tr>
<tr>
<td>47</td>
<td>4455</td>
<td>79.2024%</td>
</tr>
</tbody>
</table>

Figure 6-6: Terminal outputs of selecting window length with FAN + exponential model;

The experiment results are plotted in the following diagram (Figure 6-8):
On the following page a series of diagrams represents the experiments which have been performed by applying different classifiers and different weight variation model. (Figure 6-8) shows the terminal output and plot about prediction accuracies according to the variation of window length by using IBk classifier along with linear weight variation model, (Figure 6-89) shows the terminal output and plots prediction accuracies according to the variation of window length by using Kstar classifier along with exponential weight variation model, (Figure 6-810) shows the terminal output and plots prediction accuracies according to the variation of window length by using LMT classifier along with exponential weight variation model, (Figure 6-811) shows the terminal output and plots prediction accuracies according to the variation of window length by using SMO classifier along with exponential weight variation model, (Figure 6-812) shows the terminal output and plots prediction accuracies according to the variation of window length by using JRIP classifier along with exponential weight variation model, (Figure 6-813) shows the terminal output and plots prediction accuracies according to the variation of window length by using PART classifier along with exponential weight variation model.
<table>
<thead>
<tr>
<th>Win length</th>
<th>Correctly classified instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>410</td>
</tr>
<tr>
<td>3</td>
<td>281</td>
</tr>
<tr>
<td>4</td>
<td>131</td>
</tr>
<tr>
<td>5</td>
<td>77</td>
</tr>
<tr>
<td>6</td>
<td>42</td>
</tr>
<tr>
<td>7</td>
<td>22</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 6-8: PSAR terminal output and plot showing selecting window length (IBk + Linear).
Figure 6-9: PSAR terminal output and plot showing selecting window length (Kstar + Exp).
Figure 6-10: PSAR terminal output and plot showing selecting window length (LMT + Exp).
Figure 6-11: PSAR terminal output and plot showing selecting window length (SMO + Exp).
Chapter 6: Protein Structural Accuracy Recommender

Figure 6-12: PSAR terminal output and plot showing selecting window length (Jrip + Exp).
Figure 6-13: PSAR terminal output and plot showing selecting window length (PART + Exp).
The prediction accuracy for the protein secondary structure in this experiment is determined by several factors, e.g. the length of window frame; the weight variation model; the amino acid feature descriptors that applied in the experiment; the protein samples and the applied machine learning classifiers. Once the amino acid feature descriptors and protein samples have been determined, the prediction accuracy varies from the combinations of window length, classifiers, and weight variation model. A diagram representing this scenario is presented in (Figure 6-14).

From the above diagram, the results of IBk and Kstar are very similar to one another. Secondly, in the position of 21, almost all classifiers reported top prediction accuracies along the curves. At last, the prediction accuracies do not drop greatly while the window length approaches the other side of 49. Once the window length reaches 21, the prediction accuracies remain stable.
6.5 Selecting Linear or Exponential

In our experiment it is shown that the certain classifiers perform better in the linear weight variation model and certain classifiers perform better in the exponential weight variation model. In the design of this experiment, the combinations of classifiers and weight variation models are examined. This work was also inspired by previous experiments in looking for global optimal window length. In this experiment, the FD232 dataset was implemented; thirty-three top ranked amino acid feature descriptors, twenty-six top ranked amino acid feature descriptors and seven top ranked amino acid feature descriptors were applied sequentially. The following table (Table 6-6) shows the performances of each different learning scheme when they were applied along with linear or exponential weight variation models:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Naïve Bayes</th>
<th>FAN</th>
<th>AODE</th>
<th>RBF Network</th>
<th>IB1</th>
<th>IBk</th>
<th>KStar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>74.62%</td>
<td>81.14%</td>
<td>81.88%</td>
<td>80.70%</td>
<td>90.50%</td>
<td>90.67%</td>
<td>90.97%</td>
</tr>
<tr>
<td>Exp</td>
<td>75.45%</td>
<td>81.74%</td>
<td>82.36%</td>
<td>80.88%</td>
<td>87.67%</td>
<td>88.30%</td>
<td>88.51%</td>
</tr>
<tr>
<td>ID3</td>
<td>80.20%</td>
<td>82.67%</td>
<td>85.60%</td>
<td>81.40%</td>
<td>83.94%</td>
<td>82.10%</td>
<td>82.65%</td>
</tr>
<tr>
<td>Linear</td>
<td>82.83%</td>
<td>82.36%</td>
<td>84.32%</td>
<td>81.73%</td>
<td>81.89%</td>
<td>80.46%</td>
<td>81.95%</td>
</tr>
</tbody>
</table>

The terminal output of the PSAR infrastructure output results in the following diagram (Figure 6-17):
Chapter 6: Protein Structural Accuracy Recommender

Figure 6-15: Results that obtained when classifiers worked with linear weight variation model.

The linear weight variation model worked better in instance based classifiers, such as IB1, and IBk. The exponential weight variation models presented slightly higher results when working with Bayesian network classifiers, than with linear weight variation models. The most frequently used classifiers in PSAR were FAN, and IBk. The Forest-augmented Naïve (TAN) Bayesian classifier was used in processing vast amount of prediction tasks, due to its performance, and IBk was used to obtain the highest prediction accuracy.

Similarly, the PSAR terminal output of classifiers work with exponential weight variation model is presented in the follow diagram (Figure 6-16):
A more representative description of this experiment is presented in the following two diagrams (Figure 6-17) and (Figure 6-18). All experiments took 21 as the window length, and implemented the top seven, top twenty six, or top thirty-three attributes over the FD232 dataset. Different types of variation model were implemented in each classifier.
Chapter 6: Protein Structural Accuracy Recommender

Figure 6-17: Performances of classifiers with linear variation model.

Figure 6-18: Performances of classifiers with exponential variation model.
Not only were the prediction accuracies of the classifiers recorded, but also the consumed time of the classifiers to give information about the efficiency of the classifiers. For example, although the instance based classifiers produced the highest prediction accuracies, it took a couple of days to finish a single task and was also highly dependent on the size of the dataset. It is obvious that the instance based classifiers are not suitable for vast amount of tests in batch mode. The Forest-Augmented Naïve Bayesian network classifiers produced relatively accurate prediction accuracy, and at the same time they performed the task quickly, usually taking a couple of seconds to finish. The FAN classifier is suitable for huge amounts of batched experimental tasks.

6.6 Selecting Combinations of Amino Acid Feature Descriptors

This was the last experiment that to be performed by the PSAR infrastructure, but the heaviest. Selecting the combinations of amino acid feature descriptors, is the first step toward the further understanding about how these amino acid feature descriptors work together, and contribute to the secondary structure prediction accuracy. The number of combinations of amino acid feature descriptors is enormous. This experiment performed prediction tasks over the randomly combinations of all the 33 features, with a maximum of five feature descriptors implemented at the same time. The total number of this definition was $28480320 + 982080 + 32706 + 1056 + 33 = 29496195$. Without the distributed computational platform such a task would not be possible. There are nearly three million tests in the queue, and each one of them requires at least seven seconds in preparing the dataset and performing the prediction. Using a single computer it requires a total of 2389.7 days. But by using the PSAR infrastructure covering 96 CPUs in the Linux cluster, the required time costs have been shortened to 19.91 days (less than three weeks).

In these three million results, there are 139 combinations, which were found to report good prediction accuracies. There were also 73 combinations of feature descriptors reported with low prediction accuracies. A PSAR terminal output of the list of the feature descriptor combinations that produced prediction accuracy greater than 84% is presented in the following diagram (Figure 6-19):
In the following diagram (Figure 6-20) the occurrences of good attributes are presented. For example the “BLAM930101” descriptor appeared in each experiment which produced accuracy greater than 84%. Does that mean the “BLAM930101” is suitable for protein secondary structure prediction? Although the intrinsic reason of why it happens is not yet known, but the answer is yes.

Figure 6-20: Occurrences of features with prediction accuracies >84%.
A PSAR terminal output of the list of the feature descriptor combinations that produced prediction accuracy less than 73% is presented in the following diagram (Figure 6-21), and the occurrence plot is presented in (Figure 6-22):

Figure 6-21: Combinations of feature descriptors produced prediction accuracy less than 73%.

Figure 6-22: Occurrences of attributes with prediction accuracies <73%.
Similarly, the amino acid feature descriptor of “AVBF000105” appeared 16 times among the 73 bad combinations.

6.7 Conclusions

In this chapter the implementation of the PSAR infrastructure is presented. Not only can the PSAR be implemented to obtain the aforementioned experiment results, but it can also be used to suggest the best prediction accuracy over a particular protein. The preparation process, the discretization process, and the learning and predicting process over the dataset are automated. By using the PSAR further more experimental results could be obtained. Suggestions for future work are presented in Chapter Seven.
7 Future Work

The PAMS/PSAR infrastructure was designed to explore the intrinsic factors which are hidden in the vast amount of biological structural data. The future development of the PAMS/PSAR infrastructure can be approached from both a technical or theoretical aspect. The technical enhancements include a friendly user interface for the PSAR infrastructure, and the further development about the distribution characteristic of the PSAR.

There are over five million experimental results have been recorded in the system at the moment, and these results have not yet been fully studied. A series of further studies are proposed in this chapter.

7.1 Technical Improvements for PSAR

The PAMS has a user interface at the moment but the PSAR does not have. Developing a user interface enable the users to be able to access full functionalities of the PSAR infrastructure. The job distributor is embedded behind the user interface, receives configurations, and distributes jobs over the compute agents on the network. At the moment the PSAR is accessed by terminal user interface. In future, the PSAR is planned to be accessed both from web and X window applications:

- **X window access**: the job distributor can be launched from an X environment, as the PAMS do at the moment;

- **web-based access**: the job distributor is managed by a web interface through the apache web server. This is the most convenient method to manipulate the PSAR infrastructure. The job distributor will also be embedded with a web based results reviewer.
The future development of the PSAR's interface is represented as the following defined diagram (Figure 7-1):

![Figure 7-1: PSAR Interface Design](image)

### 7.2 Scientific Improvements for PSAR

There are still a few questions that have not been answered throughout the thesis. The original target of developing the PSAR infrastructure was pursuing the target of improving protein secondary structure prediction accuracy by data purification. However, the experiments brought a series of interesting patterns, which were worthy of further investigation. In this section a list of scenarios are described and briefly examined.

First of all, there were a few combinations of feature descriptors, which produced very good prediction results. On the other hand there were another set of combinations of feature descriptors, which produced poor prediction accuracies. There are certain patterns hidden within these results and need to be examined.
Secondly, these prediction results need to be verified with chemists using their chemical expertise. It is possible to reveal the biochemical meaning of some patterns of prediction results.

Thirdly, other biochemistry prediction problems could be applied in this PAMS/PSAR system, such as contact map prediction, subcellular activities prediction etc.

Furthermore, a study of applying different weight variation models in the protein secondary structure prediction problem can be examined. Obviously the Gaussian function can be used in this case. To perform this research, the problems of scale normalization and position resolutions need first to be solved.

Obviously the current protein secondary structure prediction methodologies predict the accuracy by implementing all amino acid feature descriptors with the same weight. Applying methods that take different amino acid feature descriptors with different weights makes more sense. The WEKA package has just started to support weighted prediction, which is a good opportunity for this project.

There are also questions such as whether these physico-chemical forces influence each other while they assembled the protein tertiary structure, and how these kinds of scenarios can be captured by the PSAR infrastructure.

7.3 From Secondary Structure to Tertiary Structure

The ultimate target is to predict the tertiary structure from the primary sequence. With good prediction accuracy in the secondary structure, what steps can we take to predict the tertiary structure? Obviously it is not an easy task, but worthy of further pursuit. From the former diagram (Figure 3-1), it is worthwhile to consider the impact of PSAR to the whole
process and how the PSAR can be improved to benefit the whole process of tertiary structure prediction.

7.4 General Conclusion

In this thesis, the author has presented a computational framework to perform automatically the tasks of datasets preparation and protein secondary structure prediction. The hypothesis is approved: along with the data purification the prediction accuracy of protein secondary structure improves. From the results listed in the Chapter Six it is confirmed that along with the procedure of data purification, the prediction accuracies of protein secondary structure increases. Apart from that the implementation of PSAR infrastructure opened a new viewpoint, to make the available biological data resources be more useful.


Dhe-Paganon, S., Duda, K., Iwamoto, M., Chi, Y. I., & Shoelson, S. E. 2002, "Crystal structure of the HNF4 alpha ligand binding domain in complex with endogenous fatty acid ligand", *J. Biol. Chem.*, vol. 277, no. 41, pp. 37973-37976.


Bibliography


Frishman, D. & Argos, P. 1996b, "Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence", *Protein Eng.,* vol. 9, no. 2, pp. 133-142.


Frishman, D. & Argos, P. 1996a, "Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence", *Protein Eng.,* vol. 9, no. 2, pp. 133-142.


Bibliography


Bibliography


Lau, W., Kuo, T. Y., Tapper, W., Cox, S., & Collins, A. 2006, "Exploiting large scale computing to construct high resolution linkage disequilibrium maps of the human genome", *Bioinformatics*.


LEVINTHA.C 1968, "Are There Pathways for Protein Folding", *Journal de Chimie Physique et de Physico-Chimie Biologique*, vol. 65, no. 1, p. 44-&.


Pauling, L. 1985, "Citation Classic - the Nature of the Chemical-Bond and the Structure of Molecules and Crystals - An Introduction to Modern Structural Chemistry", Current Contents/Engineering Technology & Applied Sciences no. 4, p. 16.


Pauling, L. & Sherman, J. 1933a, "The nature of the chemical bond. VI The calculation from thermochemical data of the energy of resonance of molecules among several electronic structures", Journal of Chemical Physics, vol. 1, no. 8, pp. 606-617.


Bibliography


