A data-driven protein-structure prediction algorithm

Clinton J. Robinson David B. Skillicorn

March 13, 2006

School of Computing
Queen's University
Kingston, Ontario, Canada K7L 3N6

Document prepared March 13, 2006

Abstract

Protein-structure elucidation is currently slow and expensive by physical means and current prediction algorithms either lack accuracy or scope. A data-driven dynamic-programming algorithm for predicting protein structures is presented. Observed conformations of short amino-acid chains in the Protein Data Bank are reduced to canonical conformations using singular value decomposition to remove components considered to be noise, and semidiscrete decomposition to form clusters. These canonical conformations are then used to generate conformations for longer sequences using dynamic programming.

The algorithm is able to extrapolate beyond the base data to provide conformations for short sequences of previously unseen amino acids. The algorithm is also able to predict significant portions of large proteins, includng complex secondary structural elements such as turns, bends and random coils. The entire structure of a small protein is predicted and presented as a 3-dimensional model.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
  - 1.1.1 Protein structure . . . 1
  - 1.1.2 Determining protein structure . . . 4
- 1.2 State of the art . . . 4
- 1.3 Objective . . . 4
- 1.4 Organization of Report . . . 5

2 Background **6**
- 2.1 Structure determination methods . . . 6
  - 2.1.1 X-Ray Crystallography . . . 6
  - 2.1.2 Nuclear Magnetic Resonance . . . 7
- 2.2 The Protein Data Bank . . . 7
- 2.3 The Ramachandran plot . . . 7
- 2.4 Structure prediction algorithms . . . 8
  - 2.4.1 Secondary structure prediction . . . 8
  - 2.4.2 Comparative modeling . . . 11
- 2.5 Matrix decompositions . . . 12
  - 2.5.1 Singular Value Decomposition . . . 12
  - 2.5.2 SemiDiscrete Decomposition . . . 13
- 2.6 Dynamic programming . . . 13

3 Methodology **15**
- 3.1 Reformatting the PDB . . . 15
- 3.2 Statistical analysis of extracted data . . . 15
- 3.3 Aside: Translating torsion angles . . . 17
- 3.4 Exploring conformational variation with SVD . . . 18
- 3.5 Obtaining canonical torsion angles with SVD . . . 19
- 3.6 Obtaining structures with SDD . . . 20
- 3.7 Using dynamic programming to predict protein structure . . . 23
  - 3.7.1 Combining clusters . . . 23
  - 3.7.2 Pseudo-code for SPAA . . . 24
  - 3.7.3 Potential problems with SPAA . . . 24
- 3.8 Comparing actual and predicted conformations . . . 28
  - 3.8.1 Multiple predictions for short sequences . . . 29

3.8.2 Predicting pieces of proteins . . . 29
3.8.3 Prediction of an entire protein . . . 29

4 Results **30**
4.1 Statistical analysis of extracted data . . . 30
4.2 Exploring conformational variation with SVD . . . 31
4.3 Obtaining canonical torsion angles with SVD . . . 37
4.3.1 Obtaining structures with SDD . . . 37
4.4 Comparing actual and predicted conformations . . . 41
4.4.1 Multiple predictions for short sequences . . . 41
4.4.2 Predicting pieces of proteins . . . 42
4.4.3 Predicting an entire protein . . . 45
4.5 Discussion . . . 47

5 Conclusions **48**
5.1 Future Work . . . 49

A Resulting singular values from SVD **53**
A.1 Sequences of Length 3 . . . 54
A.2 Sequences of Length 4 . . . 55

B 3-dimensional plots of U obtained from SVD **58**
B.1 Sequences of Length 3 . . . 59
B.2 Sequences of Length 4 . . . 61

C Ramachandran plots with clusters **64**
C.1 Sequences of Length 3 . . . 65
C.2 Sequences of Length 4 . . . 68

D Prediction of structures from the PDB **73**

E Protein 1crn **76**
E.1 Sequence . . . 76
E.2 Torsion angles . . . 77

List of Figures

1.1 A 3-dimensional representation of the protein hemoglobin, obtained from the Protein Data Bank [3]. . . . 2
1.2 A list of amino acids . . . 2
1.3 An abstract illustration of an amino acid . . . 3
1.4 An illustration showing ϕ, ψ angles . . . 3

2.1 Number of entries in the PDB . . . 8
2.2 A Ramachandran Plot . . . 9
2.3 A possible class hierarchy produced by SDD . . . 13

3.1 Methodology diagram . . . 16
3.2 Protein-structure prediction diagram . . . 17
3.3 Translated Ramachandran plot . . . 18
3.4 SVD cluster analysis . . . 19
3.5 Creating a class hierarchy with SDD . . . 21
3.6 Mapping clusters To Ramachandran plot . . . 21
3.7 Building structures from clusters . . . 24
3.8 Illustration of 'non-standard conformation' problem with SPAA . . . 27
3.9 Illustration of 'weak cluster' problem with SPAA . . . 27
3.10 Format of a sequence/structure chart . . . 28

4.1 Statistics for extracted sequences . . . 30
4.2 Scree plots of singular values from sequences of length 3 . . . 32
4.3 Scree plots of singular values from sequences of length 4 . . . 33
4.4 SVD clusters mapped to Ramachandran plots from sequences of length 3 . . . 34
4.5 SVD clusters mapped to Ramachandran plots from sequences of length 4 . . . 35
4.6 Original vs. clustered Ramachandran plots . . . 36
4.7 Plots of canonical bond angles . . . 38
4.8 Clustering results of SDD vs SVD . . . 39
4.9 SDD clusters shown on Ramachandran plots . . . 40
4.10 Multiple predicted structures for a sequence . . . 41
4.11 Sequence/structure chart for 1shr . . . 42
4.12 Sequence/structure chart for 1h47 . . . 43
4.13 Sequence/structure chart for 1qvn . . . 43
4.14 Sequence/structure chart for 1u0f . . . 44
4.15 Predicted vs Actual Structure of protein 1crn . . . 46

D.1 Sequence/structure chart for 1h9i 74
D.2 Sequence/structure chart for 1qw7 74
D.3 Sequence/structure chart for 1seq 75
D.4 Sequence/structure chart for 1v1j 75

Chapter 1

Introduction

1.1 Motivation

Proteins are complex macromolecules that are fundamental to every living organism. The physical shape of a protein, that is its conformation, is primarily what determines its function. The conformation of a protein is believed to be determined by the amino acid sequence encoded for it in DNA. Since proteins constitute the building blocks of life, understanding protein structure permits a deeper understanding of living systems. Elucidating protein conformation is therefore a compelling priority of modern science.

1.1.1 Protein structure

Protein structure is complex. A hierarchy of four interacting levels of structure has been used to describe the structures that proteins form. The primary structure is a linear strand of amino acids that form the backbone of the protein, and are believed to determine almost all of its structure. The secondary structure is the arrangement of sections of the primary sequence into structural elements, such as α-helices, β-sheets, turns, etc. The tertiary structure is the 3-dimensional arrangement of the secondary structural units. The quaternary structure is the combination of two or more tertiary protein units. See Figure 1.1 for an example of the structure of an actual protein.

There are 20 different amino acids each with a similar base and a unique chemical side-chain. See Figure 1.2 for a list of the 20 amino acids. The side-chain gives each residue unique physical and chemical properties. See Figure 1.3 for an illustration of an amino acid.

Amino acids polymerize by bonding together at their similar base structures to form a chain. The presence of side-chains means that adjacent amino acids can only be in certain relative orientations. These orientations are further constrained by longer-range effects. By using a common reference point, torsion angles between two amino acids can be calculated. Torsion angles represent the degree of physical rotation of two residues relative to each other. Two important angles are the ϕ and ψ angles. See Figure 1.4 for an illustration.

The torsion angles between amino acids are influenced by both short-range and long-range effects. One dominant influence is the physicochemical properties of the residues being joined together [24, 14]. Other relevant influences are: the effects of amino acids

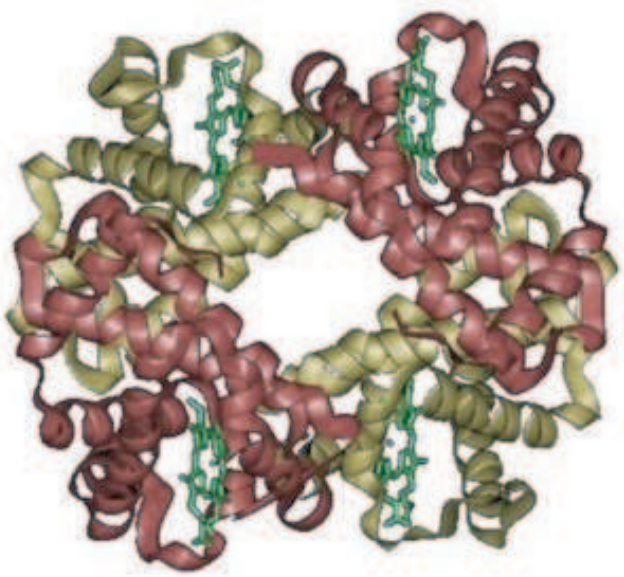


Figure 1.1: A 3-dimensional representation of the protein hemoglobin, obtained from the Protein Data Bank [3].

Amino Acid	Symbol	Symbol
Alanine	ALA	A
Cysteine	CYS	C
Aspartic Acid	ASP	D
Glutamic Acid	GLU	E
Phenylalanine	PHE	F
Glycine	GLY	G
Histidine	HIS	H
Isoleucine	ILE	I
Lysine	LYS	K
Leucine	LEU	L
Methionine	MET	M
Asparagine	ASN	N
Proline	PRO	P
Glutamine	GLN	Q
Arginine	ARG	R
Serine	SER	S
Threonine	THR	T
Valine	VAL	V
Tryptophan	TRP	W
Tyrosine	TYR	Y

Figure 1.2: A list of all the amino acids and their respective 3-letter and 1-letter abbreviations.

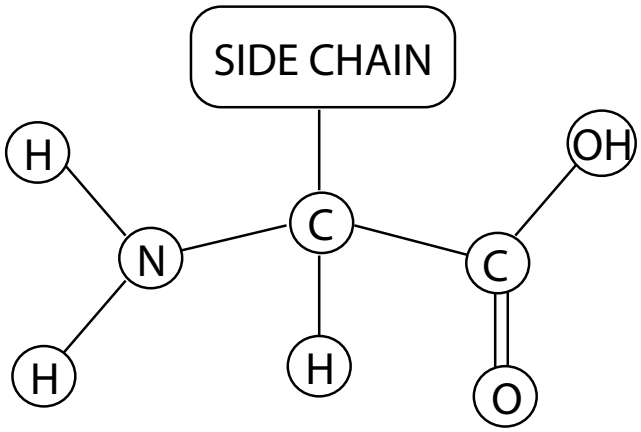


Figure 1.3: An abstract illustration of an amino acid.

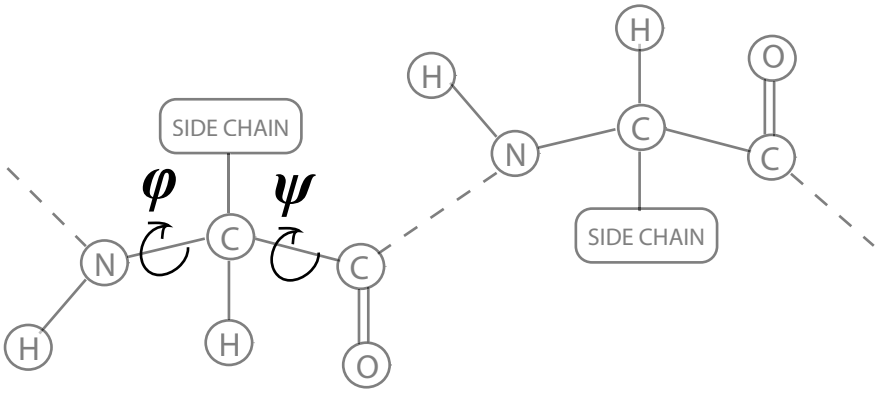


Figure 1.4: An illustration of two amino acid residues joined together to demonstrate the ϕ and ψ torsion angles.

nearby [27, 20], and the presence of amino acids that are far away along the backbone, but physically close because of tertiary structure [9, 15]. The interaction of short-range and long-range effects makes the problem of predicting structure extremely difficult.

1.1.2 Determining protein structure

Currently, protein structure is determined by physical methods. These methods are complex, expensive and slow relative to the number of possible proteins. There are millions of known protein sequences [30] but only thousands of known structures [3].

Prediction of protein structure using a conventional computational approach is intractable. Levinthal [18] pointed out that a protein that consists of a mere 100 amino acids has more than 10^{48} possible conformations, from which finding the correct one could take millions of years.

Despite the inherent complexity of finding protein structure, it is considered a world-wide priority. For instance, the world's fastest supercomputer, BlueGene, is dedicated to this cause [12]. Numerous structural prediction methods have been developed, drawing on many areas of computer, information, chemical, biological and statistical sciences [22]. Because of the vast search space, most methods employ abstraction or search space reduction. Most prediction methods also involve many assumptions. No method available today definitively solves the problem of protein structure determination. In fact, none approach the realm of a practical solution.

1.2 State of the art

Discovering protein structure is a world-wide initiative, the results of which can benefit numerous areas of modern science. However, current structural determination methods are too slow and expensive to find structures for all known proteins. The search space of protein conformation is too immense for computation from first principles, and current prediction methods lack the ability to produce real-world solutions. Any insight into determining protein structure is a worthwhile endeavor.

1.3 Objective

A reductive strategy for conformation prediction is to determine the structure of short chains of amino acids, and then use these to predict possible conformations for longer and longer chains, using a dynamic programming methodology. The problem with this strategy is that there are many observed conformations for short amino acid chains, and so many different ways in which two chains can be combined to form longer chains. The problem remains computationally intractable.

We show that the very large number of observed conformations of short amino acid chains are plausibly due to noise associated with the techniques used to determine conformations. We apply singular value decomposition to denoise observed conformations from the Protein Data Bank. We then use semidiscrete decomposition to cluster these conformation automatically into canonical conformations. Dynamic programming can then be used on this much smaller set of conformations to predict the conformations of longer and longer chains.

1.4 Organization of Report

The required knowledge to understand the methodology and results of this study is provided in Chapter 2. Current structural determination and prediction methods are covered to provide information in the current state of the art. Databases of protein structural data are explained and data-mining techniques that are utilized to prepare data for the algorithm are also detailed. Chapter 3 outlines the methodology employed in the study from data preparation to the construction of a dynamic programming algorithm. Results obtained for each step of the methodology are provided in Chapter 4 and finally conclusions are presented in Chapter 5.

Chapter 2

Background

This chapter provides background into the current state of the art in several fields. An overview of the two most utilized methods of protein structure determination are given to help understand their inherent limitations and problems. Sources of protein structural data will be discussed to explain why new formats are required for data-mining applications. The best conformation prediction algorithms will be discussed to give an overview of how this problem is currently addressed. Finally, data-mining techniques known as matrix decompositions and dynamic programming, which are used in this study, will be explained.

2.1 Structure determination methods

X-Ray Crystallography (XC) and Nuclear Magnetic Resonance (NMR) are the two primary methods of determining a proteins 3-dimensional structure. Currently, XC accounts for 85% of all known structures and NMR for 13% [3].

2.1.1 X-Ray Crystallography

X-Ray Crystallography [6] is a technique where the atomic structure of a crystal is obtained by sending x-rays through it and observing the resulting diffraction pattern. When XC is used to determine the structure of inorganic compounds, it is a relatively simple process. However, when applied to organic compounds, the procedure becomes orders of magnitudes more difficult.

XC requires that compounds be crystallized to find their structure. For organic macromolecules like proteins, this first step can be a painstaking process. There is no definitive guide to crystallizing any protein and it is considered somewhat of a 'black art'. XC introduces a bias to datasets of protein structures, as some classes of protein are easier to crystallize than others.

XC is also not a black-box method; it is not as simple as putting a crystallized protein in one end and obtaining a structure out the other. XC provides an electron density map of the crystal being analyzed. Computer and human interaction is required to refine the density map into descriptions of specific atoms and their coordinates. Errors may be introduced during this step.

The results of XC are also limited by the resolution of the process. A resolution of 2.5Å(Angstroms) is typical for protein structures. The uncertainty of the position of an atom in the structure is approximately $\frac{1}{5}$ of the resolution. A temperature factor is also supplied for each atom obtained with XC. It records the amount of thermal motion observed in the electron density map, which translates into uncertainty of that atom's position.

2.1.2 Nuclear Magnetic Resonance

Solution Nuclear Magnetic Resonance uses the magnetic property of atomic nuclei to determine molecular information. It has been applied to determine organic structures and is employed in many areas of bioscience. NMR's most commonly known application is Magnetic Resonance Imaging, or MRI.

NMR does not require a protein to be crystallized. However, since NMR is only reliable for proteins under a certain mass, a bias is introduced. NMR is also not a black-box method; it produces a series of possible models which can be refined to provide a final model.

There is an inherent error rate of a structure obtained with NMR due to the methodology. For most protein structures the error is less than 2Åper atom.

2.2 The Protein Data Bank

The Protein Data Bank (PDB) [3] is the world-wide repository for known protein structures. The PDB has grown rapidly since its beginnings in 1972 and is currently growing by more than 5,000 entries per year. Figure 2.1 demonstrates the rate of growth of the PDB. Despite the numerous contributions, the PDB still only represents a tiny fraction of known proteins. The gap between proteins that are known to exist and proteins with determined structure is actually widening [30].

While any determined protein structure can be submitted to the PDB, there are rigorous quality standards that must be met for any model to be accepted. This insures that the information in the PDB is reliable and verifiable.

The PDB format was designed to be flexible and robust in an attempt to incorporate all possible knowledge accumulated for each protein. This does have the effect that, for any specific study, the native PDB is non-optimal. Problems with the native PDB format from the point of view of this study are:

- Each PDB entry is kept as an individual file.
- Structures are represented as 3-dimensional atomic coordinates.
- Extraneous information is present, i.e. remarks about crystallization techniques.

2.3 The Ramachandran plot

Ramachandran [21] devised an area plot of the two main angles of residue-to-residue bonding in a protein chain. It is commonly referred to as a Ramachandran plot and graphs the ϕ, ψ torsion angles between pairs of amino acids. The plot was originally conceived to predict conformations of individual amino acids by examining the physicochemical constraints that limit their possibilities. The restrictions on ϕ, ψ angle pairs estimated by Ramachandran

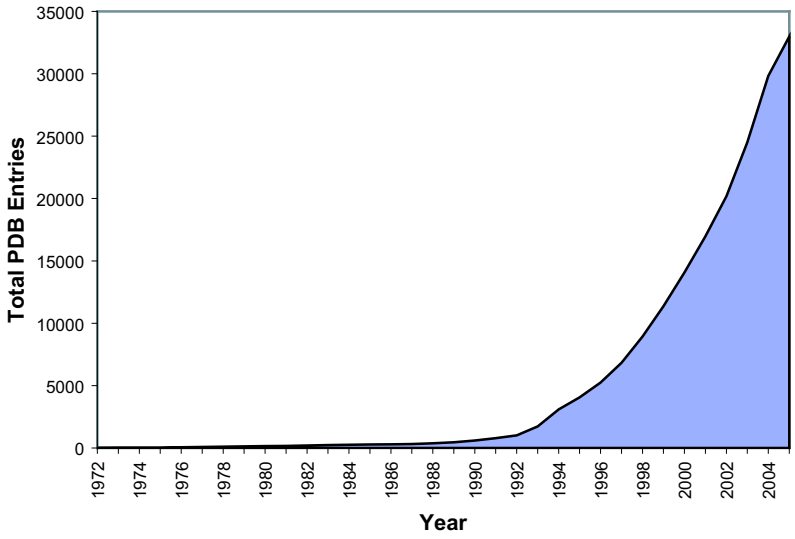


Figure 2.1: Number of structures contained in the Protein Data Bank versus time [3].

have since been experimentally verified and are remarkably accurate [11]. Figure 2.2 shows an example Ramachandran plot, from Kleywegt and Jones [16], for the amino acid GLY.

The Ramachandran plot is now commonly used as a quick method of viewing a torsion angle space in specific contexts. For example, instead of a plot for all possible conformations of the amino acid ALA, a modern Ramachandran plot will show all possible conformations of ALA in a particular protein. These types of plot can be generated in real time from the PDB website [3].

Areas on the Ramachandran plot represent possible secondary structures for that residue pair. Figure 2.2 provides labels for some common structures. Secondary structures are not necessarily apparent from just viewing a Ramachandran plot, however. This would require a set of plots that represent a sequence of amino acids. For example, one point that lies within the α-helix region may actually be part of a more complicated turn which happens to pass through that conformational space.

2.4 Structure prediction algorithms

2.4.1 Secondary structure prediction

Ab initio secondary structure prediction algorithms date back 30 years to when the first databases of protein structure were created. It was theorized that protein conformation could be defined using this limited search space by simplifying complex protein structure to basic secondary structural elements. Commonly, prediction algorithms map an input of a sequence of amino acids into members of a 3-state representation from the set {α-helix, β-sheet, coil} or more recently to members of an 8-state representation from the set {α-helix,

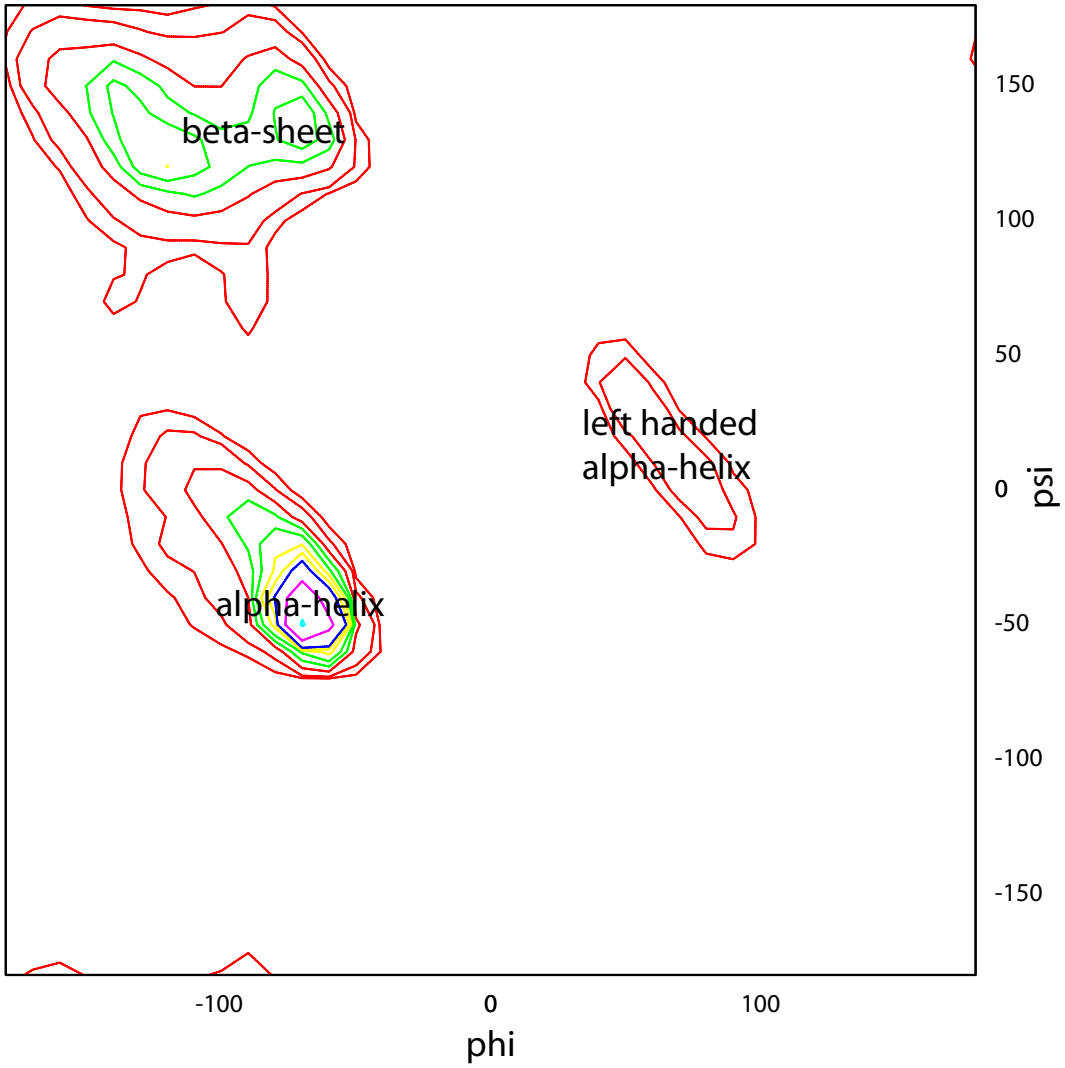


Figure 2.2: A typical Ramachandran plot for the amino acid GLY obtained from Kleywegt and Jones [16]. Areas of the plot which can correspond to common secondary structures are indicated.

β-sheet, 3-helix, π-helix, turn, bend, strand, random}). Properties of the protein can then be gleaned by comparing the arrangement of the predicted secondary structures against known structural motifs.

The Chou-Fasman [4] and GOR [8] algorithms are the earliest attempts and are the precursors to modern prediction algorithms. Chou-Fasman used a simple table of propensities for a particular amino acid to be found in secondary structural states. The propensities themselves were originally calculated using a database of 29 proteins, which represented all the available structures at the time. GOR, using a similar heuristic of structural propensity, expanded on the Chou-Fasman algorithm by introducing a sliding window which took into account the effects of amino acids at longer range. These algorithms demonstrated the validity of this mode of reasoning by achieving around 55% accuracy in 3-state prediction. Though simplistic, both algorithms could predict states at greater than random rates. However, it was obvious that more complicated techniques would have to be devised to reach useful levels of prediction accuracy.

A few selected, modern secondary structure-prediction algorithms are described below. They are generally considered the best in their field [5, 22].

PSIPRED

Created in 1999, PSIPRED [13] utilizes a simple model of two feed-forward neural networks to predict secondary structure. PSIPRED uses results from PSI-BLAST [1], a popular bioinformatics program that statistically matches amino acid sequences. When an input sequence is entered into PSIPRED, it performs a PSI-BLAST search on that sequence to obtain any similar sequences from the PDB. It then averages the resulting secondary structures using a neural network to provide a prediction for the original input. The system is remarkably simple and is able to achieve consistent accuracy rates of 76%, similar to more complicated techniques.

Predator

The Predator algorithm was originally developed by Frishman and Argos in 1996 [7], and has since been improved upon [22]. It boasts a 75% accuracy rate for 3-state secondary-structure prediction. Predator was designed to combine a local nearest-neighbour approach with the effects of long-range interactions. Hydrogen-bonding propensities calculated from a non-redundant derivative of PDB are the critical component of Predator.

Predator uses seven statistical measures per residue of the input sequence to determine secondary structure. Three measures are related to long-range hydrogen-bonding propensities. Another three are based on secondary structural propensities acquired from protein structure data. The seventh incorporates an amino acid window of length four to provide a probability of a turn. A decision tree is then used to combine the seven measures and predict the structure.

SSPro

SSPro was originally developed in 1999 as a 3-state prediction algorithm [2] and was later improved in 2002 to produce 8-state predictions (and renamed SSPro8) [19]. SSPro employs

a neural network to learn secondary structural states for amino-acid sequences from previously acquired data. The network consists of 11 bidirectional recurrent neural networks that transfer information between the input and the output sequence. It functions with a sliding window on the input sequence. Essentially this means that the output being produced from SSPro is fed back into the system to influence later results.

SSPro is trained on a derivative of the PDB which has sequence and structure homologues removed. This is done to keep the neural network from becoming over-trained on particular sequences. For example, the dataset used for the original SSPro contained only 1180 structures. SSPro and SSPro8 both have accuracy in the 80% range on independent test sets.

Secondary-structure prediction algorithms have advanced significantly since the original implementations. The results obtained by any of these methods are moderately useful in terms of providing better understanding of protein structure and possibly function. However, the usefulness of predictions supplied by these processes are severely restricted due to the inherently limited conformation space to which they map. For example, much information is lost by the abstraction that an α-helix is a single state. All of the above methods also use limited derivatives of the PDB, for reasons which may be valid related to the methodology of the approach, but nonetheless place artificial limits on the data available.

2.4.2 Comparative modeling

Comparative modeling is a class of structural-assembly algorithms that rely on determining substructures or simple structural alphabets from which proteins can be formed. This is in contrast to the *ab initio* approach.

SwissModel

SwissModel [23] is a complex 3-dimensional protein-modelling application developed by the Swiss Institute of Bioinformatics. SwissModel takes a sequence of amino acids as input and attempts to assemble a 3-dimensional representation by finding homologous structures within the PDB. The algorithm breaks the input sequence into segments of at least 20 amino acids long and performs a BLAST search in the PDB to find similar sequences [1]. 3-dimensional structure(s) for each subsequence returned from BLAST are then taken from the PDB data. The algorithm attempts to combinatorially join all structures together based on their conformations. The final stage then filters structures by physical plausibility.

SwissModel can generate very accurate 3-dimensional representations of protein structure. The major limitation of this approach is that proteins with large areas of structural homology must already exist in the PDB. SwissModel does not perform any generalization, abstraction or extrapolation.

Comparative modeling is effective in generating accurate and complete (that is, 3-dimensional) conformations. The results from a comparative modeling approach are more useful than results obtained from secondary-structure prediction (secondary structure is below 3-dimensional conformation in the protein structure hierarchy). Many comparative modeling techniques exist. However, they all suffer from the same limitation. Large structural homologues must already exist in a database. This narrows the generality of a comparative modeling approach; only conformations of proteins similar to those already discovered can be predicted.

2.5 Matrix decompositions

Matrix decompositions are relatively simple procedures which can yield powerful insights into structure within data. They are employed in the field of data mining as unsupervised methods of classification, pattern recognition and structural analysis [25]. They are termed decompositions since they involve separating an input dataset into several components, each of which contains information on aspects of the original data. Two well-known matrix decompositions used in this study are described below.

2.5.1 Singular Value Decomposition

The singular value decomposition (SVD) of a matrix A is:

$$A = USV^T \tag{2.1}$$

where T indicates the transpose. If A is an $n \times m$ matrix; U is an $n \times m$ matrix, S is a diagonal matrix of $m \times m$ with non-increasing values (the singular values σ_1,...,σ_m) and V is a $m \times m$ matrix.

SVD has the property of rotating the original space of the matrix A so that variance is maximized in the earliest dimensions. The greatest variance from the data is maximized and represented in the first column of U, the greatest remaining variance is maximized in the second column of U and so on. The magnitude of the singular values contained in S can be used as a measure of how much variation is contained in each respective column [25]. Specifically, let f be the contribution of a singular value; then:

$$f_k = \sigma_k^2 / \sum_{i=1}^{r} \sigma_i^2 \tag{2.2}$$

This property of SVD has two applications in this study. First, as the greatest amount of variance has been captured in the earliest dimensions, these dimensions can be considered to contain the minimal set of components which still accurately describe the data. Values in the later dimensions contain components which are not useful in describing the dataset or are very weakly related to the main structure of the dataset. By examining the contribution of each singular value, a value of k can be found as a dimension at which to truncate the decomposed matrices. By truncating the decomposed matrices at k and then re-multiplying them (as in Equation 2.3), a matrix of similar shape (i.e. $n \times m$) to A is found but with the effects of the weak components removed. Essentially a 'noise'-reduced version of A results.

$$A' = U_k S_k V_k^T \tag{2.3}$$

Second, SVD can also be used as a clustering technique. When $k < 4$, using the same definition of k as above, the columns of US can be directly plotted for a visual, geometrical analysis. Essentially, the Euclidean distance of two points in this new space can be used to determine the similarity between them; points which are proximal are similar. Should different classes of objects exist within the data, clusters of points will emerge.

SVD has been used as an image compression technique, as a noise filter and is often used in web-search engine algorithms [25]. It has more recently been employed in the field of bioinformatics to analyze gene-expression microarrays [28] and in protein folding dynamics simulations [26].

2.5.2 SemiDiscrete Decomposition

The SemiDiscrete Decomposition (SDD) of dimension j of an $n \times m$ matrix A is expressed:

$$A_j = X_j D_j Y_j \tag{2.4}$$

where the entries of the $X, D \epsilon \{-1, 0, +1\}$, X is $n \times j$, D is a diagonal $j \times j$ matrix and Y is $j \times m$. SDD is an iterative algorithm which attempts to find j clusters within the data. SDD is a bump-hunting technique, finding the most significant area or 'bump' within a matrix. SDD then removes the bump and then repeats this process j times, if possible. SDD creates a matrix, X, whose rows represent the original rows of matrix A but are ternary classified ($\{-1, 0, +1\}$) per column. The classes -1 and $+1$ are in some sense opposites, while 0 is a neutral (i.e. not related to class -1 or $+1$) class for a particular level [25].

This classification can be interpreted as a j-deep hierarchy of classes with 3^{j-1} different possible class labels. For example, consider the following X matrix:

$$\begin{pmatrix} +1 & +1 & +1 \\ +1 & -1 & -1 \\ -1 & +1 & 0 \\ \ldots & & \end{pmatrix}$$

which produces the class hierarchy in Figure 2.3:

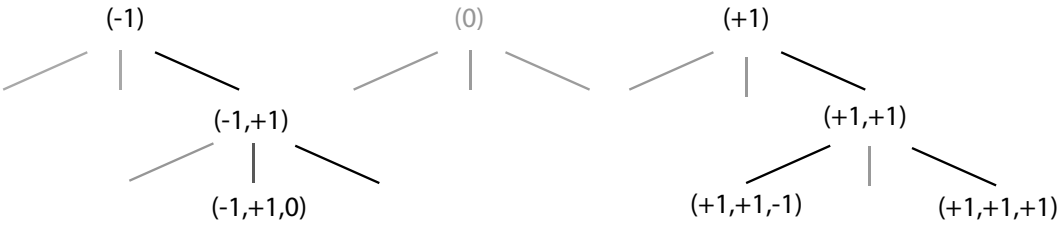


Figure 2.3: A possible class hierarchy that could be generated from SDD.

For many datasets SDD and SVD will agree on the clusters to be found in the data. This is important because the clusters are obtained from SDD much faster than from SVD. Even though SVD reduces the dimensionality of the original data, it still requires a k-dimensional distance calculation per n points to determine clusters. SDD returns a classification tree directly.

SDD is used as a memory efficient replacement for SVD in the field of image processing [31]. SDD is often used in areas similar to those of SVD.

2.6 Dynamic programming

Dynamic programming is a methodology employed to reduce the time required to find solutions to complex problems. Problems which exhibit the properties of overlapping subproblems and optimal substructure can be solved using this method. When solutions to large problems can be created by combining solutions to smaller problems, then the overall complexity depends on the method of attack. If the problem is attacked 'top-down' then

the solution of each subproblem generates a large number of smaller and simpler subproblems to be solved; and each is solved independently. However, the same subproblems are solved repeatedly because they occur as subproblems in many different contexts. Solving the problem 'bottom-up' is clearly more effective: each small subproblem is solved initially and their solutions remembered; these small solutions are then used to generate solutions to many larger subproblems; and the process continues until top-level problems have been solved. This process finds optimal solutions to top-level problems if these depend on optimal solutions to their subproblems.

Protein-structure prediction has overlapping subproblems; this has been demonstrated by comparative modeling approaches. An input sequence of a protein can be broken down into subsequences for which substructures can be generated and later recombined into a final structure. We don't know if proteins exhibit optimal substructure because, so far, there is no optimal solution to the protein folding problem.

Currently protein structure prediction is an expensive, time-consuming process that produces inaccurate results. Large databases of protein structure exist, but are not in useful formats for particular types of study. Current state of the art protein-structure prediction algorithms show that it is possible to predict structure through amino acid sequence analysis as well as assembling conformational elements into larger structures. However, both of these methods are limited.

Well-known data-mining techniques can be applied to information from the PDB. They have the potential to remove error, noise or unrelated variation from the data. They also have the potential to find conformational elements related to amino acid sequences which can then subsequently be used in a dynamic-programming algorithm to create predictions for large structures.

Chapter 3

Methodology

This chapter outlines the methods used to create a data-driven, dynamic-programming protein-structure prediction algorithm. First the PDB is re-structured to allow information to be extracted from it. We then show that the range of conformations exhibited by occurrences of particular short amino acid sequences reflect a small number of possibilities distorted by noise, rather than a wide range of possibilities. Sets of torsion angles are processed using SVD and automatically clustered using SDD. The resulting data from these steps is used in a dynamic-programming algorithm to build protein structures in a bottom-up fashion. Figure 3.1 provides an illustration of the methodology involved to acquire data for the algorithm. Figure 3.2 provides an illustration of protein-structure prediction algorithm.

3.1 Reformatting the PDB

For matrices of torsion angles to be easily extracted from the PDB, a new format is required. Each individual PDB file from the main file server located at "ftp.rcsb.org/pub/pdb/data/structures/all/pdb/" was downloaded. This process finished on October 18, 2004.

Each file was unzipped and entered into a bioinformatics program DANG [29]. DANG accepts any PDB file as input and returns each amino acid in the protein's sequence with its respective torsion and rotamer angles. The output of DANG was parsed for each PDB file and appended to a text file, where one line is one PDB entry, in the following format:

$$(pdbid)amino_1 : \phi_1 : \psi_1 : \chi_1^1; \chi_1^2; \chi_1^3; \chi_1^4; \chi_1^5, amino_2 : \phi_2 : \psi_2 : \chi_2^1; \chi_2^2; \chi_2^3; ... \tag{3.1}$$

The χ values represent rotamer angles which are not used in this study but were included for future studies.

Any PDB file which DANG was unable to process, or was not a protein, was excluded. A total of 25,288 protein files were parsed and entered into the new dataset.

3.2 Statistical analysis of extracted data

An application was developed to take a sequence of amino acids from the new data set as input and return every instance of that sequence, along with the related torsion angles, in

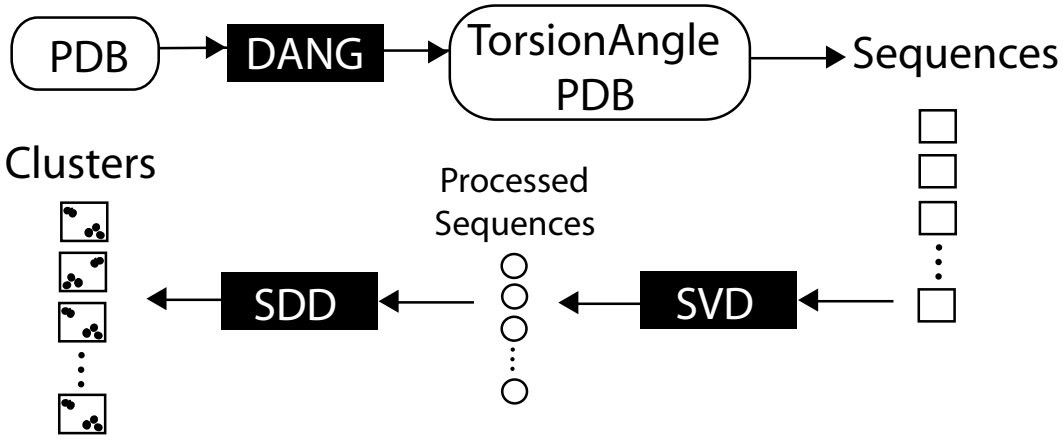


Figure 3.1: A diagram of the methodology employed.

comma delimited format. The output corresponds to an $n \times m$ matrix where n is the number of occurrences of that sequence in the data and m is the number of torsion angles. Each of the n rows is one structure and each pair of columns are the ϕ and ψ angles between a pair of amino acids. For example, a sequence of amino acids:

$$A - B - C - D \tag{3.2}$$

would produce the matrix,

$$\begin{pmatrix} \phi^1_{AB} & \psi^1_{AB} & \phi^1_{BC} & \psi^1_{BC} & \phi^1_{CD} & \psi^1_{CD} \\ \phi^2_{AB} & \psi^2_{AB} & \phi^2_{BC} & \psi^2_{BC} & \phi^2_{CD} & \psi^2_{CD} \\ \cdots & & & & & \\ \phi^n_{AB} & \psi^n_{AB} & \phi^n_{BC} & \psi^n_{BC} & \phi^n_{CD} & \psi^n_{CD} \end{pmatrix}$$

As an example of an actual instance, 3 rows of the matrix for sequence CYS-THR-ALA, which corresponds to 3 occurrences of that sequence in the data, are shown below:

$$\begin{pmatrix} -70.2 & 149.7 & -107.7 & -3.6 \\ 156.2 & 119.4 & -142.2 & 141.4 \\ -107.3 & 140.2 & -93.8 & -6.7 \\ \cdots & & & \end{pmatrix}$$

Even this small selection of torsion angles demonstrates the different values, and therefore conformations, that a short amino acid sequence can take.

Sufficient data on protein subsequences is required for a data-driven protein structure-prediction algorithm. The length of a sequence (the number of amino acids in a sequence, e.g. CYS-THR-ALA = length 4) is roughly inversely proportional to the number of occurrences it will have in the PDB. A simple assumption is that each amino acid added to a sequence will reduce the rate of occurrence by a factor of 20 (since there are 20 amino acids). For example, if A-B-C has 100 occurrences then A-B-C-D will probably have 5 occurrences. A brief, preliminary analysis of the rate of occurrences for sequences of length 5 showed the

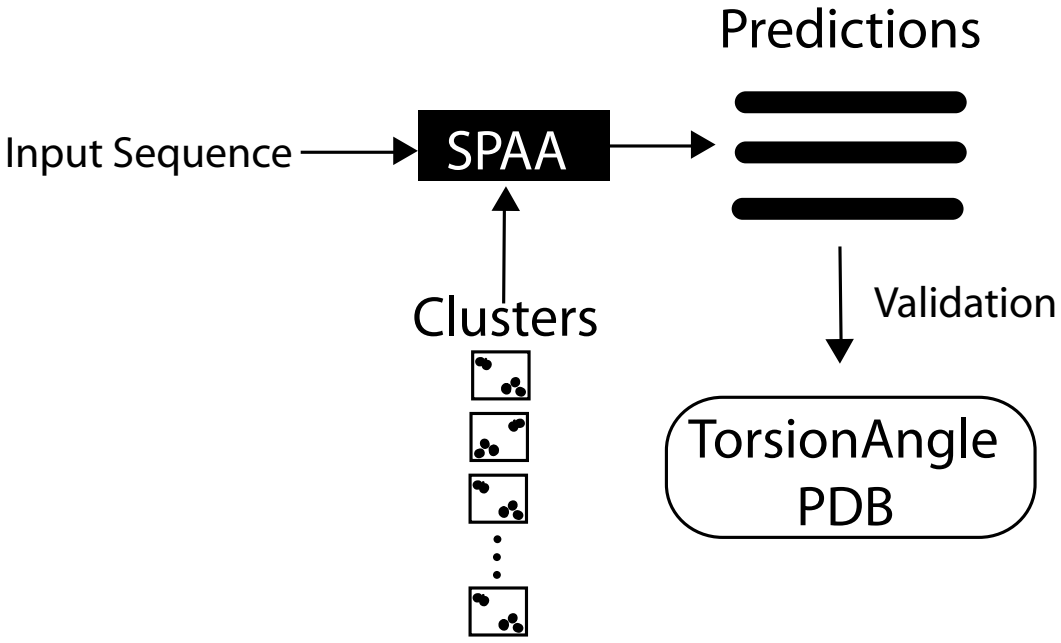


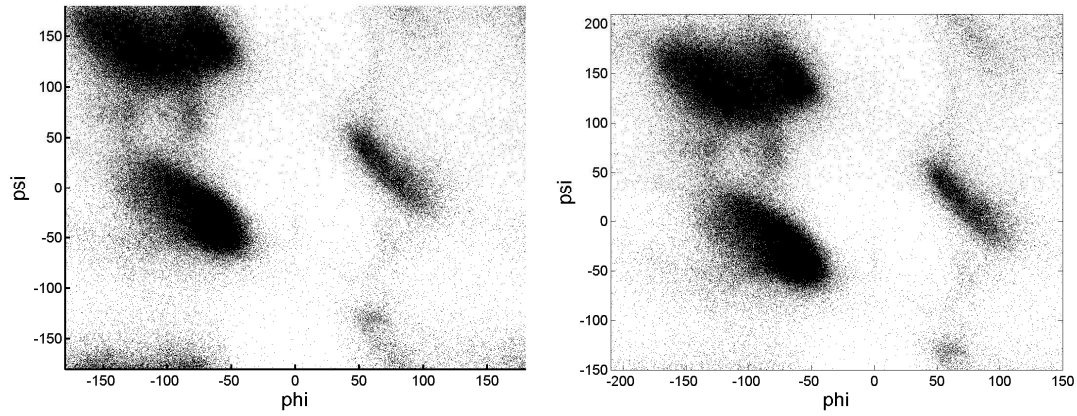
Figure 3.2: A diagram of the algorithm developed to predict protein structure.

frequency to be less than 20 for some combinations of amino acids. This was considered just below the threshold of a useful amount of data. Sequences of length 3 and length 4 were chosen as the basis of our process to maximize the availability of data.

Every possible combination of length 3 and length 4 amino acid sequences were extracted from the data set and saved. There are $20^3 = 8000$ combinations of length three and $20^4 = 160,000$ combinations of length four.

3.3 Aside: Translating torsion angles

All torsion angles are translated to ensure more appropriate clustering results before any of the matrix decompositions are performed. A known problem in clustering torsion angles is that the values wrap around; -180^o is identical to 180^o. Any numerical clustering technique will treat these values as very far apart. To counteract this effect, it is possible to shift the values of the torsion angles so that the wrap-around effect is minimized. The ϕ angle is translated -30^o and the ψ angle is translated $+30^o$ for each entry. An illustration of the effect is shown in Figure 3.3; the entire plot is shifted to place the angle discontinuities in areas of low density. While this does not completely eliminate the problem it reduces its effect. When torsion angles are subsequently analyzed after some procedure, they are shifted back $+30^o$ in the ϕ angle and -30^o in the ψ angle.



(a) **(b)**

Figure 3.3: **(a)** Ramachandran plot of 519,024 torsion angles independent of residue type. Areas of high density that correspond to common secondary structural elements can be seen to wrap around the plot. **(b)** The same Ramachandran plot translated by -30^o in the ϕ angle an $+30^o$ in the ψ angle, which minimizes the wrap-around effect.

3.4 Exploring conformational variation with SVD

The native SVD function supplied by MatLab 6.5r13 was used to calculate the decomposition of every individual length 3 and length 4 matrix. A random sample of sequences were chosen for analysis to obtain a representative result set for both length 3 and length 4.

The U matrix (resulting from the decomposition, see Equation 2.1), was inspected geometrically by plotting the first three columns. (If the magnitude of the singular values related to the U matrix drops significantly after the second value, only the first two columns were plotted.)

Clusters of points were then observed visually. An approximate 3-dimensional volume was created to represent an observed cluster by selecting points near its edge (see Figure 3.4.a). Since the U matrix produced for an input matrix A has the row order preserved (e.g. row x of U represents row x of A), the rows of the selected points map directly to the original input values (see Figure 3.4.b). The original entries in A were then used to generate an approximate area on a Ramachandran plot for each torsion angle pair in the sequence that A represents (see Figure 3.4.c). The series of Ramachandran plots obtained in this fashion were then analyzed to observe resulting structures. Since this requires extensive human interaction to complete, this step was not automated but used in an exploratory way to determine the effects of matrix decompositions on this data. Clearly this methodology is time intensive and does not scale.

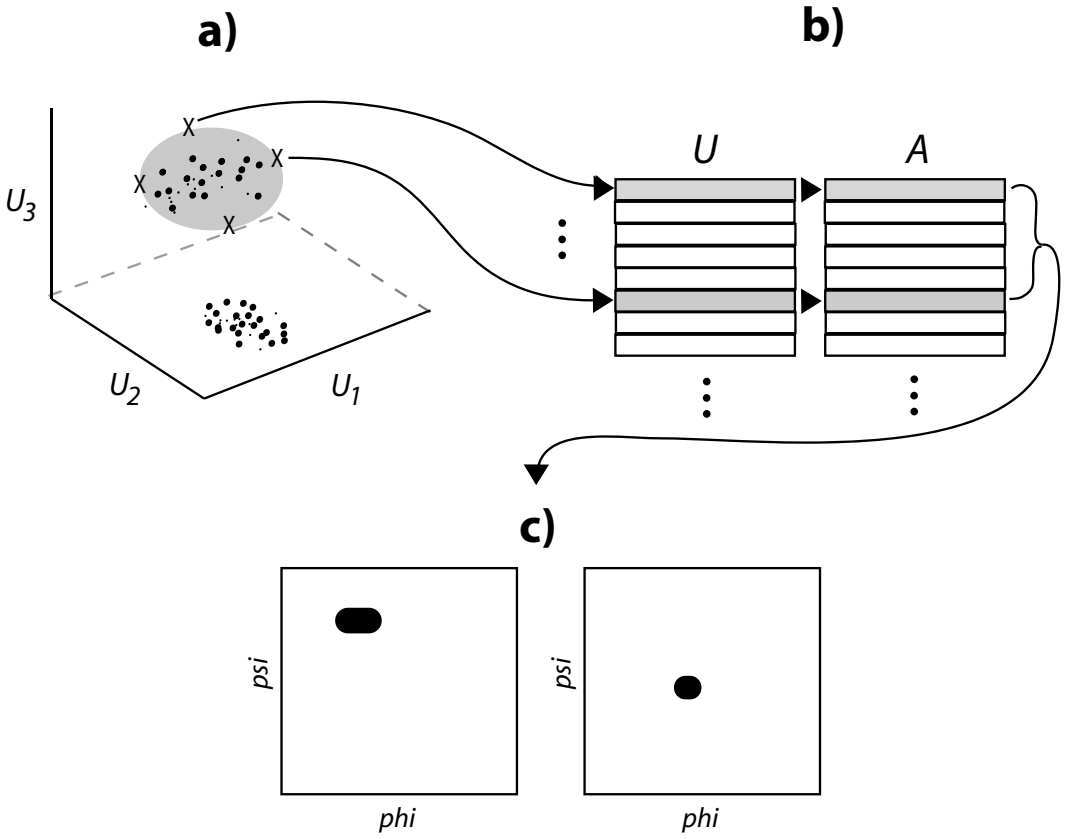


Figure 3.4: A diagram showing how an SVD cluster analysis is performed.

3.5 Obtaining canonical torsion angles with SVD

The torsion-angle data provided in the PDB contains sources of error. An inherent level of error results from the structural determination methods used to acquire PDB data. Experimental error due to the human interaction component of determination methods may also be introduced (see Section 2.1). Other sources can influence torsion angle data for specific sets, for example, long range structural interactions or effects of other residues further down the protein backbone. Sources of variation may obscure conformational structure by introducing irrelevant variation. It may be possible to remove some of these effects and produce clearer sets of torsion angles for specific amino-acid sequences.

The singular values contained in diagonal of the S matrix were examined. A quick visual analysis was performed by plotting the singular values on a scree plot to see if and where the magnitude of the singular values dropped significantly. A value, k, could then be determined at which to truncate the decomposed matrices. Dimensions equal to and less than k should contain a reduced set of components which still accurately describe the original data. The truncated, decomposed matrices were re-multiplied to obtain a matrix A_k of similar shape to A (See Equation 2.3). Ramachandran plots were then generated

from A_k and compared to Ramachandran plots generated from A to examine the effect of decomposition and truncation.

It was determined from the results that SVD was removing variation unrelated to main structure and providing a set of canonical torsion angles. The SVD truncation process was subsequently automated for the entire set of 168,000 extracted sequences. Thee truncation value k was chosen independently for each sequence by examining the contribution of each singular value using Equation 2.2. A threshold value was determined experimentally from the manual examination above; k was chosen so that $f_k < 0.05$.

3.6 Obtaining structures with SDD

The results from Section 3.4 indicated that structural possibilities can be obtained using matrix decompositions. The step in Section 3.5 produced clearer versions of the original data. Generalized conformational possibilities for every possible combination of amino acids are required as the basis of a protein assembly algorithm. The conformations determined by a decomposition can potentially be found by hand for a specific sequence, but doing so for all 168,000 sets is not feasible. A method to quickly and easily automate this process was required. Subsequently, SDDPACK by Kolda and O'Leary [17] was used in MatLab v6.5r13 to perform the decomposition of every resulting A_k matrix from Section 3.5.

The clustering results of SDD and SVD were compared to determine their similarity. The first three columns of the resulting U matrix obtained from SVD were used to create a 3-dimensional plot for geometrical interpretation. Cluster labels from SDD were used to plot the points of the 3-dimensional graph as particular colours and shapes. This allows a quick visual analysis to determine the degree to which the clustering methods agree; if the SDD colours and shapes are grouped in the SVD clusters, then they produce similar results. Only a small, randomly selected fraction of all sequences were examined in this fashion to obtain a generalized conclusion. The results indicated that SDD and SVD obtained similar clustering effects.

The first three columns of the X matrix (resulting from the decomposition, see Equation 2.4) were used to determine a class hierarchy. Since the X matrix produced for an input matrix A has the row order preserved (e.g. row x of X represents row x of A), a direct mapping of the SDD cluster labels to the original structures in A is possible (see Figure 3.5.a). Using the ternary class labels of the 1^{st} column of X, every entry in A was entered into one cluster labeled (-1), (0) or $(+1)$. Using the labels of the 1^{st} and 2^{nd} columns of X, every entry in A was entered into one cluster labeled $(-1,-1)$, $(-1,0)$, $(-1,+1)$, ..., $(+1,+1)$. Using the labels of the 1^{st}, 2^{nd} and 3^{rd} columns of X, every entry in A was entered into one cluster labeled $(-1,-1,-1)$, ..., $(+1,+1,+1)$. Since this is a hierarchical method, every member of a child class will also belong to its respective parent. Figure 3.5.b demonstrates this concept visually using only 2 columns for simplicity.

For each SDD cluster label a subset of the original matrix A is produced. For example, assume an SDD was performed on a matrix A which represented the amino acid sequence $A-B-C-D$, for each cluster the following would be obtained:

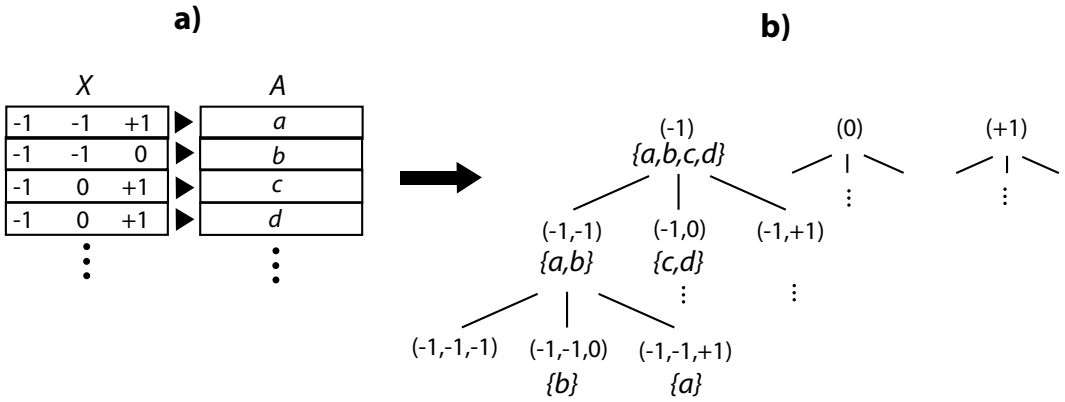


Figure 3.5: A visual representation of using SDD cluster labels to create a class hierarchy.

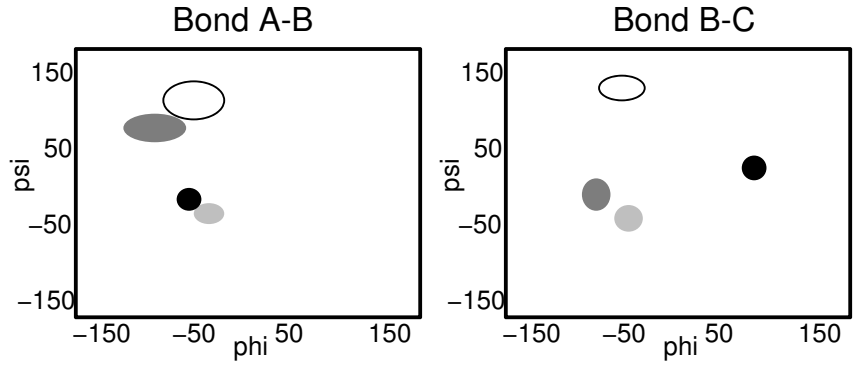


Figure 3.6: A hypothetical set of clusters obtained from a SDD decomposition on a sequence A-B-C where each different shade on the plot corresponds to an individual cluster. The ellipse has a centroid at the mean values for a cluster while the area of the ellipse represents the range of conformational possibilities provided by the respective standard deviations.

Listing 3.1 Pseudo-code for the filter applied to a hierarchy obtained using SDD as a clusterer.

```
if j < 5
   disregard this cluster and its children
if standard deviation > 20^o for any φ or ψ
   apply filter to cluster's children
else
   this cluster is a valid structure
```

$$\begin{pmatrix} \phi^1_{AB} & \psi^1_{AB} & \phi^1_{BC} & \psi^1_{BC} & \phi^1_{CD} & \psi^1_{CD} \\ \phi^2_{AB} & \psi^2_{AB} & \phi^2_{BC} & \psi^2_{BC} & \phi^2_{CD} & \psi^2_{CD} \\ \multicolumn{6}{c}{...} \\ \phi^j_{AB} & \psi^j_{AB} & \phi^j_{BC} & \psi^j_{BC} & \phi^n_{CD} & \psi^j_{CD} \end{pmatrix}$$

where j represents the number of entries in A which were found to belong to that cluster and $0 \leq j \leq n$ (see Section 3.2 for the original description of A). Independently, for each cluster, the resulting values were aggregated together by calculating the mean and standard deviation of each column. This corresponds to obtaining one general structural possibility that each cluster may represent. The mean of each ϕ, ψ pair corresponds to a point on a Ramachandran plot and the respective standard deviations define an area around that point. Figure 3.6 provides an illustration of this concept. The filter shown in Listing 3.1 was then applied to the hierarchy, starting from the top clusters.

Clusters will be eliminated because of the threshold on j in the filter. The purpose of the SDD clustering step is to find generalized structural possibilities; any cluster smaller than the threshold on j would represent a rare conformation in the PDB data.

The threshold value for the standard deviation was chosen to obtain valid structures. SDD could potentially group outliers together in some clusters, e.g. structures which are not similar to any other cluster or each other. This would correspond to points from varying locations on a set of Ramachandran plots and therefore high standard deviations. The threshold value for the standard deviation was also chosen to use the best structures available. Since SDD provides a class hierarchy, each child provides a more refined separation of structures than its parent. More accurate clusters are obtained by moving down the class hierarchy to values with lower standard deviation. For example, a parent may have children that represent subtly different conformations.

Each valid cluster left after application of the filter was saved to a data file. The output file for sequence A-B-C (and similarly for sequences of length 4), where one row represents one cluster, has the following format:

$$\phi_{AB}, \psi_{AB}, std\phi_{AB}, std\psi_{AB}, \phi_{BC}, \psi_{BC}, std\phi_{BC}, std\psi_{BC}, j, \frac{j}{n}, clusterid \tag{3.3}$$

3.7 Using dynamic programming to predict protein structure

The protein-structure prediction algorithm was developed using a dynamic-programming approach. It will be referred to as SVD/SDD Protein Assembly Algorithm, or SPAA, for ease of reference. SPAA accepts a string of amino acids as input and returns a set of predictions for that sequence in the form of torsion angles. A detailed explanation of how SPAA works can be found in the pseudo-code version in Section 3.7.2.

SPAA functions by splitting the input sequence into its respective subsequences of length 3 and 4 and then using the clusters obtained in the previous steps to build the structure in a bottom-up fashion. For example, if the input sequence is *A-B-C-D-E-F-G-H*, SPAA will split it into {*A-B-C*, *B-C-D*, *C-D-E*, *D-E-F*, *E-F-G*, *F-G-H*, *A-B-C-D*, *B-C-D-E*, *C-D-E-F*, *D-E-F-G*, *E-F-G-H*}.

SPAA begins building from the subsequences of length 4. Following the previous example, this would be {*A-B-C-D*, *B-C-D-E*, *C-D-E-F*, *D-E-F-G*, *E-F-G-H*}. SPAA iteratively attempts to extend each structure it has obtained with the appropriate clusters using sequences of length 4 first (see Combining Clusters: Section 3.7.1 for a detailed explanation of this process). If none of those clusters have conformations that agree for the overlapping amino acids, it will then attempt to use clusters from sequences of length 3. For example, starting with subsequence *C-D-E-F*, SPAA will try to extend it with *A-B-C-D* to make *A-B-C-D-E-F* and *E-F-G-H* to make *C-D-E-F-G-H*. If no matching structures in the overlapped region can be found, SPAA will try to extend *C-D-E-F* with *B-C-D* to make *B-C-D-E-F* and with *E-F-G* to make *C-D-E-F-G*. Any valid structures obtained are entered back into a set for SPAA to extend again on the next iteration.

Clusters for sequences of length 4 are used first as they contain greater context than sequences of length 3; they contain conformations related to a more specific sequence of amino acids. The clusters for length 3 contain less contextually relevant conformations and will create weaker predictions. This is a design choice resulting from lack of data. If enough sequences of length 5 had been available, SPAA would have utilized those first instead.

SPAA uses every subsequence of length 4 as the base structures. This design choice was made because early prototypes of SPAA were unable to generate entire structures for large sequences, failing at arbitrary points within an input sequence. Because of the low frequency of some conformations in the data, an early version was unable to return useful predictions. It was decided that by 'growing' every piece of the protein independently, almost complete structures could be generated leaving gaps only at weak spots. This design choice means that SPAA may return pieces of a protein rather than a whole structure.

3.7.1 Combining clusters

It is possible to assemble larger structures from the data produced in Section 3.6. Sequences can be joined together by overlapping similar amino acids. For example, $A-B-C-D$ and $C-D-E-F$ can be overlapped on $C-D$. If clusters for the common pair have similar torsion angles, then larger structures that represent $A-B-C-D-E-F$ can be built. Figure 3.7 illustrates this concept and a formal description of the procedure is found below.

Let γ represent a cluster as defined in Equation 3.3 and Γ^z represent the domain of clusters for sequence z, so that $\gamma \in \Gamma$. Let ϕ, ψ, $std\phi$, $std\psi$ represent similar variables as

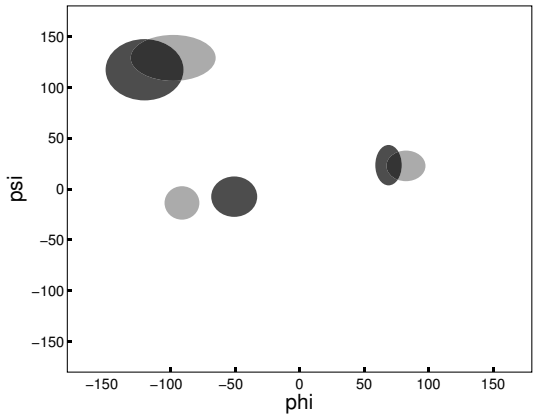


Figure 3.7: Hypothetical clusters obtained for the B-C bond of sequence A-B-C and B-C-D shown on a Ramachandran plot. The light areas represent clusters for one sequence and the dark areas for the other. Structures can be joined together where the light and dark areas overlap.

defined in Equation 3.3. Then γ^z_{XY} represents the torsion angle pair corresponding to the $X-Y$ bond of a cluster for amino acid sequence z. Then define;

$$\gamma^w_{XY} \equiv \gamma^z_{XY} \tag{3.4}$$

iff

$$std\phi^w_{XY} < |\phi^w_{XY} - \phi^z_{XY}| \quad \text{or} \quad std\phi^z_{XY} < |\phi^w_{XY} - \phi^z_{XY}| \tag{3.5}$$

and

$$std\psi^w_{XY} < |\psi^w_{XY} - \psi^z_{XY}| \quad \text{or} \quad std\psi^z_{XY} < |\psi^w_{XY} - \psi^z_{XY}| \tag{3.6}$$

Let a be any arbitrary sequence of amino acids $A-B-C-D$ and let b be any arbitrary sequence of amino acids $C-D-E-F$. Two sequences can be combined where any $\gamma^a_{CD} \equiv \gamma^b_{CD}$ with the resulting longer sequence $A-B-C-D-E-F$ taking the form:

$$\phi^a_{AB}, \psi^a_{AB}, \phi^a_{BC}, \psi^a_{BC}, \frac{\phi^a_{CD} + \phi^b_{CD}}{2}, \frac{\psi^a_{CD} + \psi^b_{CD}}{2}, \phi^b_{DE}, \psi^b_{DE}, \phi^b_{EF}, \psi^b_{EF} \tag{3.7}$$

3.7.2 Pseudo-code for SPAA

Listing 3.7.2 contains a pseudo-code version of the SPAA algorithm. The version of SPAA used in this study was implemented in Java 1.42_08.

3.7.3 Potential problems with SPAA

Based on the methodology used to acquire data for SPAA, the algorithm has potential points of failure.

Listing 3.2 Pseudo-code for SPAA

```
SPAA(string Sequence)

Sub_Sequence3[] = split Sequence into all possible sub-sequences of length 3
Sub_Sequence4[] = split Sequence into all possible sub-sequences of length 4
//Load every cluster for each sub-sequence, it is assumed BaseStructures
//retains the name of the sequence for each cluster for reference later
BaseStructures3 = loadClusters( Sub_Sequence3 )
BaseStructures4 = loadClusters( Sub_Sequence4 )
//Start the prediction using the sequences of length 4 as the basis
Predictions.addAll( BaseStructures4 )
do until (all Predictions.flag == true)
    CurrentPrediction = Predictions.removeNextWithNoFlag
    wasExtended = false
    //Get the clusters for the next relevent four amino acids in the input sequence
    Extension[] = BaseStructures.get( CurrentPrediction.getNext4 )
    GOTO ExtendR
    Extension[] = BaseStructures.get( CurrentPrediction.getPrev4 )
    GOTO ExtendL
    //If no sequence of length 4 could extend the prediction, try length 3
    if !wasExtended
        Extension[] = BaseStructures.get( CurrentPrediction.getNext3 )
        GOTO ExtendR
        Extension[] = BaseStructures.get( CurrentPrediction.getPrev3 )
        GOTO ExtendL
    end if
    //If no length 4 OR length 3 could extend the prediction, and this
    //prediction is 10 or more amino acids long flag the prediction as
    //finished and save it
    if !wasExtended and CurrentPrediction.length > 10
        CurrentPrediction.flag = true;
        Predictions.add( CurrentPrediction )
    end if
loop
output(Predictions)
end

ExtendL:, ExtendR:
    for j:=1 to size of Extension
        //Use the Combining Clusters method to see if a structure can be extended
        if endBond( CurrentPrediction ) == firstBond( Extension[j] ) //If ExtendR
        if endBond( Extension[j] ) == firstBond( CurrentPrediction ) //If ExtendL
            //Extend the current prediction with the base structure
            NewPrediction = join( CurrentPrediction, Extension[j] )
            Predictions.add( NewPrediction )
            wasExtended = true
        end if
    end for
    return
```

Weak areas in the data

Any data-driven algorithm must rely on the available data. The probability of error will increase drastically for any area in which the amount of data is below average. This is true for the SPAA algorithm; input that contains sequences of amino acids with less than average frequency within the PDB will be difficult to predict. There are amino acid sequences of length 4 for which there are no examples in the PDB. Any input sequence into SPAA containing one of these sets cannot be predicted. For input sequences with less than average frequency, the ability of SPAA to correctly suggest structures will diminish and SPAA may fail to return any prediction. This weakness is a problem with the amount of data available and can be corrected by acquiring more protein structures.

An amount of confidence in the accuracy of SPAA's predictions can therefore be related to the amount of available data for an input sequence. More specifically, it is linked to the sequences of length 4 into which an input sequence is split, since the algorithm uses these as its base and to extend structures first. Sequences of length 4 that have less than half the mean value of occurrences for all sequences of length 4 are considered *weak*. Sequences with occurrences greater than this amount are considered *normal*. As the percentage of *weak* vs. *normal* sequences increases for an input sequence, the confidence in SPAA's prediction decreases.

Non-standard conformations

Areas of potential failure are introduced by the way in which data is processed by the SPAA algorithm. Canonical bond angles provided by SVD will be translated from the original values. Conformations which are very non-standard may be eliminated from the possible range of torsion angles that SPAA can suggest. See Figure 3.8 for an illustration of this effect. This is partly an effect of the SVD truncation process, the consequences of which are not clear. The PDB web source itself makes note of irregular torsion angle entries, indicating only that there are potential errors [3]. As the source of the information used in the study is unclear as to nature of these errors, no definitive conclusion can be drawn about how well SPAA will function in a scenario like this. However, it is clear that SPAA will be unable to predict these types of torsion angles. SPAA will appear to have failed from the viewpoint of validating results against the PDB.

Some conformations which result from amino acid sequences larger than the window size of 4 amino acids may also be eliminated. Rare structures which result from long-range interactions or long amino acid sequences could potentially be removed since the context of the clusters is limited. From the viewpoint of the methodology they will appear as non-standard conformations.

Small and/or weak clusters

Due to weak distributions of particular conformations for individual sequences and the clustering step employed to process data, points of failure are introduced. Clustering, by definition, attempts to group similar objects together. However, when there are relatively few objects available they may not be recognized as a cluster, which in this study means the data will be discarded. The methods in Section 3.6 also involve discarding clusters which are deemed insignificant or unreliable due to high standard deviations. This removal of data

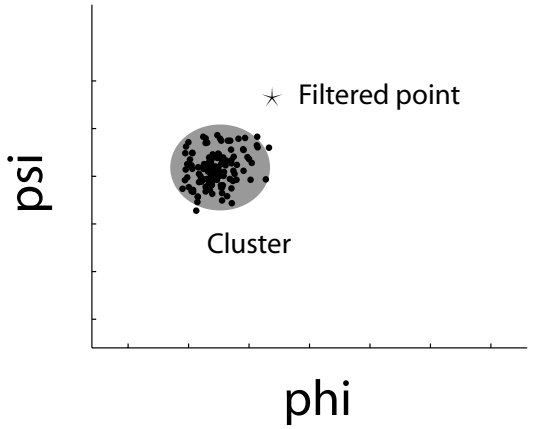


Figure 3.8: An hypothetical illustration depicting a close up of an area on a Ramachandran plot with a cluster obtained for SPAA in grey, occurrences the cluster represents and a stray occurrence which was filtered.

means that any structures which rely on these torsion angles will no longer be accurately predicted. This loss is necessary to achieve the generality of the algorithm, however, and cannot be avoided. See Figure 3.9 for an illustration of this effect.

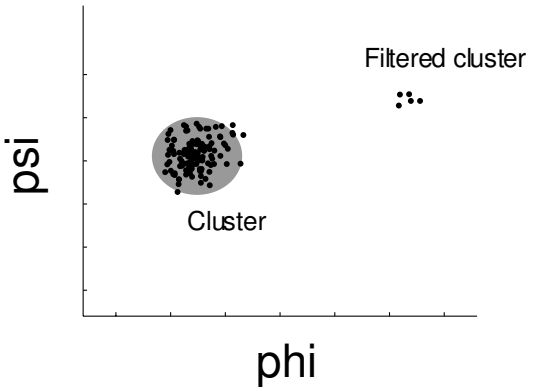


Figure 3.9: An hypothetical illustration depicting a close up of an area on a Ramachandran plot with a cluster obtained for SPAA in grey, occurrences the cluster represents and occurrences which represent another potential cluster which was eliminated.

In recognition of these sources of potential error the standard format for displaying structural conformations of proteins has been amended. Figure 3.10 demonstrates changes that have been made to accommodate SPAA structural predictions. This format is used in the results and in several Appendices for quick information display.

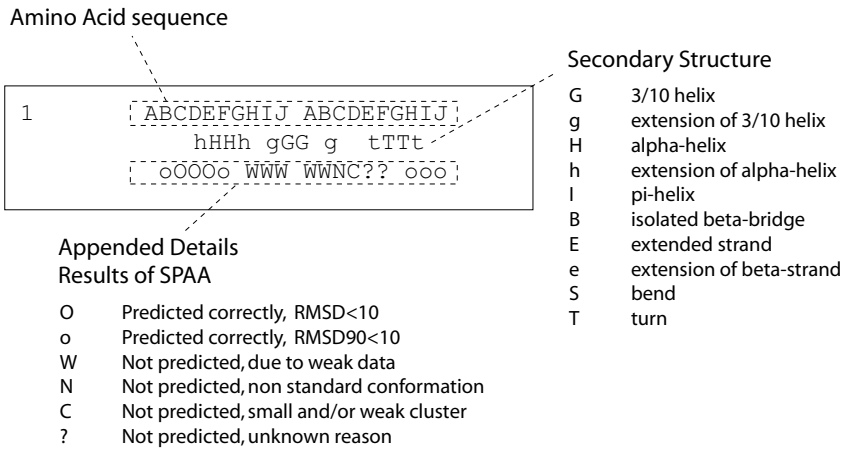


Figure 3.10: The standard sequence/structure chart employed by the PDB website [3] and others is extended by adding designations to points of failure or problems of the SPAA algorithm.

3.8 Comparing actual and predicted conformations

An application was written that can return sets of torsion angles representing the structure for any arbitrary sequence of amino acids from the PDB data file created in Section 3.1. It can also accept as input a set of torsion angles to check against the values it finds and determine their similarity. The root mean square distance (RMSD) is used to measure the difference between two sets of torsion angles. Equation 3.8 shows how the RMSD is calculated. Structures with low RMSD between actual and predicted values are considered to be similar and therefore constitute a successful prediction for that protein sequence.

Calculating the RMSD in this way will tend to underestimate the validity of the prediction should there be some predicted torsion angle pairs that are very distant from the actual angles. Thus an option to minimize this effect by ignoring the largest 10% of distances between individual torsion angle pairs. For example, for a protein sequence of 200 amino acids, up to 20 torsion angle pairs with bad predictions can be ignored. This will be referred to as RMSD90 for ease of reference. The RMSD can then be compared to the RMSD90 to further determine the validity of a prediction. If RMSD is high and RMSD90 is high, then the prediction is invalid. If RMSD is high and RMSD90 is significantly lower, then the prediction is good. If the both RMSD and RMSD90 are low then the prediction is very good.

$$RMSD = \sqrt{\frac{1}{2N}\sum_{i=1}^{N}((\phi^i_{predicted} - \phi^i_{actual})^2 + (\psi^i_{predicted} - \psi^i_{actual})^2)} \tag{3.8}$$

Root mean square distance should not be confused with root mean square deviation, a commonly used method to judge similarity between atomic coordinates of PDB entries.

3.8.1 Multiple predictions for short sequences

SPAA can potentially produce multiple predictions for an arbitrary input, particularly for short sequences. Sequences of approximately 10 amino acids long which exist in the PDB will be selected as test input. The output of SPAA will be compared against actual structures to verify that at least one of SPAA's predictions exists in the PDB. This will demonstrate SPAA's ability to predict potential and/or novel structures through dynamic extrapolation from the base data.

3.8.2 Predicting pieces of proteins

Sequences of entire proteins will be selected as input into SPAA to determine its ability to assemble complex structures. SPAA is known to have limitations and cannot be expected to produce an entire structure for large proteins (e.g. greater than 50 amino acids). It has the feature that it returns the largest substructures it was able to assemble. These substructures will be displayed using a Sequence/Structure Chart (see Figure 3.10). SPAA's ability to generate complex conformations (e.g. beyond α-helices and β-sheets determined by conventional methods) can then be easily explored, along with the areas of failure.

Sequences of proteins used in building the dataset will be used as input to SPAA. This will determine SPAA's ability to build structures about which it already has information. Sequences of proteins that have been added to the PDB after the data set was created will be used as test input for SPAA. This will determine SPAA's ability to predict structures about which it has no information.

3.8.3 Prediction of an entire protein

An attempt to predict the entire structure for a small protein with SPAA will be made. This will demonstrate the potential of the algorithm to produce entire, real-world examples of protein structure. Proteins of lengths less than 50 amino acids will be considered given SPAA's limitations.

The methodology outlined in this chapter is illustrated in Figure 3.1. Every sequence of length 3 and 4 is extracted from the newly formatted version of the PDB. SVD is then applied to selected sequences to produce canonical bond angles from which SDD is used to cluster conformational possibilities. The resulting clusters are then used in a dynamic-programming algorithm to accept amino acid sequences as input and provide predictions in the form of torsion angles.

Chapter 4

Results

This chapter provides detailed results obtained for each step outlined in the Chapter 3. An analysis of the extracted PDB data is presented to provide an understanding of the nature of the information used in this study. Results showing the usefulness of SVD as a denoising technique are demonstrated, as well as its ability to cluster structural elements of amino-acid sequences. The effects of SDD's clustering ability are parallel to SVD's, so SDD is employed as an unsupervised, automated protein-structure clustering mechanism. The results of several different input sequences of amino acids into SPAA are shown, as well as the prediction of an entire protein structure.

All run times that are provided were obtained using a Pentium-IV 2.6GHz with 2.0GB of RAM running Windows 2003 Server.

4.1 Statistical analysis of extracted data

From the newly created version of the PDB a total of 13,425,234 sequences of length three and 13,401,684 sequences of length 4 were extracted. Figure 4.1 shows a table of statistics for the extracted sequences.

Number of examples	Length 3	Length 4
Range	4 – 11,021	0 – 1705
Mean	1650	104
Median	1250	67

Figure 4.1: Statistics for the sequences extracted from the new PDB format.

The rarity of particular amino-acid sequences partly results from the encoding scheme of nucleotides. For instance the amino acid SER has six nucleotide codons associated with it and PHE has two (see Figure 1.2). The sequence SER-SER-SER has 5010 occurrences while PHE-PHE-PHE has only 651 occurrences. Of course, the physicochemical properties of each amino acid also affect the abundance of certain sequences. For instance, ALA and VAL are hydrophobic whereas PRO is hydrophilic; it would be assumed that sequences containing ALA and VAL would be naturally more abundant than sequences with ALA and

PRO. Indeed this happens: ALA-VAL-ALA has 8389 occurrences and ALA-PRO-ALA has 3429 occurrences. There is also the inherent bias of the PDB and structural determination methods favouring proteins of particular classes over others (see Section 2.2, Section 2.1).

This quick analysis of distributions of sequences in the PDB demonstrates that some sequences will cause difficulties for SPAA.

4.2 Exploring conformational variation with SVD

Computing the SVD of a matrix for a sequence of length 3 with number of occurrences near the mean took 13s. The SVD of a matrix for a sequence of length 4 with number of occurrences near the mean took 3s.

Figure 4.2 shows scree plots of singular values obtained from SVD on randomly selected sequences of length 3. Figure 4.3 is similar but for sequences of length 4. The singular values for both sequences of length 3 and 4 demonstrate a steep decline in the magnitude of the singular values in the lower dimensions. This indicates that there are a few components that account for the majority of variance present and that the remaining components account for little variance. The singular values indicate that the resulting U matrix for most sequences could be truncated at $k = 2 - 3$ for sequences of length 3 and $k = 3 - 4$ for sequences of length 4. More plots of singular values can be seen in Appendix A.

Figure 4.4 shows 3-dimensional plots of the first 3 columns of the resulting U matrix from SVD on sequences of length 3 and the subsequent areas on a Ramachandran plot they represent (see Figure 3.4 for a detailed explanation of this process). Figure 4.5 is similar but for sequences of length 4. Figure 4.6 shows a standard set of Ramachandran plots for a specific sequence contrasted with a set of Ramachandran plots with conformational areas obtained using SVD truncation. More plots of U matrices can be seen in Appendix B and subsequent Ramachandran plots can be seen in Appendix C.

Limited structural information can be discerned from Figure 4.6.a. There are the apparent structural possibilities of α-helices and β-sheets, but it is not possible to tell which individual points are related throughout the set of plots and therefore what conformations are actually present for that particular sequence. While a Ramachandran plot is only intended to give conformational ranges for a pair of amino acids, this demonstrates the difficulty of extracting generalized conformations from the raw PDB data. The SVD clusters shown on Ramachandran plots in Figure 4.6 tell a drastically different story; four distinct structures are present. One structure is a standard α-helix and one is a β-sheet. There is also one conformation which lies in a non-standard area of the α-helix region and one structure appears to be a transitional element between secondary structures. Figures 4.4 and 4.5 detail more examples of SVD's abilities. The areas on the subsequent Ramachandran plots represent generalized conformational possibilities for each respective amino-acid sequence. Therefore, the clusters present in the geometric interpretation are conformational possibilities. These conformational possibilities can be typical secondary structures, but more importantly, they can also be unique structural elements for a sequence of amino acids that may be vital for protein structure prediction.

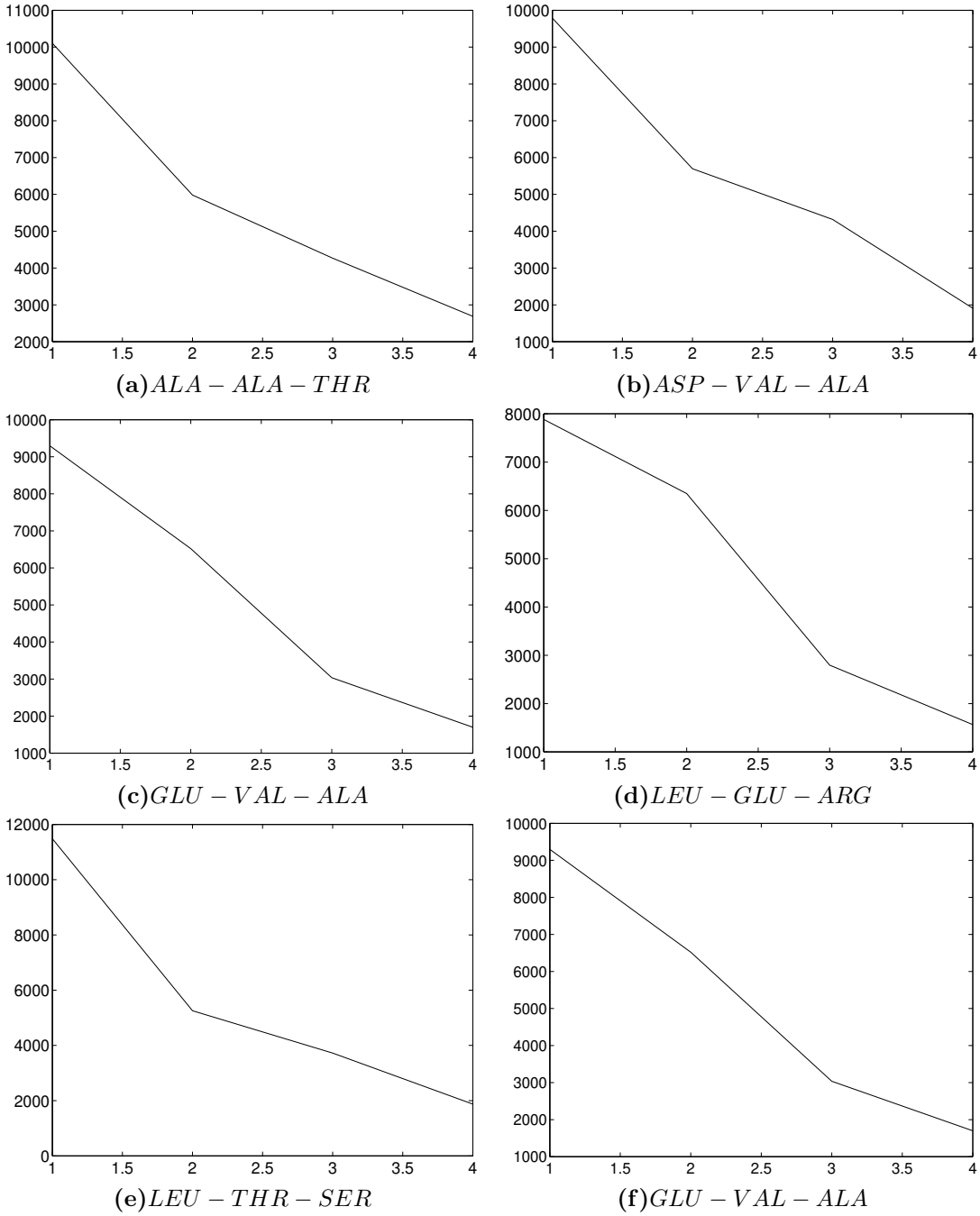
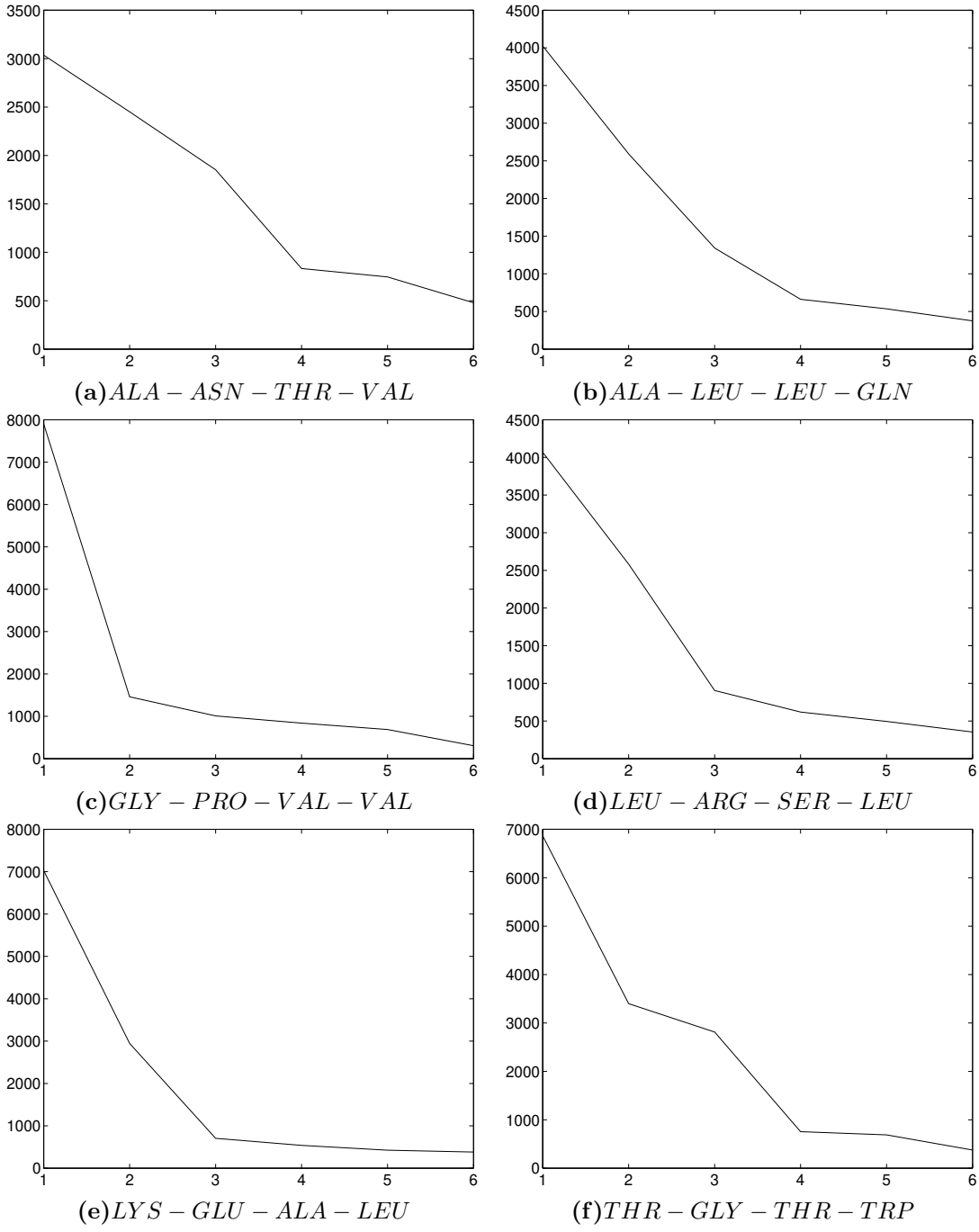


Figure 4.2: Scree plots for singular values from SVD on specific sequences of length 3.

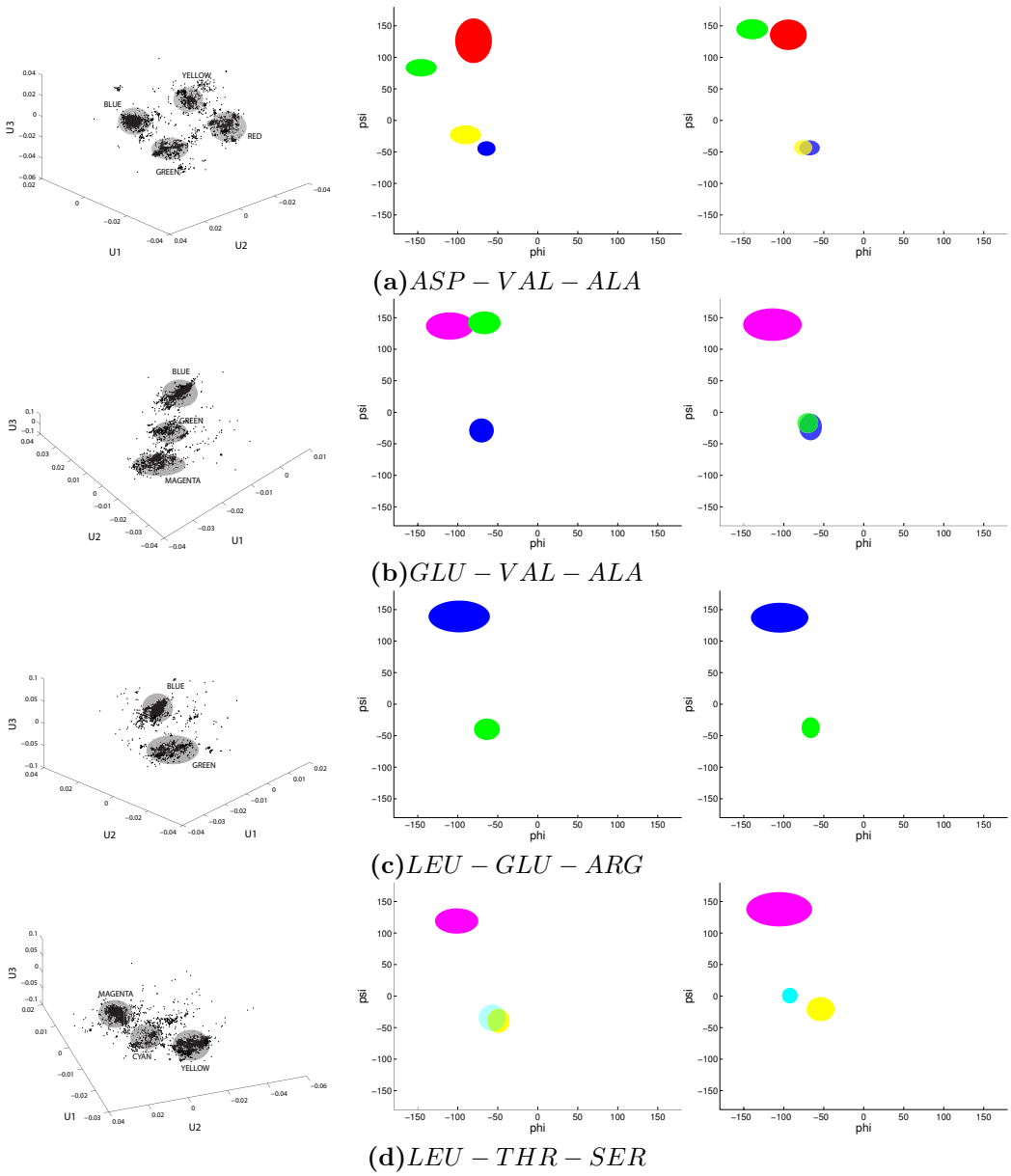


(a) $ALA - ASN - THR - VAL$ **(b)** $ALA - LEU - LEU - GLN$

(c) $GLY - PRO - VAL - VAL$ **(d)** $LEU - ARG - SER - LEU$

(e) $LYS - GLU - ALA - LEU$ **(f)** $THR - GLY - THR - TRP$

Figure 4.3: Scree plots for singular values from SVD on specific sequences of length 4.



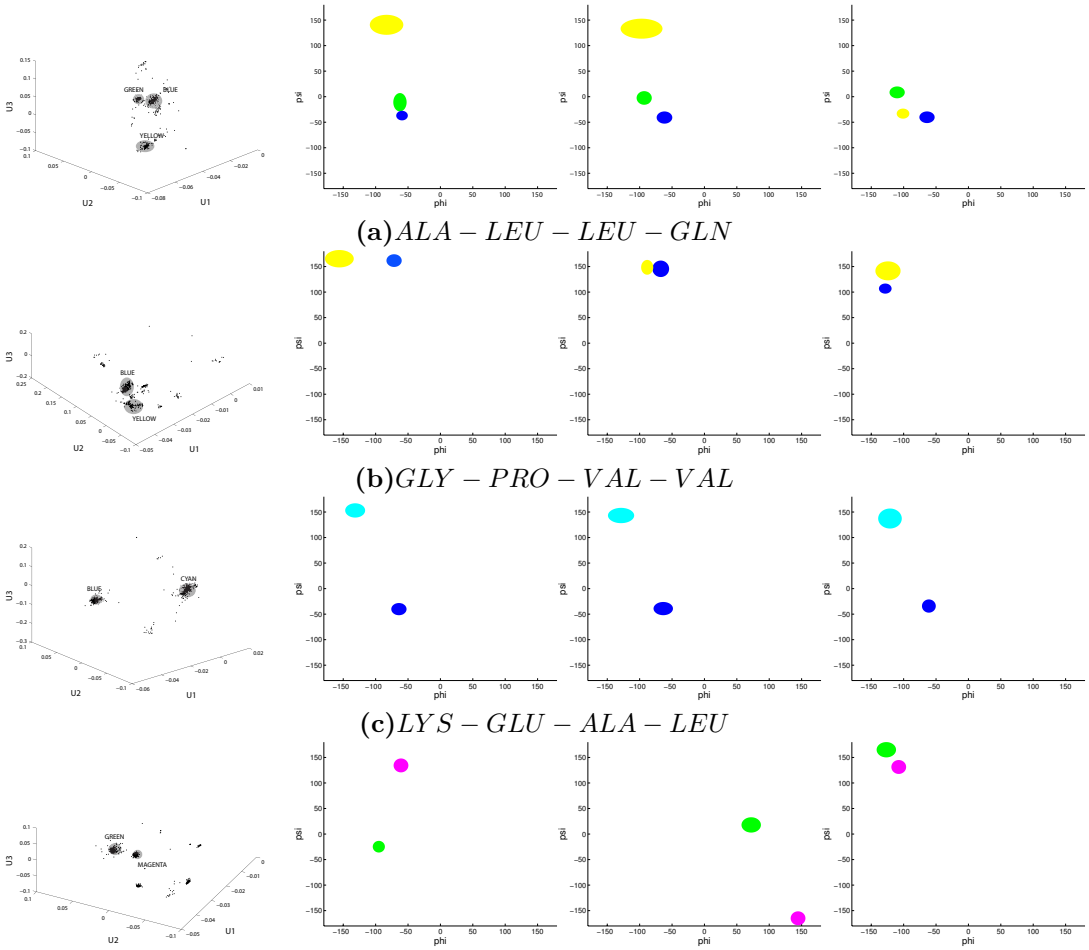
(a) $ASP - VAL - ALA$

(b) $GLU - VAL - ALA$

(c) $LEU - GLU - ARG$

(d) $LEU - THR - SER$

Figure 4.4: A series of figures showing: a graph of the first 3 columns of a U matrix resulting from SVD on a specific sequence on the left, a resulting set of Ramachandran plots showing conformational areas represented by the clusters on the right. Coloured areas represent similar clusters throughout each set. These figures are for sequences of length 3.



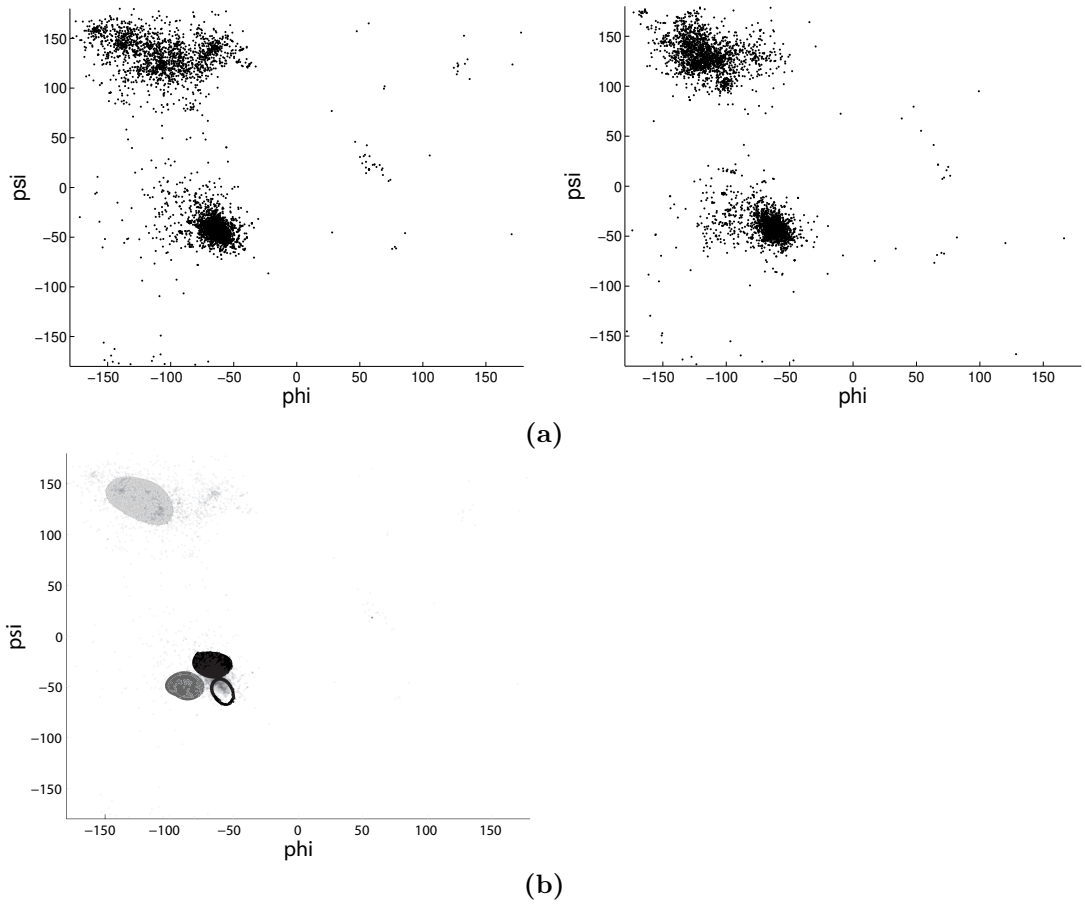
(a) $ALA - LEU - LEU - GLN$

(b) $GLY - PRO - VAL - VAL$

(c) $LYS - GLU - ALA - LEU$

(d) $THR - GLY - THR - TRP$

Figure 4.5: A series of figures showing: a graph of the first 3 columns of a U matrix resulting from SVD on a specific sequence on the left, a resulting set of Ramachandran plots showing conformational areas represented by the clusters on the right. Coloured areas represent similar clusters throughout each set. These figures are for sequences of length 4.



(a)

(b)

Figure 4.6: **(a)** A set of Ramachandran plots for the sequence LEU-VAL-ARG. **(b)** A set of Ramachandran plots which has been created from clusters found using SVD on the sequence LEU-VAL-ARG. The different shades indicate similar clusters throughout the set. The dark grey cluster suggests an α helix between $LEU - VAL$ and a sheet between $VAL - ARG$. The black cluster suggest an α-helix with a different range of torsion angles than the grey cluster between $LEU - VAL$ and remains an α-helix between $VAL - ARG$. These types of conformation are not intuitive from the original Ramachandran plot

4.3 Obtaining canonical torsion angles with SVD

Truncating a resulting set of matrices produced from SVD at a value k and re-multiplying them will produce a canonical version of the original matrix (see Equation 2.3). The canonical version can than be investigated to see the effects of truncation.

Figure 4.7 compares a standard Ramachandran plot for selected sequences to a Ramachandran plot generated with data that has been decomposed, truncated and re-multiplied using SVD. The 'SVD-Ramachandrans' show that many points have moved towards areas commonly identified as having higher probabilities (see Figure 2.2). Points also show a tendency to exhibit tighter grouping. This result implies that SVD successfully removes components which are unrelated to the local structure of a particular amino-acid sequence. Therefore, the resulting values can be considered canonical torsion angles from which components have been removed that do not effect the main structure of the data.

The source of the weak components cannot be readily deduced but can be speculated upon. There is an inherent error rate introduced by the physical determination methods used to obtain protein structures. It is plausible that such errors are normally distributed so that SVD can distinguish them from the structural components. Another possible source of variation is the interaction of amino acids outside the window size of the sequence; and physicochemical influences propagating through the protein backbone. Minor variations in the torsion angles introduced from amino acids outside the window would introduce variance unrelated to the most significant structure. There may be other potential sources for the removed components, but the results of viewing the canonical torsion angles on Ramachandran plots indicates that these sources are best accounted for as noise.

Denoising a set of torsion angles for a specific sequence of amino acids can be viewed as obtaining a clearer picture of the fundamental structural possibilities for that sequence. Using denoised sets therefore provides a more reliable framework for assembling proteins from subsequences.

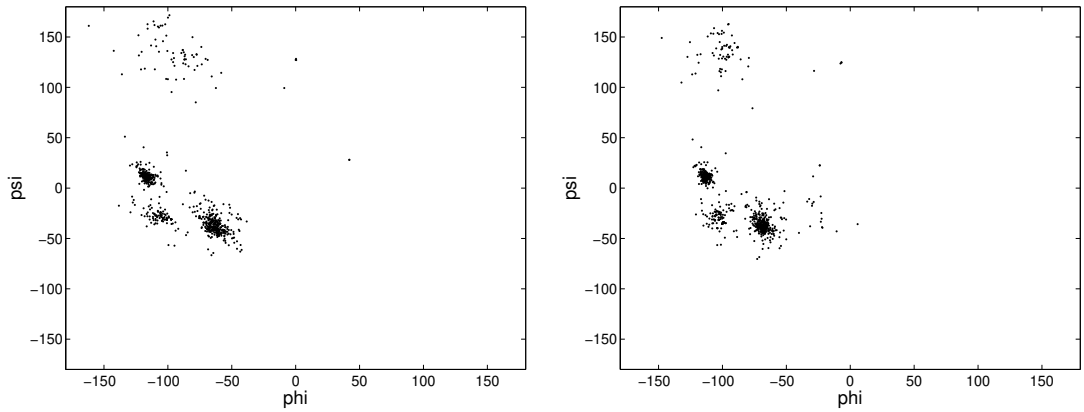
4.3.1 Obtaining structures with SDD

The SDD of a matrix for a sequence of length 3 with number of occurrences near the mean took 25s. The SVD of a matrix for a sequence of length 4 with number of occurrences near the mean took 5s.

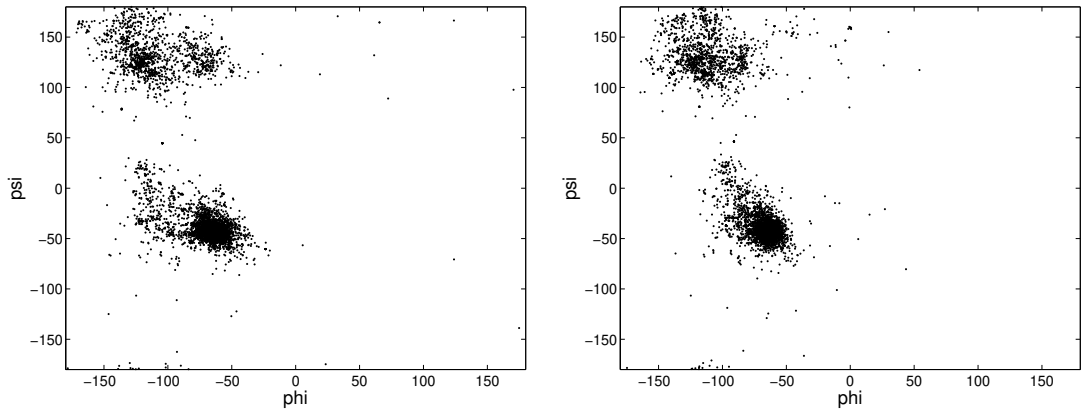
Figure 4.8 shows the results of SDD compared to the results of SVD for selected sequences. SDD obtains near identical results for clustering conformations. SDD actually provides a more refined separation then is immediately available from SVD since only 3 dimensions can be visualized at a time in SVD. More clusters with refined structures are sometimes possible with SDD over SVD.

A typical clustering result from SDD performed on a SVD-denoised dataset is shown in Figure 4.9. SDD finds distinct conformations for sets of amino acids. Like the results from the SVD clusters, these conformations can represent standard secondary structural elements or non-typical conformations which may be important to protein structure. The examples are general results of randomly selected data.

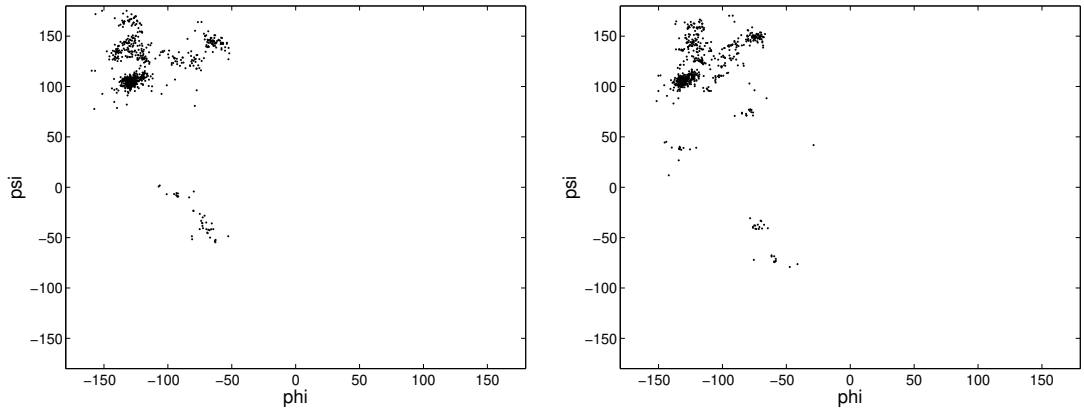
SDD finds conformational possibilities for sequences of amino acids, which have been denoised using SVD, and which are not obvious from the original data. The structures that are obtained can represent standard secondary structural possibilities, but more importantly



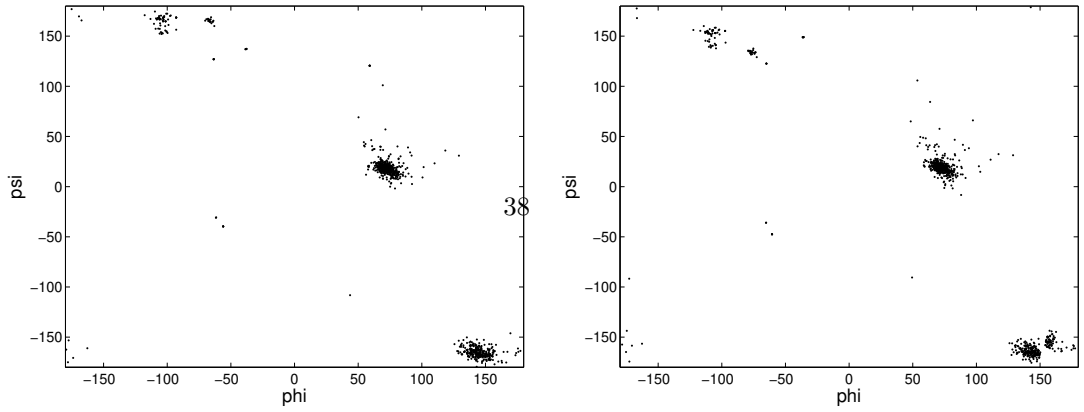
(a) $ALA - LEU - LEU - GLN$



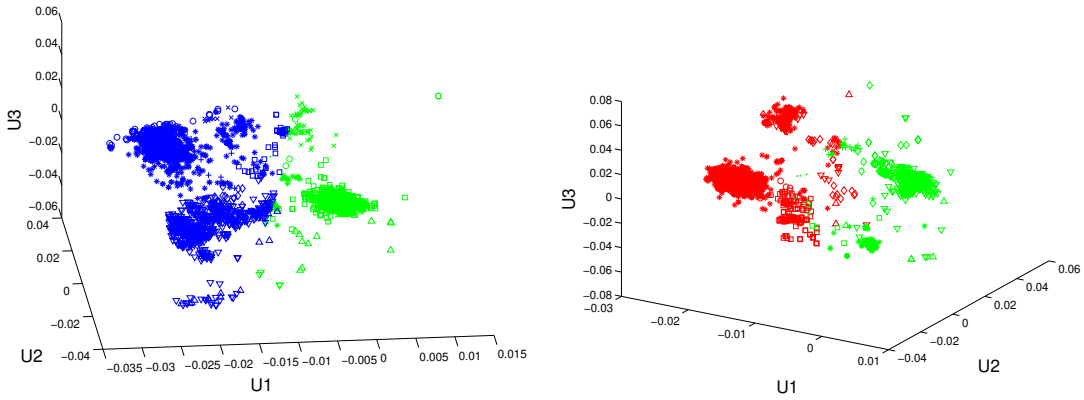
(b) $GLU - VAL - ALA$



(c) $GLY - PRO - VAL - VAL$



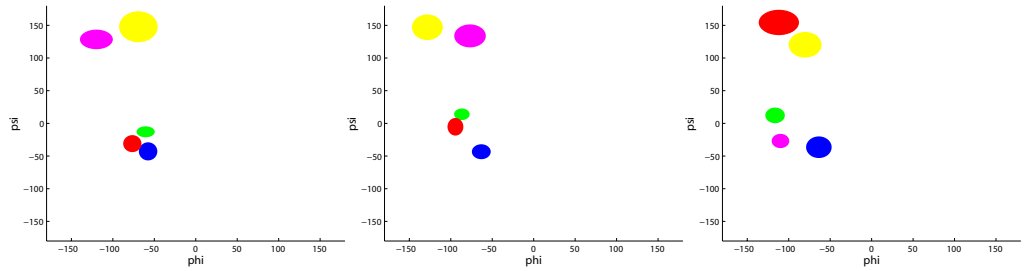
(d) $THR - GLY - THR - TRP$



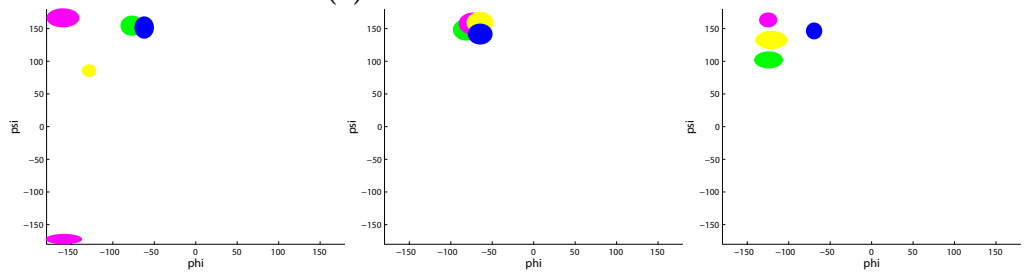
(a) $ASP - VAL - ILE$ **(b)** $ILE - ALA - VAL$

(c) $LEU - THR - GLU - ALA$ **(d)** $SER - GLY - SER - LEU$

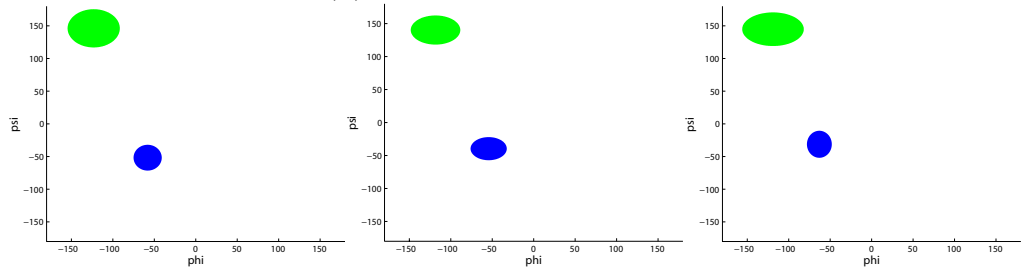
Figure 4.8: Demonstration of the clustering ability of SDD vs SVD for selected sequences of amino acids. The results of SVD are used for position and cluster labels obtained from SDD are denoted by colours and shapes.



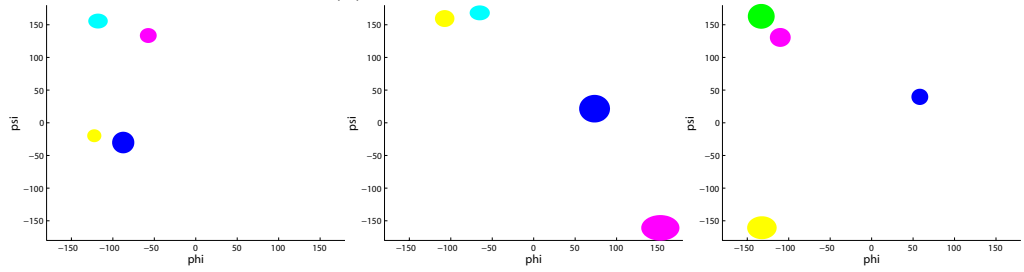
(a) $ALA - LEU - LEU - GLN$



(b) $GLY - PRO - VAL - VAL$



(c) $LYS - GLU - ALA - LEU$



(d) $THR - GLY - THR - TRP$

Figure 4.9: Sets of Ramachandran plots depicting the conformation possibilities obtained from SDD for the respective sequences. The coloured areas represent similar clusters throughout each set.

can represent conformations which may be structurally significant in protein-structure prediction.

4.4 Comparing actual and predicted conformations

The SPAA algorithm uses the structural possibilities obtained for each sequence and attempts to predict protein structure by combining sequences into longer units.

4.4.1 Multiple predictions for short sequences

Figure 4.10 shows predicted torsion angles for the sequence **FVAALNAGDL** predicted by SPAA, as well as its actual structure. SPAA suggests 7 distinct possibilities, one of which very closely resembles the structure determined by traditional methods. The other 6 structures may exist in nature, but have not yet been observed. It took SPAA 35s to generate these results.

PREDICTIONS FOR **FVAALNAGDL**

A:	**(-60,-54)**	**(-59,-44)**	**(-60,-44)**	**(-63,-37)**	**(-63,-46)**	...
P1:	**(-65,-45)**	**(-62,-41)**	**(-63,-41)**	**(-64,-41)**	**(-64,-40)**	...
P2:	(-65,-45)	(-62,-41)	(-63,-41)	(-64,-41)	(-64,-40)	...
P3:	(-65,-45)	(-62,-41)	(-63,-41)	(-64,-41)	(-61,-37)	...
P4:	(-65,-45)	(-62,-41)	(-59,-41)	(-66,-18)	(-86,-6)	...
P5:	(-65,-45)	(-62,-41)	(-65,-39)	(-144,141)	(-119,127)	...
P6:	(-65,-45)	(-62,-41)	(-65,-39)	(-144,141)	(-118,132)	...
P7:	(-132,119)	(-121,120)	(-126,145)	(-115,124)	(-113,140)	...

A:	... **(-63,-31)**	**(-87,-8)**	**(67,30)**	**(-89,83)**
P1:	... **(-62,-30)**	**(-91,6)**	**(60,32)**	**(-91,85)**
P2:	... (-64,-32)	(-76,-14)	(75,36)	(-114,5)
P3:	... (-102,16)	(-72,3)	(-91,12)	(66,31)
P4:	... (60,36)	(-67,-18)	(75,36)	(-114,5)
P5:	... (-82,117)	(-94,5)	(60,32)	(-91,85)
P6:	... (-80,158)	(-57,134)	(85,2)	(-67,129)
P7:	... (-80,158)	(-57,134)	(85,2)	(-67,129)

Figure 4.10: Several predicted structures for the amino acid sequence FNAALNAGDL are displayed along with the actual structure. The actual structure and the closest prediction are in bold.

SPAA can predict more than one structural possibility for any arbitrary input sequence. In the PDB only one structural occurrence of this sequence may exist. This is a clear indicator that the algorithm is able to not only utilize the original data, but extrapolate from it to theorize novel structures which may exist.

While it may at first seem trivial to obtain several structures and then choose the most correct version as evidence of validity, it is actually non-trivial for SPAA to return relatively

```
1          VHLTPEEKTA VNALWGKVNV DAVGGEALGR LLVVYPWTQR FFESFGDLSS
               hHHHHH HHHHHgG  t TThHHHHHHH HHHH gGGGG GGgGG
            oooOOOOOO oooooooooo oooooooooo oooooooooo oooooooooo

51         PDAVMGNPKV KAHGKKVLGA FSDGLAHLDN LKGTFSQLSE LHCDKLHVDP
           hHHHHtThHH HHHHHHHHHH HHHHHHttTT hHHHthHHHH HHiII    t
           oooooooooo ooooo OOOOO OOOOOONNN     NNNoooo oooooooooo

101        ENFRLLGNVL VCVLARNFGK EFTPQMQAAY QKVVAGVANA LAHKYH
           hHHHHHHHHH HHHHHHHHgG G  hHHHHHH HHHHHHHHHH HtgGG
           oooooooooo oooooooooo ooooooCCC
```

Figure 4.11: A chart which details the amino acid sequence of the protein 1shr and the prediction of this sequence from SPAA following the sequence/structure chart format.

few possibilities of which one is actually present in the PDB. The number of conformational possibilities for an arbitrary set of amino acids is not necessarily known. However, the probability of the algorithm accidentally obtaining a correct structure for a sequence even as small as length 10 is low.

4.4.2 Predicting pieces of proteins

Sequences from a protein present in the PDB before the download date were selected for the algorithm to attempt to predict. This will determine SPAA's ability to assemble structures about which it knows information implicitly. Figure 4.11 shows the a resulting prediction from SPAA for the protein **1shr**. It took SPAA 62s to generate results for this input. The algorithm was able to accurately predict most of the protein's fundamental structure within an accuracy of 10 RMSD90. The algorithm succeeded in predicting complex turns leading out of α-helices and also 3/10 helices and π-helices. Failures occurred near the 80^{th} amino acids due to a complex turn that was determined to be very non-standard. Another failure occurred near the end of the protein due to filtering of a cluster for a particular amino-acid sequence. More predictions from SPAA on this type of data can be found in Appendix D. See Figure 3.10 for a review of the sequence/structure chart format.

Sequences from proteins that were added to the PDB after the download date were selected for the algorithm to attempt to predict. This measures SPAA's ability to predict structures that have never been encountered before. Figure 4.12 shows results for the protein **1h47**. It took SPAA 56s to generate results for this input. SPAA has correctly predicted a large portion of the protein's interior sequence. 33 of the 152 ($\approx$22%) sequences of length 4 into which the input was split were *weak*. Notably, at the 88^{th} and 122^{nd} amino acid marks, turns have correctly been predicted to within 10 RMSD. The 102^{nd} to 107^{th} amino acids mark a random join between an α-helix and β-sheet, including a bend at the 103^{rd} amino acid mark, that has been accurately predicted. A complete prediction failed due to weak areas in the data.

Figure 4.13 shows results for the protein **1qvn**. It took SPAA 30s to generate results for this input. SPAA correctly predicts a large interior portion of the protein sequence. 21 of the 65 ($\approx$32%) sequences of length 4 into which the input was split were *weak*. The most

```
51        LGAAALGDIG KLFPDTDPAF KGADSRELLR EAWRRIQAKG YTLGNVDVTI
          HHHTT   HH HHS SS GGG TT  HHHHHH HHHHHHHHTT  EE EEEEEE
                               WW WWOOOOOOOO OOOOOOOOOO OOOOOOOOOO

101       IAQAPKMLPH IPQMRVFIAE DLGCHMDDVN VKATTTEKLG FTGRGEGIAC
          E SSS  HHH HTHHHHHHHH HTT  GGGEE EEEE  TT H HHHTTSEEEE
          OOOOOOOOOO OOOOOOOOOO OOOOooWWWW

151       EAVALLIKAT K
          EEEEEEE
```

Figure 4.12: A chart which details the amino acid sequence of the protein 1h47 and the prediction of this sequence from SPAA following the sequence/structure chart format.

```
1         APTSSSTKKT QLQLEHLLLD LQMILNGINN YKNPKLTRML TFKFYMPKKA
              HHHHHH HHHHHHHHHH HHHHHHHHHT S HHHHHHHT TS B  BS
                                       CC Cooooooooo oooooooooo

51        TELKHLQCLE EELKPLEEAL NLAQSKNFHL RPRDLISNIN VIVLELKGSE
           SGGGGHHHH TTHHHHHHHH HH          HHHHHHHHH HHHHHHT SS
          oooooooooo oooooooooo oWWWW

101       TTFMCEYADE TATIVEFLNR WITFCQSIIS TL
                B SS  B HHHHHHH HHHHHHHHHH TT
```

Figure 4.13: A chart which details the amino acid sequence of the protein 1qvn and the prediction of this sequence from SPAA following the sequence/structure chart format.

```
...
101  VALRNRSNTP IKVDGKDVMP EVNRVLDKMK SFCQRVRSGD WKGYTGKSIT
     HHHTTTT     EETTEESHH HHHHHHHHHH HHHHHHHTT  SB TTS B
           CCCo oOOOOOOOOO OOOOOOOOOO OOOOOOOOOO OOOOOOOOoC

151  DIINIGIGGS DLGPLMVTEA LKPYSKGGPR VWFVSNIDGT HIAKTLASLS
     EEEEE  GGG THHHHHHHHH TGGGTTTS E EEEE  SSHH HHHHHHTTTT
     CC

201  PETSLFIIAS KTFTTQETIT NAETAKEWFL EAAKDPSAVA KHFVALSTNT
     GGGEEEEEE  SSS  HHHHH HHHHHHHHHH HHH  GGGGG GTEEEEES H
                NNNoooooooo oooooooooo oooooooooo oooo???

251  AKVKEFGIDP QNMLEFWDWV GGRYSLWSAI GLSIALHVGF DHFEQLLSGA
     HHHHHHT  G GGEEE  TTT TGGGTTTTGG GHHHHHHHTH HHHHHHHHHH
                                                WWWoooooooo

301  HWMDQHFLKT PLEKNAPVLL ALLGIWYINC YGCETHALLP YDQYMHRFAA
     HHHHHHHHHS  GGG HHHHH HHHHHHHHHT T   EEEEEE S STTTTHHH
     oooooooooo oooooooooo ooooooooooW WW

351  YFQQGDMESN GKYITKSGAR VDHQTGPIVW GEPGTNGQHA FYQLIHQGTK
     HHHHHHHHHH    EETTS B  SS    EEE    TTGGGGT THHHHHHSS


401  MIPCDFLIPV QTQHPIRKGL HHKILLANFL AQTEALMKGK LPEEARKELQ
     EEEEEEEES B S   GGGH HHHHHHHHHH HHHHHHHH B  HHHHHHHHH
                                           NNNoo oooooooooo

451  AAGKSPEDLE KLLPHKVFEG NRPTNSIVFT KLTPFILGAL IAMYEHKIFV
     HTT  HHHHH HHHGGG B      EEEEEES B  HHHHHHH HHHHHHHHHH
     oooooooooo oooooooooo oooooooooo oooooooooo oooooooooo

501  QGIMWDINSF DQWGVELGKQ LAKKIEPELE GSSAVTSHDS STNGLISFIK
     HHHHTTB TT TTGGGHHHHH HHHHHHHHHS SS    SS H HHHHHHHHHH
     oooo??

551  QQRDTKLEHH HHHH
     HHTT
```

Figure 4.14: A chart which details the amino acid sequence of the protein 1u0f and the prediction of this sequence from SPAA following the sequence/structure chart format.

interesting feature of this prediction is from the 40^{th} to the 56^{th} amino acid mark. SPAA accurately predicts a segment 16 amino acids long that involves random areas, turns, and bends leading into a 3/10 helix. This shows that SPAA has the ability to predict complicated structures longer than the sequences it uses as its basis.

Figure 4.14 shows results for the protein **1u0f**. It took SPAA 89s to generate results for this input. SPAA correctly predicts much of the complicated conformations of the protein. 156 of the 550 (≈28%) sequences of length 4 into which the input was split were *weak*. Near the 100^{th} amino acid mark, SPAA correctly predicts a long α-helix structure which eventually turns into a complicated turn and random segment near the 140^{th} amino acid. SPAA was successful in predicting many structures beyond the standard α-helix and β-sheet structures. There were points of failure due to lack of data and non-standard conformations. However, SPAA was able to predict a significant portion of this large protein.

SPAA was not able to produce accurate predictions for every test input sequence. The results for protein **1ceo**, **1nq7** and **1q31** were not accurate. However, they had the following percentage of *weak* sequences of length 4 respectively: 110 of 232 (≈47%), 102 of 237 (≈43%), 107 of 212 (≈50%).

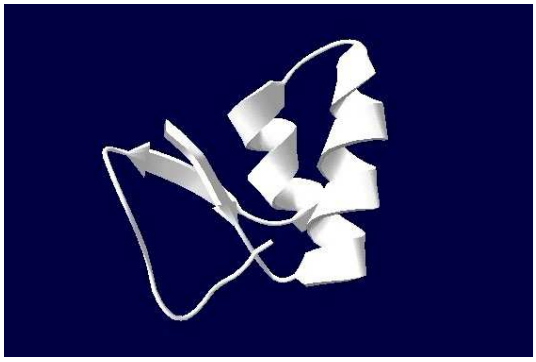
More predictions from SPAA on new proteins from the PDB can be seen in Appendix D.

The SPAA algorithm is able to go beyond the bounds of the original data, accurately proposing possible conformations for amino-acid combinations which have not yet been determined. SPAA demonstrates the ability to produce complex structures well beyond typical 3-state predictions. Torsion angles for turns, bends and random regions have been predicted accurately as well as standard α-helices and β-sheets. The limitations that were expected in Section 3.7.3 do indeed show up as problems in predicting entire structures. However, SPAA is still able to predict large portions of proteins despite this. Some input sequences did not return accurate results. These proteins tend to be those containing sequences where the available data is far less than average. It appears that when the fraction of *weak* sequences (see Section 3.7.3 for definition) is near 25% predictions are good, but when it approaches 50%, SPAA's ability diminishes.

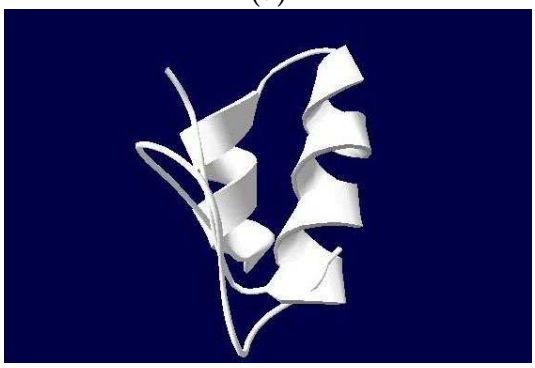
4.4.3 Predicting an entire protein

The potential exists for the algorithm to predict entire structures in addition to the ability to predict pieces of proteins. Protein **1crn** is a simple plant protein, 46 residues in length. Though short, the protein involves α-helices, turns and random bends. When the amino-acid sequence of protein **1crn** is used as input, four distinct structural possibilities are returned, one of which is similar to the actual conformation. (It should be noted that the first and last residue were omitted from the input sequence.) SPAA took 32s to produce the results for **1crn**. The predicted torsion angles were input in SwissProt3D to generate a hypothetical structure for visual comparison to the actual structure. The results are shown in Figure 4.15. The entire sequence of **1crn**, along with the actual and predicted torsion angles are given in Appendix E.

The algorithm was able to correctly generate torsion angles of turns leading into and out of α-helices with little deviation from the actual values, as well as correctly predict the length of the interior α-helices. One point of major error in the prediction is identified at a PRO-GLU bond (see Appendix E for residues), which coincides with a turn joining the two α-helices. The algorithm incorrectly suggests a torsion angle pair of $\phi' = -57$, $\psi' = -34$ where the correct values are $\phi = -56$, $\psi = 146$. The actual value for the torsion angle



(a)



(a)

Figure 4.15: **(a)** A 3-dimensional representation of the actual structure of protein 1crn. **(b)** A 3-dimensional model generated with predicted torsion angles from the SPAA algorithm. The images were generated with Swiss-PdbViewer [10].

pair in question lay outside of the clusters obtained for the subset of amino acids at the PRO-GLU bond. The sequence and combination of events that leads to this type of error were examined in Section 3.7.3.

4.5 Discussion

Applying SVD to torsion-angle data has been demonstrated as a useful denoising technique. SVD removes components unrelated to the main conformational possibilities of a set of amino acids and produces conformations that seem to represent fundamental structures more appropriately. The noise has many potential sources, some of which may be; error from structural determination methods used to create the PDB, or structural influences that propagate throughout the protein backbone.

SDD is an effective clustering technique that finds distinct conformational possibilities for a sequence of amino acids. The clusters represent structures that extend beyond the conventional secondary structural elements and provide a more accurate basis for protein structure prediction by assembly of subsequences.

A dynamic-programming, data-driven approach is a valid approach to protein structure prediction. SPAA has demonstrated the ability to produce accurate predictions for many, different protein sequences. The algorithm is able to generate results for complex structural elements that are trouble areas for other prediction algorithms (see Section 2.4). The approach taken in this study also has the advantage of returning torsion angles as opposed to generic secondary structure from a predefined set. Detailed models such as the one for **1crn** can be created from the torsion angle output.

There are flaws and limitations inherent to SPAA's prediction techniques. Errors have been introduced through data-processing methods. Structures can be discarded due to rarity in the PDB. Some conformational possibilities are eliminated as structures are refined into generalized versions. Of course, SPAA's overall ability is linked the amount of data available in the PDB.

Chapter 5

Conclusions

Understanding protein function and structure is a world-wide initiative with benefits that reach into many aspects of modern science. Elucidation of protein structure is currently performed by structural determination methods that are slow and expensive. Modern protein-structure prediction methods lack either the necessary accuracy or scope to be employed as viable solutions. Predicting protein structure from first principles is computationally intractable.

The objective of this work was to create a data-driven, protein-structure prediction algorithm with improved accuracy and scope. Specifically, the algorithm should meet the following criteria:

- Utilizes all of the available protein structure data that is available.
- Based on processed data that provides a more reliable framework.
- Successfully extrapolates from the available data to provide novel solutions for input sequences.
- Provides more detailed predictions than is possible with many existing prediction algorithms.

SPAA, and the data it uses, have been shown to fulfill the above requirements. The ability to construct canonical torsion angles and find canonical conformations for short subsequences was shown to give SPAA its predictive capability. The ability to assemble conformations of small sequences into larger structures would not be possible using the original data. SPAA proved to be useful in providing accurate predictions for pieces of large proteins and for providing full, detailed structures of small proteins. It produced more detailed structures than secondary-structure prediction methods and has greater generalization than comparative modelling approaches, producing results for a greater range of input.

Though demonstrating the potential to advance the state of the art, there are particular limitations due to the nature of the methodology employed. These could possibly be resolved in the future, particularly as the available data grows.

5.1 Future Work

Acquiring more data to use as the basis of SPAA will of course improve the accuracy and scope of its predictions.

Using SPAA in tandem with other protein-structure prediction algorithms may provide better results. An existing method which has the ability to generate entire structures for larger proteins could be coupled with SPAA's ability to produce very accurate predictions. A method that is not as inherently data-bound as SPAA would be an apt choice. Some of the predictions returned from SPAA were mostly complete, lacking only small pieces that connect the structures into the final protein. Using a molecular simulation method just for the gaps would require relatively little computational power, compared to generating an entire structure, and provide a very detailed and complete conformation. Even using secondary-structure predictions to fill in the gaps would provide a better idea of the final overall structure of the entire protein.

Using an amino-acid substitution method to generate sequences of amino acids may yield better results than generating every combination. Amino acids can be grouped into different categories based on similar physicochemical properties. They can also be grouped together by the likelihood of one amino acid being replaced by another due to point mutations. Combining sequences that share similar properties may allow some areas of weak data to be extended and provide better structural information. For example, LEU-LEU-LEU and LEU-LEU-ILE may share similar conformational possibilities, but are treated as separate in SPAA. There is, however, the inherent problem of using a suitable amino-acid substitution heuristic, which continues to be an unsolved problem in bioinformatics.

The potential also exists to expand SPAA past just ϕ and ψ angles to include all possible torsion angles that exist between two amino acid residues and provide results of even greater detail. The ω and χ rotamer angles could be included for amino acid sequences to extend the clustering results into a higher- dimensional space. However, the number of rotamer angles are variable and depend on the amino acid residue, making it non-trivial to extend the methodology to include this information.

Bibliography

[1] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Anang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLST: a new generation of protein database search programs. *Nucleic Acids Research*, 25, 1997.

[2] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15:937–946, 1999.

[3] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne. The Protein Data Bank. *Nucleic Acids Research*, pages 235–242, 2001.

[4] P.Y. Chou and G.D. Fasman. Prediction of the secondary structure of proteins from their amino acid sequence. *Advances in Enzymology and Related Areas of Molecular Biology*, 47:45–48, 1978.

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

[6] J. Drenth. *Principles of Protein X-Ray Crystallography.* Springer-Verlag Inc, 1999.

[7] D. Frishman and P. Argos. 75% accuracy in protein secondary structure prediction. *Proteins*, 27:329–335, 1997.

[8] J. Garnier, D.J. Osguthorpe, and B. Robson. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology*, 120:97–120, 1978.

[9] M.M. Gromiha and S. Selvaraj. Importance of long-range interactions in protein folding. *Biophysical Chemistry*, 77:49–68, 1999.

[10] N. Guex and M.C. Peitsch. SWISS-MODEL and the Swiss-PdbViewer: An environment for comparative protein modeling. *Electrophoresis*, 18:2714–2723, 1997.

[11] S. Hovmoller, T. Zhou, and T. Ohlson. Conformations of amino acids in proteins. *Biological Crystallography*, D58:768–776, 2002.

[12] IBM. Describing protein folding kinetics by molecular dynamics simulations. *The Journal of Physical Chemistry B*, 108(21):6571–6581, 2004.

[13] D.T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292:195–202, 1999.

[14] E. Kabat and T. Wu. The influence of nearest-neighboring amino acid residues on aspects of secondary structure of proteins. attempts to locate alpha-helices and beta-sheets. *Biopolymers*, 12(4):751–774, 1973.

[15] J. Klein-Seetharaman, M. Oikawa, S.B. Grimshaw, J. Wirmer, E. Duchardt, T. Ueda, T. Imoto, L.J. Smith, C. Dobson, and H. Schwalbe. Long-range interactions within a nonnative protein. *Science*, 295:1719–1722, 2002.

[16] G. Kleywegt and T. Jones. Phi/psi-chology: Ramachandran revisited. *Structure*, 4:1395–1400, 1996.

[17] T. Kolda and D.P. O'Leary. SDDPACK. `http://www.cs.umd.edu/oleary/SDDPACK/` (accessed: Nov. 2004), 1999.

[18] C. Levinthal. *How to fold graciously*. University of Illinois Press, 1969.

[19] G. Pollastri, D. Przybylski, B. Rost, and P. Baldi. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins*, 47:228–235, 2002.

[20] R. Doyle R, K. Simons, H. Qian, and D. Baker. Local interactions and the optimization of protein folding. *Proteins*, 29(3):282–91, 1997.

[21] C. Ramakrishnan and G.N. Ramachandran. Stereochemical criteria for polypeptide and protein chain conformations. II. allowed conformations for a pair of peptide units. *Biophysics Journal*, 5:909–933, 1965.

[22] B. Rost. Review: Protein secondary structure prediction continues to rise. *Journal Structural Biology*, 134:204–218, 2001.

[23] T. Schwede, J. Kopp, N. Guex, and M.C. Peitsch. SWISS-MODEL: an automated protein homology-modeling server. *Nucleic Acids Research*, 31:3381–3385, 2003.

[24] S. Sheik, P. Ananthalakshmi, G. Ramya Bhargavi, and K. Sekar. CADB: Conformation angles database of proteins. *Nucleic Acids Research*, 31:448–451, 2003.

[25] D.B. Skillicorn, S.M. McConnell, and E.Y. Soong. Handbook of data mining using matrix decompositions. 2003.

[26] C.D. Snow, H. Nguyen, V.S. Pande, and M. Gruebele. Absolute comparison of simulated and experimental protein-folding dynamics. *Nature*, 420:102–106, 2002.

[27] R. Unger and J. Moult. Local interactions dominate folding in a simple protein model. *Journal of Molecular Biology*, 259:988–994, 1996.

[28] M.E. Wall, A. Patricia, and T.S. Brettin. SVDMAN - singular value decomposition analysis of microarray data. *Bioinformatics*, 17:566–568, 2001.

[29] J. Word. DANG. `http://kinemage.biochem.duke.edu/software/dang.php` (accessed: Jan. 2005), 2000.

[30] C. Wu, H. Huang, L. Arminski, J. Castro-Alvear, Y. Chen, Z. Hu, R.S. Ledley, K.C. Lewis, H. Mewes, B.C. Orcutt, B.E. Suzek, A. Tsugita, C.R. Vinayaka, L. Yeh, J. Zhang, and W.C. Barker. The protein information resource: an integrated public resource of functional annotation of proteins. *Nucleic Acids Research*, 30:35–37, 2002.

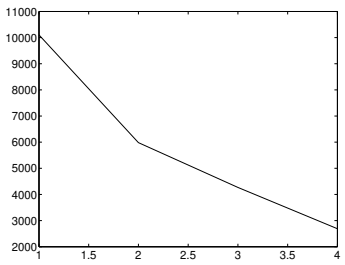
[31] S. Zyto, A. Grama, and W. Szpankowski. Semi-discrete matrix transforms (SDD) for image and video compression. *IEEE Data Compression Conference*, pages 484–487, 2002.

Appendix A

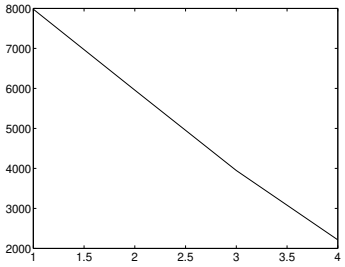
Resulting singular values from SVD

Here are screen plots of singular values obtained from Singular Value Decomposition on sets of torsion angles for the respective amino-acid sequences. Note how the singular values decrease in later dimensions.

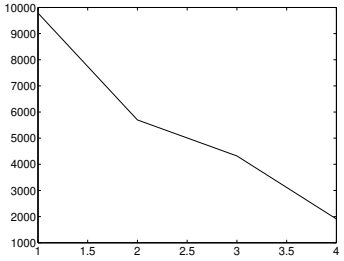
A.1 Sequences of Length 3



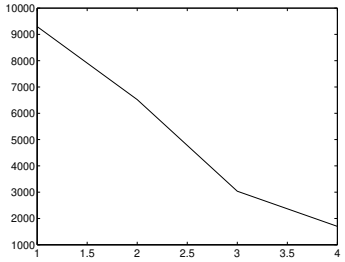
$ALA - ALA - THR$



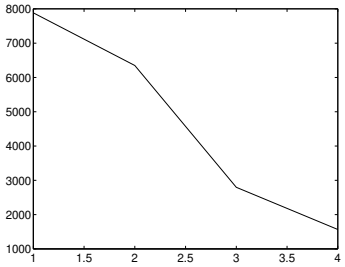
$ASP - LYS - LEU$



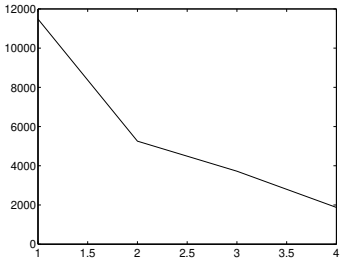
$ASP - VAL - ALA$



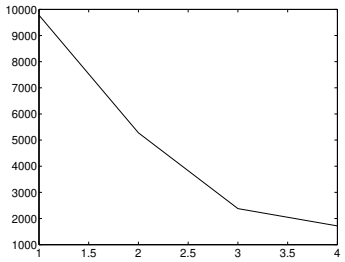
$GLU - VAL - ALA$



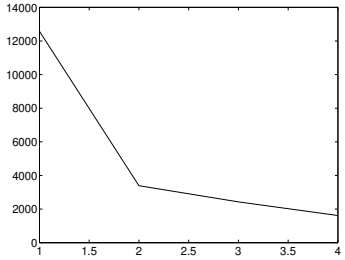
$LEU - GLU - ARG$



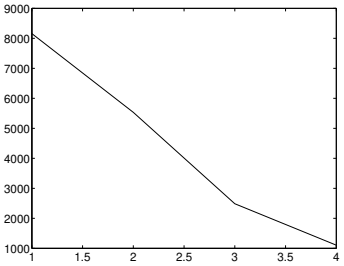
$LEU - THR - SER$



$LYS - LEU - ILE$

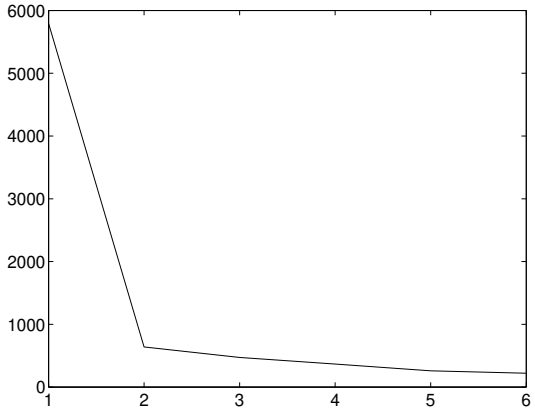


$LYS - PRO - VAL$

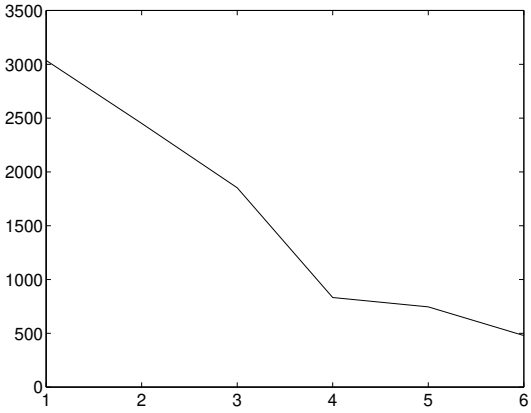


$PHE - LEU - GLU$

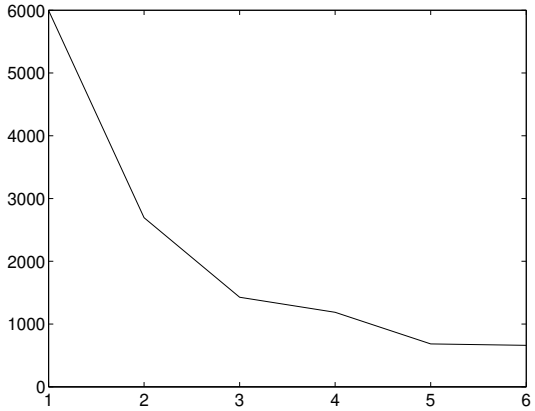
A.2 Sequences of Length 4



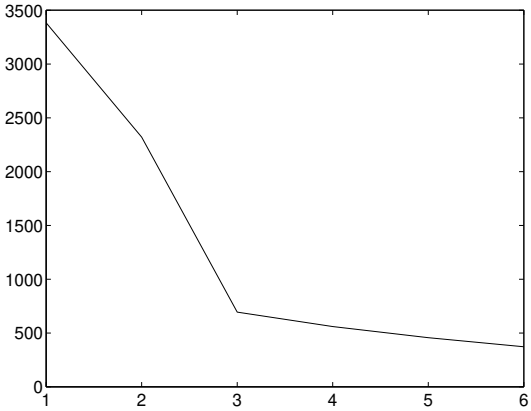
$ALA - ALA - HIS - CYS$



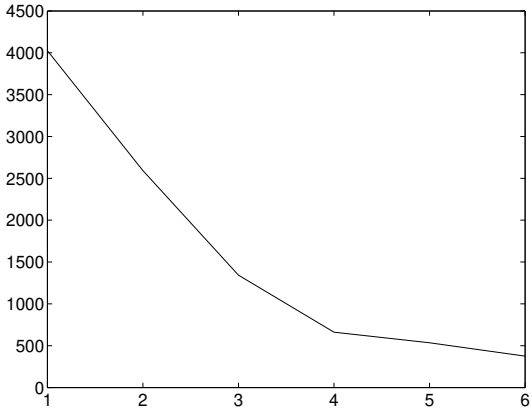
$ALA - ASN - THR - VAL$



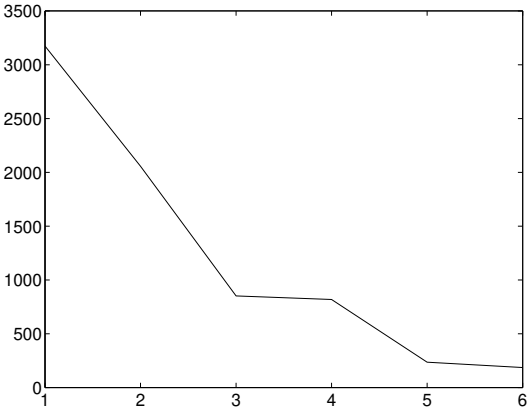
$ALA - ASP - ALA - ALA$



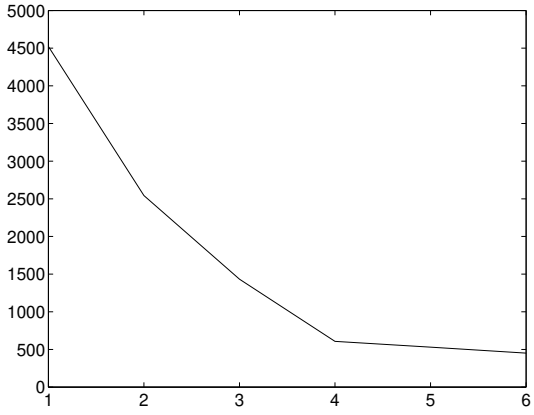
$ALA - GLU - ARG - LEU$



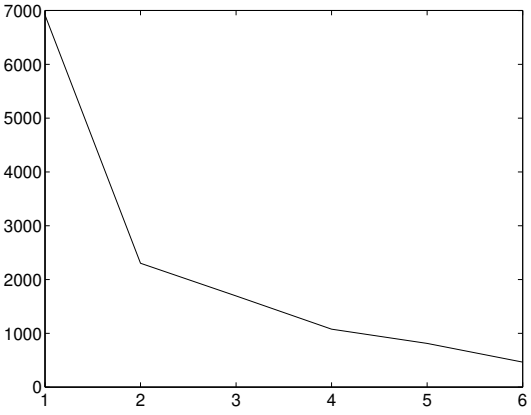
$ALA - LEU - LEU - GLN$



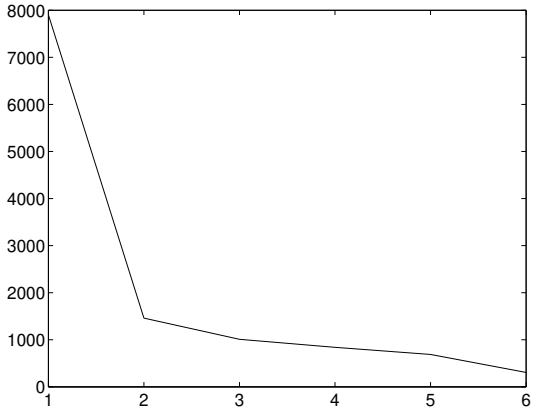
$CYS - SER - ALA - LEU$



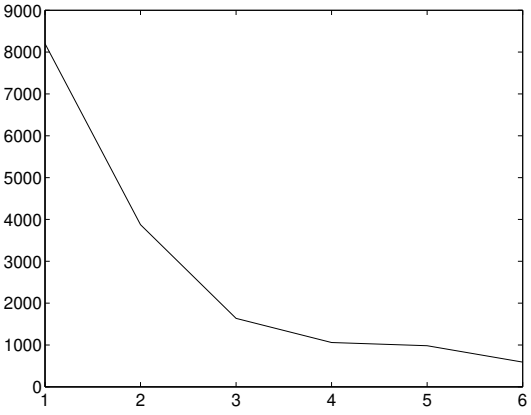
$GLU - GLU - VAL - GLU$



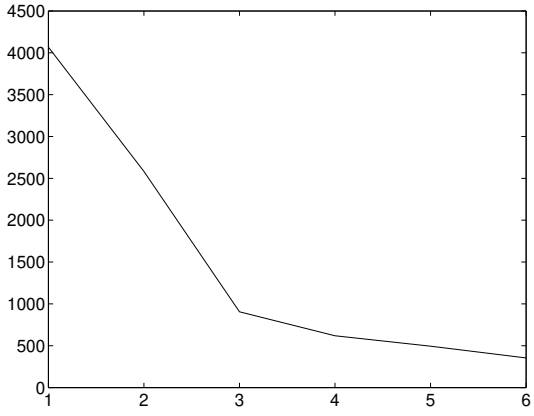
$GLY - LYS - PRO - LEU$



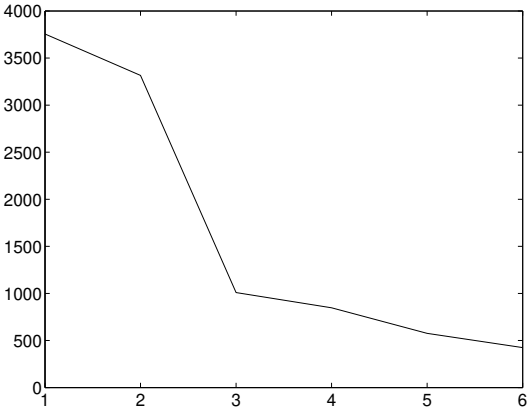
$GLY - PRO - VAL - VAL$



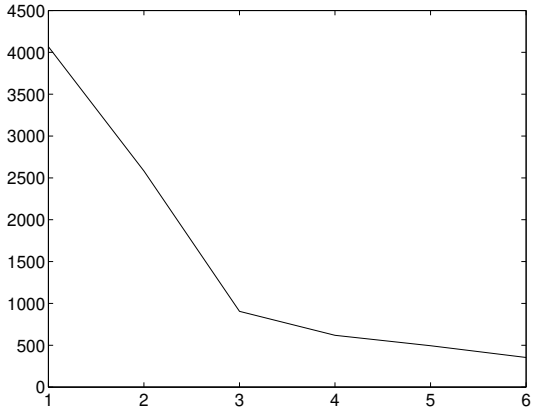
$GLY - VAL - ILE - THR$



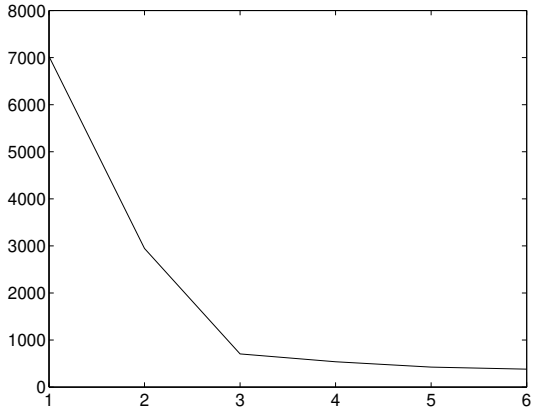
$LEU - ARG - SER - LEU$



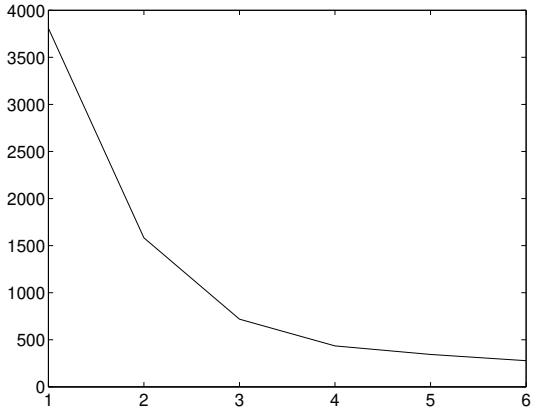
$LEU - LEU - ASP - LEU$



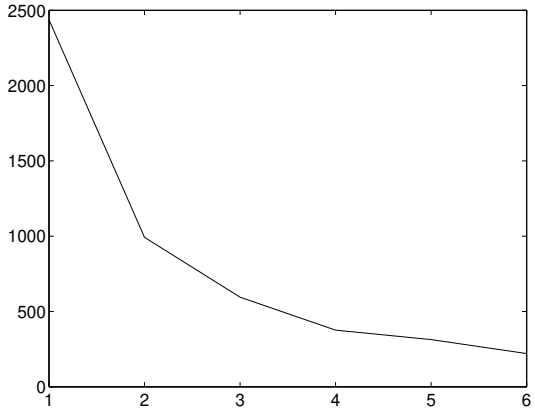
$LEU - ARG - SER - LEU$



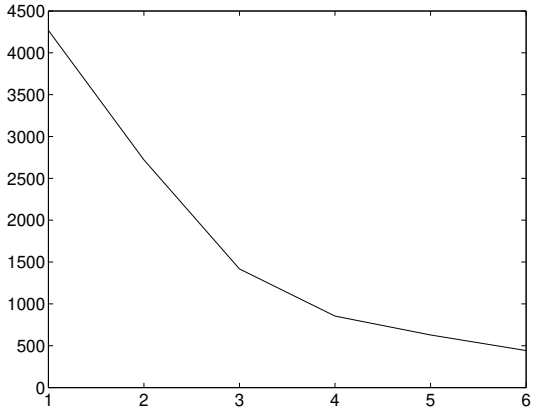
$LYS - GLU - ALA - LEU$



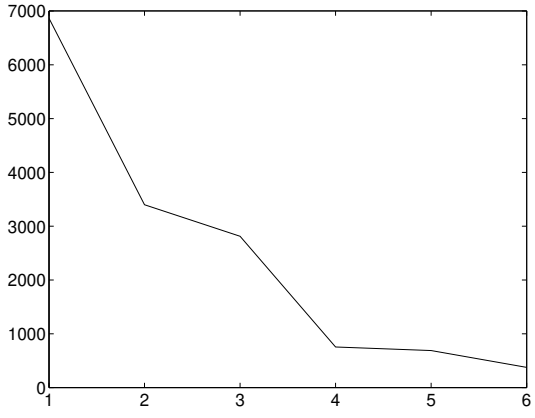
$LYS - LEU - PHE - ASN$



$PHE - VAL - SER - SER$



$PRO - GLU - THR - LEU$



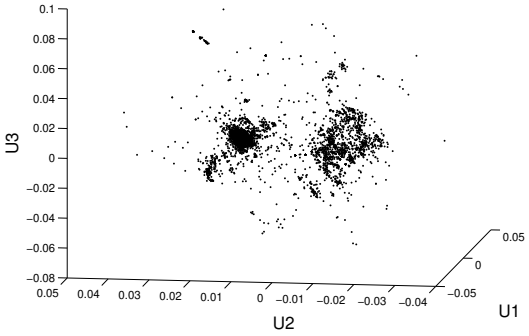
$THR - GLY - THR - TRP$

Appendix B

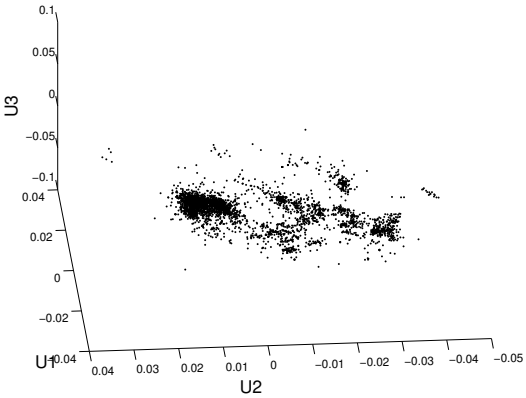
3-dimensional plots of U obtained from SVD

Here are plots of the first 3 dimensions of U matrices from Singular Value Decomposition on sets of torsion angles for the respective sequences of amino acids. The clusters observed in the plot correspond to conformational possibilities for each sequence.

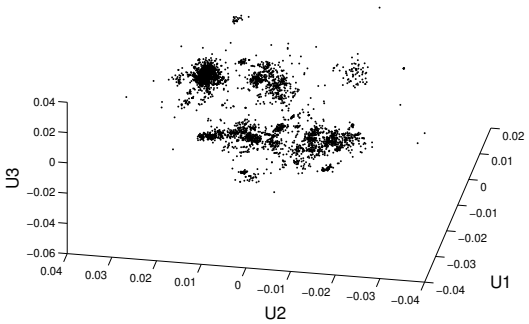
B.1 Sequences of Length 3



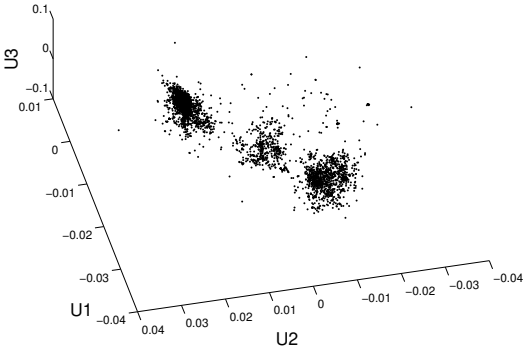
$ALA - ALA - THR$



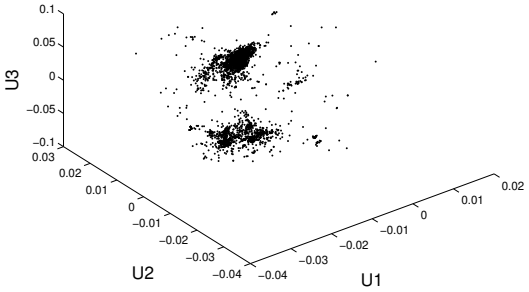
$ASP - LYS - LEU$



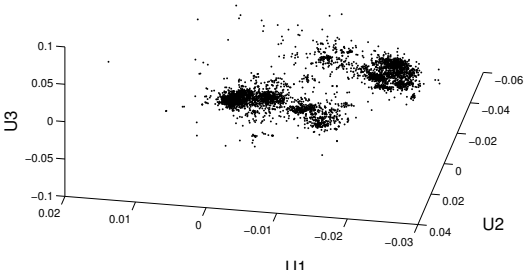
$ASP - VAL - ALA$



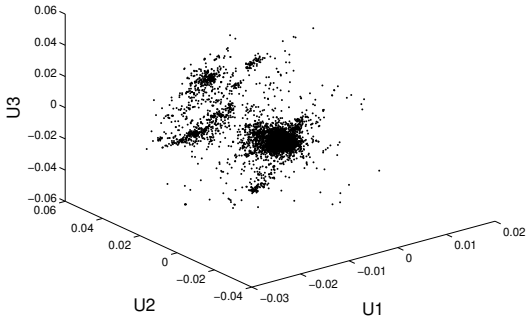
$GLU - VAL - ALA$



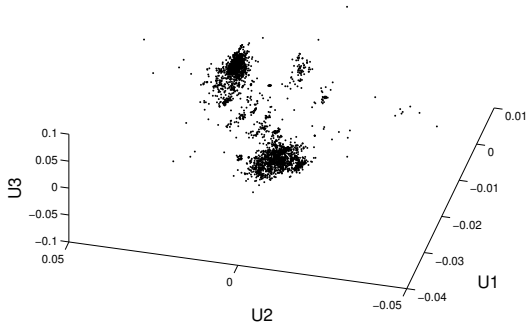
$LEU - GLU - ARG$



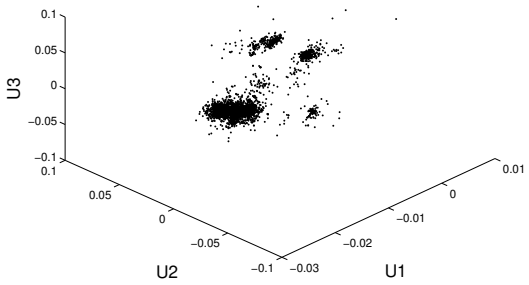
$LEU - THR - SER$



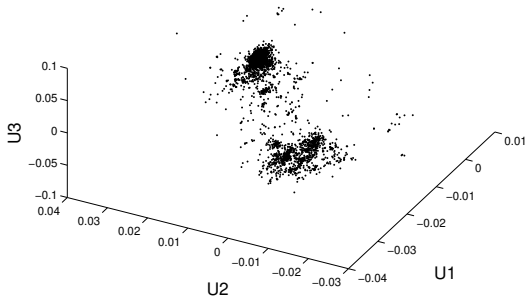
$LYS - GLU - LEU$



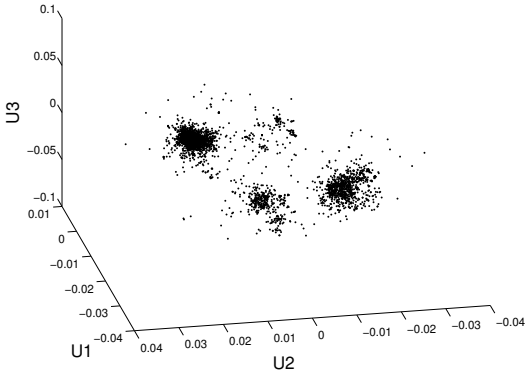
$LYS - LEU - ILE$



$LYS - PRO - VAL$

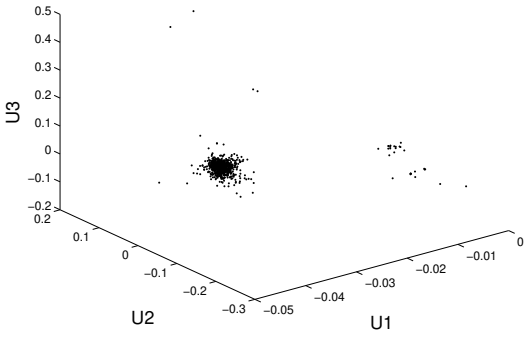


$PHE - LEU - GLU$

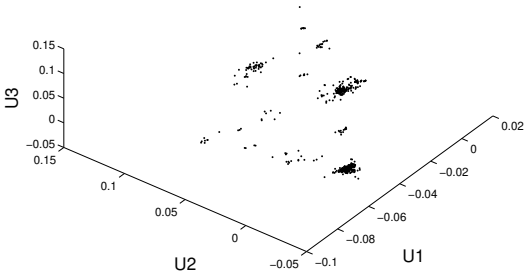


$VAL - GLU - GLU$

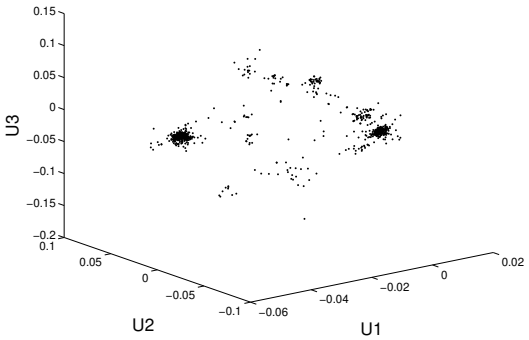
B.2 Sequences of Length 4



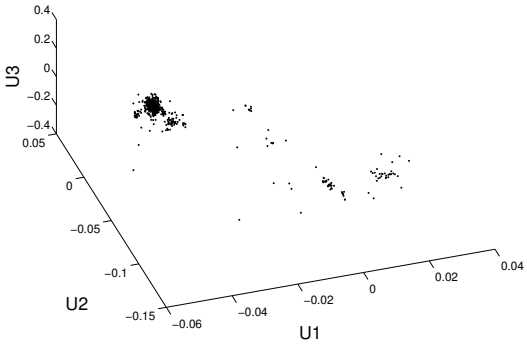
$ALA - ALA - HIS - CYS$



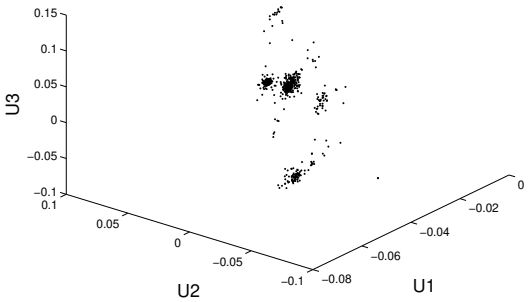
$ALA - ASN - THR - VAL$



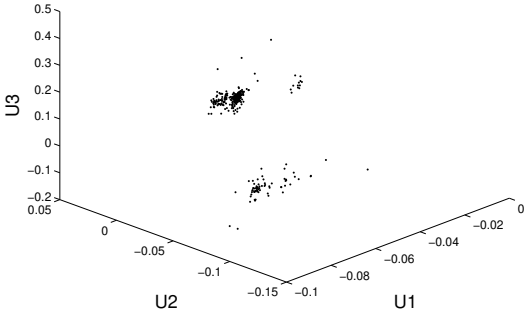
$ALA - ASP - ALA - ALA$



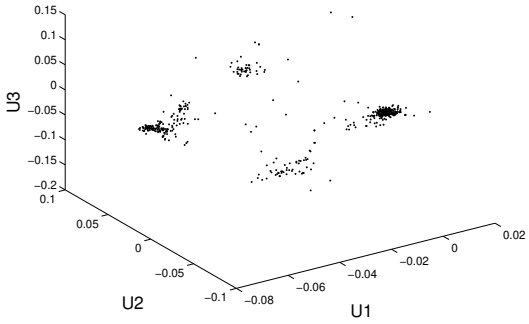
$ALA - GLU - ARG - LEU$



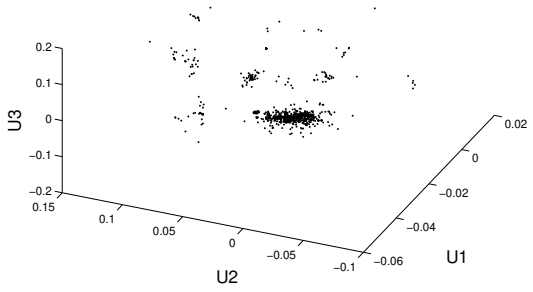
$ALA - LEU - LEU - GLN$



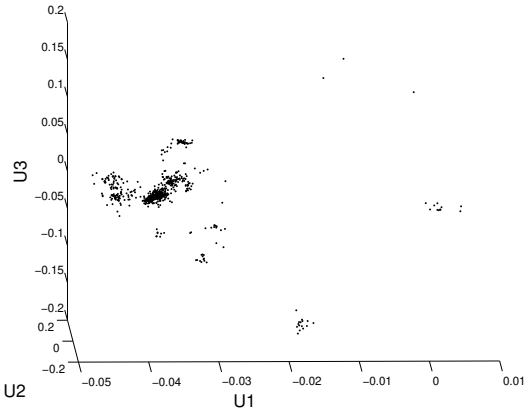
$CYS - SER - ALA - LEU$



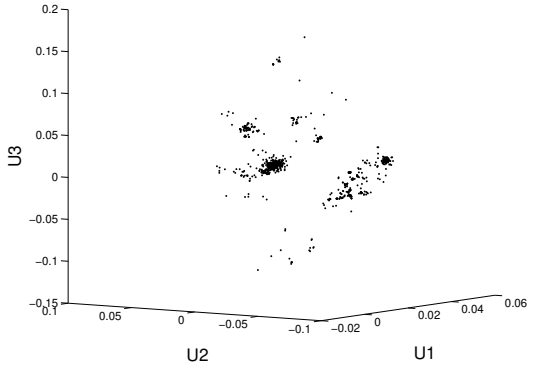
$GLU - GLU - VAL - GLU$



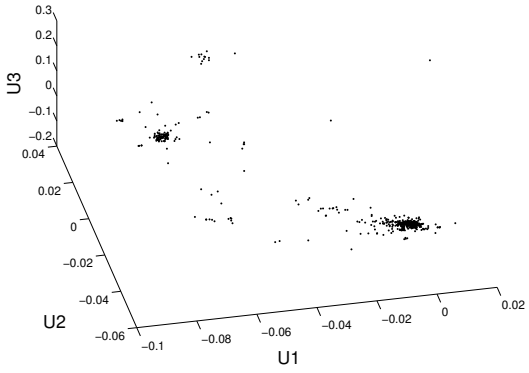
$GLY - LYS - PRO - LEU$



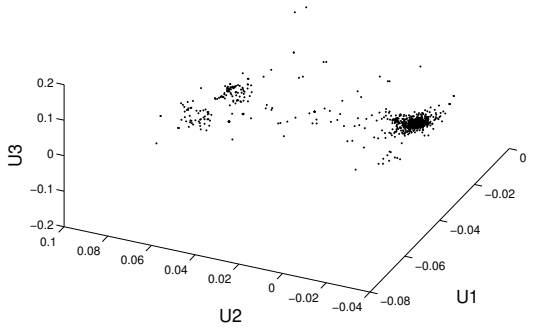
$GLY - PRO - VAL - VAL$



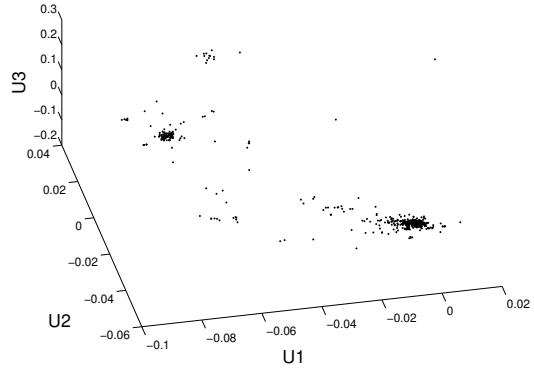
$GLY - VAL - ILE - THR$



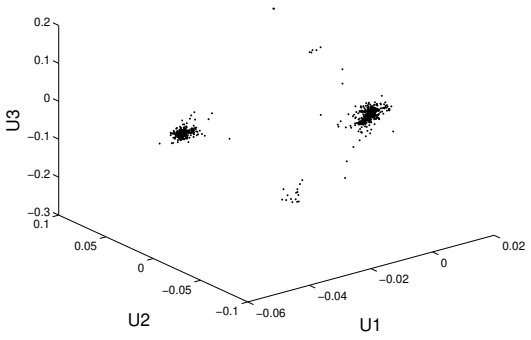
$LEU - ARG - SER - LEU$



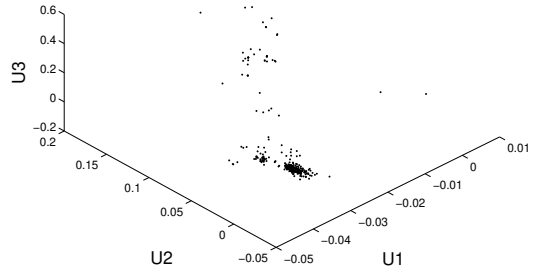
$LEU - LEU - ASP - LEU$



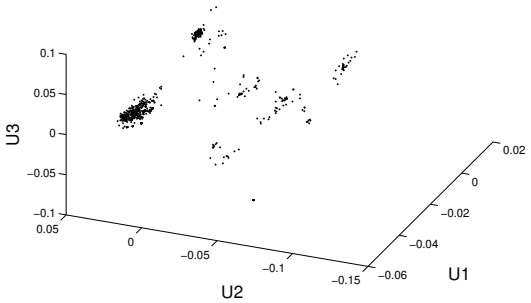
$LEU - ARG - SER - LEU$



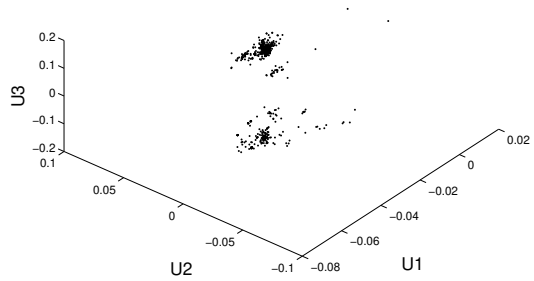
$LYS - GLU - ALA - LEU$



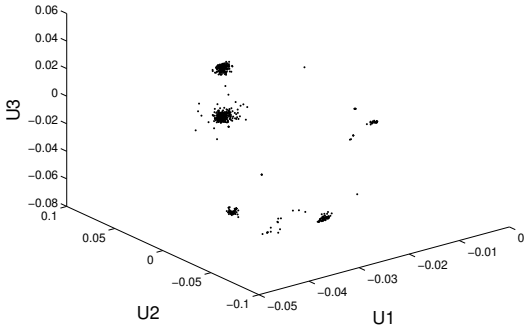
$LYS - LEU - PHE - ASN$



$THR - LEU - GLU - ASP$



$PRO - GLU - THR - LEU$



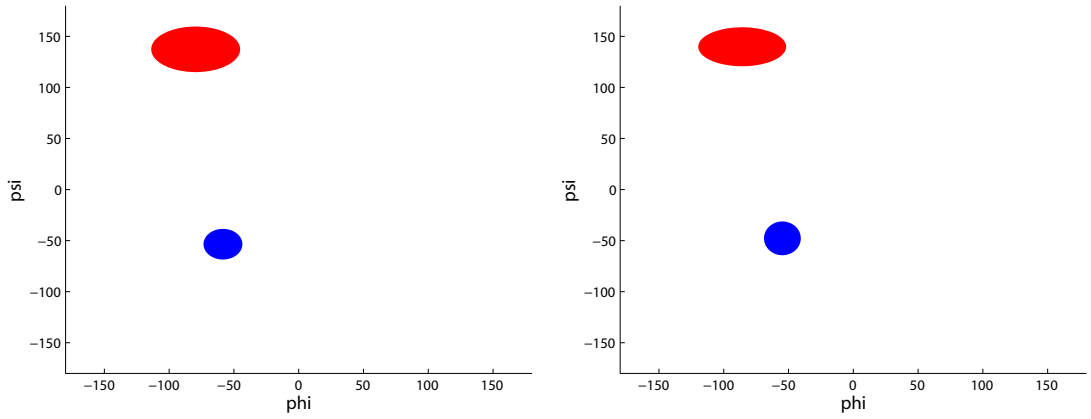
$THR - GLY - THR - TRP$

Appendix C

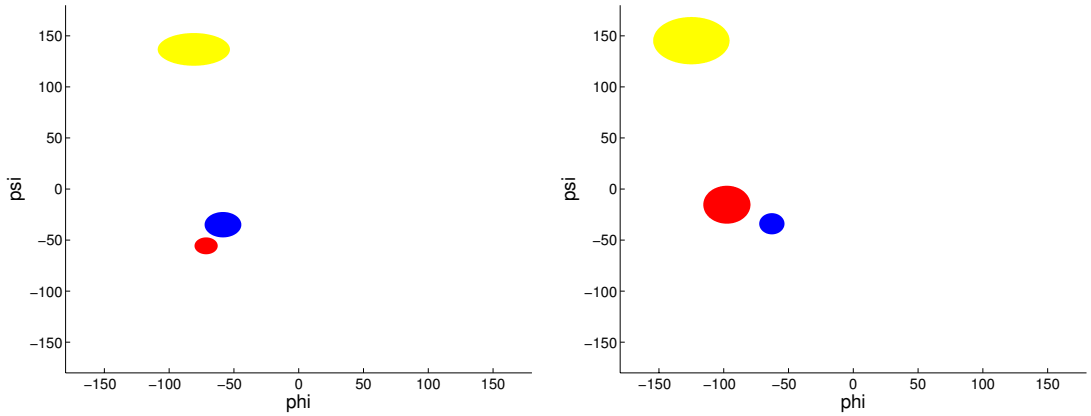
Ramachandran plots with clusters

Here are areas on Ramachandran plots corresponding to conformational possibilities determined by using SVD on the respective amino acid sequences.

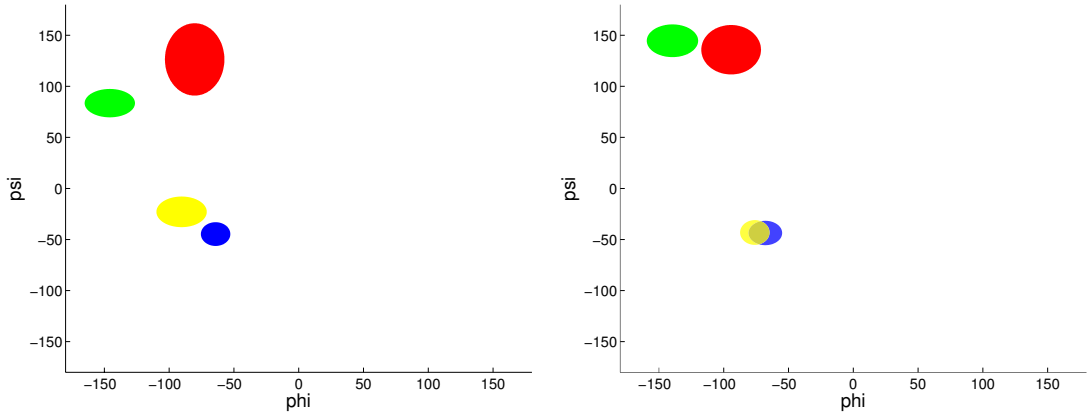
C.1 Sequences of Length 3



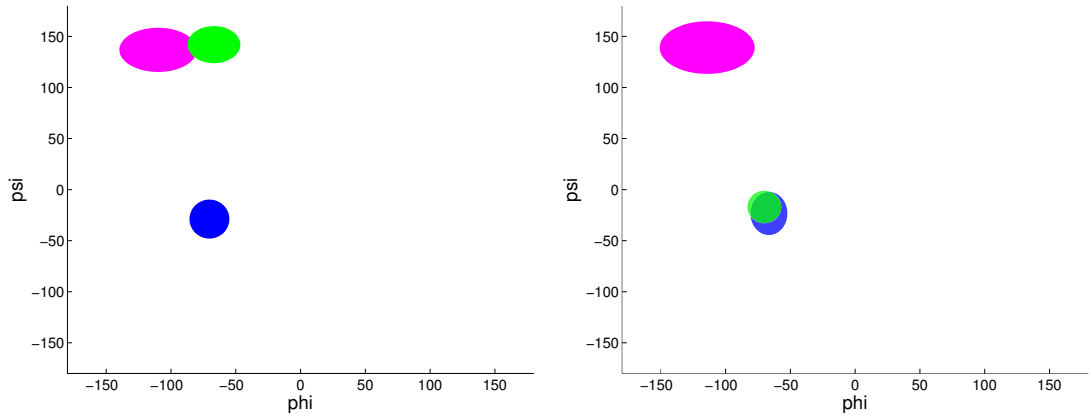
$ALA - ALA - THR$



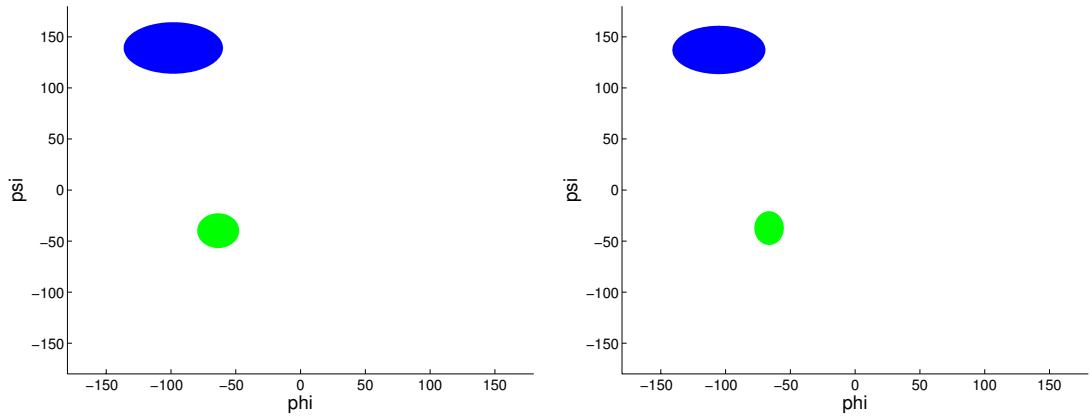
$ASP - LYS - LEU$



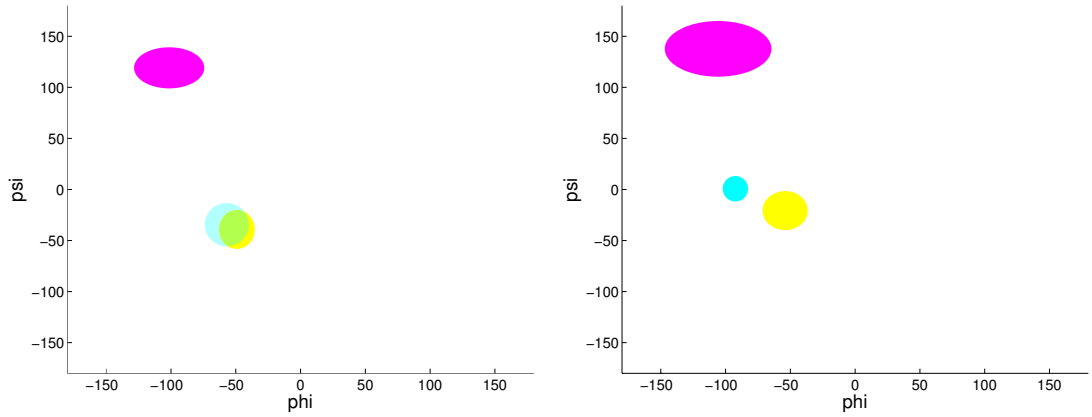
$ASP - VAL - ALA$



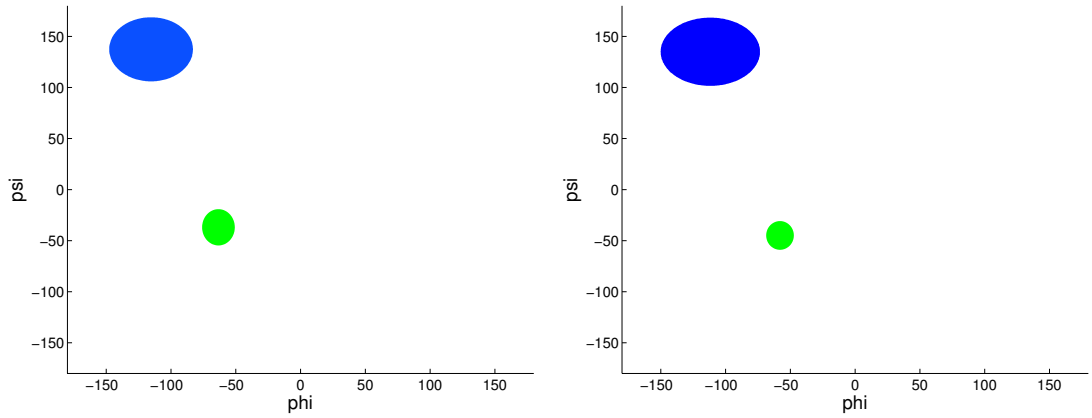
$GLU - VAL - ALA$



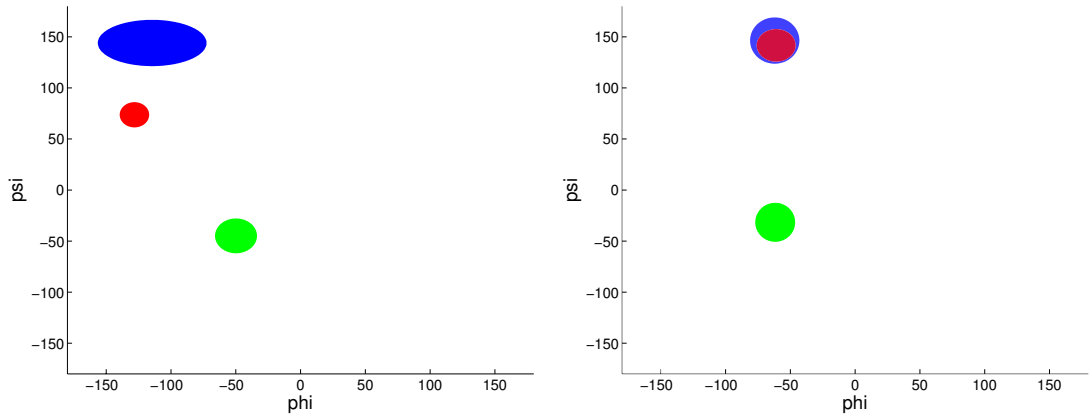
$LEU - GLU - ARG$



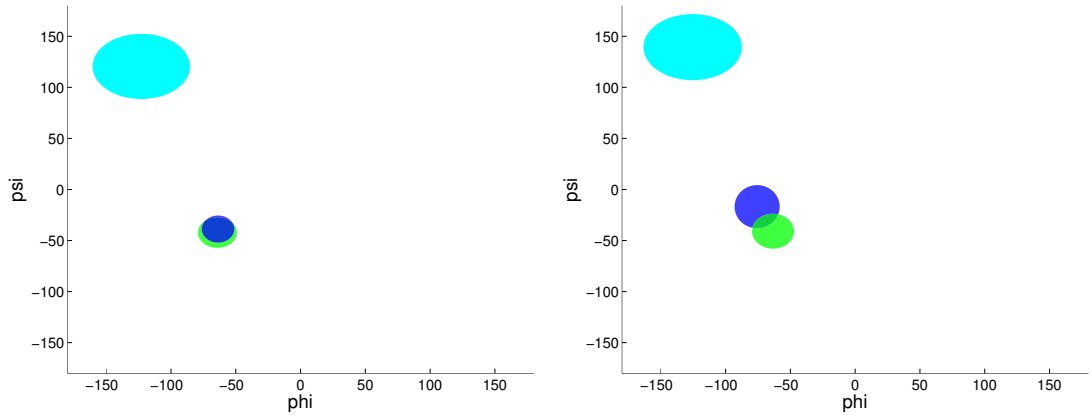
$LEU - THR - SER$



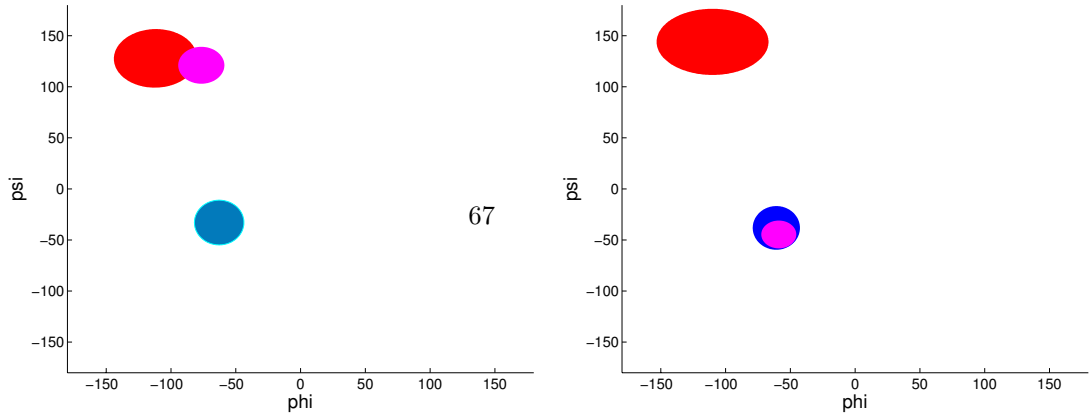
$LYS - LEU - ILE$



$LYS - PRO - VAL$

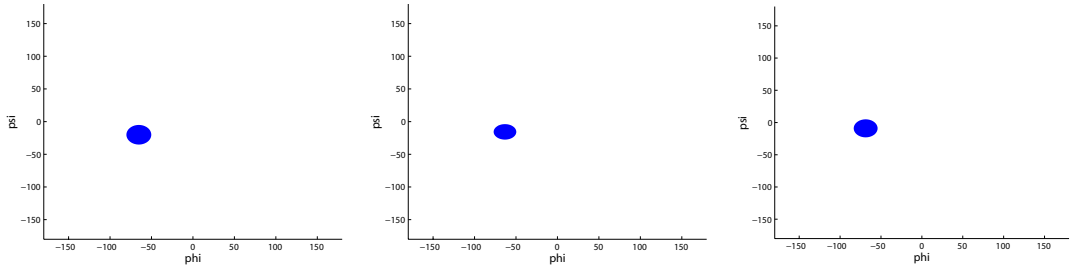


$PHE - LEU - GLU$

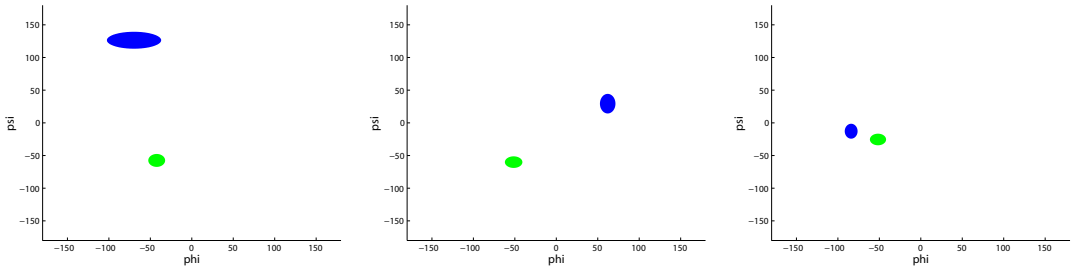


$VAL - GLU - GLU$

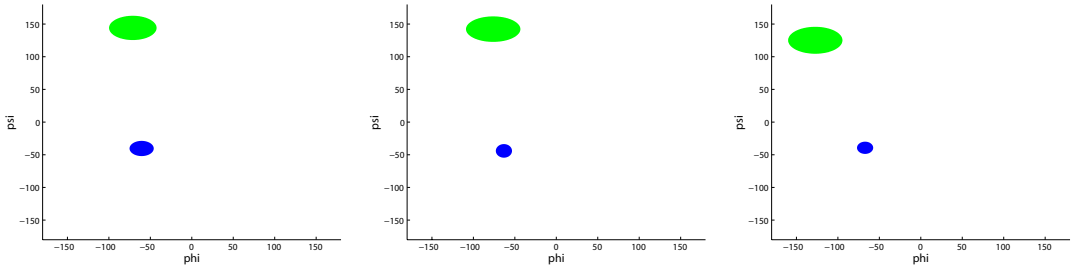
C.2 Sequences of Length 4



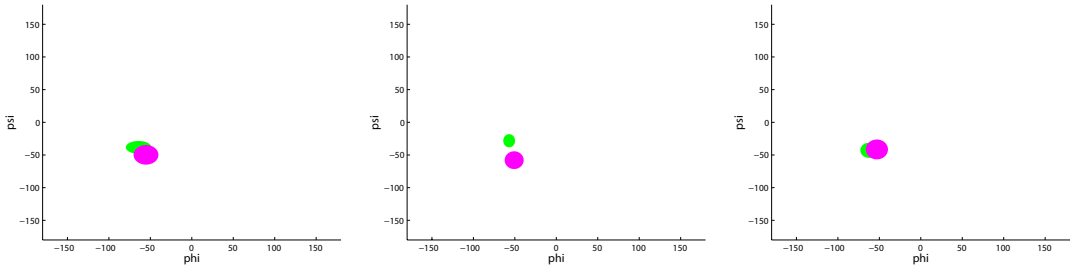
$ALA - ALA - HIS - CYS$



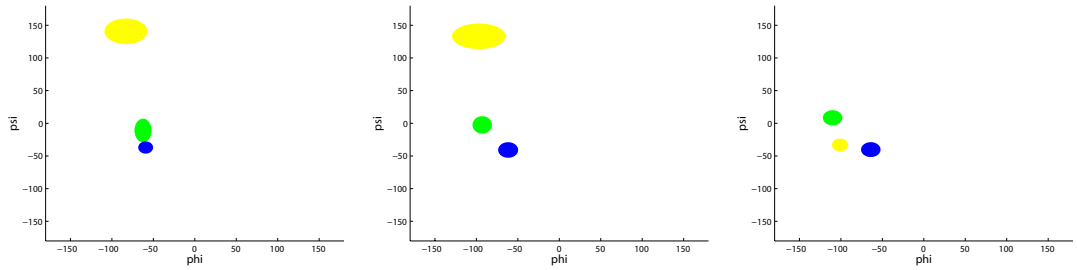
$ALA - ASN - THR - VAL$



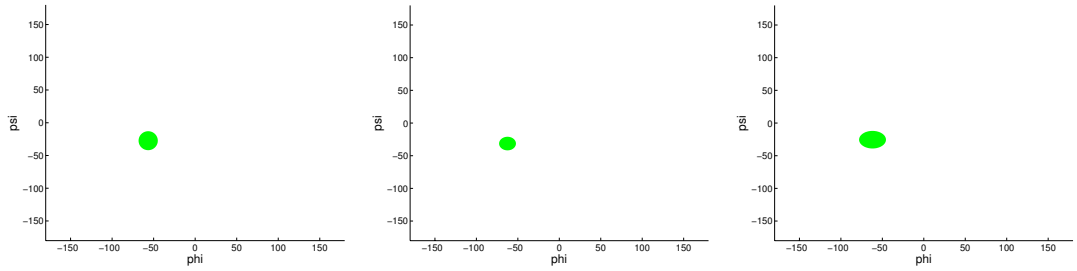
$ALA - ASP - ALA - ALA$



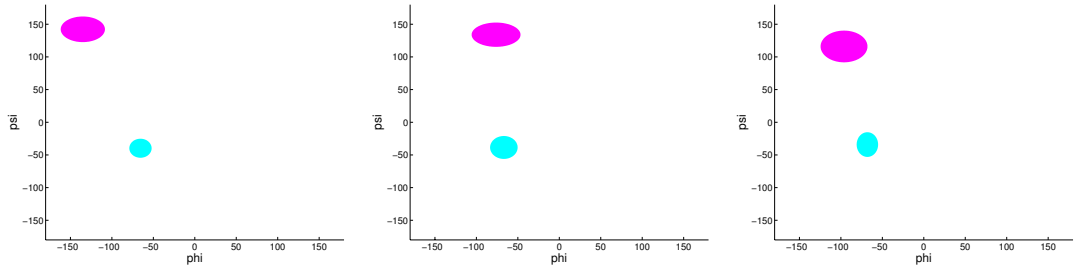
$ALA - GLU - ARG - LEU$



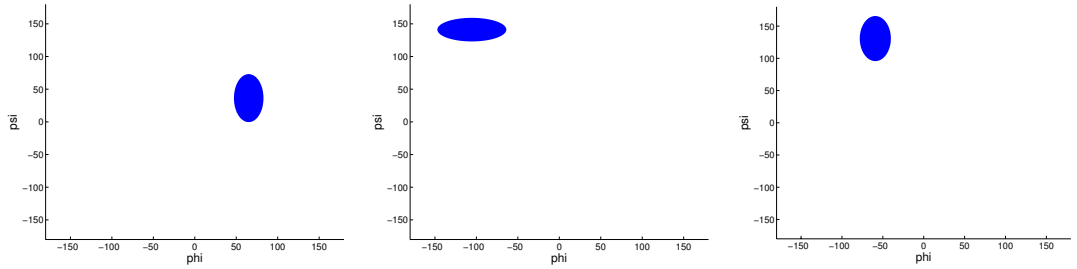
$ALA - LEU - LEU - GLN$



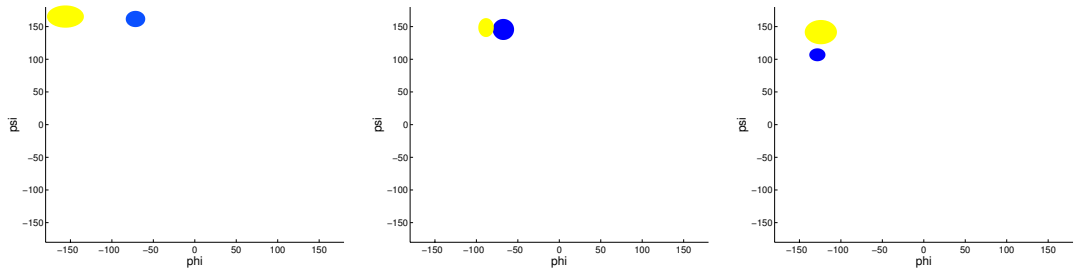
$CYS - SER - ALA - LEU$



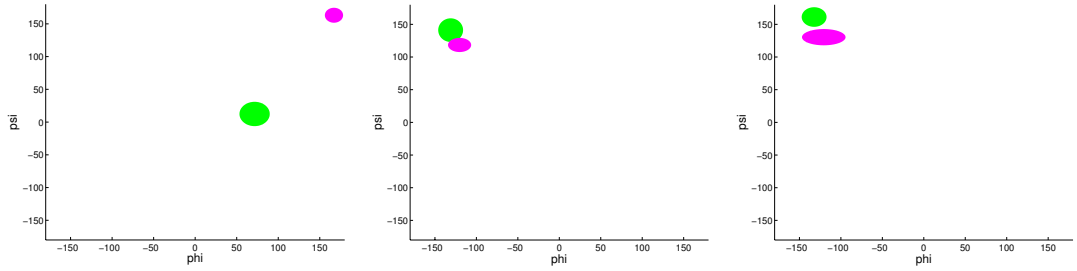
$GLU - GLU - VAL - GLU$



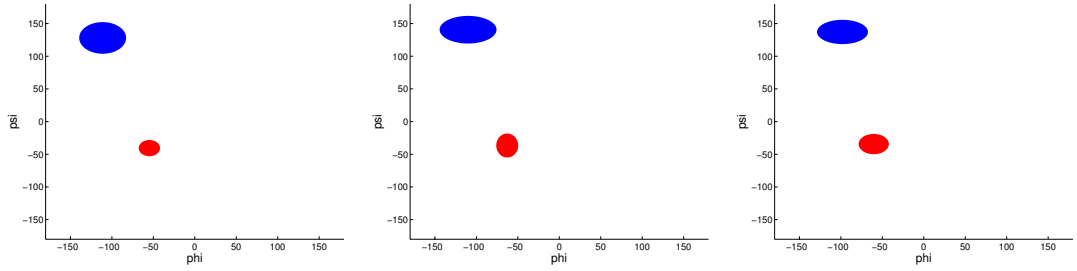
$GLY - LYS - PRO - LEU$



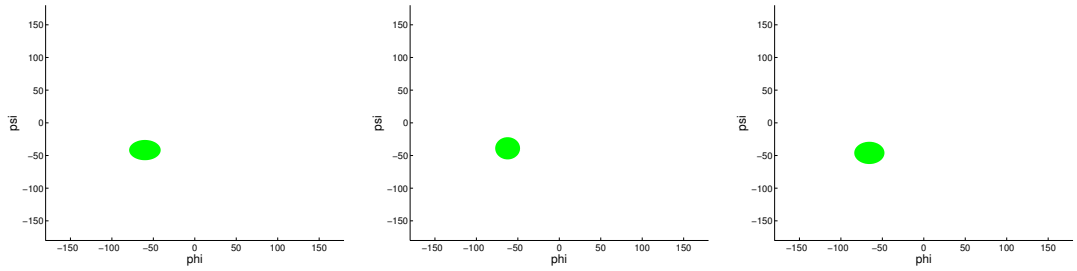
$GLY - PRO - VAL - VAL$



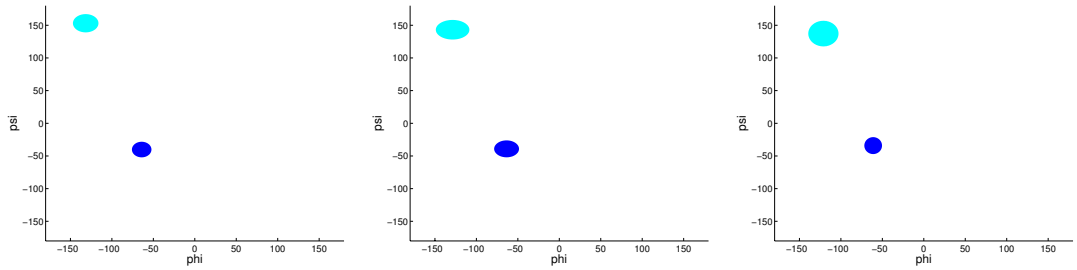
$GLY - VAL - ILE - THR$



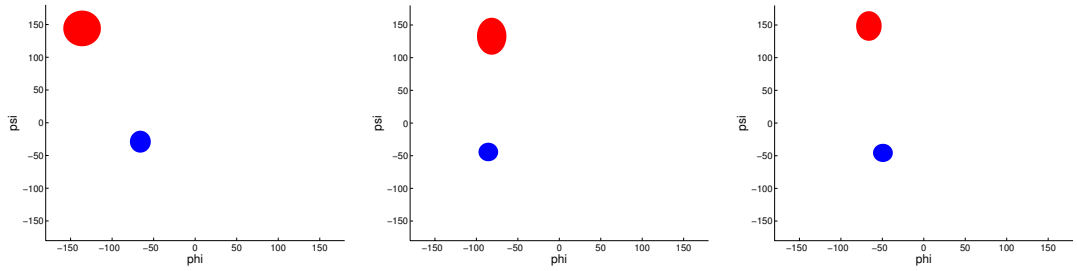
$LEU - ARG - SER - LEU$



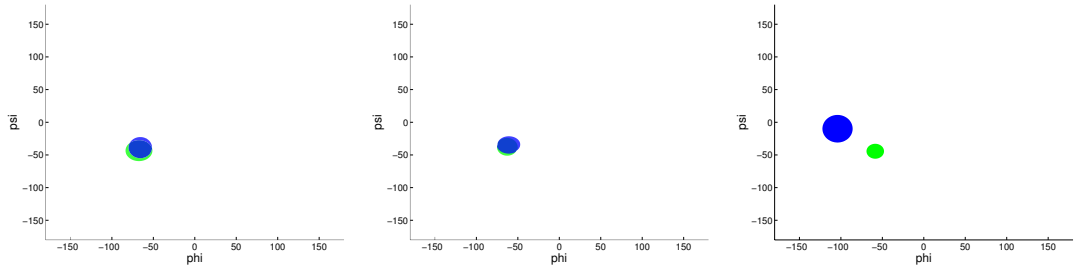
$LEU - LEU - ASP - LEU$



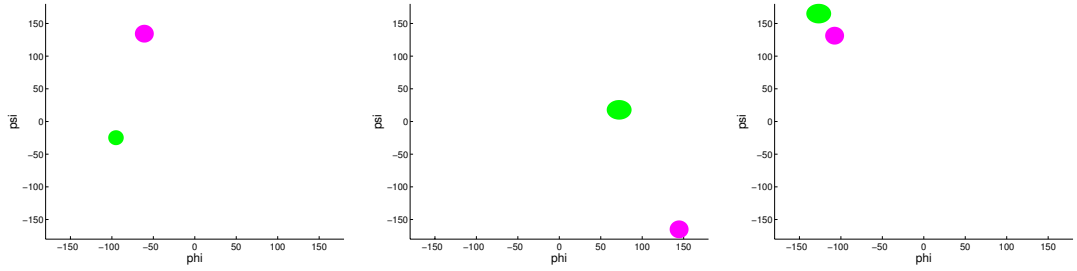
$LYS - GLU - ALA - LEU$



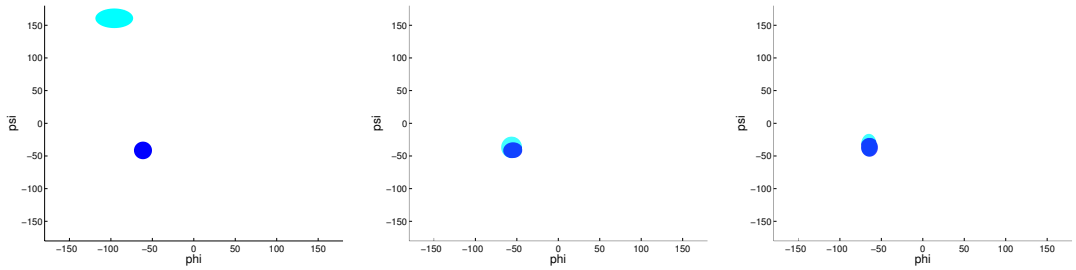
$LYS - LEU - LYS - PRO$



$LYS - LEU - PHE - ASN$



$THR - GLY - THR - TRP$



$THR - LEU - GLU - ASP$

Appendix D

Prediction of structures from the PDB

Here are sequence structure charts of predictions of proteins entered into the PDB after October 18, 2004. The format of the chart is explained in Figure 3.10.

```
1          IVGGYTCAAN SIPYQVSLNS GSHFCGGSLI NSQWVVSAAH CYKSRIQVRL
            B  EE  TT TTTTEEEEES SSB EEEEE  BTTEEEE GG G  SS EEEE
            ooOOOOOOO OOOOOOOOOO OOOOOOOOOO OOOOOOOOOO OOOOOOOOOO

51         GEHNIDVLEG NEQFINAAKI ITHPNFNGNT LDNDIMLIKL SSPATLNSRV
           S SBTTS  S   EEEEESEE EE TT  TTT  TT  EEEEE SS    SSSS
           OOOOOOOOOO OOOOOOOOOO OOOOOOOOOO OOOOOOOOOO OOOOOOOOoo

101        ATVSLPRSCA AAGTECLISG WGNTKSSGSS YPSLLQCLKA PVLSDSSCKS
              EE SS    TT EEEEEE SS   SSS     SB EEEEE EE  HHHHHH
           oooooooooo oooooooooo oooooooooo WWWW???

151        SYPGQITGNM ICVGFLEGGK DSCQGDSGGP VVCNGQLQGI VSWGYGCAQK
           HTTTT  TTE EEES TT S  B  TTTTT E EEETTEE EE E B SSSS T

201        NKPGVYTKVC NYVNWIQQTI AAN
           T  EEEEEGG GSHHHHHHHH HTT
                         ???
```

Figure D.1: A chart which details the amino acid sequence of the protein 1h9i and the prediction of this sequence from SPAA following the sequence/structure chart format.

```
1          SIGTGDRINT VRGPITISEA GFTLTHEHIC GSSAGFLRAW PEFFGSRKAL
            EESSSEEEE TTEEEEHHHH  SEE SB SE E  TTHHHH  GGGGS HHHH
              CCCoooo oooooooooo oooooooooo oooooooooo oooooooooo

51         AEKAVRGLRR ARAAGVRTIV DVSTFDIGRD VSLLAEVSRA ADVHIVAATG
           HHHHHHHHHH HHHTT  EEE E   GGGT   HHHHHHHHHH HT EEE EEE
           oooooooooo oooooooooo oooooooooo ooooNNN    NNNoooooooo

101        LWFDPPLSMR LRSVEELTQF FLREIQYGIE DTGIRAGII( LCX)VATTGK
             S   HHHH T  HHHHHHH HHHHHHT ST TT    SEEE EE SSS  HH
           oooooooooo oooNNN

151        ATPFQELVLK AAARASLATG VPVTTHTAAS QRDGEQQAAI FESEGLSPSR
           HHHHHHHHHH HHHHH  EEE EE  GGGTHH HHHHHHHHHT T  GGGEEE
...
```

Figure D.2: A chart which details the amino acid sequence of the protein 1qw7 and the prediction of this sequence from SPAA following the sequence/structure chart format.

```
  1 EVKLVESGGG LVQPGGSLKL SCAASGFTFS TYTMSWARQT PEKKLEWVAY
      EEEEE  E EE TT  EEE EEEEESS GG GS EEEEEEE TT  EEE EE
       wwwoooo oooooooooo oooooooooo oNNN

 51 ISKGGGSTYY PDTVKGRFTI SRDNAKNTLY LQMSSLKSED TALYYCARGA
    E TTSS EEE  TTTTTTEEE EEEGGGTEEE EEE S  GGG  EEEEEEE
        NNNooo oooooooooo oooooooooo oooooooooo ooooooooCC

101 MFGNDFKYPM DRWGQGTSVT VSSAATTPPS VYPLAPGSAA QTNSMVTLGC
    EEETTEEE S BS    EEEE E SS  B  E EEEE          SEEEEEE
    CC
...
```

Figure D.3: A chart which details the amino acid sequence of the protein 1seq and the prediction of this sequence from SPAA following the sequence/structure chart format.

```
  1 MPRSLANAPI MILNGPNLNL LGQRQPEIYG SDTLADVEAL CVKAAAAHGG
        TTTS E EEEE TTGGG TTSS HHHH  S  HHHHHHH HHHHHHHHT
           CCC oooooooooo oooooooooo oooooooooo oooooooooo

 51 TVDFRQSNHE GELVDWIHEA RLNHCGIVIN PAAYSHTSVA ILDALNTCDG
     EEEEE S H HHHHHHHHHH HHH SEEEEE  GGGGTTTHH HHHHHHHTTT
    oooooooooo oooooooooo oooooooooo oooooooooo oooooooooo

101 LPVVEVHISN IHQREPFRHH SYVSQRADGV VAGCGVQGYV FGVERIAALA
      EEEEESS  GGGTTGGGS   SHHHH SEE EESSTTHHHH HHHHHHHHHH
    oooooooooo oooooooooo ooooooooC CC

151 GAGSARA
```

Figure D.4: A chart which details the amino acid sequence of the protein 1v1j and the prediction of this sequence from SPAA following the sequence/structure chart format.

Appendix E

Protein 1crn

A sequence/structure diagram is provided for protein **1crn** to demonstrate the complexity of the secondary structural elements. The actual torsion angles of the entire sequence are shown along with a set of predicted torsion angles from the SPAA algorithm. Angles which SPAA incorrectly predicted are marked in bold.

E.1 Sequence

```
1 TTCCPSIVAR SNFNVCRLPG TPEAICATYT GCIIIPGATC PGDYAN
   EE SSHHHH HHHHHHHHTT   HHHHHHHH S EE SSS    TTS
```

E.2 Torsion angles

```
A  : (-107,144)  (-131,133)  (-118,151)  (-76,-18)  (-157,166)  (-63,-42)
P  : (-107,143)  (-131,136)  (-122,148)  (-75,-20)  (-156,167)  (-58,-49)

A..: (-55,-44)   (-61,-43)   (-63,-43)   (-61,-42)  (-64,-39)   (-59,-47)
P..: (-60,-42)   (-61,-51)   (-59,-44)   (-60,-42)  (-69,-34)   (-58,-46)

A..: (-62,-35)   (-69,-41)   (-56,-36)   (-77,-16)  (-53,-46)   (-77,-7)
P..: (-66,-34)   (-67,-43)   (-57,-36)   (-76,-23)  (-55,-45)   (-76,-6)

A..: (106,7)     (-52,136)   (-56,146)   (-56,-36)  (-63,-34)   (-74,-37)
P..: (96,8)      (-69,149)   (-57,-34)   (-64,-36)  (-64,-42)   (-67,-39)

A..: (-64,-31)   (-62,-54)   (-68,-25)   (-67,-36)  (-108,-18)  (91,-3)
P..: (-64,-35)   (-64,-52)   (-67,-27)   (-68,-35)  (-108,-20)  (91,-3)

A..: (-69,164)   (-129,157)  (-111,129)  (-124,158) (-78,-24)   (-89,-161)
P..: (-67,167)   (-134,158)  (-112,128)  (-123,156) (-77,-26)   (-83,-168)

A..: (-120,1)    (-114,104)  (-75,145)   (-71,162)  (-61,-23)
P..: (-118,1)    (-112,100)  (-73,142)   (-75,165)  (-62,-23)
```