ADVANCES IN COMPUTATIONAL PROTEIN STRUCTURE PREDICTION

ASHWIN SUBRAMANI

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
CHEMICAL AND BIOLOGICAL ENGINEERING
ADVISER: PROFESSOR CHRISTODOULOS A. FLOUDAS

NOVEMBER 2011
© Copyright by Ashwin Subramani, 2011.

All rights reserved.
Abstract

The thesis is premised on the application of optimization theory, and algorithms based on it, to the protein structure prediction problem. The protein structure prediction problem can be expressed simply as, “Given the amino acid sequence of the protein, what is its three dimensional structure?”. A number of theories suggest alternate pathways for a protein to undertake the folding process. One such theory is the hierarchical theory of protein folding, which proposes that local secondary structure of proteins is formed prior to their three dimensional arrangement. Hence, the thesis aims to address a number of the intermediate problems to the tertiary structure prediction problem.

Given an amino acid sequence of a protein, we first aim to predict the location of the secondary structure elements. To address this issue, a novel mixed-integer linear optimization model has been developed. The model divides a given protein sequence into overlapping nonapeptides, and evaluates the likelihood of the central amino acid to be in an $\alpha$-helix. This likelihood is expressed as a weighted linear sum of the pairwise probabilities of neighboring amino acid pairs to form hydrogen bonds. In addition, chemical shift based data from a large database is used to reduce the superstructure of possible helical locations in the protein. Having gathered information on the location of $\alpha$-helices and $\beta$-strands through different means, a novel mixed-integer optimization model to predict $\beta$-sheet topology has been developed. Accurate prediction of the topology of a protein would provide important distance constraints between amino acids separated in the primary sequence. The model aims to maximize the pseudo-contact energy between $\beta$-strands in the protein. In addition, a model based on torsion angle dynamics and clustering aims to re-rank the shortlisted set of topologies in order to identify the native topology of the protein.

In addition to constraints derived out of location and mutual contacts of secondary structures, it is important to impose distance and angle constraints on the disordered loop regions of the protein. Unlike the previous methods, the flexible nature of loops precludes successful prediction using
only database-based methods. Hence, a novel loop structure prediction framework has been developed, which incorporates non-linear non-convex optimization, along with dihedral angle sampling and discrete side-chain optimization. An iterative approach is introduced to sequentially reduce the predicted bounds on the dihedral angles. All of these preceding steps are used to generate constraints, which are incorporated into the three dimensional structure prediction. The tertiary structure prediction algorithm combines deterministic global optimization, stochastic conformational space annealing and torsion angle dynamics to generate structural conformers which satisfy the constraints. For a blind case study, it is difficult to determine the native structure from an ensemble. Hence, a new, traveling-salesman problem (TSP) based clustering approach has been introduced. The method iteratively eliminates low quality structures from the ensemble, and eventually helps select five conformers which are closest to the native structure from the generated ensemble.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abstract</strong></td>
<td>iii</td>
</tr>
<tr>
<td><strong>List of Tables</strong></td>
<td>x</td>
</tr>
<tr>
<td><strong>List of Figures</strong></td>
<td>xii</td>
</tr>
<tr>
<td><strong>List of Publications</strong></td>
<td>xv</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>2 Helical Prediction in Proteins</strong></td>
<td>11</td>
</tr>
<tr>
<td>2.1 Background</td>
<td>11</td>
</tr>
<tr>
<td>2.1.1 Residue based approaches</td>
<td>13</td>
</tr>
<tr>
<td>2.1.2 Hydrogen Bonding</td>
<td>14</td>
</tr>
<tr>
<td>2.2 Database Analysis</td>
<td>17</td>
</tr>
<tr>
<td>2.2.1 Classification of Protein into regions</td>
<td>19</td>
</tr>
<tr>
<td>2.2.2 Helix Beginnings and Helix Ends</td>
<td>20</td>
</tr>
<tr>
<td>2.2.3 Objective function and Constraints</td>
<td>21</td>
</tr>
<tr>
<td>2.3 Helix Prediction Model</td>
<td>24</td>
</tr>
<tr>
<td>2.3.1 Additional Constraints</td>
<td>25</td>
</tr>
<tr>
<td>2.4 Chemical Shift Data-based constraints</td>
<td>30</td>
</tr>
<tr>
<td>2.4.1 Collection of Chemical Shift data</td>
<td>30</td>
</tr>
<tr>
<td>2.4.2 Filtering of Chemical Shift Cluster Results</td>
<td>31</td>
</tr>
</tbody>
</table>
3.5.3 Accuracy Evaluation ............................................. 71
3.5.4 Results: PDBSelect25 Data Set .................................... 71
3.5.5 Results: CASP9 .................................................. 76
3.6 Discussion and Conclusions ........................................ 79

4 Loop Structure Prediction ........................................... 81
  4.1 Challenges in Loop Structure Prediction ........................... 82
    4.1.1 Flexible stem and Fixed stem loop structure prediction ....... 83
  4.2 Background .................................................... 84
  4.3 Methods ...................................................... 87
    4.3.1 Derivation of Dihedral Angle Propensities ..................... 87
    4.3.2 Generation of Initial Structures ................................ 89
    4.3.3 Rotamer Optimization ......................................... 90
    4.3.4 Energy Minimization .......................................... 95
    4.3.5 Clustering ................................................ 96
    4.3.6 Generation of improved bounds and iterative approach ........ 98
  4.4 Results and Discussion ........................................ 98
    4.4.1 Selected CASP9 targets ..................................... 103
  4.5 Conclusions .................................................. 107

5 Near-native structure identification ................................. 108
  5.1 Introduction .................................................. 108
  5.2 Methods ...................................................... 112
    5.2.1 TSP Implementation of the OREO Approach ..................... 113
    5.2.2 Cluster boundary definition and analysis ..................... 116
    5.2.3 Evaluating cluster medoids and average spherical radii .......... 123
    5.2.4 Eliminating clusters based on cluster properties .............. 124
5.2.5 Selection of near-native structures ............................................. 126
5.2.6 Automated Implementation ..................................................... 126
5.3 Computational Results .............................................................. 127
  5.3.1 High Resolution Data Set ..................................................... 127
  5.3.2 Medium to low resolution data set: Decoys ‘R’ Us ................. 131
  5.3.3 Medium to low resolution data set: ASTROFOLD ................. 133
  5.3.4 Selected CASP8 and CASP9 Targets ..................................... 133
  5.3.5 Stage-wise Enrichment of ICON ........................................... 134
5.4 Discussion and Conclusions ....................................................... 136

6 Tertiary structure prediction: Computational Studies ..................... 138
  6.1 Introduction ............................................................................. 138
  6.2 ASTRO-FOLD 2.0 framework .................................................. 142
    6.2.1 Secondary structure prediction .......................................... 142
    6.2.2 \( \beta \)-sheet topology prediction ..................................... 144
    6.2.3 Contact Prediction ............................................................. 145
    6.2.4 Loop Structure Prediction .................................................. 146
  6.3 Tertiary structure prediction ................................................... 146
    6.3.1 Energy function ................................................................. 147
    6.3.2 Problem Formulation .......................................................... 148
    6.3.3 Deterministic Global Optimization using \( \alpha \)BB ..................... 148
    6.3.4 Torsion Angle Dynamics .................................................... 150
    6.3.5 Conformational Space Annealing ........................................ 151
    6.3.6 Improved distance and dihedral bound generation using chemical shifts . 153
    6.3.7 Parallel implementation of ASTROFOLD 2.0 framework ......... 154
  6.4 Computational Studies ............................................................. 155
6.4.1 T581:3NPD .................................................. 158
6.4.2 T602:3NKZ .................................................. 163
6.4.3 T562:2KZX .................................................. 165
6.4.4 T580:3NBM .................................................. 169
6.4.5 T596:3NI7 .................................................. 172

6.5 Conclusions .................................................. 175

7 Conclusions and Future work .................................. 176

7.1 Conclusions .................................................. 176

7.1.1 Prediction of α-helical regions ......................... 176
7.1.2 β-sheet topology prediction .......................... 177
7.1.3 Loop Structure Prediction .............................. 178
7.1.4 Near-native structure identification ................ 178
7.1.5 Tertiary Structure Prediction .......................... 179

7.2 Future Work .................................................. 180

7.2.1 β-strand and β-sheet topology prediction ............. 180
7.2.2 Supersecondary structure prediction .................. 181
7.2.3 Conformational Search Strategies ..................... 182
7.2.4 Energy Function ......................................... 182
7.2.5 Hybrid approach and Future Blind Studies .......... 183

Bibliography .................................................... 185
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Helix Beginning and End Propensity Coefficients</td>
<td>38</td>
</tr>
<tr>
<td>2.2</td>
<td>Q3 and SOV accuracy for helix predictions for targets from CASP8 and CASP9 experiments</td>
<td>42</td>
</tr>
<tr>
<td>3.1</td>
<td>The number of motifs possible for a protein with ( n ) strands</td>
<td>49</td>
</tr>
<tr>
<td>3.2</td>
<td>Comparison of Average Precision and Recall values of top solutions of BeST and BetaPro</td>
<td>77</td>
</tr>
<tr>
<td>3.3</td>
<td>The distribution of the number of proteins in the blind target set of CASP9 with ( n ) strands</td>
<td>79</td>
</tr>
<tr>
<td>5.1</td>
<td>Results for Medium Resolution Data Set generated by ATRO-FOLD first principles framework</td>
<td>133</td>
</tr>
<tr>
<td>5.2</td>
<td>Results for selected targets from the CASP8 and CASP9 experiments</td>
<td>134</td>
</tr>
<tr>
<td>5.3</td>
<td>Table showing enrichment factor for different RMSD regions over number of stages</td>
<td>136</td>
</tr>
<tr>
<td>6.1</td>
<td>Structure evaluations of CASP9 targets for ASTRO-FOLD 2.0.</td>
<td>156</td>
</tr>
<tr>
<td>6.2</td>
<td>Predicted secondary structure information for T581 by ASTRO-FOLD 2.0.</td>
<td>160</td>
</tr>
<tr>
<td>6.3</td>
<td>Predicted contact results for T581</td>
<td>161</td>
</tr>
<tr>
<td>6.4</td>
<td>T602: Predicted Contacts</td>
<td>165</td>
</tr>
<tr>
<td>6.5</td>
<td>Secondary structure prediction results for T562</td>
<td>166</td>
</tr>
</tbody>
</table>
6.6 Predicted Contact results for T562 ........................................... 167
6.7 Predicted Results for Protein T580 ........................................... 170
6.8 Secondary structure prediction results for T596 ......................... 173
6.9 Contact Prediction Results for T596 ........................................... 174
List of Figures

1.1 The general structure of an amino acid .............................................. 1
1.2 Backbone torsion angles in amino acids ............................................. 3
1.3 Naturally occuring amino acids ......................................................... 5
1.4 α-helices and β-strands ................................................................. 6

2.1 Patterns of hydrogen bonding observed in proteins ......................... 18
2.2 Specific classification of nonapeptide segments in proteins .......... 20
2.3 Example of Helix Beginning Propensity Evaluation ....................... 21
2.4 Hydrogen bonding pairs in Nonapeptides .................................... 22
2.5 Helix Propensity Thresholds for individual amino acids .......... 35
2.6 Helix Pair Propensity Coefficient Values .................................... 37
2.7 Distribution of Protein Lengths observed in PDBSelect25 data set ........ 40
2.8 Distribution of Q3 and SOV values for proteins in PDBSelect25 data set .... 41
2.9 Distribution of Q3 and SOV values for proteins from the CASP experiments .... 43

3.1 Parallel and Antiparallel β-sheets ................................................ 48
3.2 Complete flowsheet of the β-sheet topology prediction algorithm .... 54
3.3 Distribution of proteins in the PDBSelect25 data set versus the number of strands ... 70
3.4 Variation of average precision, recall and mcc over the number of solutions generated .... 72
3.5 PDBSelect25 Data set results, classified by number of strands .... 73
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.6</td>
<td>PDBSelect25 Data set results classified by class of proteins</td>
<td>75</td>
</tr>
<tr>
<td>3.7</td>
<td>Accuracy Results for proteins in the blind target test set from CASP9</td>
<td>78</td>
</tr>
<tr>
<td>4.1</td>
<td>Illustrative example of variation in distribution of Loop Residue Angles</td>
<td>88</td>
</tr>
<tr>
<td>4.2</td>
<td>Distribution of number and type of loops in the Loop Test Set</td>
<td>99</td>
</tr>
<tr>
<td>4.3</td>
<td>Average best structure RMSD distribution over loop length</td>
<td>100</td>
</tr>
<tr>
<td>4.4</td>
<td>Average best structure RMSD distribution over loop length</td>
<td>102</td>
</tr>
<tr>
<td>4.5</td>
<td>Average Accuracy of Loop Bounding Box for selected CASP9 targets</td>
<td>104</td>
</tr>
<tr>
<td>4.6</td>
<td>Variation of bounding box accuracy with Secondary structure accuracy</td>
<td>105</td>
</tr>
<tr>
<td>4.7</td>
<td>Variation of bounding box accuracy with J-Score</td>
<td>106</td>
</tr>
<tr>
<td>5.1</td>
<td>Flowsheet representing the ICON Algorithm</td>
<td>113</td>
</tr>
<tr>
<td>5.2</td>
<td>Cluster boundary determination using Expectation-Maximization</td>
<td>119</td>
</tr>
<tr>
<td>5.3</td>
<td>Cluster boundary determination using Integer Optimization</td>
<td>120</td>
</tr>
<tr>
<td>5.4</td>
<td>Overall results of ICON algorithm on individual test sets</td>
<td>127</td>
</tr>
<tr>
<td>5.5</td>
<td>Ranks of selected structures of ICON in High resolution data set</td>
<td>129</td>
</tr>
<tr>
<td>5.6</td>
<td>Difference in RMSD values of structures selected by ICON and SPICKER</td>
<td>129</td>
</tr>
<tr>
<td>5.7</td>
<td>Percentiles of structures by ICON in the high resolution data set</td>
<td>130</td>
</tr>
<tr>
<td>5.8</td>
<td>Performance of ICON algorithm on the Decoys ‘R’ Us dataset</td>
<td>132</td>
</tr>
<tr>
<td>5.9</td>
<td>Stage-wise enrichment in ICON</td>
<td>135</td>
</tr>
<tr>
<td>6.1</td>
<td>Flowsheet representing the ASTRO-FOLD 2.0 Approach</td>
<td>143</td>
</tr>
<tr>
<td>6.2</td>
<td>Native structure of T581 (3NPDA)</td>
<td>159</td>
</tr>
<tr>
<td>6.3</td>
<td>Structure overlay for T581</td>
<td>163</td>
</tr>
<tr>
<td>6.4</td>
<td>Structure overlay for T602</td>
<td>164</td>
</tr>
<tr>
<td>6.5</td>
<td>Structure overlay for T562</td>
<td>166</td>
</tr>
<tr>
<td>6.6</td>
<td>Structure overlay for T562 segment</td>
<td>169</td>
</tr>
</tbody>
</table>
6.7  Structure overlay for T580  ......................................................... 170
6.8  Structure overlay for T596  ......................................................... 173
List of Publications


Book Chapters


Conference Presentations

2. Subramani, A., Wei, Y. and C.A. Floudas, “A first principles based structure prediction algorithm for $\beta$ and mixed $\alpha/\beta$ proteins”, to be presented at AIChE Annual Meeting, 2011, Minneapolis MN.


Chapter 1

Introduction

Proteins are among the most essential molecules for life. They are instrumental in most biological activities, including transportation, growth, development and reproduction. Having such a wide range of functions, they are structurally one of the most complicated and elaborate macromolecules.

A protein is typically a linear chain of monomeric units, the amino acids. There are twenty naturally occurring amino acids. Each amino acid is made up of a central carbon atom, which is tetrahedrally bonded to an amine group, a carboxyl group, a hydrogen atom and a side chain, as shown in Figure 1.1.

Figure 1.1: The general structure of an amino acid
Amino acids link with each other through a peptide bond, which is formed through a dehydration reaction between the amino group of one amino acid, and the carboxyl group of the amino acid next in sequence. Because of the net loss of a water molecule from each amino acid due to the formation of two peptide bonds with immediate neighbors on either side, the terms “residue” is often used interchangeably with amino acid. Each amino acid is encoded in the DNA sequence using a three base codon. Through the translation process, the messenger RNA which complements a DNA strand generates the correct amino acid sequence of a protein.

The geometric parameters of a peptide unit consist of chemical bond lengths, chemical bond angles and torsion angles. The torsion angle is defined as the angle between the planes of two chemical bonds, which are separated along the sequence by one chemical bond. In other words, by taking a sequence of four contiguous atoms along the backbone of a protein, and by looking down the chemical bond formed by the two central atoms, the torsion angle would be the angle observed between the other two chemical bonds. Bond lengths are typically found very close to the ideal value, as the restoring forces are quite large. These restoring forces ensure that the time scale associated with the fluctuation is short, and the range of variation is small. Bond angles allow for a little more flexibility, but are still found in very dense clusters around the minimum energy values.

Hence, if the bond lengths and bond angles are assumed to be fixed at their optimum values, the entire structure of a protein can be defined solely on the basis of the torsion angles, including all the backbone torsion angles and any side chain torsion angles of the amino acid. Backbone dihedral angles are common across all the naturally occurring amino acids, as the source of differentiation between amino acids is the side chain attached to the backbone. The three backbone torsion angles of any amino acid, commonly referred to by the greek alphabets $\phi$, $\psi$ and $\omega$ are shown in Figure 1.2. As shown, the $\phi$, $\psi$ and $\omega$ dihedral angles are formed using the bonds $N_i - C_{\alpha,i}$, $C_{\alpha,i} - C_i$ and $C_i - N_{i+1}$ as the reference bonds, respectively. Given the partial double bond nature of the peptide
bond \( (C_i - N_{i+1}) \), the \( \omega \) dihedral angle is almost always observed to be within 10° of 180°, and is hence assumed to be fixed for structure prediction purposes.

![Torsion Angle Definitions](image.png)

**Figure 1.2:** An example of the torsion angle definitions for the backbone of a protein, created using the RasMol [177]

Based on observed structure, proteins are typically classified as globular, membrane or fibrillar proteins. Globular proteins are normally soluble in polar solvents, are usually found in structures which ensure that hydrophobic parts of the protein chain are buried in the interior of the protein. Being the easiest to examine experimentally, globular proteins have the largest representation in the Protein Data Bank (PDB) [20]. Membrane proteins are found within the membrane lipid-bilayer of cells. Given that they lie in a hydrophobic environment, their structures do not necessarily require the burial of hydrophobic regions. Membrane proteins typically act as receptors and channels which permit small molecules and ions to transfer across the cell membrane. Fibrillar proteins are typically found as aggregates in nature, wherein peptides of same or similar sequences agglomerate and result in an insoluble aggregate. In this thesis, we focus on the structure prediction techniques applied to the most abundant category of proteins, that is, the globular proteins.
As mentioned previously, there are twenty naturally occurring amino acids, shown in Figure 1.3. In terms of polarity, the set spans a wide range of properties, ranging from hydrophobic alliphatic and aromatic side chains, to long, linear charged side chains. As one would expect, hydrophobic amino acids typically try to shield their side chains from polar solvents, while hydrophilic and charged amino acids are found to be much more exposed to solvent. Amino acids Glutamine, Serine, Threonine and Asparagine are classified as hydrophilic. Amino acids Leucine, Isoleucine, Alanine, Glycine, Proline, Methionine, Phenylalanine, Valine, Cysteine, Tyrosine and Tryptophan are categorized as hydrophobic. Finally, Aspartic acid and Glutamic acid are negatively charged amino acids, while Lysine, Arginine and Histidine are positively charged amino acids. The structures of Glycine and Proline allow them to have very specific unique behaviors, making them very vital for protein folding and for protein structure prediction algorithms. Since the side chain of Glycine is just a hydrogen atom, this makes it the only achiral amino acid. The presence of only a hydrogen atom on the side chain also allows a much larger degree of conformational flexibility than any other amino acid. In contrast, the bond between the side chain atoms and the backbone for Proline restricts the conformational flexibility of the amino acid much more than all other amino acids. As a result of the bond, the $\phi$ torsion angle ends up being almost fixed at a value of around $-70^\circ$. In addition to these special cases, the side chains of Cysteine and, to a lesser extent, Methionine permit the formation of disulfide bonds between sulfur atoms present in reactive thiol groups. Weber and Miller [211] have presented a list of factors involved in the selection of the 20 amino acids based on a darwinian evolution premise. Selection parameters based on availability in the primitive ocean, function and stability in peptides, stability to racemization and stability of the corresponding transfer RNA have been included. Based on their analysis, they conclude that if organic life were to be present on another planet, about 75% of the amino acids would be the same as on earth.

The understanding of the structure of a protein can be broken down into four stages. The primary structure of a protein is its amino acid sequence, without any information about how
Figure 1.3: All naturally observed amino acids. Nitrogen: Blue, Oxygen: Red, Hydrogen: Gray, Carbon: Green, Sulfur: Orange
the residues interact in space. The primary sequence is usually defined by starting at the amino-terminal, and ending at the carboxy-terminal of the linear chain. The secondary structure of a protein is defined by patterns of local and non-local hydrogen bonding. The most common types of observed secondary structures are \( \alpha \)-helices and \( \beta \)-strands. \( \alpha \)-helices are defined by a repeated pattern of local hydrogen bonds, where the carbonyl oxygen of any amino acid \( i \) forms a hydrogen bond with the amine group of amino acid \( i+4 \). This is made possible by the fact that an ideal helical turn involves about 3.6 amino acids, and thus this hydrogen bond network runs approximately parallel to the axis of the helix. The \( \beta \)-strand is an extended conformation which is not stable by itself. The strand is stabilized by hydrogen bonds formed with other \( \beta \)-strands. Hence, a minimum of two \( \beta \)-strands are required in a protein complex with any \( \beta \) configuration. Further, the connection between two \( \beta \)-strands is classified as antiparallel if the primary sequence is increasing along the two strands in opposite directions, or parallel, if the primary sequence is incrementing in the same direction along both strands. A typical example of hydrogen bonds in helices and strands is shown in Figure 1.4.

![Example of hydrogen bonding in \( \alpha \)-helices](image1.png)

![Example of hydrogen bonding between \( \beta \)-strands](image2.png)

Figure 1.4: Commonly observed secondary structure elements and their hydrogen bonding patterns

Secondary structure elements come together in space to form the tertiary structure of a protein. The tertiary structure of a protein is governed by factors such as interaction of secondary structure...
backbone elements, salt bridge formation, ion-pair interactions, side chain packing and solvent interactions. The final stage of protein structure, known as quaternary structure, presents the interactions between different chains of a protein. These chains interact with each other to form larger aggregates or complexes, which function together as a single unit. The methods developed in this thesis are targeted towards the prediction of the tertiary structure of globular proteins.

Nuclear Magnetic Resonance (NMR) and X-Ray Crystallography are the two most common methods of determining protein structure experimentally. In X-Ray crystallography, an isolated and crystallized protein molecule is irradiated with X-Rays, and the diffraction patterns are collected to develop an electron density map. Given the amino acid sequence of the protein, all atoms of the protein are fit to the electron density map. Recently, NMR has found an increased usage among experimentalists looking to elucidate structures of proteins. The NMR method requires that the protein be soluble in an aqueous medium, which is not always necessary. Based on the type of experiment, a large set of constraints on atom-atom distances between many pairs of atoms in the protein are generated using the experimental results. Finally, modeling techniques like molecular dynamics are used to find the set of best-fit structures to the set of constraints provided.

Most algorithms which target the prediction of tertiary structures of proteins rely on the commonly accepted belief that there is a direct relationship between the amino acid sequence and the final tertiary structure of a protein. Given the complex structures and interactions in proteins observed in nature, the protein structure prediction problem is widely regarded as one of the holy grail problems in computational chemistry, molecular and structural biology. As the final folded structure of a protein is directly linked to its function, it becomes very important for the prediction of function in new proteins as well. Apart from the satisfaction of scientific curiosity, the ability to predict a protein’s final tertiary structure would help in the cure of diseases which impact the body due to its inability to produce a particular protein. By knowing the protein’s final structure, artificial proteins can be created which can act as drug delivery devices. Knowing the final structure of the protein can also help in drug design, creation of sensory proteins and industrial
catalysts. Along with this, a number of other fields in biological research, which depend on the relationship between a protein’s structure and function would benefit from this knowledge [29]. The ultimate goal of protein modeling is to predict a structure from its sequence with an accuracy and consistency that is comparable to the best results achieved experimentally. This would allow safe usage of \textit{in silico} protein models in all contexts where currently only experimental structures provide a solid basis: structure-based drug design, analysis of protein function, interactions, antigenic behavior and rational design of proteins with increased stability or novel functions.

One of the basic tenets of all structure prediction algorithms is Anfinsen’s hypothesis [8, 9]. The hypothesis states that in nature, a protein would be found in a structure or a structural ensemble which minimizes the global free energy of the system. This, however, leads to the two main challenges associated with all structure prediction algorithms, irrespective of their dependence on the PDB [20]. The first challenge is associated with the energy function which defines the structural landscape of the protein. Since the protein is expected to lie at the free energy minimum of the system, the force field would be required to ensure that minimum energy conformations correspond to the native protein structure, across a wide variety of protein topologies. A large body of work is available in literature, which tackle this challenging problem [25, 65, 136]. In addition to the force field, the structural landscape of the protein provides a significant challenge. A famous articulation of this problem is the Levinthal’s paradox, which raises the question of the impossibility of searching all possible conformations before reaching the global minimum structure, given that protein folding occurs in the timescale of a few milliseconds or microseconds [113, 234]. A number of approaches such as molecular dynamics, advanced monte carlo methods, genetic algorithms, simulated annealing and global optimization have been used to address this challenge [46, 60, 62, 226].

This thesis premises itself on the hierarchical theory of protein folding, which states that the secondary structure is formed much faster than the tertiary structure of the protein [18, 19]. Based on this theory, once the secondary structure is determined, these come together in space, thus
eliminating the contact of hydrophobic regions with the solvent and forming the tertiary structure of the protein. Hence, we address the problem of protein structure prediction in a similar manner. Firstly, prediction of the regions of $\alpha$-helices would be presented in Chapter 2. Following this, the prediction of the arrangement of isolated $\beta$-strands to form a network, commonly known as $\beta$-sheet topology is presented in Chapter 3. The prediction of loop regions is vital to reduce the conformational search space, and a method aiming to provide tight dihedral angle bounds on amino acids in the intermediate coil region is presented in Chapter 4. All of these steps are used to generate constraints which would be used to determine the final three dimensional structure of a protein. On the other end of the spectrum, most algorithms produce a large ensemble of possible structures for a target protein. Without the knowledge of the native, it is not possible to consistently identify the structure nearest to the native. To address this issue, Chapter 5 presents a novel traveling salesman based algorithm to identify a small subset of structures nearest to the native, from a given ensemble. Constraints generated from each of the aforementioned steps are introduced into the tertiary structure prediction algorithm, presented in Chapter 6.

In Chapter 2, a novel integer optimization framework has been presented for the prediction of $\alpha$-helical regions in proteins. Using a large, non-redundant training set, and by dividing each protein into overlapping oligopeptides, the propensity for any amino acid to be in an $\alpha$-helix is evaluated. For any new target protein, assignment of helical regions is made by comparing the evaluated propensity to threshold cutoff propensities, which were derived using the large database. A number of constraints are implemented to yield biologically meaningful results. Further, a chemical shift database is used to eliminate regions of the sequence which are unlikely to have $\alpha$-helices, thus reducing the search space for the optimization algorithm.

Using prediction of $\beta$-strand regions by a variety of methods, the next stage is to predict how these strands arrange themselves in space, which is presented in Chapter 3. A novel integer optimization algorithm aims to maximize a pseudo-contact potential between all pairs of strands, to determine the best arrangement of the strands. A wide variety of constraints have been introduced,
ranging from sequential and geometric to structural restrictions, which limit the possible arrangements of the strands. Further, a new torsion angle dynamics and clustering based algorithm has been developed, which re-ranks the predicted β-sheet topologies to ensure that the most accurate prediction lies within the topmost set of topology predictions.

For all regions of the sequence that are not part of the secondary structures, the structure prediction algorithm is classified under Loop Structure Prediction, and is presented in Chapter 4. Here, a large database of observed loop angles for all amino acids is used to generate initial structures for an ensemble of predicted loop structures. After steric clash removal, the structures are subject to nonlinear local optimization to get local minimum structures. These structures are iteratively used to derive new bounds on the dihedral angles, which are used to narrow down the set of possible structures to the ones closest to the native structure. The result from this stage of the algorithm is used to derive lower and upper bounds on the dihedral angles in the loop regions.

For a large ensemble of predicted structures, it is not possible to consistently identify the structures nearest to native, without the knowledge of the native structure itself. In Chapter 5, a novel iterative Traveling Salesman problem (TSP) based clustering algorithm is presented, which aims to identify a small subset of near-native structures from a predicted ensemble. Once the optimal path through each of the structures, represented as nodes on a TSP path, is determined, an integer optimization based algorithm identifies the densest clusters of predicted structures.

Constraints developed from each of the previous stages are incorporated into the tertiary structure prediction algorithm. The tertiary structure prediction algorithm combines deterministic global optimization, stochastic conformational space annealing and torsion angle dynamics in a hybrid approach to generate an ensemble of predicted structures. The tertiary structure prediction algorithm has been applied to a number of targets in the recently concluded community-wide experiment for protein structure prediction (CASP9). The results of these applications have been presented and discussed in detail.
Chapter 2

Helical Prediction in Proteins

Secondary structure prediction aims at providing a hierarchical first step towards tertiary structure prediction. Most first principles-based methods aim to predict secondary or supersecondary structures as an initial step, before attempting to predict the final three dimensional structure of the protein. By predicting secondary structures like $\alpha$-helices and $\beta$-strand regions in proteins, it is possible to apply distance and torsion angle restrictions on individual residues and on pairs of residues belonging to a certain secondary structure. In particular, it is instructive to note that an $\alpha$-helix has approximately 3.6 residues per turn, and each residue pair at positions $(i, i + 4)$ is found at approximately 6 Å from each other. The Ramachandran plot \[161\] is used to place angle constraints on amino acids, depending on the secondary structure they are found in.

2.1 Background

Secondary structure prediction has grown over three decades to its current state. In the 1960s and 1970s, methods based on single amino acid propensities were used. These methods analyzed the propensities for individual residues to act as helix breakers, helix supporters or being neutral to helix formations, based on their frequency of occurrence in and around $\alpha$-helices. The second
generation methods, which were popular till the early 1990s, evaluated similar propensities for segments of adjacent residues, which were varying in size. The newer methods in secondary structure prediction use evolutionary information to try and improve the prediction accuracy. Based on the idea that structure is more conserved than sequence, it is believed that mutations which occur in proteins are not neutral or random. Position specific profiles describing the residues which can be exchanged for each other at specific positions are commonly incorporated. This is done by using automatic database search methods like PSI-BLAST [6] and Hidden Markov Models. PSI-BLAST in particular is extremely useful in identifying distant relationships between proteins, using an iterative procedure. Detailed reviews of Protein Secondary structure prediction are available in literature [141, 167]. A review on various methods which use multiple sequence alignment has also been presented, which shows the strengths and weaknesses of the most commonly used multiple sequence alignment packages [146].

Secondary structure prediction based on residue nature, inter-residue potentials and knowledge based potentials have been a subject of great interest for a long time. A novel knowledge-based potential was presented, which demonstrated that a statistical approach can produce a successfully transferable all-atom potential. The potential combined a realistic all-atom representation with a contact potential and hydrogen-bonding function, and propagates through MC dynamics [83]. It was concluded that for a protein to achieve a realistic tertiary structure, it was necessary to include a geometrically and spatially realistic side chain and backbone representation, an accurate representation of H-bonding, and a potential that represents specific hydrophobic and other atom-atom interaction in a physically appropriate manner. In a continuing work, it was shown that the protein fold universe arises from compact conformations of hydrogen-bonded, secondary structures [227]. This was done by creating two models, a detailed atomistic model and a reduced model, with a minimal potential consisting of Hydrogen bonding, excluded volume and a uniform pairwise attraction potential between side chains. The conformational search schemes were based on replica exchange Monte Carlo sampling.
Based on the above analyses, the viewpoints on explanation of protein folding mechanisms have begun to take a form which is an intermediate of the traditional beliefs. The classical viewpoint advocates that folding is a hierarchical process, with secondary structure elements forming quickly, followed by a much slower arrangement into a three dimensional fold. The opposing perspective suggests the idea that a hydrophobic collapse results in the formation of secondary and tertiary features concurrently. An emerging viewpoint combines the above, professing that a combination of local interactions (which form local secondary structures) and long range hydrophobic contacts (which form $\beta$-sheets and their topologies) drive the protein to its folded structure. In analyzing secondary structure, the role of these local forces comes under scrutiny. Local forces which are believed to drive the formation of $\alpha$-helices are primarily hydrogen bonding, with some support from alternating hydrophobic and polar interactions due to the amphiphilic nature of the $\alpha$-helix.

### 2.1.1 Residue based approaches

As was mentioned previously, some of the earliest research on predicting helical regions in proteins were based on identifying individual residue propensities in helices. This was followed by the analysis of pairs and groups of residues separated from each other by short segments of sequences. A significant work in this direction looked to model short range interactions between residues separated by a maximum of six residues [143]. This was done by creating four prediction functions using a linear combination of statistical quantities representative of the separations. A parameter estimation model was created to maximize the number of correct assignments. Recently, a number of efforts have been directed to incorporate the much larger database information from the PDB into such models. Cochran et al. present an analysis of residue preferences at N1 and N2 positions of $\alpha$-helices [35] [36]. They measure the free energy change caused by the replacement of the N1 and N2 residues in a short polypeptide. Using the modified Lifson-Roig helix-coil theory [200],
and the measurement of helix content using circular dichroism (CD) spectroscopy, they provide a rank ordered list for free energies of the various residues, including the acidic and basic residues in their charged and uncharged forms. In continuation to the specificity of the N1 residues, Engel et al. presented statistics over the PDB Select 2002 database, which showed the position dependent propensities up to a length of 15 residues from either terminus of the helix [53]. They showed that helix lengths have a high propensity to be a multiple of the number of turns (or an approximate multiple of 3.6 residues). Further it was shown statistically that the N1 side of the helix tends to be hydrophobic in nature, while the other side is polar in nature. Hydrophobic residues were seen to have a preference for the N4, N7, N11-N12 and N15 positions, while polar residues preferred N6, N9-N10 and N13 positions. Specific pairs of residues have been seen to have a higher propensity to be at the N and C terminals of helices [71]. Here, the relative entropy of an amino acid at a helical position was calculated based on its propensity to be at that position in helices. Based on the values of these entropies, it was shown that the residue before the beginning of a helix (called the NCap position [15]) is the most selective in the residues which can be there, followed by the fourth residue from the N-terminal. The C-terminal was shown to be comparatively much less selective. It has also been hypothesised, to a considerable extent of success that a fold could be determined by its binary sequence of hydrophobic and polar amino acids [93]. In this instance, a 4-helix bundle was designed via a degenerate library of binary sequence genes expressed in Escherichia Coli and a number of correctly folded 4-helix bundles were found in the resulting systems.

### 2.1.2 Hydrogen Bonding

A point of debate for a long time is the role and importance of hydrogen bonding in protein folding, and hence its relative importance in protein structure prediction. There are rival theories which either claim that the formation of secondary structures through local interactions precedes the three dimensional structure folding, or claim the presence of hydrophobic collapse of the protein, with
the formation of the hydrophobic core driving the folding of the protein. It is widely accepted that the internal hydrogen bonding in secondary structures of proteins compensate for the hydrogen bonds which might have formed with the solvent in an unfolded state. Hydrogen bonding can be accepted to be a stabilizer, which means that although it may not help in reducing the overall potential energy \(^{[95]}\) of the system when seen in helices, it ensures that the energy of the folded protein is not higher than the unfolded state because of buried polar residues.

A number of surveys have been done to analyze the hydrogen bonding patterns in proteins. Rose and co-workers have presented a complete analysis of the various types of hydrogen bonds present in proteins, with conclusions on the nature and range of hydrogen bonds \(^{[196]}\). It was shown that most hydrogen bonds are local in nature, further buttressing the theory that local hydrogen bonds contribute to the stability of secondary structures. Further, most hydrogen bonds are between backbone residues and most among them between residues at positions \((i, i+4)\) and \((i, i+3)\). Based on the local nature of most hydrogen bonds, it was concluded that the absence of local hydrogen bonding would support denaturation because of the possibility of hydrogen bonding with the solvent. It was postulated that local hydrogen bonds are formed between nearby partners at an early stage in folding.

Hydrogen bonds are particularly repetitive and sequential in \(\alpha\)-helices. A number of efforts have been made to use and analyse hydrogen bonding patterns in helices. Frishman and Argos have presented a method to predict secondary structures in proteins using non-local information in the form of hydrogen bonding patterns. Hydrogen bonding patterns were elucidated from all secondary structure types, by evaluating the frequency of occurrence of every residue pair in each of the secondary structures \(^{[66]}\). This was used in conjunction to the primary sequence around an amino acid to determine its secondary structure, by evaluating thresholds for each residue to lie in a secondary structure.

Hydrogen bonding patterns are observed to be residue dependent in both \(\alpha\)-helices and \(\beta\)-sheets. The propensity of certain residue combinations is observed to be much higher than others.
This is because of the suitability of their structure and nature to a particular secondary structure. In \(\alpha\)-helices, side chain hydrogen bonding is mostly observed at the beginnings and ends of helices \[15\]. This can be attributed to the fact that at the ends of the helix, the residue does not have a backbone hydrogen bond with either residues before it (in the case of N-terminus) or after it (in the case of C-terminus). Analysis has been carried out to observe hydrogen bonds involving one or both side chains. Argos and co-worker presented the significance of side chain to main chain hydrogen bonds as formers and stabilizers of a protein fold, by analyzing the side chain to main chain hydrogen bonds formed in both helices and sheets \[24\]. Based on a statistical analysis of a non-redundant data set, it was observed that certain residues prefer to form short range side chain hydrogen bonds in some secondary structures, while their preference is reversed in other secondary structures. It was further shown that helices require a strong stop or start signal at their N-terminus for initiation. At the same time, the C-terminus showed no observable patterns. Most side chain hydrogen bonds in helices were seen to involve charged residues. In a further study, specific focus was given to intrahelical sidechain-sidechain hydrogen bonding contacts \[209\]. It was observed that most sidechain-sidechain contacts were between residues with a sequence spacing \((i, i + 4)\), which can be attributed to the absence of \(g^-\) conformation in helices for the dihedral angle \(\chi_1\). It was further shown that most \((i, i + 3)\) side chain contacts were between polar residues, while most \((i, i + 4)\) contacts were between hydrophobic ones. Specific analysis has been carried out recently on the stabilization effects of side-chain side-chain ionic interactions \[132\]. Ionic side chains tend to form a hydrogen bond frequently, locally or long range, known as a salt-bridge. In this study, it was observed that there was a correlation between the frequency of residue configurations able to form ionic interactions, with their probability of actually forming the interactions. It was further observed that some configurations have a high probability of forming an ionic interaction, and are frequently seen to form one. These are believed to be the most stabilizing salt-bridges. Such residue specific behaviors can be used to model the beginning and ends of helices. Aurora et al. \[16\] provide rules on the termination of \(\alpha\)-helices by Glycine. Based on a statistical analysis, cou-
pled with geometrical reasoning for the termination of the helix based on the position of Glycine relative to other residues on its side of the helix turn, simple rules on the nature of nearby residues of glycine were presented. Constraints using such restrictions can be used to develop methods to predict helical regions. Based on this, a method has been presented in this thesis, which uses hydrogen bonding propensities of residue pairs to predict helical regions in proteins, based on a training model using a non-redundant data set.

An interesting idea was presented [81], which showed that the native-state folds of proteins can emerge on the basis of considerations of geometry and symmetry in the protein. It was shown that a simple model that encapsulates the general attributes common to all polypeptide chains, such as steric hindrances, hydrogen bonding and hydrophobicity was sufficient to give rise to a free energy landscape for globular proteins. The identical nature of the protein was incorporated by representing each amino by just its $C_\alpha$ atom, lying along the axis of a self-avoiding flexible tube. Hydrogen bonding and hydrophobicity are modeled using simple geometrical rules, based on local coordinate systems origining at the $C_\alpha$ atoms. It was shown that a limited number of folds emerge by the application of these restrictions. In the model presented in this chapter, we extend this idea to the secondary structure of a protein. The thesis put forward here is that hydrogen bonding and hydrophobicity based considerations are sufficient to predict the $\alpha$-helical locations in a protein.

2.2 Database Analysis

As was mentioned in the previous section, the primary signature of the $\alpha$-helix is the presence of backbone hydrogen bonds between amino acids separated by three or four along the sequence. In order to validate this, Figure 2.1 shows the distribution of hydrogen bonds in helices as compared to the entire protein for a large, non-redundant protein data set. As demonstrated in the figure, the majority of the hydrogen bonds formed between amino acids separated by three or four residues lie in the helical region.
Since a helix is defined by this hydrogen bonding ladder, the propensity for any amino acid to be in an $\alpha$-helix would be mirrored in the ability of its sequential neighbors to form $i, i + 3$ and $i, i + 4$ backbone hydrogen bonds. Therefore, propensity for any pair of amino acids $a$ and $b$ separated by three or four amino acids to form a hydrogen bond in an $\alpha$ helix can be defined as:

$$P(a, b) = \frac{N_H(a, b)}{N(a, b)} \frac{\sum_{a,b} N_H(a, b)}{\sum_{a,b} N(a, b)}. \quad (2.1)$$

Here, $N_H(a, b)$ represents the frequency of occurrence of amino acid pair $a, b$ in $\alpha$-helices, separated by the specified distance (three or four amino acids). Similarly $N(a, b)$ represents the frequency of occurrence of the same pair of amino acids, with the same specified sequential distance,
in the entire protein. By dividing a given protein into overlapping nonapeptides, the propensity of
the central amino acid to be in an \( \alpha \)-helix is given by the linear combination of five \( i, i + 4 \) terms,
and four \( i, i + 3 \) terms, as shown in Figure 2.4.

2.2.1 Classification of Protein into regions

As a further classification of residues inside helices, we separate out the residues which belong
to the beginning and end of helices, from the ones in the middle of a helix and the ones which
lie at both the beginnings and ends of either the same or different helices. The thesis proposed is
that the cooperativity of hydrogen bonding brought about by adjacent residue pairs in the middle
of helices will be stronger than the cooperativity brought about by residue pairs found at the be-
ginnings and ends of helices. As an example, Proline appears at the beginning of helices, with no
hydrogen bonding to the fourth residue before it. However, its presence in the helix is supported
by its own forward hydrogen bond, along with other hydrogen bonds around it. Hence, unique
criteria for establishing the beginnings and ends of helices have been incorporated. This is shown
diagrammatically in Figure 2.2.

In Figure 2.2, the possible regions surrounding the central amino acid of a nonapeptide are
demonstrated. Each amino acid marked ‘H’ is present within a helix, while each amino acid
marked ‘N’ is present outside a helix. Part (a) of the figure shows that the central amino acid is
present within a helix, with each of its neighbors inside the same helix. Similarly, part (b) shows a
central residue with all its neighbors non-helical. Part (c) demonstrates a central amino acid at the
start of a helix, with all amino acids preceding it marked as non-helical. Part (d) demonstrates the
amino acids at the C-terminus of a helix, where all residues after the central amino acid are marked
as non-helical. Finally, part (e) demonstrates a central amino acid which lies between two helices,
and can therefore be classified to lie both in the beginning and end regions of helices.
2.2.2 Helix Beginnings and Helix Ends

Hydrophobicity and helix capping has been taken into account at the helix beginnings and ends using a unique set of expressions for helix ends. For the seven positions surrounding the helix beginnings and ends, as shown in Fig 2.3, propensities for the 20 amino acids to be in each of the positions is evaluated. These propensities are given as:

\[
P_{beg}(i; k) = \frac{N(i; k) / \sum_{i=1}^{20} N(i; k)}{N(i) / \sum_{i=1}^{20} N(i)}
\]

(2.2)

A similar equation is also created for the helix ends. Here, \(k\) is the position at the helix beginning or end, and \(i\) is the particular amino acid.
2.2.3 Objective function and Constraints

The objective function of the Training Model is the minimization of slacks:

\[
\text{MIN} \sum_{i,p} (S(i,p) + S_{\text{beg}}(i,p) + S_{\text{end}}(i,p))
\]

Here, \(S(i,p)\) are the slack variables corresponding to the helix existence equations, while \(S_{\text{beg}}(i,p)\) and \(S_{\text{end}}(i,p)\) are the slack variables corresponding to the beginning and end equations respectively. The set \(p\) refers to the set of proteins, while the set \(i\) is the residues in each protein.

**Propensity Equations**

Propensity equation 2.3 for each residue relates the propensity of the central residue of the nonapeptide (Fig 2.4) to residue pairs

\[
P(i,p) = \sum_{j=1}^{5} A(j) \cdot P_{\text{pair}}(i + j - 5, i + j - 1) + \sum_{j=6}^{9} A(j) \cdot P_{\text{pair}}(i + j - 9, i + j - 6) \quad (2.3)
\]

The first term in the equation refers to the \((i, i + 4)\) pairs, while the second term refers to the \((i, i + 3)\) pairs. The propensity equations for the beginning and the end regions of a helix, are given...
as:

\[ P_{beg}(i, p) = \sum_{k=1}^{7} \mu_{beg} P_{beg}(i + k - 4; k) \]  \hspace{1cm} (2.4)

\[ P_{end}(i, p) = \sum_{k=1}^{7} \mu_{end} P_{end}(i + k - 4; k) \]  \hspace{1cm} (2.5)

Figure 2.4: Hydrogen bonding pairs in any nonapeptide. The figure shows the five pairs of \((i, i+4)\) hydrogen bonds, as well as the four \((i, i+3)\) hydrogen bond pairs.

The classification of the entire protein into four regions is given by the four terms \(y_{begreg}(i, p)\), \(y_{endreg}(i, p)\), \(y_{both}(i, p)\) and \(y_{none}(i, p)\), representing (respectively), the beginning, end, both beginning and end; and neither beginning nor end. For the training model, these terms are parameters which become variables for the helix prediction model.

The variables for the Training model are:

- The 4 set of coefficients for the helix existence equation, representing the 4 different regions of the proteins: \(A_{beg}(j)\), \(A_{end}(j)\), \(A_{both}(j)\) and \(A_{none}(j)\).
- Threshold propensity cutoffs for the helix existence equation: \(T_{beg}(r)\), \(T_{end}(r)\), \(T_{none}(r)\) and \(T_{both}(r)\).
- Two coefficient sets for the helix beginnings and ends: \(\mu_{beg}(k)\), \(\mu_{end}(k)\)
- Two threshold values for these equations: \(beg\_T\), \(end\_T\)
Constraints

The sum of weights for terms in the expression defining helix beginnings and ends should sum to 1:

\[
\sum_{k} \mu_{\text{beg}}(k) = 1 \quad (2.6)
\]

\[
\sum_{k} \mu_{\text{end}}(k) = 1. \quad (2.7)
\]

Similarly, for the equations which define the propensity for an amino acid to be in a helix, the sum of the coefficients should sum to 1.

\[
\sum_{j=1}^{9} A(j) = 1 \quad (2.8)
\]

where

\[
A(j) = \begin{cases} 
A_{\text{beg}}(j) : & y_{\text{begreg}}(i,p) = 1 \\
A_{\text{end}}(j) : & y_{\text{endreg}}(i,p) = 1 \\
A_{\text{both}}(j) : & y_{\text{both}}(i,p) = 1 \\
A_{\text{none}}(j) : & y_{\text{none}}(i,p) = 1.
\end{cases} \quad (2.9)
\]

The propensities are evaluated such that they should lie above the threshold values if they are helical residues, and should be below it if they are non-helical. The threshold equations for the helical existence are:

\[
P(i,p) + S(i,p) \geq T(r) - M \times (1 - y_{\text{hel}}(i,p)) \quad (2.10)
\]

\[
P(i,p) - S(i,p) \leq T(r) + M \times y_{\text{hel}}(i,p) \quad (2.11)
\]
where

\[
T(r) = \begin{cases} 
T_{\text{beg}}(j) & : y_{\text{begreg}}(i,p) = 1 \\
T_{\text{end}}(j) & : y_{\text{endreg}}(i,p) = 1 \\
T_{\text{both}}(j) & : y_{\text{both}}(i,p) = 1 \\
T_{\text{none}}(j) & : y_{\text{none}}(i,p) = 1 
\end{cases}
\]  

(2.12)

### 2.3 Helix Prediction Model

The parameters for the helix prediction model are the set of variables of the Training model. The variables for the helix prediction model are:

- **Binary Variables** \( y_{\text{hel}}(i) \): 1 if the residue \( i \) is in a helix, 0 otherwise

- **0-1 continuous variables** \( y_{\text{beg}}(i) \) and \( y_{\text{end}}(i) \), which are 1 if the residue \( i \) is the beginning or end of a helix, respectively.

- **0-1 continuous variables** to represent the 4 regions of the protein: \( y_{\text{begreg}}(i) \), \( y_{\text{endreg}}(i) \), \( y_{\text{both}}(i) \) and \( y_{\text{none}}(i) \).

The Propensity and threshold equations for the helix beginnings and helix ends are similar to the ones presented in the Training Model. For the helix existence equations, we define 4 dummy variables \( P_1(i) \), \( P_2(i) \), \( P_3(i) \) and \( P_4(i) \) to represent the helix propensity for any amino acid in the four regions around a helix, as shown in Figure 2.2. The equations describing the propensity evaluations are as shown:
\[ P_1(i) = \sum_{j=1}^{5} A_{beg}(j) \cdot P(i + j - 5, i + j - 1) + \sum_{j=6}^{9} A_{beg}(j) \cdot P(i + j - 9, i + j - 6) \]

\[ P_2(i) = \sum_{j=1}^{5} A_{end}(j) \cdot P(i + j - 5, i + j - 1) + \sum_{j=6}^{9} A_{end}(j) \cdot P(i + j - 9, i + j - 6) \]

\[ P_3(i) = \sum_{j=1}^{5} A_{both}(j) \cdot P(i + j - 5, i + j - 1) + \sum_{j=6}^{9} A_{both}(j) \cdot P(i + j - 9, i + j - 6) \]

\[ P_4(i) = \sum_{j=1}^{5} A_{none}(j) \cdot P(i + j - 5, i + j - 1) + \sum_{j=6}^{9} A_{none}(j) \cdot P(i + j - 9, i + j - 6) \]

such that

\[ P(i) = P_1(i) \cdot y_{begreg}(i) + P_2(i) \cdot y_{endreg}(i) + P_3(i) \cdot y_{both}(i) + P_4(i) \cdot y_{none}(i) \quad (2.13) \]

\[ y_{begreg}(i) + y_{endreg}(i) + y_{both}(i) + y_{none}(i) = 1 \quad (2.14) \]

For any amino acid at position \( i \), the first equation above assigns its helical propensity to be the sum of the products of the propensity to be in a given region and the binary variable representing the same region. Further, any position \( i \) can only be in one of the four regions around a helix. This is presented mathematically in Equation 2.14.

### 2.3.1 Additional Constraints

A number of additional constraints have been added to the helix prediction model in order to obtain biologically meaningful helix predictions. These are presented below:

1. The number of helix beginnings should equal the number of helix ends in a protein

\[ \sum_i (y_{beg}(i) - y_{end}(i)) = 0 \]
2. A helix should be at least four amino acids long, and any two helices should be separated by at least one amino acid.

\[ \sum_{i'=i-1}^{i+2} y_{\text{end}}(i') \leq 1 - y_{\text{beg}}(i) \]

\[ \sum_{i'=i-2}^{i+1} y_{\text{beg}}(i') \leq 1 - y_{\text{end}}(i) \]

The first equation claims that if the variable \( y_{\text{beg}}(i) \), representing the beginning of a helix, is active at any residue position \( i \), then the variable \( y_{\text{end}}(i) \), representing the end of a helix, would be zero for the preceding position, and for all positions up to \( i + 2 \), thus ensuring a minimum length of four amino acids for any helix.

3. This constraint presents the definition of the \( y_{\text{hel}}(i) \) variables. At any amino acid position \( i \), if the difference between the total number of \( y_{\text{beg}}(i) \) and \( y_{\text{end}}(i) \) variables is 1, then the variable \( y_{\text{hel}}(i) \) is active (i.e, the residue lies inside a helix). By defining the mathematical equation as shown in Equation 2.15 it is also ensured that the difference between the total number of \( y_{\text{beg}}(i) \) and \( y_{\text{end}}(i) \) variables till any given residue position \( i \) can only be 0 or 1.

\[ \sum_{i'=1}^{i} (y_{\text{beg}}(i') - y_{\text{end}}(i' - 1)) = y_{\text{hel}}(i) \] (2.15)

4. This constraint is introduced as a tightening constraint, which relates the binary variable \( y_{\text{hel}}(i) \) at any residue position \( i \) to the binary variables \( y_{\text{hel}}(i - 1), y_{\text{beg}}(i) \) and \( y_{\text{end}}(i - 1) \). The constraint implements the idea that for any residue position \( i \), the binary variable \( y_{\text{hel}}(i) \) is active only if it is the beginning of a helix \( (y_{\text{beg}}(i) = 1) \), or that the previous amino acid is helical and not the end of a helix \( (y_{\text{hel}}(i - 1) = 1 \text{ and } y_{\text{end}}(i) = 0) \).

\[ y_{\text{hel}}(i) = y_{\text{hel}}(i - 1) + y_{\text{beg}}(i) - y_{\text{end}}(i - 1) \]
5. The next constraints relate the regions $y_{none}(i)$ and $y_{endreg}(i)$ to the variables $y_{beg}(i)$ and $y_{end}(i)$. The constraints introduce the definitions of these regions, by enforcing that the variables associated with these regions are fixed to zero for any region of a protein surrounding a position with active $y_{beg}(i)$ or $y_{end}(i)$ variables.

$$y_{none}(i), y_{endreg}(i) \leq 1 - y_{beg}(i') \forall i - 3 \leq i' \leq i + 4$$

$$y_{none}(i), y_{begreg}(i) \leq 1 - y_{end}(i') \forall i - 4 \leq i' \leq i + 3$$

6. In exactly the same manner, the two remaining regions represented by $y_{begreg}(i)$ and $y_{both}(i)$ are related to the variables representing the beginnings and ends of helices. The constraint enforces that if all the residues in a given sequential neighborhood do not include helix beginnings and ends, then the amino acids of this neighborhood cannot lie in the “beginning” and “both beginning and end” regions, as defined previously.

$$y_{begreg}(i), y_{both}(i) \leq \sum_{i' = i - 3}^{i + 4} y_{beg}(i')$$

$$y_{endreg}(i), y_{both}(i) \leq \sum_{i' = i - 4}^{i + 3} y_{beg}(i')$$

As described previously, the propensity for any amino acid inside a helix is required to be above the threshold value evaluated through the training model. However, in order to account for discrepancies in this assignment, we introduce slack variables into the equation. Slack variables ($S(i)$) are positive variables which compensate if the value of the propensity ($P(i)$) falls below the value of the threshold ($T(r)$). These variables are minimized in the objective function. In the equations shown below, the propensity of any residue $i$ is required to be more than the threshold
propensity in the region that the residue is found to be in, if the amino acid lies within a helix (represented by an active \( y_{hel}(i) \) variable).

\[
P(i) + S(i) \geq T_{beg}(r).y_{begreg}(i) + T_{end}(r).y_{endreg}(i) + T_{both}(r).y_{both}(i) \\
+ T_{none}(r).y_{none}(i) - M(1 - y_{hel}(i)) \quad (2.16)
\]

\[
P(i) - S(i) \leq T_{beg}(r).y_{begreg}(i) + T_{end}(r).y_{endreg}(i) \\
T_{both}(r).y_{both}(i) + T_{none}(r).y_{none}(i) + M \cdot y_{hel}(i) \quad (2.17)
\]

**Helical Content**

Constraints are used to predict the number of helical residues in the protein. The helical content of a protein is defined by the equation:

\[
Content = \frac{N_{hel}}{N_{res}} \quad (2.18)
\]

where \( N_{hel} \) are the number of helical amino acids in the protein, and \( N_{res} \) is the total number of amino acids in the protein. A number of machine learning based methods, which use sequence alignment information have been used to determine the secondary structure content of a protein \[171\]. Recently, a multiple linear regression model to estimate the helical and strand content of a protein, without the need of multiple sequence alignment, has been presented in literature \[80\]. Using a number of features ranging from average molecular weight of the amino acid sequence to component moment vectors of the individual residue types in the sequence, the authors develop a linear regression model of the form

\[
y_{\alpha} = a_0 + \sum_{i=1}^{n_{\alpha}} a_i f_i \quad (2.19)
\]
CHAPTER 2. **Helical Prediction in Proteins**

where \( y_\alpha \) denotes the predicted \( \alpha \)-helix content, \( f_i \) represents the feature selected and \( a_i \) represents the coefficient of the feature. Using input data from the original model (direct communication from authors), a linear regression training model for the evaluation of useful features \( f_i \) and their coefficients \( a_i \) was established.

The objective of the model is to determine the values of the coefficients \( a_i \). Mathematically, the model is expressed as:

\[
\min \sum_{j=1}^{n_p} (y_{\alpha,j} - a_0 - \sum_{i=1}^{n} a_i f_i)^2
\]

The resulting model provided 49 active features, including the constant term \( a_0 \). Using an error range of 10% around the predicted helical content value, the following constraints are imposed on the model:

\[
\sum_i y_{hel}(i) \geq 0.9 \times y_\alpha
\]

\[
\sum_i y_{hel}(i) \leq 1.1 \times y_\alpha
\]

**Helix Capping Constraints**

In addition to the above constraints, a set of additional constraints were applied to model the most frequently observed helical caps. At the N-terminus, it was observed that at least one between \( N_{Cap} \) and \( N4 \) residues have to be favorable at that position. An amino acid at a position in the N-terminus of a helix was considered favorable at that position if the amino acid has a propensity of greater than 1 to be in the position. This was applied by fixing the \( y_{beg}(i) \) at the unfavorable residue positions to 0.

At the C-terminus, the Schellman motif and \( \alpha \)-L motif was applied as constraints [16]. For the Schellman motif, the constraint requires that if Glycine occurs at C’ position, followed by a
hydrophobic residue at C\textsuperscript{\textalpha}, a hydrophobic residue at C3 inside the helix and a polar residue at C1, then any active helix must be terminated. For the \( \alpha \)-L motif, the constraint requires that if a Glycine at C\textprime position is followed by a polar residue, then it causes termination of the helix unless there is a favorable complementary residue in the final turn of the helix preceding it. Here, complementarity of amino acids is defined with respect to charge. Any pair of oppositely charged amino acids are considered complementary to each other.

### 2.4 Chemical Shift Data-based constraints

Chemical shifts of atomic nuclei in or close to a protein backbone have been recognized to be extremely sensitive to the local environment they find themselves in [180]. In particular, a strong correlation between the chemical shift of atomic nuclei and the secondary structure of the amino acid [201, 215]. Hence, the aim of these constraints is to identify supersets of \( \alpha \)-helical regions in a protein. This is tantamount to identifying specific amino acids in the protein sequence which can be assured to be absent from any \( \alpha \)-helices. Through this analysis, the binary variable \( y_{hel}(i) \) representing the presence of the residue \( i \) in a helix can be fixed to 0 for non-helical regions of the protein directly.

#### 2.4.1 Collection of Chemical Shift data

For the purposes of \( \alpha \)-helical prediction, chemical shift data is collected from the SPARTA database available through literature [180]. Further, specific atomic nuclei have been seen to correlate well with \( \alpha \)-helices, while other nuclei are seen to correlate with \( \beta \)-strands [201]. It has been observed that the chemical shift data of \( ^1\text{H}_\alpha, ^1\text{H}_\text{N},^\text{C}_\alpha \) and N are most correlated with \( \alpha \) helices in proteins. In order to collect the SPARTA database, each protein in the database has been divided into overlapping tri-peptides. For each overlapping tri-peptide of a target protein, the chemical shift data corresponding to all instances of its occurrence in the SPARTA database are collected. In addition,
we allow for single point substitutions in the tri-peptide up to a substitution score of 2, where the substitution score is similar to the one used to generate the TALOS algorithm \[180\]. At this stage, we have a matrix of dimension \(N \times 4\), where \(N\) is the number of entries from the database. The columns of this matrix correspond to the chemical shifts associated with each entry collected from the database. Two additional columns are added to this matrix, corresponding to the substitution score and one the secondary structure of the collected tripeptide, respectively. The substitution score represents the “weight” of each row of the matrix, represented by the row’s similarity to the target tripeptide. The column corresponding to secondary structure is assigned a value +1 for helices, -1 for strands and 0 for coils.

The traveling salesman problem based clustering algorithm ICON \[198\] is used to cluster the generated matrix for each tripeptide of the target sequence. The algorithm creates an optimal traveling salesman path through each of the rows of the matrix, represented as nodes on a traveling salesman path. This method uses an integer optimization algorithm to define cluster boundaries in the optimal traveling salesman path. Further details of the ICON algorithm are presented in Chapter \[5\]. At the end of the clustering algorithm, we collect the medoids of the top 5 clusters sorted by the size of the clusters. A consensus of the secondary structures of the 5 clusters is assigned to the central amino acid of the target tripeptide. The first and the last amino acids of the target sequence are always assigned as ‘Coil’, since they are almost always seen to lie outside any ordered secondary structure.

### 2.4.2 Filtering of Chemical Shift Cluster Results

Following the secondary structure assignment obtained from the chemical shift data, we apply filters to the results in order to identify the super-structure of helical regions for the target protein. It is important to note that for certain specific tri-peptides, especially ones involving Glycine and Proline, there could be no clustering results, since data for these amino acids is missing from
the chemical shift database. Hence, the assignments are brought down to 4 classes: H, E, C and N, where N refers to non-assigned residues. The following filters are then applied to the entire sequence in order to develop the super-structure.

1. Start with the residues assigned as N. The order of preference for neighbors is $H > E > N > C$. Hence, whichever side has a higher preference assignment, 2 residues on that side are converted to H. If neither side contains an H or E for 2 residues, then N is converted to C

2. All ‘H’ segments separated by 1 ‘E’ or ‘C’ residue are joined together

3. All ‘H’ segments greater than 2 in length, separated by 2 or 3 ‘E’ or ‘C’ residues in any combination, are joined together

4. All ‘H’ segments are extended for 4 residues on either side if they have atleast 2 residues assigned as ‘E’, and no residue assigned as ‘H’. The 2 ‘E’ residues represent the hydrophobic face of the helical wheel. If the extension of the helix results in the presence of a Proline at greater than 2 amino acids from either end of the helix, the change is avoided.

5. All helical regions are extended by 1 residue on either side.

Once these filters are applied, we go through all the helical regions of the protein, by ensuring that all $i, i+4$ or $i, i+3$ residues in the regions are of the same nature (hydrophobic/polar). This is to ensure the continuous nature of the helical wheel. If such a situation does not happen, we break the helix into two parts at the residue whose threshold $T_{both}(r)$ is the highest, thus ensuring that it is the residue which is most likely to be between two helices in the coil region.

All residues apart from the helically marked residues from the above algorithm are now fixed as “non-helical”, by fixing the binary variable $y_{hel}(i)$ to be 0.
2.5 Protein Data Sets, Cross Validation and Accuracy Evaluation

In order to carry out the training, to evaluate parameter values, 989 pure α-helical proteins from the October 2008 release of PDBSelect25 were used. The PDBSelect25 data set consists of single chain proteins with less than 25% pairwise sequence similarity. This final set was collected, after eliminating proteins with incomplete backbone information from the original data set.

A five fold cross validation process has been carried out, by dividing the protein set into five parts in an 80%-20% division, that is, 80% of the proteins would be used for training the parameters, and 20% would be used as the test set. For each test set, the parameters obtained by training on the remainder of the proteins are used for secondary structure prediction. However, in order to address the prediction of helical regions in blind targets, like the targets of the CASP experiments, the training was carried out over the entire set of 989 proteins. A comparison between the parameter values obtained in each of the training processes has been presented subsequently in the chapter.

The Test sets derived out of the CASP experiments provide the best test cases for the algorithm, since they are blind structure predictions in the true sense. However, in order to ensure that an independence is maintained between the training and test sets, the CASP9 proteins were checked for sequence similarity against the PDBSelect25 data set using the culling software PISCES [210]. Using cutoff thresholds of 25% and 30%, blind test sets were derived for an independent test of the helix prediction algorithm.

2.5.1 True Secondary Structure Assignment

The true secondary structure for any protein was determined using the dictionary of secondary structure of proteins, DSSP [92]. The default DSSP algorithm presents secondary structure infor-
mation by classifying it into eight categories: H (α helix), G (3_{10} helix), I (π-helix), E (β strand), B (isolated β-bridge), T (turn), S (bend) and the rest as random coil. Since this algorithm deals with the prediction of the location of α-helices, the predictions are reduced to two classes only: H (for helical regions) and the rest. Therefore, the DSSP output is reduced to a two state form by the following transformation: all H, G and I assignments are converted to helices (H), and the remainder are converted to coil (C).

### 2.5.2 Helix Prediction Accuracy

A very simple metric for predicting the accuracy of helical regions in proteins is the $Q_α$ score. It is represented by the mathematical equation:

$$Q_α = \frac{\sum_i Pred_i}{N} \times 100$$  \hspace{1cm} (2.23)

where $N$ is the number of amino acids in a given sequence, and $Pred_i$ is a binary parameter which is set to 1, if any amino acid $i$ is assigned correctly to the state it actually belongs to (H or C, see previous section).

In addition to the simplistic check of accuracy, a good measure for the accuracy of a secondary structure prediction algorithm is the segmental overlap evaluation, known as SOV. Simplistically, it provides a measure of overlap between the predicted and the actual segments of secondary structure for any prediction [225]. Mathematically, it is given by the formula:

$$SOV = 100 \times \frac{1}{N} \sum_{i \in \{H,C\}} \sum_{S_i} \left( \frac{\minov(S_1,S_2) + \delta(S_1,S_2)}{\maxov(S_1,S_2)} \right) \times \text{Len}(S_1)$$  \hspace{1cm} (2.24)

Here, $N$ is the total number of residues in the protein, $S_i$ is the set of all overlapping pairs of segments (s1,s2) in conformation state i, Len(S1) is the number of residues in segment S1, minov(S1,S2) is the length of the actual overlap and maxov(S1,S2) is the total extent of the segment.
\( \delta(S_1, S_2) \) is the minimum of \( \max_{\mathrm{ov}}(S_1, S_2) \), \( \min_{\mathrm{ov}}(S_1, S_2) \), \( \min_{\mathrm{ov}}(S_1, S_2) \), \( \text{int}[0.5 \times \text{len}(S_1)] \) and \( \text{int}[0.5 \times \text{len}(S_2)] \).

### 2.5.3 Results: Training Stage

This section presents the parameter values evaluated during the Training stage. As was described in the model previously, a number of parameters were trained in order to be used for blind test predictions. Among these, one of the most critical ones are the threshold propensity values for each individual amino acid. The threshold propensity values for each individual amino acid, in each region of a protein is presented in Figure 2.5.

![Figure 2.5: Helix Propensity Thresholds for individual amino acids](image)
As would be expected, the threshold values for Glycine and Proline to be in helices is much lower than all the other amino acids. Glycines and Prolines are very rarely observed to be inside helices, and are commonly termed as “helix breakers”. Hence, pair propensity calculations which evaluate their propensity to form hydrogen bonds (either as donors or acceptors) produce very low values. It should be noted that even though the threshold propensity values for glycine and proline are much lower than the rest of the amino acids, their significantly lower pair propensity values ensure that even these values are rarely surpassed. The more interesting fact to come out of this result is that significantly different threshold values are observed for each of the amino acids, thus invalidating the possibility of using a single threshold value across all amino acids. As would be discussed later, this is not the case with the helix end propensity calculations, where a single value has been used across all amino acids.

One of the pioneering articles studying the patterns of hydrogen bonding at helix ends has been carried out by Aurora and Rose [15]. In this article, the authors provide detailed analysis of patterns of hydrogen bonding observed at the ends of helices in proteins, and rationalize it based on energetic arguments. The study also provides an analysis of the most observed amino acids at the helix ends. In conjunction with the article of the text, the highest occurrence of amino acids at the N-terminus of a helix is seen for Aspartic Acid (D), Threonine (T), Glutamic Acid (E) and Proline (P). Since their occurrence at the N-terminus exceeds other amino acids, pair propensities involving each of these amino acids as the donor would be very high. Hence, the threshold values for these amino acids to be at a helix beginning should be higher than the threshold values at other positions relative to a helix. As can be seen by the black line in Figure 2.5 this is in fact the case. Similarly, the most observed amino acids at the C-termini of helices is Glycine (G). Hence, the helix end threshold propensity of glycine (shown with the red line) is seen to be higher than the other threshold values.

Figure 2.6 shows the relative weights of the individual terms in the helix propensity equation, Equation 2.3.
As can be seen from the figure, depending on the region of the protein under consideration, a different set of pair propensities seem to contribute the most to the helical propensity of the central amino acid of the nonapeptide. For instance, in the beginning regions of a helix, it is seen that the contributions of the hydrogen bonding in the first turn, that is, the hydrogen bonds between the amino acid pairs \( (i, i + 3) \) and \( (i, i + 4) \), where \( i \) is the central amino acid of the nonapeptide, is the most significant. This makes biological sense, since the stability of the first turn of the helix depends significantly on the backbone hydrogen bonds in the first turn. Further, some of the least contributing terms in the beginning of helices are hydrogen bonds between the initial residues and residues in the preceding turn, which is completely intuitive as well. Similarly, at the end of helix
Table 2.1: Helix Beginning and End Propensity Coefficients

<table>
<thead>
<tr>
<th>Helix Position</th>
<th>Helix Beginning Coefficient</th>
<th>Helix End Coefficient</th>
<th>Helix End Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N''$</td>
<td>0.116</td>
<td>0.13</td>
<td>$C_4$</td>
</tr>
<tr>
<td>$N'$</td>
<td>0.223</td>
<td>0.21</td>
<td>$C_3$</td>
</tr>
<tr>
<td>$N_{cap}$</td>
<td>0.082</td>
<td>0.333</td>
<td>$C_2$</td>
</tr>
<tr>
<td>$N_1$</td>
<td>0.078</td>
<td>0.18</td>
<td>$C_1$</td>
</tr>
<tr>
<td>$N_2$</td>
<td>0.142</td>
<td>0.086</td>
<td>$C_{cap}$</td>
</tr>
<tr>
<td>$N_3$</td>
<td>0.195</td>
<td>0.047</td>
<td>$C'$</td>
</tr>
<tr>
<td>$N_4$</td>
<td>0.164</td>
<td>0.012</td>
<td>$C''$</td>
</tr>
</tbody>
</table>

(shown in red), the contributions of the hydrogen bonds in the last turn of the helix (represented by the terms “1-5” and “2-5”) contribute the most, while all contributions after the end of the helix are negligible. In the central regions of the helix, the results suggest that the highest contribution comes from hydrogen bonds involving the central amino acid itself. Similarly, for regions which lie between two helices in close sequential proximity, it is suggested that the hydrogen bonding contributions of the central amino acid are most vital. This makes biological sense as well, as the role of the residue would appear to be vital when breaking a longer helix into two.

In addition to the coefficients involved in the helix propensity evaluation, coefficients for helix beginning and end propensities were also evaluated. These are shown in Table 2.1. In this table, the position $N''$ refers to the residue three positions prior to the start of the helix. Hence, $N_{cap}$ refers to the residue immediately preceding the helix. Similarly the $C_{cap}$ residue follows the end of the helix, while $C_1$ refers to the last residue in the helix.

As shown in Table 2.1, the most significant contributors to the helix beginning and end expressions are the amino acids inside the helix. Hence, at the N-terminus, the highest contributions come from the residues $N_2 - N_4$, while at the C-terminus, the highest contributions come from the residues at positions $C_4 - C_1$. Further, this is in complete agreement with the coefficient contributions of the pairwise terms observed in Figure 2.6 where it was seen that the highest contributions to helix propensity came from hydrogen bonds formed between amino acids which were within
the helix. At the N-terminus, a major contribution is seen by the term at the position \( N' \). This has been explained in the work of Aurora and Rose [15], where they explain the pervasive prevalence of a side chain to main chain hydrogen bond at the N-terminus of helices. This side chain to main chain hydrogen bond is seen to contribute significantly to the stability of the helix, as observed by its destabilization when the residue is mutated to one without a side chain hydrogen donor. Thus, as can be seen, the primary objective of introduction of the helix beginning and end propensities, which was to capture end stabilization effects of helices, is realized.

### 2.5.4 Results: PDBSelect25 Data set

Figure 2.7 shows the distribution of lengths of the \( \alpha \)-helical proteins in the PDBSelect25 data set. The average Q3 and SOV accuracy for the application of the helix prediction model on the PDBSelect25 data set was observed to be 82.2\% and 77.0\%, respectively. Most secondary structure prediction algorithms depend on profile information. This means that most successful algorithms check the similarity of a given target sequence to the entire PDB, and elucidate input parameters to machine learning algorithms based on the degree of similarity found. In spite of this, most algorithms have found themselves with an average accuracy of about 75\% to 85\%, still about 3\% short of the predicted theoretical limit [168]. In this work, the \( \alpha \)-helix prediction method is not based on deriving sequence profiles or position specific scoring matrices for the amino acids of a target sequence. Given that the algorithm aims to model the standard patterns of hydrogen bonding and hydrophobicity in \( \alpha \)-helices, it is likely to be more reliable for \( \alpha \)-helix prediction in proteins which have a low sequence similarity to the protein data bank.

Figure 2.8 shows the distribution of Q3 and SOV accuracies for all the proteins the PDBSelect25 data set. As can be expected, a fairly wide range of accuracy values are observed. However, it is seen that the Q3 values do not fall below 55\% for any of the proteins in this data set. Further, 163 proteins are seen to have a Q3 accuracy score above 90\%. A wider distribution is seen in
the SOV values. The segment overlap metric punishes any inconsistencies in the location of ends of secondary structure much more than the Q3 metric. Hence, if a long helix is predicted as two smaller helices with one amino acid in between, the SOV penalty would be much larger in this instance than the Q3 penalty.

2.5.5 Results: CASP8 and CASP9 data sets

The helix prediction method presented in this chapter was also applied on a number of blind targets from the recently concluded CASP8 and CASP9 community-wide experiments. For each target provided during the CASP8 and CASP9 events, the helix and strand content algorithms presented
Figure 2.8: Distribution of Q3 and SOV values for proteins in PDBSelect25 data set
previously were used to determine if the target was purely \( \alpha \)-helical or if it contained any extended \( \beta \)-strand regions. Any protein with \( \beta \)-strand content predicted to be less than 10% was identified as purely \( \alpha \)-helical. For each of these targets, the helix prediction algorithm was applied to identify the location of \( \alpha \)-helices. The accuracy of the prediction was measured using the same metrics as presented previously. These results are presented in Table 2.2.

The overall average Q3 and SOV were observed to be 80.8% and 80.6%, respectively. As a test towards the possibility of the extension of the \( \alpha \)-helical prediction model to include mixed \( \alpha/\beta \) proteins, the algorithm was tested on all \( \alpha \) and mixed \( \alpha/\beta \) proteins provided during the CASP8 and CASP9 experiments. The distribution for Q3 and SOV for 172 proteins is shown in Figure 2.9. The overall average Q3 and SOV were observed to be 79.4% and 73.3%, respectively. Given that the parameters of the model were not trained on any mixed \( \alpha/\beta \) proteins, the marginal drop in performance can be seen as an encouraging sign of the possibility of extending the model to predict the secondary structure elements for mixed \( \alpha/\beta \) proteins as well.

Table 2.2: Q3 and SOV accuracy for helix predictions for targets from CASP8 and CASP9 experiments

<table>
<thead>
<tr>
<th>Protein</th>
<th>Q3</th>
<th>SOV</th>
<th>Protein</th>
<th>Q3</th>
<th>SOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>3NO6</td>
<td>73.2</td>
<td>67.1</td>
<td>3MPX</td>
<td>93.0</td>
<td>82.5</td>
</tr>
<tr>
<td>2L3W</td>
<td>82.5</td>
<td>72.0</td>
<td>2KY4</td>
<td>75.2</td>
<td>70.3</td>
</tr>
<tr>
<td>2L06</td>
<td>84.5</td>
<td>80.4</td>
<td>2XSE</td>
<td>78.7</td>
<td>89.9</td>
</tr>
<tr>
<td>3NRG</td>
<td>76.5</td>
<td>76.0</td>
<td>3NKZ</td>
<td>82.5</td>
<td>75.8</td>
</tr>
<tr>
<td>3NMD</td>
<td>94.3</td>
<td>98.4</td>
<td>3NNR</td>
<td>85.0</td>
<td>74.0</td>
</tr>
<tr>
<td>3NQW</td>
<td>69.7</td>
<td>66.7</td>
<td>3OQL</td>
<td>78.2</td>
<td>66.9</td>
</tr>
<tr>
<td>2X3O</td>
<td>72.6</td>
<td>94.1</td>
<td>3D7I</td>
<td>73.2</td>
<td>76.2</td>
</tr>
<tr>
<td>3DAI</td>
<td>86.2</td>
<td>83.1</td>
<td>3DEW</td>
<td>74.0</td>
<td>89.0</td>
</tr>
<tr>
<td>2K5E</td>
<td>76.7</td>
<td>74.9</td>
<td>3DJB</td>
<td>90.2</td>
<td>92.4</td>
</tr>
<tr>
<td>2K53</td>
<td>75.7</td>
<td>80.5</td>
<td>3DKA</td>
<td>78.5</td>
<td>87.8</td>
</tr>
<tr>
<td>2K5K</td>
<td>96.8</td>
<td>94.3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 2.9: Distribution of Q3 and SOV values for proteins from the CASP experiments
2.6 Discussion

The single sequence prediction ability of the helix prediction algorithm presented in this chapter provides it with a distinct advantage over most secondary structure prediction algorithms. In the absence of reliable profile information for any target protein, most secondary structure prediction algorithms have very low confidence in their own predictions. Such a situation can occur if either a protein has no matches of significance in the database, or if it matches equally well with more than one sequences have different secondary structure elements. With increasing number of proteins in the data bank, the number of cases in which the first scenario is observed are reducing. However, the second scenario continues to be an important source of discrepancy among secondary structure prediction methods which rely on profile information. One of the critical stages in any method relying on profile information is the alignment of a target sequence with the database. While methods such as PSI-BLAST [6] are commonly employed for this purpose, the interpretation of the data is left to the individual algorithm. Depending on the similarity cutoff implemented in the algorithm, a particular sequence match may or may not be rejected.

The helix prediction algorithm presented in this chapter aims to replicate the hydrogen bonding fingerprint observed in helices. Almost all backbone-to-backbone hydrogen bonds between amino acids in helices take place between residues separated by three or four residues, depending on the turn of the helix. It has been observed that this hydrogen bond network stabilizes the helix, to the extent of ensuring the stability of a single helix in solution [23]. At the ends of each helix, the model incorporates end effects caused by the presence of side chain and hydrophobic interactions by evaluating propensities for each residue type to be at specific locations at the ends of helices. As has been discussed previously in literature [15], specific positions at the N and C termini of helices provide an ideal environment for hydrogen bond donors and acceptors to exist, thus ensuring the additional stability at the end of the helix. While the probability parameters were developed from a large non-redundant data set, the lack of employment of a sequence similarity procedure like
PSI-BLAST offers a distinct advantage to the helix prediction algorithm when dealing with target proteins of low sequence similarity. Given that the confidence of sequence similarity is not a metric involved in the prediction algorithm, the location of helical regions would depend solely on the ability of amino acid pairs to form the hydrogen bonding network representative of a helix.

In a manner similar to the prediction of $\alpha$-helix locations in purely helical proteins, a model can be developed for the prediction of $\alpha$-helices in mixed $\alpha/\beta$ proteins. However, the challenge of secondary structure prediction is larger in such proteins, since it would also be required to predict the location of the $\beta$-strands of the protein. The prediction of $\beta$-strands is an even more challenging problem for methods which do not rely on profile information. Unlike $\alpha$-helices, $\beta$-strands are not stabilized by local forces only. As would be described in detail in Chapter 3, the stability of $\beta$-strands is attributed to hydrogen bonding ladders formed between amino acids which are quite separated in the primary sequence of the protein. For any algorithm not relying on profile information to predict the secondary structure of a protein, it is conceivable that a sufficient degree of accuracy can be achieved only by simultaneously predicting the location of $\alpha$-helices, $\beta$-strands and their arrangement in a $\beta$-sheet topology. As will be shown in Chapter 3, the prediction of sheet topology, given the location of $\beta$-strands is a very challenging problem. This makes the helical prediction in mixed $\alpha/\beta$ proteins a greater challenge than pure $\alpha$-helical proteins.

2.7 Conclusions

This chapter presents a new single sequence $\alpha$-helix prediction method. Instead of relying on profile information for any target protein, the algorithm aims to capture the hydrogen bonding fingerprint present in $\alpha$-helices. In addition, end effects commonly seen in $\alpha$-helices, attributed to hydrophobicity and side chain interactions, are also incorporated. The training model has been implemented as a linear programming formulation, while the prediction algorithm for any target protein is implemented as a mixed integer linear optimization model.
The model was tested on two large, independent data sets. The first test set consists of proteins from the PDBSelect25 databank, with pairwise sequence similarity under 25%. The average Q3 accuracy for this data set was seen to be 83.8%. The second data set consists of proteins from the recently concluded community-wide CASP9 experiment. The Q3 accuracy for this data set was seen to be 81.8%. Given that the method does not rely on profile information, it is seen to perform consistently similar to the proteins which rely on sequence similarity information. The algorithm would hence provide a distinct advantage when run on proteins with very low sequence similarity to the protein data bank.
Chapter 3

β-sheet topology prediction

While the prediction of the location of secondary structure elements like α-helices and β-strands is an important initial step towards the final three dimensional structure prediction of proteins, the knowledge of these secondary structure elements provide guidance on local structures only. Hence, when translated into mathematical constraints to be imposed on the tertiary structure of a protein, this information presents relation between amino acids which are located close together in sequence. Further, while α-helices are intrinsically stable because of the presence of local hydrogen bonds \[184\], isolated β-strands are not stable by themselves. They are stabilized because of the presence of lateral non-local hydrogen bonds. Non-local hydrogen bonds in this work will refer to the formation of hydrogen bonds between separated parts of the sequence. The arrangements of β-strands in such a hydrogen bonding network is termed as the β-sheet topology of a protein. In terms of developing constraints for the final three dimensional structure prediction algorithm (presented in Chapter 6), the accurate prediction of the β-sheet topology would provide distance constraints between pairs of amino acids separated in sequence, by restricting the distance between them to ensure that a hydrogen bond is formed between the relevant atoms of these amino acids. Wako and Scheraga \[208\] have evaluated the impact of non-local distance constraints between
pairs of atoms on the quality of the predicted structure. An empirical relation was found between the number, quality and type of distance constraints to the RMSD value of the structures generated.

Figure 3.1 shows a typical representation of a parallel and antiparallel \( \beta \)-sheet (Figure 3.1(a) shows a Parallel \( \beta \)-sheet, while Fig 3.1(b) shows an Antiparallel \( \beta \)-sheet). As can be seen in both of these diagrams, the contacts formed between \( \beta \)-strands (represented by each long flat strip) are nonlocal in nature. Further, Figure 3.1(a) shows that all the strands are arranged in a single sheet, while Figure 3.1(b) shows the arrangement of the eight strands in two sheets. Since most algorithms which are not based on first principles aim to use machine learning, they mostly rely on local information for the training of models. The presence of multiple nonlocal contacts is hence only marginally captured in most algorithms.

(a) Example of Parallel \( \beta \)-sheet
(b) Example of AntiParallel \( \beta \)-sheet

Figure 3.1: Cartoon representation of the two most commonly found types of flat \( \beta \)-sheets

3.1 Computational Complexity of \( \beta \)-sheet topologies

The model presented below only deals with flat \( \beta \)-sheets, and avoids the prediction of \( \beta \)-barrels. A flat \( \beta \)-sheet can be described as one for which a circular path cannot be drawn from any \( \beta \)-strand back to itself by tracing the path of the hydrogen bonding network. In order to make sure that the
Table 3.1: The number of motifs possible for a protein with $n$ strands

<table>
<thead>
<tr>
<th>Strands</th>
<th>Number of Motifs</th>
<th>Strands</th>
<th>Number of Motifs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>96</td>
<td>5</td>
<td>960</td>
</tr>
<tr>
<td>6</td>
<td>11520</td>
<td>7</td>
<td>161280</td>
</tr>
<tr>
<td>8</td>
<td>2580480</td>
<td>9</td>
<td>46448640</td>
</tr>
<tr>
<td>10</td>
<td>928972800</td>
<td>11</td>
<td>$2.0437 \times 10^{10}$</td>
</tr>
</tbody>
</table>

The same $\beta$-sheet topology is not counted twice by rotating our view of the sheet, the following pair of restrictions are introduced:

1. The position of the first strand in the sequence, starting from the N terminus is within the first $[(n + 1)/2]$, where $n$ is the total number of strands. This would ensure that the rotation of the image would not result in the counting of a new $\beta$-sheet topology.

2. The orientation of the first strand is taken to be fixed to, without loss of generality, “up”. This way, the rotation of the protein on the other axis does not result in the duplication of count of the $\beta$-sheet.

Using the aforementioned rules, for any $\beta$-sheet of $n$ strands, there are $n!$ factorial ways of arranging them. By including the $2^n$ ways of arranging the $\beta$-sheets, and including the elimination of the axes of symmetry, we would have

$$\frac{1}{4} \times n! \times 2^n = n! \times 2^{n-2}$$

possibilities for an $n$-stranded $\beta$-sheet. Table 3.1 thus provides the number of possible arrangements of $\beta$-strands in proteins with number of strands between 2 and 11.
3.2 Background

Predicting the topology of alignment of $\beta$ strands in a protein has proved to be harder than the prediction of secondary structure, primarily because of the presence of non-local hydrogen bonds which connect residues in different strands. In order to address this problem, a significant amount of work has been carried out towards understanding patterns, and deriving biological logic behind the existence of a fairly small subset of all possible $\beta$-sheet topologies.

3.2.1 Conformational and Biological restrictions on $\beta$-sheet topologies

In order to determine rules based on conformational and biological observations of proteins, $\beta$-sheet topologies observed in nature have been categorized into a broad set of categories. Some of the earliest work in this direction classified proteins based on tertiary structure patterns [34] [165]. Subsequently, protein structures have been classified in large databases like SCOP and CATH, based on the structural family that they belong to [79] [142] [147] [148].

A number of surveys in literature highlight specific properties of the presence and stabilizing role of side chain hydrogen bonds in the various secondary structures. For antiparallel $\beta$-sheets, alternate pairs of residues are hydrogen bonded. Hence, the stability of adjacent beta strands forming hydrogen bonds is dependent on the residue pairs forming the hydrogen bonds. The analysis of side chain interactions and pair wise correlations has been carried out in detail by Wouters and Curmi [218]. They found a correlation between the directionality in the packing interactions of non hydrogen bonded $\beta$ and $\gamma$ branched residues, and the handedness of the twist in the $\beta$ sheet. Likewise, amino acid pairing preferences in parallel $\beta$ sheets has also been carried out [64]. In any residue pair, it was shown that propensity of sheet formation was different, depending on the which residue actually forms the hydrogen bonds. The favorability of certain residue pairs was explained on the basis of the preference of the $\chi_1$ angle of the side chain rotamer. The interactions
between charged residues was shown to be dependent on if the negatively charged residue formed the hydrogen bond.

Considerable work has been carried out over the years, aiming to determine conformational and structural restrictions in $\beta$ and mixed $\alpha/\beta$ proteins. Orengo and Thornton \cite{149} classified mixed $\alpha/\beta$ proteins into broad categories: the $\alpha - \beta$ sandwich where $\alpha$ helices and $\beta$-strands form unique layers like a sandwich, and the $\alpha - \beta$ rolls where the $\beta$-sheet forms folds or rolls, thus creating a cradle for the $\alpha$-helices. Similarly, extensive analysis on the extraction and classification of the greek key motif in $\beta$-sheets has been presented by Hutchinson and Thornton \cite{84}. Research has also aimed to eliminate certain $\beta$-sheet arrangements based on topological arguments. It was observed that crossover arrangements (i.e, connections between consecutive parallel strands in a given sheet, irrespective of whether they are actually contacting each other) are right handed in nature \cite{163,192}. Aside from elaborate topological studies which present generic rules for the elimination of strand arrangements, pointers were provided towards elimination of topologies under specific conditions or preferences towards specific arrangements of $\beta$-strands. One of the most significant reductions in the allowed topologies comes from the contribution by Richardson \cite{164}, who presented a series of simple rules which eliminate a large number of topologies of proteins depending on handedness of connections and the elimination of “knots”, or crossing loops, in the structure. More recently, an exhaustive analysis of $\beta$-sheets with upto 6 strands was presented \cite{172}. A detailed analysis of the small $\beta$-sheets displayed preference of $\beta$-sheets with the same type of contact between pairs of $\beta$-strands, along with a strong rejection of $\beta$-strand arrangements which caused the formation of knots or pretzel-like structures.

A number of approaches have been used to combine the secondary structure prediction, and the $\beta$-sheet topology prediction problems. These algorithms take as input the primary sequence of the protein, and provide the locations of the $\beta$-strands in addition to the arrangement of these strands in the three dimensional space. Klepeis and Floudas \cite{103} presented an integer linear optimization based framework, which produces a rank-ordered list of $\beta$-strand arrangements, along
with the locations of cysteine-cysteine disulphide bridges. Starting from an amino acid sequence, and following the separation of all \(\alpha\)-helical residues, their approach creates a superset of possible \(\beta\)-strand regions. Using binary variables to represent residue-to-residue and strand-to-strand contacts, the algorithm predicts the locations and arrangements of the \(\beta\)-strands by maximize the hydrophobic contact potential of contacting amino acids. Other methods have used database driven algorithms like conditional random fields [119] for the simultaneous prediction of \(\beta\)-strands and \(\beta\)-sheets.

The \(\beta\)-sheet topology prediction problem is a more specific instance of the contact prediction problem, a field where a number of varied approaches have been presented in literature. In general, the protein contact prediction problem aims to predict contacts between non-local residues of a protein that would be expected to be in proximity in the final three dimensional structure. Some of the most common approaches to contact prediction use homology modeling [128], correlated mutation analysis [109], machine learning techniques [220] and optimization based approaches [159].

### 3.2.2 Approaches to \(\beta\)-sheet topology prediction

A number of methods have employed data mining based methods to derive contact potentials for pairs of residues which are present in \(\beta\) strands [30, 195, 231]. Initial work in this direction aimed to use residue pair potentials to determine the alignment of strands [82]. The authors used a combination of neural network based secondary structure prediction, a pair potential, and hidden markov models for fold recognition. Other researchers presented work where tripeptides were used to derive potentials for the prediction of \(\beta\)-sheets [13]. Similarly, stochastic tree grammar was used for the identification of \(\beta\)-sheets [127], although the test set for this algorithm was very limited. Steward and Thornton [195] used an information theoretic approach to develop sets of tables with pair information values. Similarly, residue pairwise potentials have been derived for
residue pairs in contact, as well as offset by up to two amino acids \[231\]. These pairwise potentials were used to derive a weighted contact potential between $\beta$-strands, and to derive a rank-ordered list of predicted $\beta$-sheet topologies. Cheng and Baldi \[30\] presented an algorithm BetaPro, which predicts the arrangement of $\beta$-strands in a three stage approach. 2D recursive neural networks were trained to predict the contact potential between amino acid pairs. These pseudo contact potentials are used in a dynamic programming framework to determine the best alignment between pairs of strands. Finally, a greedy algorithm is used to predict the arrangement of $\beta$-strands, while keeping basic biological constraints. Two approaches were further presented which combined the BetaPro approach with integer optimization and an enhanced greedy approach to accommodate folding cooperativity \[88\]. In this work, any contact formed between pairs of $\beta$-strands resulted in an increase in the strand-to-strand contact potentials of neighboring strands, thus mirroring a zipper-like cooperativity in the formation of contacts between strands that are not sequentially continuous.

Bayesian approaches were introduced for the prediction of $\beta$-sheet topologies \[17\]. Separate algorithms were presented for proteins up to six strands, and for proteins with more than six strands. Given the larger amount of available training data, proteins with up to six strands have been modeled using a probabilistic framework by combining residue pairing potentials derived out of apriori knowledge of known $\beta$-sheet architectures. For proteins with more than six strands, a modified approach to that of Cheng and Baldi \[30\] was proposed, by introducing penalties for gaps in strand alignments, and by accounting for the formation of $\beta$-bulges.

### 3.3 Mathematical Model

Figure 3.2 shows the entire flowsheet of the $\beta$-sheet topology prediction model. It should be noted that the method does not involve any training, and hence any protein can be used as a test protein for the prediction of the arrangement of $\beta$-strands. As an input for any given protein, the algorithm only expects the amino acid sequence, as well as a pre-determined secondary structure,
i.e. the location of the $\alpha$-helices and $\beta$-strands. The first precursor for the prediction of the $\beta$-sheet topology of the protein is the identification of the $\beta$-strand regions in the protein. We used the Dictionary of secondary structure of proteins (DSSP) for the identification of $\beta$ strands [92].

Figure 3.2: Complete flowsheet of the $\beta$-sheet topology prediction algorithm

### 3.3.1 Residue-to-Residue Contact Potential

This secondary structure information (including positions of helices in the protein) was used for the generation of residue-residue contact potential generation from the method of Cheng and Baldi [30]. The model uses an input vector of 251 elements to determine the contact potential between
any pair of amino acids. For any amino acid pair \(i, j\) a local window of five amino acids around each of the amino acids is selected. Each position in the local window accounts for 20 positions representing the amino acids, 3 representing the secondary structure (helix, strand and coil) and 2 representing the binary nature of solvent accessibility to the amino acid (a cutoff of 25% is used for this purpose). In addition, a term representing the sequential separation of the pair of amino acids is used.

### 3.3.2 Strand-to-Strand Contact Potential

For any pair of strands the best alignment score was determined by sliding one strand across the second in parallel and antiparallel fashion. While doing so, it is ensured that the selected alignment between any pair of strands does not result in more than two amino acids on either of the strands to be unsatisfied with respect to hydrogen bonding. Further, for \(\beta\)-strands up to three amino acids, it is required that at least two amino acids should form a contact in the selected optimal alignment.

Given that the psuedo contact potentials are derived from database driven methods, it is expected to have a bias towards local contacts. Since the availability of training data for long range contacts is much lesser than local contacts, the algorithm is biased towards providing a higher weightage to local contacts \([88]\). To correct for this bias, a new parameter is introduced. Let the strands of a protein be defined starting from 1, 2, \ldots, \(N\), where \(N\) is the total number of strands in the protein. The modified strand-to-strand antiparallel contact potential \(E_{AP}(si, sj)\) can be given as:

\[
E_{AP}(si, sj) = (1 + 0.5 \times (sj - si - 1)) \times E_{AP,n}(si, sj) \tag{3.1}
\]

\[
E_{P}(si, sj) = (1 + 0.5 \times (sj - si)) \times E_{P,n}(si, sj) \tag{3.2}
\]

where \(E_{AP,n}(si, sj)\) and \(E_{P,n}(si, sj)\) are the nominal values of the strand-to-strand antiparallel and parallel contact potentials (respectively) evaluated previously.
We define three sets of binary variables, as defined under:

\[
y(i, j) = \begin{cases} 
1 & : \text{if residues } i \text{ and } j \text{ contact} \\
0 & : \text{otherwise} 
\end{cases} \quad (3.3)
\]

Here, residues \( i \) and \( j \) belong to different strands.

\[
w_{AP}(s_i, s_j) = \begin{cases} 
1 & : \text{if strands } s_i \text{ and } s_j \text{ contact in antiparallel fashion} \\
0 & : \text{otherwise} 
\end{cases} \quad (3.4)
\]

\[
w_P(s_i, s_j) = \begin{cases} 
1 & : \text{if strands } s_i \text{ and } s_j \text{ contact in parallel fashion} \\
0 & : \text{otherwise} 
\end{cases} \quad (3.5)
\]

Since all contacts are commutative, all binary variables are set up such that the second index is greater than the first. The objective of the model is to maximize the contact potential of the predicted \( \beta \)-sheet topology, and takes the form:

\[
\text{OBJECTIVE} = \sum_{s_i} \sum_{s_j} E_{AP}(s_i, s_j)w_{AP}(s_i, s_j) + \sum_{s_i} \sum_{s_j} E_P(s_i, s_j)w_P(s_i, s_j) \quad (3.6)
\]

### 3.3.3 Relational Constraints

Several constraints are included to ensure that we obtain physically realistic \( \beta \)-sheet topologies. The first set of constraints link the binary variables for residue-residue contacts \((y(i, j))\) to the binary variables for strand-strand contacts\((w_{AP}(s_i, s_j) \text{ and } w_P(s_i, s_j))\). By evaluating the strand-strand contact potentials \(E_P(s_i, s_j)\) and \(E_{AP}(s_i, s_j)\), we know the best alignment of any strand pair. We hence define two binary matrices \(\text{ResidueContactAP}(i, j)\) and \(\text{ResidueContactP}(i, j)\),
wherein entries are 1 if $i$ and $j$ can form a contact at all. In addition, we define parameters $Strand(i)$ which represent the strand to which residue $i$ belongs. Of course, this contact would depend on whether the strands they belong to are in contact. This condition can be expressed as:

$$y(i, j) = w_{AP}(si, sj) \ast ResidueContactAP(i, j) + w_{P}(si, sj) \ast ResidueContactP(i, j)$$

$$\forall Strand(i) = si, Strand(j) = sj, sj > si$$  \hspace{1cm} (3.7)

The constraint expresses the relation between the sets of binary variables by enforcing that the binary variable $y(i, j)$ is active if the amino acids can form a contact (represented by $ResidueContactAP(i, j)$ and $ResidueContactP(i, j)$) and the corresponding strands are in contact (represented by $w_{AP}(si, sj)$ and $w_{P}(si, sj)$). Any two strands $si$ and $sj$ can at most form one type of contact with each other, which becomes:

$$w_{AP}(si, sj) + w_{P}(si, sj) \leq 1 \hspace{1cm} \forall sj > si.$$  \hspace{1cm} (3.8)

The following section presents a large number of constraints added to the optimization model, aimed at providing biologically meaningful results. For each constraint, the performance of the constraint on the PDBSelect25 is also presented.

### 3.3.4 Biological Constraints

A strand residue can have a maximum of two contacts. However, this does not mean that the strand itself can only have two contacts. It is possible for a long strand to pair up with more than one strand on one side. Hence, the maximum number of contacts a strand can make is taken as 3. In the entire set of proteins, only four proteins had one strand with four contacts and none had
more than four contacts. At the same time, it is required that each strand have at least one contact. These constraints can be represented as:

$$\sum_{j \neq i} y(i, j) \leq 2 \quad \forall i, \text{Strand}(i) \neq \text{Strand}(j) \quad (3.9)$$

$$\sum_{s_j \neq s_i} w_{AP}(s_i, s_j) + \sum_{s_j \neq s_i} w_P(s_i, s_j) \leq 3 \quad \forall s_i \quad (3.10)$$

$$\sum_{s_j \neq s_i} w_{AP}(s_i, s_j) + \sum_{s_j \neq s_i} w_P(s_i, s_j) \geq 1 \quad \forall s_i. \quad (3.11)$$

**PDBSelect25 Statistics:** No amino acid was seen to have more than two contacts for any protein. A minimum of one contact is true for all strands in PDBSelect25. Only four proteins were seen to have one strand each with four contacts. No protein was seen to have any strand with more than four contacts.

For a non barrel protein structure, the total number of contacts does not exceed $$N_{str} - 1$$, where $$N_{str}$$ is the total number of strands in the protein. This is expressed as:

$$\sum_{s_i} \sum_{s_j} w_{AP}(s_i, s_j) + \sum_{s_i} \sum_{s_j} w_P(s_i, s_j) \leq N_{str} - 1 \quad (3.12)$$

**PDBSelect25 Statistics:** This constraint is very general, and is satisfied for all proteins which do not have any strand with four contacts. Further, the upper bound is reached only for proteins with one $$\beta$$-sheet.

Since hydrogen bonding and hydrophobic collapse are believed to be the driving force for $$\beta$$-strands to form sheets, the strands aim to minimize exposed area [192, 193]. Moreover, since $$\beta$$ sheets typically form the core of the protein, the possibility of unsatisfied side chains forming hydrogen bonds with the solvent reduces. This exposed area comes about when two unequal strands form a contact, or when a contact is off-centre. In order to ensure that strands with similar lengths form contacts, and that the hydrogen bonding requirements of the strand are satisfied, we
enforce that the total residues contacting a given strand should lie between \((len_{si} - 2)\) and \((2len_{si} + 3)\), where \(len_{si}\) is the length of the strand \(si\). We introduce parameters \(N_{contactAP}(si, sj)\) and \(N_{contactP}(si, sj)\), defined as:

\[
N_{contactAP}(si, sj) = \sum_{i \in si} \sum_{j \in sj} ResidueContactAP(i, j) \quad \forall sj > si
\]

\[
N_{contactP}(si, sj) = \sum_{i \in si} \sum_{j \in sj} ResidueContactP(i, j) \quad \forall sj > si
\]

The constraint can hence be written as:

\[
\sum_{sj \neq si} w_{AP}(si, sj) \times N_{contactAP}(si, sj) + \sum_{sj \neq si} w_{P}(si, sj) \times N_{contactP}(si, sj) \geq len_{si} - 2 \quad (3.15)
\]

\[
\sum_{sj \neq si} w_{AP}(si, sj) \times N_{contactAP}(si, sj) + \sum_{sj \neq si} w_{P}(si, sj) \times N_{contactP}(si, sj) \leq 2 \times len_{si} + 3 \quad (3.16)
\]

**PDBSelect25 Statistics:** For all \(\beta\)-strands in the PDBSelect25 data set (including all \(\beta\) and mixed \(\alpha/\beta\) proteins), we find a success rate of 99.71% for the lower bounding expression shown above. Even among the remaining 0.29% of \(\beta\)-strands, 0.22% fall within an error of one amino acid of the allowed lower bound. For the upper bound, we find a success rate of 99.92% among all \(\beta\)-strands in the PDBSelect25 data set. By reducing the upper bound to \(2len_{si} + 1\), the success rate goes down to 99.85%, and can be used as a means to tighten the upper bounding constraint.

In a number of instances, it is seen that a longer strand pairs with more than one smaller strand on one side. While Equation 3.9 ensures that any strand residue does not have more than 2 contacts, there could still be a possibility wherein the third contacting strand is predicted to wrap around the first strand, thus satisfying criteria for maximum number of strand and residue contacts. In order to avoid this, we introduce parameters \(Overlap(si, sj, sk)\), which measure the overlap...
in contacting residues of strands $s_i$ and $s_j$, when both contact strand $s_k$. Thus, for any triplet of strands $(s_i, s_j, s_k)$ contacting a fourth strand $s_l$, we impose that the overlap of at least one pair be zero. This is written as:

$$w_{AP}(s_i, s_l) + w_{AP}(s_j, s_l) + w_{AP}(s_k, s_l) \leq 2$$

$$\forall \text{Overlap}(s_i, s_j) \cdot \text{Overlap}(s_j, s_k) \cdot \text{Overlap}(s_i, s_k) \geq 1. \quad (3.17)$$

**PDBSelect25 Statistics:** This is a basic constraint that is satisfied by all proteins.

Similar constraints can be written involving parallel contacts. Further, it was observed that for strands making three antiparallel contacts, at least one contact was made with its neighbors, or one of the edge strands. A number of strands forming 3 contacts made their third contact with a very small strand, which was typically either its own neighbor (by merely proving to be a small extended region following a $\beta$-turn) or at either end of the protein sequence, thus resulting in a much smaller impact on entropy loss. This constraint can be written as

$$\sum_{s_j \neq s_i} w_{AP}(s_i, s_j) \leq w_{AP}(1, s_i) + w_{AP}(s_i - 1, s_i)$$

$$+ w_{AP}(s_i, s_i + 1) + w_{AP}(s_i, N) + 2. \quad (3.18)$$

**PDBSelect25 Statistics:** In the PDBSelect25 data set, 396 $\beta$-strands were seen to have three contacts. Among these, 374 (94.44%) $\beta$-strands were seen to satisfy the constraint mentioned above.

Based on the idea presented by Przytycka et al. recently [154], non-local contacts can be classified into specific classes. In this article, the authors are able to re-create 80% of existing topologies using a small set of rules for bringing sequentially distant strands together. At each
implementation of a rule, strands ended up forming new neighbors (i.e. a new set of strands could potentially come together to form a contact). Hence, for any non local contact to form (here, we define a non-local contact to be a contact between strands $s_i$ and $s_j$ such that $s_j \geq s_i + 3$), the constraint is expressed as:

$$w_{AP}(s_i, s_j) \leq w_{AP}(s_i-1, s_j+1) + w_{AP}(s_i-1, s_j+1) + w_{AP}(s_i+1, s_j-1) + w_{AP}(s_i+1, s_j+1).$$ \hspace{1cm} \text{(3.19)}$$

A few qualifiers for the validity of Equation 3.19 have been put in place. A circular definition of neighbors has been employed, (i.e. the strand preceding the first strand is taken as the last strand). Similarly, the strand following the last strand is the first one in the sequence. A similar approach was used previously while determining the rules of formation of $\beta$-sandwich topologies in pure $\beta$ proteins \cite{32, 98}. Further, if a neighbor of a given strand is of length two or three, we move further along in the sequence in the same direction till we identify a valid neighbor to the current strand. The rationale behind this idea is that a very small strand is not influential enough to actually bring sequentially separated parts of the protein together in space. For strands $i$ and $j$ such that $j = i + 2$, we add two additional terms to the equation, representing the contact of strand $i + 1$ with strands $i$ and $j$. A similar set of equations is written out for parallel contacts.

**PDBSelect25 Statistics**: A success rate of 93.18\% was seen for this set of constraints.

Driven by hydrophobic collapse, it is expected that the most hydrophobic strands would form the core of the $\beta$ sheet, while the less hydrophobic and shorter strands would form the terminals on both sides \cite{194}. This would mean that the less hydrophobic and shorter strands are likely to have one contact, while the more hydrophobic or longer strands are likely to have more than one contact. The strands are first sorted by length. Within a given length, the strands are sorted by the number of hydrophobic residues. Starting from the smallest strand, we postulate claim that atleast one of the first two would have just one contact. We continue to grow this set in a similar manner, (i.e. atleast 2 of the first four would have one contact each, and so on). The number of such sets
created depends on the total number of strands, and one such set is added for every five strands in
the entire protein.

A number of additional constraints are used to further narrow the search space of possible β-
sheets. The first set of constraints restrict the total number of hydrogen bonds, or “contacts”, that
are formed between pairs of amino acids in β-strands. Studying hydrogen bonding patterns in
globular proteins [196], the absolute number of hydrogen bonds ($N_{HB}$) formed in a protein could
be expressed as:

$$N_{HB} = 1.49f_{\alpha} \ast N + 0.65f_{\beta} \ast N + 0.5 \ast (1 - f_{\alpha} - f_{\beta}) \ast N. \quad (3.20)$$

where $f_{\alpha}$ and $f_{\beta}$ are the fractions of α-helical and β-strand residues in the protein, respectively.
From the expression, we can see that the three terms on the right hand side represent the expected
contributions of the helical, extended and coil regions, respectively, to the total number of hydrogen
bonds in the protein. Using this expression, restrictions are introduced on the total number of
hydrogen bonds (or “contacts”) between amino acids in β-strands, by allowing a 15% error range
around the predicted value. Mathematically, this is written as:

$$N_{HB,min} \leq \sum_{i} \sum_{j} y(i, j) \leq N_{HB,max} \forall i \in s_i; j \in s_j, s_j > s_i. \quad (3.21)$$

**PDBSelect25 Statistics:** The success rate for the lower and upper bounds in the constraint
shown above are 93.21% and 95.14%, respectively. The main aim of this set of constraints is to
reduce erroneous over-prediction of contacts, thus reducing the number of false positives predicted.

One of the arrangements of β-strands conspicuous by its absence is commonly referred to as
the “pretzel” [38], and were used recently [72]. For any quartet of β-strands ($s_i, s_j, s_k, s_l$) which
lie in the same β-sheet, this constraint prevents the possibility of arrangements which result in the
four strands lining up as ($s_k, s_i, s_l, s_j$) or ($s_j, s_l, s_i, s_k$). This restriction is written as:
\[ w_{AP}(s_i, s_k) \leq 2 - w_{AP}(s_i, s_l) + w_{AP}(s_j, s_l). \] (3.22)

**PDBSelect25 Statistics:** The success rate for this constraint is observed to be 99.21% among all proteins in the PDBSelect25 data set.

A similar equation can be written for parallel contact between strands, and for combinations of the two kinds of contacts. Another class of contacts which have been observed to be constantly avoided is known as the layer crossover [72]. According to this rule, for two pairs of sequentially consecutive \( \beta \)-strands starting at strand positions \( s_i \) and \( s_j \), the strand combination \((s_i + 1, s_j + 1, s_i, s_j)\) is avoided. This condition is an indirect consequence of the loop crossing conditions presented by Richardson [164] and Gelfand and co-workers [32, 98]. This can be mathematically presented as:

\[ w_{AP}(s_i + 1, s_j + 1) \leq 2 - w_{AP}(s_i, s_j + 1) + w_{AP}(s_i, s_j). \] (3.23)

**PDBSelect25 Statistics:** The success rate for this constraint is observed to be 97.14% among all proteins in the PDBSelect25 data set.

As before, a circular nature for the identification of strands is used, wherein the first strand is assumed to be “following” the last strand of the primary sequence.

\[ \sum_{s} k \sum_{s} lw_{AP}(sk, sl) \geq \sum_{s} i \sum_{s} lw_{AP}(si, sj) \] (3.24)

This constraint also encompasses the additional requirement of each non-local contact to be a part of exactly one “interlock”, also observed previously in literature [63]. Recent work has shown specific patterns that have emerged out of the analysis of \( \beta \)-sandwich proteins. These proteins are characterized by a pair of \( \beta \)-sheets packed against each other like a sandwich [63, 97]. The first observation was the absence of parallel contacts between strands. Further, it was observed that for
any non-local strand pairing \((s_i, s_j)\) in one sheet, a counter-balancing non-local contact between \(s_i + 1\) and \(s_j + 1\) is observed in the opposite sheet, thus forming an “interlock”. These constraints cannot be directly applied to our model, since the aim is to able to develop a prediction algorithm for any kind of \(\beta\) or mixed \(\alpha/\beta\) protein. Hence, we generalize this condition to include any quartet of strands \((s_i, s_j, s_k, s_l)\) such that \(s_i < s_k < s_j < s_l\) and postulate that an interlock is formed between strand pairs \((s_i, s_j)\) and \((s_k, s_l)\), given by the following constraint:

\[
\sum_{s_k} \sum_{s_l} w_{AP}(s_k, s_l) \geq \sum_{s_i} \sum_{s_l} w_{AP}(s_i, s_j).
\]  

\[(3.25)\]

**PDBSelect25 Statistics:** The success rate for this constraint is observed to be 91.81% among all proteins in the PDBSelect25 data set. The highest number of instances when the constraint is not satisfied involves \(\beta\)-strands which are contacting three other \(\beta\)-strands of the protein.

### 3.3.5 Integer Cut Constraints

The advantage of creating an integer linear optimization based model is the facility to create of a rank-ordered list of solutions. We aim to predict a small subset of topologies for each protein. In a number of cases, the objective function value of two topologies are highly similar to each other. By enlisting a small subset of top solutions, it enables us to differentiate between the topologies using a more detailed force field at the final stage. This can be achieved through the introduction of integer cuts. Since we are fixing the anchor points for contacts between two strands, the integer cuts would not involve the residue specific binary variables \(y(i, j)\). At each iteration, the addition of an integer cut eliminates the current top solution from the feasible set, thus forcing the model to look for the next best solution. We divide the set of strand-strand binary variables into two subsets: \(A(x)\) defines the subset of variables \(x\) which are assigned value 1, while \(I(x)\) comprises of all contacts which were not active. Let \(N_A\) be the cardinality of the subset \(A(x)\). The index \(x\) runs
over all antiparallel and parallel contacts between strands. The integer cut constraint can be written as:

\[
\sum_{(s_i, s_j) \in A(x)} w_{AP}(s_i, s_j) + \sum_{(s_i, s_j) \in A(x)} w_{P}(s_i, s_j) - \sum_{(s_i, s_j) \in I(x)} w_{AP}(s_i, s_j) - \sum_{(s_i, s_j) \in A(x)} w_{AP}(s_i, s_j) \leq N_A - 1. \tag{3.26}
\]

### 3.3.6 Elimination of circular paths

Since the objective is to maximize the contact potential between strands, most solutions would be cyclic in nature. Since the fraction of proteins which form \(\beta\) barrels is much smaller than proteins which don’t, we choose to eliminate the possibility of all barrel-like structures. On the other hand, if barrel structures were permitted, it was observed that barrel-like results were obtained for 96.4% of proteins. It is not possible to eliminate all circular solutions in the form of constraints. This is because we look to eliminate cyclic, as well as sub-cyclic solutions. For a protein with 20 strands, the number of possibilities exceed \(10^5\). The addition of such a large number of constraints would be extremely detrimental to the speed and performance of the algorithm. Hence, each solution obtained is checked for circular nature. If it is so, we do not increase the solution counter, and instead directly add an integer cut to eliminate this solution. The algorithm used for this purpose is described below.

We first acknowledge the fact that the arrangement of \(\beta\)-strands, obtained as a solution to the optimization model previously presented, is an undirected graph. This means that if one represents the nodes of the graph as the strands, and the edge of the graph as the presence of a contact, then the contact is commutative in nature. A depth-first search (DFS) algorithm can create a tree structure to the solution, starting from the root (say strand 1). We define a node edge structure for each strand with the following parameters: boolean \(Visited\) to represent if we have traversed through
this node in the graph, integer \textit{parent} determining the predecessor of the current node, an array of contacts \textit{Contacts} representing the set of contacts for the strand, along with the boolean array \textit{ContactVisited}, representing if we reached the contact through the current strand, and finally the type of edge between the current strand and the one we move onto from here (\textit{tree} or \textit{back}). The description below explains these types of edges.

A DFS iteration can be represented by the following steps:

1. If this is the first step, take strand 1 as current strand. Otherwise select the current active strand

2. If one or more strands of \textit{ContactVisited}(i) are 0, let strand \textit{j} be the first of such strands.

   - If \textit{Visited}(j) = 1, mark edge as \textit{back} and return true for cyclic graph.
   - Otherwise set \textit{parent}(j) = i, \textit{ContactVisited}(i, j) = true, \textit{ContactVisited}(j, i) = true. Set current strand as \textit{j}.

3. Otherwise set strand \textit{k} as current strand, where \textit{parent}(i) = k.

4. If current strand is strand 1, and it is not the first iteration, terminate and return false for subcycle.

If there are multiple sheets in the final prediction, we then repeat the above algorithm with a strand from the second sheet as the starting strand.

\subsection*{3.3.7 Prohibited Contacts}

In order to ensure that the number of contacts are not over-predicted, and to keep the total number of true positives and true negatives approximately equal to the actual number, each valid result is analyzed to check for over-predictions. Each predicted contact is identified by two strands, say \textit{i} and \textit{j}. If both strands \textit{i} and \textit{j} have other contacts with contact potentials greater than the current
value, with the current value itself below a threshold of 0.2, we fix this contact to be invalid, and re-run the optimization model with this restriction. The need for re-running the model arises from the constraints represented by Equation 3.19. By removing a non-local contact, it is possible that we may be rendering other non-local contacts invalid.

Since the prediction of strand pairings from the algorithm forms a set of unordered pairs of integers, the verification of a set of basic biological consistencies is rendered difficult. As discussed previously in literature [88], one of the primary features of observed β-sheet topologies is the consistency of contact type along any given face of a β-strand, i.e. all contacts of a given β-strand along one of its two faces are either antiparallel or parallel. This cannot be verified trivially by the prediction of the algorithm. Hence, in order to ensure that a consistent assignment is possible for a given topological prediction, each predicted topology is checked for two-colorability, i.e. we check if the predicted β-sheet topology can be re-drawn as a two-colorable graph. To do this, all contacts between strands in the predicted topology are re-cast as nodes of a graph. Two “nodes” are connected if the corresponding contacts share a β-strand. In addition, the two contacts should either be of opposing natures (i.e. one should be parallel, and the second antiparallel) or they should share at least one amino acid of the common strand. This algorithm has been presented in greater detail previously [88]. The two-colorability of a graph is a well established problem, and can be solved by a breadth-first search algorithm.

### 3.4 Re-ranking Predicted Topologies

At the end of the algorithm, a large number of predicted sheet topologies for the target protein are received, which are ranked by the total strand-to-strand contact potential defined previously. However, in a number of cases, it was observed that the difference between the objective function values of the top few solutions was extremely low, perhaps falling into error tolerance limits. Hence, it becomes important to provide an improved ranking of the predicted sheet topologies
using a detailed, atomistic level approach. Hence, we have developed a re-ranking strategy based on torsion angle dynamics and clustering, which would identify the top set of predicted topologies.

While a number of algorithms for the prediction of feasible structures satisfying a sparse set of distance and dihedral angle constraints have been presented in the literature \cite{41,139}, torsion angle dynamics provide a very attractive alternative. Unlike classical molecular dynamics simulations, torsion angle dynamics algorithms combine steric-based energy terms with constraint violation based penalty expressions, thus allowing for faster calculations. Moreover, the primary idea moves from energy minimization to identification of feasible structures. For our algorithm, the CYANA package \cite{75} proves to be a very useful tool for carrying out torsion angle dynamics simulations.

For each predicted sheet topology, the predicted residue-to-residue contacts are converted into lower and upper bounding distance constraints, by using a small error tolerance on the hydrogen bond that would be formed between contacting amino acids. These set of bounds, along with dihedral angle bounds on the amino acids in the $\beta$-strands restricting them to the correct region of the Ramachandran plot, are provided as input to the torsion angle dynamics package. Using CYANA, we generate 200 feasible structures for each predicted sheet topology.

In order to separate out the topologies from each other, we need to assemble a small subset of representative structures from each predicted topology. To this end, we use a traveling salesman problem based clustering algorithm, ICON \cite{48,49,129,198}. Here, each feasible structure generated by CYANA is considered as a node on a traveling salesman path. The problem is then reduced to one of identifying the globally optimal path to navigate through each of the “nodes”. Once such a path is established, it is partitioned into clusters such that the resulting clusters minimize the global sum of intra-cluster errors.

The computational time for the algorithm depends on the number of strands in a protein, and on the number of amino acids in the $\beta$-strands of the protein. For a typical eight strand protein, the generation of residue-to-residue contact potential by the method of Cheng and Baldi \cite{30} takes 10 minutes. The mixed-integer linear optimization formulation for the prediction of 100 $\beta$-sheet
topologies takes 5 minutes. The re-ranking algorithm involving torsion angle dynamics and clustering requires 10 minutes per topology to generate 200 structures. When implemented in parallel on a cluster of nodes, the entire set of topologies can be handled in much quicker time, depending on the number of processors available.

The algorithm is available as a web tool to the scientific community at \( \text{http://selene.princeton.edu/BeST} \). The web tool offers the possibility of using the \( \beta \)-sheet prediction algorithm with and without the knowledge of the native structure of the protein. When a PDB file is provided to the web tool script, sequence and secondary structure files are generated, which can be uploaded for the prediction of the \( \beta \)-sheet topology. In the absence of a PDB file, the provision of a sequence file to the web tool script allows the script to submit the sequence for secondary structure prediction to web servers like PSI-PRED [89], which is subsequently used for \( \beta \)-sheet topology prediction.

### 3.5 Computational Results

This section presents the results of the \( \beta \)-sheet topology prediction algorithm, BeST, applied to a number of varied data sets. We first describe the data sets that were used as test sets, followed by the assignment of secondary structure, and the evaluation of prediction accuracy.

#### 3.5.1 Protein Data Sets

A total of 2602 proteins from the PDBSelect25 (October 2008 release) have been selected as the first test set. The PDBSelect25 data set is a collection of single chain proteins with pairwise sequence similarity up to 25%. This set of proteins is further divided into 796 \( \beta \) proteins, and 1806 mixed \( \alpha/\beta \) proteins. For this data set, the distribution between the number of proteins and the number of actual \( \beta \) strands in proteins is presented in Figure 3.3.
The second set of proteins consists of blind targets obtained from the recently concluded CASP9 experiment. A set of 42 \(\beta\) and mixed \(\alpha/\beta\) proteins that were provided as blind case studies during the CASP9 structure prediction experiment have been used to validate the accuracy of the \(\beta\)-sheet topology prediction algorithm.

### 3.5.2 Secondary structure assignments

The true secondary structure assignments for the proteins in the test sets was made using the dictionary of secondary structure of proteins, DSSP [92]. The method uses hydrogen bonding based criteria to identify the regions of \(\beta\)-strands in any protein. Based on the DSSP algorithm, PROMOTIF [85] determines the native arrangement of \(\beta\)-strands in the protein.
3.5.3 Accuracy Evaluation

Three metrics (Precision, Recall and Matthews Correlation Coefficient, MCC) have been used to evaluate the accuracy of the predicted $\beta$-sheet topologies. These can be defined by the following equations:

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{3.27}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{3.28}
\]

\[
MCC = \frac{TPXTN - FPXFN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.29}
\]

In each of these equations, the expressions $TP$, $TN$, $FP$ and $FN$ stand for true positives, true negatives, false positives and false negatives, respectively.

3.5.4 Results: PDBSelect25 Data Set

For the presentation of the results of the application of the $\beta$-sheet topology prediction algorithm on the data set, all proteins with two strands were removed from the test set. Given that we are looking to generate a large rank-ordered list of solutions, it was felt that the presence of proteins with two strands would bias the results. The weighted average precision, recall and MCC results for the entire data set, for the top 25 generated solutions are presented in Figure 3.4. The weighted average precision for any given number of solutions is given by:

\[
\frac{\text{Precision}}{\text{Precision}} = \frac{\sum_{n=1}^{N} \text{prec}_n \times N\text{prot}_n}{\sum_{n=1}^{N} N\text{prot}_n} \tag{3.30}
\]
Here, $\overline{prec}_n$ is the average precision observed among all proteins with $n$ strands, while $N_{prot,n}$ is the number of proteins with $n$ strands. Similar expressions were used for the evaluation of the weighted average recall and correlation coefficient.

![Figure 3.4: Variation of average precision, recall and MCC over the number of solutions generated](image)

As can be seen from the results, we achieve a top solution precision, recall and MCC of about 63%, 62% and 0.48 respectively. When the top five solutions from the model are considered, the average precision, recall and MCC increase to about 79%, 77% and 0.71 respectively. As the number of solutions considered are increased to 25, we see that the average precision and recall values increase gradually, and take up a value close to 84% and 81% respectively. Table 3.1 shows that the number of arrangements of $\beta$-strands increase significantly with the number of strands in the protein. Even with the large number of arrangements of strands possible in proteins, we observe a very large degree of accuracy in average precision and recall values in the top 25 generated solutions over the entire data set.
Figure 3.5 shows the distribution of the average precision and recall results for varying number of strands, when the number of solutions considered are the top 1, 5, 10, 15, 20 and 25, respectively.

![Figure 3.5: PDBSelect25 Data set results, classified by number of strands](image)

It is observed that proteins with smaller number of strands (i.e. less than or equal to 7) reach high values of precision and recall within the top five solutions. As expected, proteins with three strands reach almost 100% precision and recall within the top five solutions. While a small degree of fluctuation is seen with respect to the precision and recall values for proteins with large number
of strands, these could be classified as outlier points, given that the number of proteins that these bars represent are very few. Further, as would be expected, we see an almost monotonic change in average precision and recall percentage values as the number of \( \beta \)-strands (upto 20\( \beta \) strands) in the proteins increase.

In a different classification of the results, Figure 3.6 shows the accuracy of results in \( \beta \) and mixed \( \alpha/\beta \) proteins.

It is observed that the performance of precision and recall are seen to be superior in the case of the mixed \( \alpha/\beta \) proteins. A number of reasons are possible for this. The first observation that was made was that more local contacts were observed in the case of the mixed \( \alpha/\beta \) proteins in the PDBSelect25 data set, when compared to the pure \( \beta \) proteins. This could be explained by the presence of \( \alpha \)-helices in these proteins, which would cause a certain degree of compartmentalization in the \( \beta \)-strand register, thus encouraging the formation of local contacts. A second explanation can be postulated based on the derivation of the machine learning based pseudo-contact potential that is used in the optimization model. The number of mixed \( \alpha/\beta \) proteins exceed the number of pure \( \beta \) proteins quite significantly, and the model may be biased towards the mixed \( \alpha/\beta \) set. This, in fact, turns out to be true for the particular model being used here. Out of the 916 proteins used in the training of the machine learning model by Cheng and Baldi [30], only 187 could be considered pure \( \beta \) proteins. Finally, it was seen that the mixed \( \alpha/\beta \) proteins formed a smaller number of sheets than the pure \( \beta \) proteins, when the same number of strands were considered. Given that our model aims to maximize contacts between strands, it is expected that indirectly, the model would aim at minimizing the number of \( \beta \)-sheets formed. This could potentially also be contributing to the improved performance in the mixed \( \alpha/\beta \) proteins.

The performance of the \( \beta \)-sheet topology prediction algorithm, BeST, was compared to the state-of-the art \( \beta \)-sheet topology prediction algorithm BetaPro [30]. The BetaPro algorithm produces only one prediction for the \( \beta \)-sheet topology of a protein. The comparison of the top solutions produced by the algorithm presented in this chapter and BetaPro is presented in Table 3.2.
Figure 3.6: PDBSelect25 Data set results classified by class of proteins
The average precision and recall for the solutions of the BetaPro algorithm is 60.9% and 60.1%, respectively. Since our algorithm permits the generation of more than one solution, an improvement of about 18% is observed when five solutions predicted from BeST are compared to the solution of BetaPro.

### 3.5.5 Results: CASP9

The model was also tested on a set of blind targets, provided during the recently concluded critical assessment of structure prediction techniques (CASP9) experiment. Table 3.3 provides a distribution of the number of proteins over the number of strands observed in the proteins.

The precision and recall observed in the top solution, and the top five solutions have been presented in Figure 3.7.

As can be seen from the results, the top solution is seen to have an average precision and recall of 66.1% and 65.8%, while the top five solutions have the corresponding values of 75.1% and 74.4%. This shows that the approach produces similar results when tested on a set of blind targets.

The results mentioned previously were based on the actual secondary structure assignments, generated out of DSSP [92]. However, in a blind target structure prediction experiment, the true secondary structure assignments are unavailable. To address this problem, we carried out secondary structure prediction using CONCORD [212], an integer linear optimization based consensus secondary structure prediction approach. The predicted secondary structure for any target protein can contain more, less or the same number of β-strands as the native secondary structure assignment. In order to evaluate the accuracy of the β-sheet topology prediction algorithm, a map between predicted and actual β-strands is established. All strands which were seen to have a mapped partner are included in the evaluation of results. Based on the predicted secondary structure, the top solution is seen to have an average precision and recall of 62.4% and 61.7%, ...
### Table 3.2: Comparison of Average Precision and Recall values of top solutions of BeST and BetaPro

<table>
<thead>
<tr>
<th>Nstr</th>
<th>Prec</th>
<th>Recall</th>
<th>BetaPro Prec</th>
<th>BetaPro Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.78</td>
<td>0.78</td>
<td>0.71</td>
<td>0.71</td>
</tr>
<tr>
<td>4</td>
<td>0.71</td>
<td>0.72</td>
<td>0.66</td>
<td>0.65</td>
</tr>
<tr>
<td>5</td>
<td>0.67</td>
<td>0.66</td>
<td>0.62</td>
<td>0.62</td>
</tr>
<tr>
<td>6</td>
<td>0.65</td>
<td>0.64</td>
<td>0.60</td>
<td>0.61</td>
</tr>
<tr>
<td>7</td>
<td>0.59</td>
<td>0.58</td>
<td>0.59</td>
<td>0.58</td>
</tr>
<tr>
<td>8</td>
<td>0.58</td>
<td>0.54</td>
<td>0.57</td>
<td>0.53</td>
</tr>
<tr>
<td>9</td>
<td>0.58</td>
<td>0.57</td>
<td>0.55</td>
<td>0.55</td>
</tr>
<tr>
<td>10</td>
<td>0.59</td>
<td>0.59</td>
<td>0.59</td>
<td>0.61</td>
</tr>
<tr>
<td>11</td>
<td>0.49</td>
<td>0.5</td>
<td>0.47</td>
<td>0.51</td>
</tr>
<tr>
<td>12</td>
<td>0.53</td>
<td>0.56</td>
<td>0.53</td>
<td>0.6</td>
</tr>
<tr>
<td>13</td>
<td>0.5</td>
<td>0.5</td>
<td>0.51</td>
<td>0.55</td>
</tr>
<tr>
<td>14</td>
<td>0.56</td>
<td>0.58</td>
<td>0.59</td>
<td>0.66</td>
</tr>
<tr>
<td>15</td>
<td>0.55</td>
<td>0.61</td>
<td>0.55</td>
<td>0.61</td>
</tr>
<tr>
<td>16</td>
<td>0.52</td>
<td>0.53</td>
<td>0.51</td>
<td>0.56</td>
</tr>
<tr>
<td>17</td>
<td>0.53</td>
<td>0.55</td>
<td>0.58</td>
<td>0.62</td>
</tr>
<tr>
<td>18</td>
<td>0.55</td>
<td>0.53</td>
<td>0.47</td>
<td>0.53</td>
</tr>
<tr>
<td>19</td>
<td>0.47</td>
<td>0.47</td>
<td>0.47</td>
<td>0.47</td>
</tr>
<tr>
<td>20</td>
<td>0.55</td>
<td>0.57</td>
<td>0.54</td>
<td>0.64</td>
</tr>
<tr>
<td>21</td>
<td>0.69</td>
<td>0.64</td>
<td>0.68</td>
<td>0.75</td>
</tr>
<tr>
<td>22</td>
<td>0.16</td>
<td>0.17</td>
<td>0.29</td>
<td>0.33</td>
</tr>
<tr>
<td>23</td>
<td>0.44</td>
<td>0.42</td>
<td>0.55</td>
<td>0.58</td>
</tr>
<tr>
<td>24</td>
<td>0.38</td>
<td>0.42</td>
<td>0.38</td>
<td>0.45</td>
</tr>
<tr>
<td>25</td>
<td>0.5</td>
<td>0.54</td>
<td>0.47</td>
<td>0.6</td>
</tr>
<tr>
<td>26</td>
<td>0.77</td>
<td>0.85</td>
<td>0.72</td>
<td>0.9</td>
</tr>
<tr>
<td>27</td>
<td>0.81</td>
<td>0.87</td>
<td>0.77</td>
<td>0.9</td>
</tr>
<tr>
<td>28</td>
<td>0.8</td>
<td>0.84</td>
<td>0.73</td>
<td>0.9</td>
</tr>
<tr>
<td>30</td>
<td>0.67</td>
<td>0.7</td>
<td>0.66</td>
<td>0.83</td>
</tr>
<tr>
<td>33</td>
<td>0.42</td>
<td>0.44</td>
<td>0.42</td>
<td>0.44</td>
</tr>
<tr>
<td>35</td>
<td>0.7</td>
<td>0.73</td>
<td>0.68</td>
<td>0.88</td>
</tr>
<tr>
<td>42</td>
<td>0.42</td>
<td>0.44</td>
<td>0.42</td>
<td>0.44</td>
</tr>
</tbody>
</table>
Figure 3.7: Accuracy Results for proteins in the blind target test set from CASP9
Table 3.3: The distribution of the number of proteins in the blind target set of CASP9 with strands

<table>
<thead>
<tr>
<th>Strands</th>
<th>Number of Proteins</th>
<th>Strands</th>
<th>Number of Proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>12</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>22</td>
<td>2</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td>27</td>
<td>1</td>
<td>31</td>
<td>1</td>
</tr>
</tbody>
</table>

respectively. The best solution among the top five solutions predicted have precision and recall values of 72.8% and 71.3%, respectively.

3.6 Discussion and Conclusions

In this chapter, a novel integer linear optimization based framework for the prediction of β-sheet topologies in β and mixed α/β proteins has been presented. The algorithm uses a modified machine learning based contact potential to define the contacts between β-strands. Using a large set of physical, structural, steric and biological constraints, the predictions for any target protein are restricted. The integer linear optimization framework permits for the generation of a rank-ordered list of topologies for any target. Using a more detailed, atomistic force field and torsion angle dynamics, structural ensembles are generated for each predicted topology. These are eventually used to re-rank the predicted strand arrangements, thus providing a small subset of possible solutions. The algorithm was tested on a large, non-redundant data set of single chain proteins, and predicted sheet topologies with an average precision of above 77% in the top five solutions. The results were
further seen to hold up in consistency when the algorithm was tested on a set of proteins from the recently concluded CASP9 experiment.

The set of constraints that have been introduced are vital to elucidating biologically and structurally meaningful topologies for any given protein. A number of these constraints are based on literature study of existing \( \beta \) and mixed \( \alpha/\beta \) proteins, and can be explained on the basis of steric, entropic or energetic considerations. In particular, a significant improvement was seen in the prediction of non-local contacts. This was brought about in part by restricting the total number of local contacts, as well as the introduction of hierarchical constraints defining the possible superset of non-local contacts. This idea of co-operation between the set of strand contacts is synchronous with the idea of the zipping and assembly model of protein folding [151]. Dill and co-workers presented this approach to protein folding, wherein the presence of a given set of non-local contacts restricts the movement of the remainder of the chain, thus bringing other non-sequential parts of the primary sequence into spatial proximity [47].

One of the key advantages of the proposed approach is its ability to produce a rank-ordered list of \( \beta \)-sheet topologies for any target protein. Hence, instead of analyzing just the “best” solution, one would be able to analyze a small set of potential topological solutions. For blind target proteins where the \( \beta \)-sheet topology is unknown, the knowledge of the top set of solutions, in conjunction with human intervention, would be helpful in narrowing down the possible set of topology solutions drastically.
Chapter 4

Loop Structure Prediction

The structure prediction algorithms presented in the previous chapters address the ordered parts of a protein. In Chapter 2, the problem of prediction of the helical regions of the proteins was addressed. A number of methods have been presented in literature for the prediction of the location of $\beta$-strands [213]. Chapter 3 addressed the problem of predicting the arrangement of $\beta$-strands in space, thus defining the topology of the $\beta$-sheets formed in pure $\beta$ and mixed $\alpha/\beta$ proteins. The regions which lie between the ordered secondary structures constitute the loop regions of the protein.

Loops are seen to typically be shorter in lengths than the ordered secondary structure components of a protein, since the secondary structure elements and their interactions that gives a protein its final compact topology. However, despite this shortcoming, loops are extremely vital to proteins. The presence of loops permits the formation of the secondary structure topology of a protein. The amino acids in loop regions are seen to be vital to the formation of $\beta$-hairpins [73]. Further, loops are typically exposed on the surface of the proteins, thus making them directly accessible to the outside environment of the protein. Loops provide a means for sheltering the hydrophobic core of the protein from the external solvent, especially in globular proteins. Loops have also seen to be
key participants in active and binding sites on the protein [214]. Detailed reviews of loop structure prediction techniques have been presented elsewhere [57, 60, 62].

4.1 Challenges in Loop Structure Prediction

Since loops form the parts of the protein sequence that do not fall under any secondary structure, they possess greater randomness and structural flexibility than the secondary structure components of the protein. Most successful methods for the prediction of secondary structure regions in a protein have employed the explicit or implicit use of the protein data bank [20]. By observing and deriving patterns of sequential identity from previously existing proteins, successful predictions regarding the locations of secondary structure elements are made. The prediction of the arrangement of $\beta$-strands in a protein is also orderly, and follows a definite, albeit currently incomplete, set of rules. This permits the use of physical and database derived constraints which can be used to enhance the $\beta$-sheet topology prediction algorithm. The absence of an ordered structure in loops suggests that exclusively database driven techniques cannot be employed consistently for the prediction of loop regions in proteins. In addition, given their much increased exposure to the outside environment, loops are observed to have relatively fewer and inconsistent contacts with the remainder of the protein structure, thus making it significantly more challenging to predict their structure.

Given the flexibility associated with loop structures, the problem has previously been described as a mini ab initio protein folding problem [223]. However, a number of techniques used for the full protein structure prediction problem are rendered futile for the loop structure prediction problem. Given the very weak correlation between sequence and structure in loop structures [37], the comparative modeling techniques widely used in protein structure prediction become unsuitable. Fold recognition techniques are based on the idea that folds of proteins are conserved much more than sequence. However, loop structures do not have any observable patterns which can be categorized into the standard “folds”. Further, one of the main outstanding challenges in fold recognition al-
algorithms is the prediction of loop structures \[86, 116\], thus making the approach unsuitable for the prediction of the structure of loops. The consensus in the field of protein structure prediction has been to tackle the problem of loop structure prediction using pure or knowledge-based first principles approaches \[44, 138, 166\].

The final aim of loop structure prediction algorithms is vastly different to secondary structure prediction algorithms. For the secondary structure prediction stage, the target of any algorithm is the determination of the locations of the secondary structure elements. Similarly, in the $\beta$-sheet topology prediction stage, the target is to determine the best arrangement of known or predicted extended fragments of the protein. The main aim of the loop structure prediction stage is the determination of tight bounds on the backbone dihedral angles of the amino acids in the loop. The prediction of the exact structure of a loop is extremely challenging. Loops are the most flexible parts of proteins, and move constantly under interactions with the environment. Thus, the “exact” structure of a loop is very hard to determine, and what is observed in experimentally elucidated structures is a snapshot of the structure of the loop. For an optimization-driven approach towards predicting the final three-dimensional structure of a target protein, the provision of tight dihedral angle bounds on loop residues is much more beneficial to finding feasible initial structures than the provision of the “exact” structure of the loop.

### 4.1.1 Flexible stem and Fixed stem loop structure prediction

Most loop structure prediction algorithms can be broadly classified into flexible stem and fixed stem structure prediction algorithms based on the input to the algorithm. The fixed stem geometry problem assumes that the structure of flanking secondary structure elements to a given loop is known. Thus, the flanking secondary structure residues, or “stems” can be fixed to the experimentally determined structure, and the problem is narrowed down to one of determination of the structure of the intermediate loop. The flexible stem geometry problem does not assume the knowl-
edge of the structure of the flanking secondary structure elements. The only information available
to a flexible stem geometry problem is the identity of the type of secondary structures which flank
the loop, that is, $\alpha$-helix or $\beta$-strand. It can be seen that the flexible stem geometry problem is a
more challenging version of the loop structure prediction problem. Recent work has demonstrated
the differences in the challenges facing flexible stem and fixed stem loop structure prediction prob-
lems [178]. In this work, the authors perturbed 6-12 residues away from their crystal conformation
and placed all side chains in non-native, but low energy conformations. Even for such small per-
turbations, it was seen that the resulting regeneration problem was much more challenging than
the loop reconstruction problem. In this chapter, an approach to the flexible stem loop structure
prediction problem has been proposed.

4.2 Background

Most successful methods for loop structure prediction have targeted to follow a series of steps con-
sisting of improved dihedral angle sampling, removal of steric clashes, energy minimization and
clustering of predicted structures to identify the best representative structures from the predicted
ensemble. While the overall approach is not novel [27], major developments have been made in
each of the individual steps.

A large part of the recent success in dihedral angle sampling can be attributed to the improved
computational resources available, allowing finer discretizations of the dihedral angle space avail-
able to amino acids. Additional computational resources have also permitted the generation of a
larger number of initial structures, resulting in an improved coverage of the structural space. The
common theme of dihedral angle sampling in loop structure prediction techniques is the generation
of initial dihedral angles from a large database of known structures. Xiang et al. have applied this
process to a test set of 553 loops, ranging between five and twelve amino acids [223]. As the ap-
proach was developed for fixed-stem geometry problems, tighter dihedral angle distributions were
used to generate the initial loop structures. Similarly, de Pristo et al. generate 1000 initial structures by sampling backbone dihedral angles from a large database of dihedral angles generated from known loop structures, by using varying degrees of coarseness [44]. Based on their analysis, it was concluded that smaller number of samples generated from a finer distribution outperform the generation of a large number of initial structures from a coarser distribution. In a varied implementation of the previous algorithm, elimination criteria based on minimum number of occurrences in a bin were used to filter initial structures generated from a database-derived dihedral angle distribution [86]. Further, coarser discretization of the dihedral angle plane was used as a means to discard conformers that were too similar to ones previously generated. A number of additional criteria have also been used to discard initial structures generated from probability distributions. One approach towards improving initial structure generation in fixed-stem geometry problems has been to reject the structure if it becomes apparent that side-chain atoms cannot be fit, or if a closure of the loop to the fixed stem residues is not possible [223]. A recent approach has derived a pseudo potential to deduce the quality of an initial structure derived from a probabilistic database [117]. The authors use a pareto optimal searching (POS) method to span the search space of a large number of contact potentials, to derive a diverse initial conformational ensemble. Choi and Deane have used environment specific scoring parameters to improve the sampling for their loop structure prediction algorithm, FREAD [33]. By including parameters specific to the environment and flanking secondary structures, it was shown that the initial structures generated were much superior, both in terms of steric clashes and in terms of proximity of dihedral angles to the native structure. Initial structures generated have been shown to be directly correlated with the quality of the database they are derived from [43, 56, 135]. The quality of the database includes parameters such as the number of loop structures, pairwise sequence similarity of the database, variation in loop lengths and experimental method used to derive the native structure of the protein. Given the lack of correlation between loop sequence and structural similarity, it is still believed that the most successful initial structure generation algorithms are based on ab initio methods [112, 138].
A number of successful approaches presented in literature have combined database driven and *ab initio* initial structure generation procedures [206]. Recent methods have used energy based criteria to eliminate initial structures before carrying out local optimization [232]. The approach identifies and removes initial structures with low hydrophobic and hydrogen bonding interactions before implementing side chain and all atom optimization procedures to generate an ensemble of predicted loop structures.

A number of energy functions have been used for the structure optimization stage of loop structure prediction algorithms. These include database derived force fields, as well as first principles based energy functions. Most first principles based energy functions are modified forms of the AMBER [39], CHARMM [124] or ECEPP/3 [145] force fields, with parameters modified to target the loop structure prediction problem. In addition, knowledge derived statistical potentials have been utilized for the problem of fixed stem loop structure prediction [33, 89]. Terms representing energetic contributions due to solvation, steric clashes, hydrogen bonding and short and long range contacts have been seen to be included in knowledge derived force fields. Hierarchical methods, which incorporate all-atom physical potentials with explicit and implicit solvent models, have been presented for the case of fixed stem loop structure prediction [86]. The algorithm applies a multiscale approach, starting from a coarse model which explicitly incorporates crystal packing, and refines the initial structures using all atom potentials. Other methods have incorporated corrective terms to account for discrepancies in the hydrophobic expressions found in all atom potentials [232].

The structure optimization stage of loop structure prediction algorithms requires the energy minimization of the initial structures previously derived. A vast variety of methods have been used to efficiently navigate the tertiary structure space of a target loop. The optimization algorithms that have been employed for the purposes of loop structure prediction include molecular dynamics [26], simulated annealing [78], torsion angle mechanics [55, 189] and nonlinear optimization [138]. In addition, a number of side chain optimization algorithms have been introduced to alleviate steric
clashes between the randomly generated side chain and backbone of the initial loop structures.

Most side chain optimization algorithms use a large database of known side chain angles [138]. A combination of side chain dihedral angles for any amino acid is known as a rotational isomer, or rotamer, of the residue. A number of rotamer libraries have been presented, which document all observed combinations of side chain dihedral angles. Two of these rotamer libraries have been used in the loop structure prediction algorithm presented in this chapter. In addition, methods have been presented in literature which solve the side chain optimization problem using detailed atomistic or knowledge based potentials, by incorporating additional effects like ionization and solvation [233]. Here, the dielectric constant of interaction between side chain atoms is allowed to vary as a function of the interacting residues to account for these effects.

The next section presents the derivation of dihedral angle propensities for the purposes of generation of initial structures. This is followed by a description of the generation of initial loop structures, combined with checks to ensure uniqueness of the generated structures. Three rotamer optimization steps are presented, which alleviate local steric clashes between the side chains and the backbone generated. This is followed by a description of the constrained non-linear optimization stage for loop structure prediction, and an overview of a traveling salesman based clustering algorithm for the identification of tighter bounds on the dihedral angles of loop residues.

### 4.3 Methods

#### 4.3.1 Derivation of Dihedral Angle Propensities

The generation of initial structures is a crucial step when local optimization techniques are employed. For the algorithm presented in this chapter, a large repository of structures from the PDB-Select25 data set was used as a library for generation of loop angle probability distributions. This data set contains 4092 single chain proteins, with pairwise sequence similarity below 25%. We
collect loop segments between the lengths of 4 and 20 from this database, as longer lengths provide a very sparse distribution of amino acid dihedral angles. For each amino acid, we discretize the Ramachandran plot into a grid of size $10^\circ \times 10^\circ$. Based on the database of collected loop segments, we count the frequency of backbone dihedral angle occurrences for each amino acid in each dihedral angle bin. A similar distribution is generated for each kind of loop, that is, separate distributions are generated from loops between helices, strands and any combination thereof. An example of the difference in dihedral angle distributions generated is shown in Fig 4.1. For any target loop, a set of 2000 initial structures are generated using these probability distributions. The process of generating initial structures is described as follows.

Figure 4.1: Illustrative example of variation in distribution of Loop Residue Angles
4.3.2 Generation of Initial Structures

For each amino acid in each type of loop, each discretized bin in the Ramachandran plot is assigned a number $n_i$ that corresponds to the frequency of dihedral angle occurrences observed. Hence the first, second and in general $i^{th}$ bin can be represented by the numbers:

\begin{align}
    b_1 : & 1, \ldots, n_1 \\
    b_2 : & (n_1 + 1), \ldots, (n_1 + n_2) \\
    b_i : & i-1 \sum_{j=1}^{i-1} n_j, \ldots, n_i + i-1 \sum_{j=1}^{i-1} n_j
\end{align}

(4.1)

By generating a random number between 1 and $\sum_j n_j$, and identifying the bin it corresponds to, we can assign a backbone angle to an amino acid. However, since the initial dihedral angles of each amino acid are generated by unique distributions, possibilities of backbone steric clashes in the generated structure are quite high. In order to alleviate backbone steric clashes, we re-sample pairs of amino acids which are identified as having clashing backbones.

Uniqueness of Initial Structures

In order to ensure that initial structures generated are not very similar to each other, any new structure is required to be unique to each of its predecessors. Uniqueness is defined by the following equation

$$
\sum_{i=1}^{N_{dih}} ((\phi_{i,k} - \phi_{i,j}) + (\psi_{i,k} - \psi_{i,j})) \geq 5 \quad \forall j < k.
$$

(4.2)

where $\phi_{i,k}$ and $\psi_{i,k}$ refer to the backbone dihedral angles of amino acid $i$ of loop conformer $k$. Here $N_{dih}$ represent all the dihedral angles of the loop. The index $j$ runs over all loop conformers previously generated, while the index $k$ refers to the new loop structure generated from the probability distribution. The equation requires that when comparing the new loop structure to
its predecessors, at least one dihedral angle is found in a bin different to the previously generated structure.

4.3.3 Rotamer Optimization

Rotamer optimization is seen to be an important intermediate step in the loop structure prediction framework. Given an initial loop backbone, there is a very high likelihood that strong steric clashes exist between side chains of loop residues and the backbone of the loop. The objective of introducing a rotamer optimization step is to identify a better starting point for the full atom local optimization of the loop structure. It is also crucial to note that the rotamer optimization step is an intermediate stage used for steric clash removal only. Hence, a fast rotamer optimization algorithm essentially behaves as an efficient local minimization step.

Most successful rotamer optimization algorithms, especially for loop structure prediction, use rotamer libraries to carry out local energy minimization. Rotamer libraries consist of combinations of side chain angles observed in the database. The computational time required for a rotamer optimization algorithms depends on two factors: the energy function being considered and the size of the rotamer library being employed [45, 222]. Hence, the aim is to devise rotamer optimization algorithms which use an energy function resembling an atomic force field and a search method that is exhaustive, while ensuring that the algorithm does not become computationally prohibitive.

Since the primary objective of the rotamer optimization step is the removal of steric clashes between the side chains, the use of an all atom force field may not be judicious. Further, the implementation of the rotamer optimization step is targeted towards removing steric clashes in the initial structures, without guarantees that the resulting side chains would remain fixed or feasible after the complete structure optimization. The use of an approximate energy function is the best tradeoff for the purposes of rotamer optimization. Most combinatorial rotamer optimization algorithms divide the energy of a conformation into two parts, given by the following equation:
\[
\min \sum_{(i,k) \in R_i} E_{ik}^{self} + \sum_{(i,k) \in R_i} \sum_{(j,l) \in R_j} E_{ijkl}^{pair}.
\] (4.3)

Here \(E_{ik}^{self}\) represents the contribution of the interaction of a rotamer with all fixed atoms during the rotamer optimization. In other words, the energy of interaction of movable atoms of the side chain of an amino acid \(i\) with all immovable atoms, including backbone atoms and \(C^\beta\) atoms, is expressed by this term. The second term \((E_{ijkl}^{pair})\) represents the pair energy, and is a representation of the energy of interaction between rotamer \(k\) of residue \(i\) and rotamer \(l\) of residue \(j\).

The efficiency of any rotamer optimization algorithm depends on the energy function, and can hence be significantly improved by using an approximate energy function. The chosen energy function should closely resemble the all atom energy function it is derived from, while being computationally inexpensive. The energy function used in the rotamer optimization algorithms presented in this chapter is a piecewise linear approximation of the repulsive part of the Lennard Jones and hydrogen bonding potential terms in the ECEPP/3 force field. The rationale for choosing these terms comes from the main aim of the rotamer optimization step, i.e. the alleviation of steric clashes. The repulsion terms are approximated by piece-wise linear functions that intersect the original expression at 2, 5, 10, 20, 50 and 100 kcal/mol. All energetic contributions above 100 kcal/mol are approximated by the last piecewise expression, while all energetic contributions less than 2 kcal/mol are ignored.

In the loop structure prediction algorithm presented in this chapter, we have incorporated three rotamer optimization algorithms. The algorithmic details of the rotamer optimization steps have been presented previously [131]. Here, the overview of the algorithms, along with changes made to the algorithms have been presented.
Rotamer Optimization: FASTER

The first rotamer optimization algorithm, known as FASTER [45], has been shown to produce nearly identical results to the global optimization dead end elimination algorithm, while being nearly 100-1000 times faster than it. The key steps of this rotamer optimization algorithm are:

1. Insert all backbone and $C_\beta$ atoms onto a fixed grid. All backbone and $C_\beta$ atoms are assumed to be immovable during the rotamer optimization step. This is to ensure that the rotamer optimization focuses only on identification of rotamer combinations which minimize the energy for the initial backbone generated using the probability distributions.

2. Load rotamers of each loop amino acid from the Penultimate library [51].

3. Pre-compute the self energy energy terms for each possible rotamer of each amino acid in the loop. In order to do this, we take each possible rotamer of any given amino acid, and evaluate the energy of the loop when this rotamer is used. This is possible since the self energy term is evaluated against all non-movable atoms of the loop. Similarly, we pre-compute the pair energies of combinations of rotamers of all amino acids in the loop. If the distance of a pair of rotamers is above a threshold, this computation is ignored.

4. Starting from the first amino acid of the loop, we iterate over all amino acids of the loop. For any amino acid, we iterate over all the possible rotamers of the amino acid. If the sum of the self and pair energy for any rotamer results in a total energy that is better than the current energy, we replace this structure to be the current active structure. This structure is referred to as the Backbone determined minimum structure (BMEC). For each of the subsequent steps presented in this algorithm, the overview of the step is presented, and the reader is referred to the original work for details [45].

5. The first pass of the FASTER algorithm, called iterative batch relaxation (iBR), is split into 3 main steps. First, the total energy for all of the rotamers in each residue for the current...
configuration $i$ is evaluated and stored. Next, the rotamer position $k$ in each residue $i$ that yields the minimum energy for the current loop configuration is saved. Finally, the total energy of the new conformation is calculated. This procedure is carried out until the energy of the loop configuration stabilizes or starts oscillating.

6. The next phase of FASTER is called the conditional iterative batch relaxation (ciBR). This step is similar to the previous step, except that the rotamer positions with lower energy are only accepted with an 80\% probability. 10 iterations of 10 optimization cycles each are performed and the lowest energy conformation is retained.

7. The final phase is the single-residue perturbation/relaxation (sPR) phase. This is a final iteration of the iBR phase, carried out by fixing the rotamer of one amino acid at a time and iterating over the remaining residues of the loop.

**Rotamer Optimization: Cyclical Search Algorithm**

The second rotamer optimization algorithm is a cyclic search method, and uses the enhanced rotamer library from Xiang and Honig [223]. The main steps of the algorithm are presented below:

1. Insert all backbone and $C_\beta$ atoms onto a fixed grid. All backbone and $C_\beta$ atoms are assumed to be immovable during the rotamer optimization step.

2. Load rotamers of each loop amino acid from the Xiang and Honig library [223].

3. Randomize the order in which amino acids would be visited by the algorithm. For each amino acid in this randomized list, we carry out the following steps.

4. Randomly rearrange the order of the residues to be visited by the rotamer optimization algorithm. For each residue $i$, the energy of the each rotamer $k$ is evaluated using the approximate energy function. This includes the intra-chain and inter-chain interactions. In addition, the
total ECEPP/3 energy of the original rotamer of the amino acid $i$ is evaluated. If the approximate energy value of the new rotamer is within a cutoff value of the true ECEPP/3 energy of the original rotamer, the total ECEPP/3 energy of the new rotamer is evaluated. If this new energy is lower than the previous existing rotamer for this amino acid, we replace the rotamer of amino acid $i$ with this new rotamer $k$.

5. This procedure is repeated until all amino acids of the loop have been addressed.

**Rotamer Optimization: Random Rotamer Search**

For the third phase of the rotamer optimization stage, we employ a procedure similar to the cyclic search algorithm presented in the previous section. However, instead of using the rotamer library by Xiang and Honig, we choose to restrict the rotamer search to the local neighborhood of the rotamers identified at this stage. It is assumed that the previous algorithms would bring the rotamers close to the local minima for the given fixed backbone. Hence, we create a rotamer library using a narrow Gaussian distribution around the current available rotamer. The mean of this distribution is the current rotamer itself, while a standard deviation of $10^\circ$ is used. The aim of this step is to provide additional refinement to the rotamers in the neighborhood of the recorded value in the library. The algorithmic steps are outlined below.

1. Insert all backbone and $C_\beta$ atoms onto a fixed grid. All backbone and $C_\beta$ atoms are assumed to be immovable during the rotamer optimization step.

2. Load rotamers of each loop amino acid from a library of 50 rotamers created around the current existing rotamer, as explained previously.

3. Randomize the order in which amino acids would be visited by the algorithm.

4. For each amino acid visited, carry out the cyclical search algorithm presented in the previous section.
The advantage of using the three step procedure to rotamer optimization has been presented in literature [131]. Here, the authors demonstrate a 35%-65% improvement in the energy of a structure at the end of the rotamer optimization steps.

4.3.4 Energy Minimization

Once the rotamer optimization step has been carried out, the loop structures are subjected to all atom energy minimization. The energy function used for this purpose is the all-atom ECEPP/3 potential, given by the expression

\[
E_{ECEPP/3} = \sum_{(i,j) \in ES} \frac{q_i q_j}{r_{ij}} + \sum_{(i,j) \in NB} F_{ij} \frac{A_{ij}}{r_{ij}^{12}} - \frac{C_{ij}}{r_{ij}^6} + \sum_{(i,j) \in HB} \left( \frac{A'_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{10}} + \frac{E_{0,k}}{2} \left(1 + c_k \cos n_k \theta_k \right) \right) \tag{4.4}
\]

In Equation 4.4, \( r_{ij} \) represents the distance between a pair of atoms \( i \) and \( j \), given that both the atoms fall into the set of atoms over which the summation is carried out. The parameter \( F_{ij} \), which represents the relative impact of the repulsive part of the Lennard-Jones expression, is taken as 0.5 for \( 1-4 \) interactions, and 1.0 for \( 1-5 \) interactions. Non-bonding parameters such as \( A_{ij}, A'_{ij}, B_{ij} \) and \( C_{ij} \) are atom pair dependent. The sets \( ES, NB \) and \( HB \) are defined over the set of pairs of atoms \( i \) and \( j \) that can have electrostatic, non-bonded and hydrogen bonding interactions, respectively. The set \( TOR \) runs over all torsion angles of the protein that can contribute to the last term of the expression.

The aforementioned problem can be represented as a constrained nonlinear programming problem. The constraints to the model are the backbone dihedral angle bounds. These bounds are refined at each stage, as is described in the next section. While most implementations of constrained nonlinear optimization are similar, some packages provide additional advantages to certain appli-
ations like the protein folding problem. One such implementation of the Sequential Quadratic Programming (SQP) method is the NPSOL package [69]. The package is attractive for protein structure prediction problems because it requires fewer evaluations of the objective function, which is computationally expensive.

4.3.5 Clustering

The main challenge behind the clustering step is the identification of a subset of predicted structures, which can be considered representative of the better structures of the ensemble. In the loop structure prediction framework, we address this challenge by implementing an iterative novel Traveling Salesman Problem (TSP) based clustering approach, known as ICON [198]. By considering each conformer generated from the ASTRO-FOLD 2.0 framework as a node on a traveling salesman path, we identify the globally optimal path through each of these nodes. Once the optimal path is determined, this path is partitioned into clusters such that the clusters minimize the global sum of intra-cluster differences in values. An overview of each of the steps of ICON is presented below, and the details can be found in Chapter 5 and literature [49, 198].

With each conformer of the target loop as a node on the TSP path, we define binary variables $y_{i,i'}$ for any pair of nodes $i$ and $i'$ as:

$$
y_{i,i'} = \begin{cases} 
1 & : \text{if node } i' \text{ immediately precedes node } i \\
0 & : \text{otherwise}
\end{cases}
$$

(4.5)

The objective function is then defined as [48]:

$$
\min \sum_i \sum_{i'} y_{i,i'} \phi_{m_i,m_{i'}}
$$

(4.6)
where $\phi_{m_{i,j}, m'_{i,j}}$ is given by:

$$
\phi(m_i, m'_i) = \sum_j \min(m_{i,j} - m'_{i,j}, 360 - (m_{i,j} - m'_{i,j}))^2
$$

(4.7)

Here, the index $j$ runs over all the pairs of $(\phi, \psi)$ angles of each amino acid of the target loop. Constraints which ensure that each node has exactly one node preceding and following it on the TSP path are implemented. In addition, efficient TSP solvers like Concorde [10] introduce additional cuts which eliminate circular tours and subtours. Once the optimal path through all conformers is determined, we propose an integer linear programming (ILP) model to determine the cluster boundaries for a given optimal ordering [49]. Since for any node on the TSP path, we know the immediate neighbors on the path, the aim is to simply determine the points on the TSP path where immediate neighbors on the path fall into separate clusters. This would be sufficient to identify the boundaries of clusters. In order to do this, we generate a distribution of $\phi_{i,i+1}$ (where $\phi_{i,i+1}$ are defined as in Equation 4.7). For any local window of $x$ elements, we identify nodes where the neighbor distance falls below one standard deviation of the global average of this distribution. In addition, this distance would be the minimum in its local window, so as to ensure that we do not separate out elements that are very similar. By selecting local minima points of this distribution as cluster “seeds”, we now have the problem of placing the remaining “outlier” points with the cluster seed element immediately before or after them in the optimal TSP path. This has been modeled as an integer linear programming (ILP) model, with binary variables assigning the outlier points to either the cluster before or after them. The objective function includes terms which account for the fixed cost (distance between an outlier and the seed of the cluster) and variable cost (distance between two outliers both assigned to the same cluster seed). Constraints are introduced to ensure that there are no crossovers, i.e. for any pair of outliers $i, i+1$, the assigned cluster of element $i+1$ should be greater than or the same as that of element $i$. Details of the mathematical implementation of the model can be found elsewhere [49]. Subsequently, the cluster centroids for each cluster are
identified by determining the cluster element with the minimum distance to all other elements of the cluster, with the distance being defined again as in Equation 4.7. Following this, we eliminate loosely bound clusters by analyzing cluster densities. All clusters with cluster densities greater than the median value are retained for future iterations. At the end of 10 iterations or when left with half the initial number of conformers, we re-rank the final list of cluster centroids using high resolution distance dependent force fields [156, 157]. The lowest energy structures are identified as the structures nearest to the native.

### 4.3.6 Generation of improved bounds and iterative approach

Using the loop decoys in the top 10 clusters sorted by cluster density, we develop new backbone dihedral angle bounds for each amino acid in the loop. These new bounds replace previously existing bounds only if they are tighter. If this is not the case, we continue with the old bounds for the next iteration. Using the existing probability distribution, we re-generate initial structures for the next iteration of the loop structure prediction algorithm. However, if the initial backbone dihedral angles do not lie within the updated bounds, the value is rejected and the angle is re-generated from the probability distribution. The entire procedure consisting of rotamer optimization, all-atom energy minimization and clustering is repeated for five iterations.

At the end of the five iterations, the structures of the top 10 clusters sorted by cluster density are used to generate the final set of bounds that would be useful in the tertiary structure prediction of the protein, presented in Chapter 6.

### 4.4 Results and Discussion

The loop structure prediction algorithm was tested on a large number of loops, ranging from five to fourteen amino acids in length. The distribution of the number of loops for each loop length is given in Figure 4.2. Three amino acids on either side of the loop were taken as the stem residues.
As has been described previously, information regarding the type of secondary structure of the stem residues was available, while their structure was unknown to the algorithm. Figure 4.2 also shows the fraction of loops that were found in each kind of neighborhood, i.e. helices on both sides, strands on both sides or any combination thereof. As shown in the figure, loops belonging to all four classes have been included in the test set to avoid bias to any specific type of loop.

Figure 4.2: Distribution of number and type of loops in the Loop Test Set

Figure 4.3 shows the distribution of the best predicted loop structure against the number of amino acids in the loop. The best loop structure is defined to be the one with the lowest root mean squared deviation (RMSD) to the native structure. For each loop, the RMSD of the best structure in the predicted ensemble was evaluated, and the RMSD values were averaged across all loops of the same length.

Figure 4.3 also shows the average RMSD distribution following the exclusion of the three stem residues attached to each loop. As shown in the figure, the algorithm achieves an average best
structure RMSD of 0.63 Å for five residue loops (1.10 Å when including the stem residues). For loops as long as 14 residues, an average best structure RMSD of 2.12 Å without the stem residues, and 2.61 Å when they are included is observed. It is noteworthy that when the stem residues are included, a 14 residue loop structure prediction problem is equivalent to the ab initio prediction of the structure of a twenty amino acid peptide, with the knowledge of the type of secondary structure for three amino acids at either terminus.

Two additional observations can be made based on the results shown in Figure 4.3. First, we see an almost consistent separation between the two lines representing the average best RMSD values with and without the stem residues, respectively. This suggests that the contribution of the stem residues to the RMSD is almost consistent across all loops. The stem residues of the loop are the only regions which are ordered, that is, have a secondary structure pattern associated. The dihedral angle bounds imposed on the stem residues are therefore much more stringent than their
counterparts on the loop residues. Hence, the contribution of the stem residues to the RMSD would be expected to be lower and be consistent across different lengths of loops considered. Second, we see an almost linear growth in the average best RMSD with respect to the loop length. The RMSD data for the entire loop can be fit to the line given by the equation

\[
RMSD = 0.3026 + 0.1684 \ast n_r
\] (4.8)

where \( n_r \) represent the number of residues in the loop region only. The rate of growth of average best RMSD to the number of residues in the loop can be fit to a straight line. With increasing number of amino acids in the loop, the tertiary structure space increases exponentially. However, a combination of iteratively improving dihedral angle sampling, rotamer optimization, all atom energy minimization and near-native structure identification ensures that the RMSD of the best structure of the predicted ensemble continues to grow linearly.

As discussed previously, the loop structure prediction algorithm is used to predict tight dihedral angle bounds on the backbone angles of the loop residues. Hence, while the prediction of low RMSD structures in the ensemble is encouraging, it is important that bounds generated on the backbone dihedral angles are as tight as possible. Figure 4.4 represents the width of the bounding box represented by the bounds on the backbone dihedral angles of the loop residues. The bounding box is defined as the difference between the upper and lower bounds on the dihedral angles of a loop.

As shown in Figure 4.4, the bounds predicted on the \( \phi \) backbone dihedral angle are much tighter than the ones predicted on the \( \psi \) backbone angle. It has frequently been observed that the variation in \( \phi \) is much smaller than the variation in the \( \psi \) dihedral angle, given that the \( \phi \) values for the \( \alpha \)-helical and \( \beta \)-strand residues are much closer than their corresponding values of the \( \psi \) dihedral angle [161]. For the \( \phi \) dihedral angle, the lowest average bounding box width is 48° for loops of length six, and largest of 72° for loops of length eleven. For the \( \psi \) dihedral angle,
corresponding values of $74^\circ$ for loops of length five, and $133^\circ$ for loops of length fourteen are observed. The bounding box width for both dihedral angles are seen to increase monotonically with loop length. With larger number of residues in a loop, the predicted loop structures are seen to span a wider range of RMSD values. The structures in the top clusters, which are used for the prediction of dihedral angle bounds tend to diverge, resulting in wider bounding box constraints on the dihedral angles. It is noteworthy that the bounding box range plateaus very early with the increase in number of amino acids in the loop. Beyond loops of length eleven, the average bounding box width for the backbone dihedral angles are seen to be almost constant. Given that the three dimensional search space expands significantly with increasing number of amino acids, the dihedral angle sampling, all atom optimization and clustering procedures in the loop structure prediction are seen to be successful in providing tight backbone dihedral angle bounds on all the
residues of longer loops as well. Further, an average accuracy of 86.14% was seen for the bounding boxes of all amino acids in all loops. The accuracy of the bounding box was defined by:

\[
\text{Accuracy} = \frac{\text{Number of dihedral angles within bounding region}}{\text{Total number of dihedral angles}}.
\]

An analysis of the amino acids with erroneous predicted bounds shows that the average error in the prediction of the \( \phi \) and \( \psi \) angles were 25.6\(^\circ\) and 11.9\(^\circ\), respectively. The error in the prediction of \( \psi \) angles is seen to be smaller, given that the average bounding box width is much larger. The error in prediction was calculated by evaluating the difference between the true dihedral angle and the closest predicted dihedral angle bound. Therefore, while the predicted bounds for more than 86% of loop amino acids include the true dihedral angle, the average error for the remaining residues are seen to be significantly smaller than the size of the Ramachandran plot itself.

### 4.4.1 Selected CASP9 targets

The loop structure prediction algorithm was applied to targets provided during the recently concluded CASP9 event. In order to define the loop regions of the protein, secondary structure prediction was first carried out to determine the regions of secondary structure in the protein. A consensus between the helix prediction method, presented in Chapter 2, and CONCORD [212], was used to determine the locations of \( \alpha \)-helices and \( \beta \)-strands for any target protein.

Figure 4.5 shows the variation in the accuracy of the bounding box for the backbone dihedral angles of the selected CASP9 targets. As shown in Figure 4.5, the weighted average \( \phi \) and \( \psi \) accuracies were seen to be 75.9% and 73.8% respectively. The accuracy of the bounding boxes are seen to vary between 55% for T576 and 96.4% for T600.

No correlation was observed between the length of the target protein and the accuracy of the bounding boxes. Other external factors were seen to affect the quality of bounding box predictions for the CASP9 targets. Figure 4.6 shows the variation of the bounding box accuracy with the
accuracy in secondary structure prediction. As was described in Chapter 2, the secondary structure accuracy is measured by the Q3 parameter, which evaluates the fraction of amino acids correctly assigned to one of three classes of secondary structure (helix, strand and coil).

It is noteworthy that a positive correlation value of 0.6543 and 0.5291 (for $\phi$ and $\psi$ respectively) was observed between the bounding box accuracy and the accuracy of the secondary structure prediction. The accuracy of the secondary structure prediction is a reflection of the length of loop, the type of loop (i.e., the secondary structure elements on either side of the loop) and the residue types of the stem residues at the ends of the helix. Each of these parameters can affect the loop structure prediction process significantly. As discussed previously, the three dimensional search space expands exponentially with increasing amino acids in the loop sequence. The determination of the true length of the loop is crucial towards the loop structure prediction process. Similarly, in the description of the model, it was shown that separate probability distributions were created
for loops depending on the secondary structure elements they are found between. Given that the
distributions vary significantly, the initial structures, and therefore, the final predicted structures
would be affected by accurate identification of the type of loop. Finally, the nature of the stem
residues is vital since the side chain atoms of the stem residues affect the rotamer optimization of
the loop residues. As the rotamer optimization is only carried out on the loop residues, the inclusion
or exclusion of an amino acid in the loop can affect the output of the rotamer optimization stage.

A separate study on the accuracy of the dihedral angle bounding boxes is presented in Figure 4.7. The figure shows the variation of the bounding box accuracy with the J-Score of target
proteins.

J-Score is a metric of the similarity of a target protein to the Protein Data Bank [20]. The metric
is evaluated by the 3-D Jury server [70], using a multiple sequence alignment approach involving
BLAST and PSI-BLAST [6]. As shown in Figure 4.7 a correlation study between the accuracy
of the predicted loop dihedral angle bounding boxes and the J-Score of the target protein shows a very small correlation for both the $\phi$ and $\psi$ backbone dihedral angles. A low correlation of the bounding box accuracy with the J-score of a protein indicates a low degree of dependence on the sequence similarity between a target protein and the protein data bank. Even though the initial structures are generated from a database derived probability distribution, the rotamer optimization and all atom physical potential based nonlinear optimization procedure ensure that the similarity of a target loop to the database has a minimal impact on the quality of prediction. Further, a lack of correlation between the J-score and bounding box accuracy is understandable given that the initial structures are not generated by using the loop structures of the top hits from the PSI-BLAST search. Homology modeling methods have been seen to have limited success in loop structure prediction, owing to the highly flexible and varied structures of loops. The model presented in this
chapter is hence able to avoid the pitfalls of database driven loop structure prediction, even with the use of a probability distribution driven initial structure sampling procedure.

4.5 Conclusions

In this chapter, a novel iterative algorithm for loop structure prediction with flexible stems was introduced. The algorithm only requires the knowledge of the sequence and the secondary structure type of the three stem residues at each end of the loop. Loop structure prediction is a critical intermediate step towards the tertiary structure prediction of proteins, as it provides tight dihedral angle bounds on the backbone dihedral angles of the residues in the loop regions of proteins. The algorithm employs an initial structure generation procedure based on a derived probability distribution. Three rotamer optimization procedures are incorporated to alleviate steric clashes between rotamers of the amino acids and the generated backbone of the loop. A full atom physics based energy function, ECEPP/3, is used to carry out nonlinear constrained optimization to collect predicted structures of the loop. A traveling salesman based clustering algorithm, ICON, is used to identify a subset of representative structures which are used to develop tight bounds on the backbone dihedral angles of the residues in the loop. The algorithm was applied on two data sets: a large number of loops from the PDBSelect25 data set, and loop regions of blind target proteins provided during the recently concluded CASP9 community-wide experiment. The algorithm was seen to predict low resolution structures for a large number of loops, and was able to derive tight dihedral angle bounds for amino acids in the loops. Further, a low degree of correlation was seen between the quality of the dihedral angle bounds and the similarity of a target protein to the Protein Data Bank.
Chapter 5

Near-native structure identification

Protein structure prediction encompasses two major challenges: (1) the generation of a large ensemble of high resolution structures for a given amino acid sequence and (2) the identification of the structure closest to the native structure for a blind prediction. In this chapter, the second of these challenges is addressed. Any protein structure prediction approach typically provides a large ensemble of predicted structures. In the absence of knowledge of the native structure itself, it is hard to consistently identify the structure closest to the native from the given ensemble.

5.1 Introduction

Most structure prediction algorithms base themselves on Anfinsen’s hypothesis [9]. According to the hypothesis, the native structure of a protein would lie at the global free energy minimum of the system. While this hypothesis is widely accepted, this does not help identify the structure nearest to the native structure from a given ensemble. This is because, while the native structure itself may lie at the global free energy minimum, there is no assurance of monotonicity between the nearness of a structure to the native, and its free energy. Since it is almost improbable to find the exact
native structure of a protein, there is no assurance that the use of an energy based criteria would help identify the structure closest to the native.

A number of techniques, spanning a wide variety of fields, have been used for the identification of near native folds. These can be broadly classified as force field based techniques and clustering techniques. Force field based techniques aim at capturing the energetic interactions that occur in proteins either through physics based energy functions, or through knowledge based potentials. CHARMM [124], AMBER [39], ECEPP [137], ECEPP/3 [145], UNRES [121] and ECEPP-05 [11] are examples of some physics based potentials. A significant amount of research has been dedicated towards optimizing the weight parameters of the physics based force fields, in order to increase the correlation between the potential energy of the protein and the nearness to the native fold [12, 219]. Knowledge based force fields are usually calculated using two different approaches. One approach uses the Boltzmann equation, which is based on the idea that lower energy states are more frequently observed. The second approach is based on parameter estimation, which aims to represent the amino acids of a protein either as a single atom, or as a group of atoms. Parameters are estimated either based on distance between specific atom pairs or on the identity of amino acids, which are trained to ensure that the native structure has an energy value much lower than the decoy structures. Distance based force fields, based on C$_\alpha$-C$_\alpha$, Centroid-Centroid [134, 156, 157, 205, 207] or all atoms [123, 155], have been shown to be successful in identifying the native structure from a large ensemble of near-native structures.

A second approach to the identification of near native folds is clustering. Problems of data clustering and organization are pervasive over a number of disciplines. The most common approaches can be classified as either hierarchical [52] or partitioning [77] clustering. A number of other frameworks for clustering have also been proposed, including model-based clustering [217], neural networks [87], simulated annealing [99], genetic algorithms [21], and data classification [28]. Most algorithms use heuristics for their searching procedures, which may result in suboptimal clustering because of analysis of only local comparisons. Recent works have presented a
novel clustering approach based on global optimum search [61], which includes a procedure to
determine the optimal number of clusters to be used [202, 203, 204]. Clustering methods have
been previously also used as a part of loop structure prediction algorithms [138], where the aim
is to eliminate loop structures which are unlikely to be close to the native structure in an iterative
manner.

The field of rearrangement clustering has emerged as a very effective technique for \textit{optimally}
minimizing the dissimilarity metric between the data points in large distance matrices. Recently,a rigorous global optimization method for biclustering biological data was introduced [48]. This
method, denoted as OREO, is based on optimal re-ordering of the rows and columns of a data
matrix to globally minimize the dissimilarity metric, by formulating the physical permutations
of rows and columns as either a network flow problem or the traveling salesman problem [10].
Highly favorable results were presented when the method was tested on several sets of biological
and image reconstruction data.

A number of methods have been proposed, which cluster the decoys of a protein based on
some mutual distance metric, and then aim to find the structure which has the highest number
of similar structures. Shortle \textit{et al.} [185] proposed a pairwise RMSD based clustering method
to this effect. The authors presented a scoring function to rank the quality of the decoys. This
scoring function aims to predict the probability that the given sequence would fold into the decoy
structure. The prior probability of the structure is derived from excluded volume and packing
terms. The likelihood term in the Bayesian expression is derived from hydrophobic and pairwise
interactions such as salt bridges and disulfide bonds. Based on such an elimination criterion, the
method reduces the working ensemble to the 1000 best scoring structures. Based on a (1000 X
1000) pairwise RMSD matrix, the structure with the most "neighboring" structures is selected.
SPICKER [229], a state of the art simple and efficient method to identify near-native folds, also
follows a similar idea. This method takes into consideration the fact that depending on whether
it is a new fold, or an existing one, most structure prediction techniques are likely to produce a
wide or narrow ensemble of structures, respectively. Hence, SPICKER modifies the “radius of cut off” for the definition of a cluster. The top 5 clusters in terms of size are selected, and the cluster centroids and medoids are suggested as the structures closest to the native.

The idea of dihedral angle based clustering of protein structures was also investigated by researchers. Dihedral angles provide a good representation of the protein structure itself, since in the dihedral angle space, we can assume two degrees of freedom for each amino acid. Circular clustering is the most effective way of handling dihedral angles [50]. The idea of circular clustering is to identify the fact that for a dihedral angle, $+180^\circ$ and $-180^\circ$ are the same. Hence, objective functions defining the dissimilarity metric should reflect this property.

A potential source of error in structure prediction algorithms comes from the mis-prediction of the topology of the target protein. This is especially a problem for structure prediction using homology based algorithms, which rely on sequence and structural homologs found by meta-servers. In such cases, any clustering method which uses the predicted structural ensemble as the starting point, without any prior knowledge of the native structure, is likely to concentrate, and hence predict, an incorrect decoy structure as the one most likely to be near-native. Situations such as these are especially detrimental to iterative algorithms, as they rely on the assumption that the previous stages of the algorithm would have initiated the search technique in the correct set of directions. For an incorrect topology based ensemble, the concept of correct set of directions fails to hold significant meaning, and can hence end up driving the algorithm towards a poor prediction.

In this chapter, an Iterative Clustering Approach for Optimal selection of near-Native structures, (ICON), is presented. We use the idea of rigorous global rearrangement clustering as presented by DiMaggio et al. (2008)[48] to cluster ensembles of protein structures in a blind case manner, that is, without the knowledge of the native structure. An objective function that reflects the dihedral angle properties is introduced. Furthermore, a combination of statistical and analytical techniques is used to eliminate structures which are unlikely to be close to the native structure. This is presented as an iterative framework, and appropriate termination criteria are introduced.
The main thesis behind ICON is that if two conformers of a protein are very similar to the native structure, they are likely to be similar to each other as well. However, if two protein structures are very dissimilar to the native structure, it is not necessary that they would be similar to each other. We implement this thesis by eliminating clusters of protein structures which are very dissimilar to each other. The algorithm and its implementation is presented in detail in the following sections. The method has been tested on an extensive data set of 1400 proteins containing high resolution decoys. The proteins in this data set have a pairwise sequence similarity of less than 35%. It has further been tested on a number of medium to low resolution conformer sets. The first data set in the medium resolution data set involves structures from the Decoys ’R’ Us dataset [174]. The second dataset in this regime is generated from the first principles protein folding framework ASTROFOLD [104]. Finally, the method has been tested on select targets of the recently concluded CASP8 and CASP9 experiments.

5.2 Methods

In this section, we introduce the novel iterative clustering method, ICON [198]. A flow diagram for the algorithm is shown in Figure 5.1.

At each stage, all the dihedral angles of all the conformers of the protein in the working set are put into the evaluation matrix $M(i, j)$, where $i$ represents the row number of the conformer, and $j$ represents the particular dihedral angle. Since the end residues of the protein do not define dihedral angles, we end up with a matrix of dimensions $N \times K$, where $N$ is the number of conformers in the working set at this stage, and $K = 2 \times (N_p - 2)$, where $N_p$ is the length of the protein. Based on this Cost Matrix, we implement the Traveling Salesman Problem (TSP) formulation of the novel bi-clustering method OREO [48], which is described below.
5.2.1 TSP Implementation of the OREO Approach

The aim of this section is to provide a detailed description of the variables and the objective function of the TSP model, which provides the optimal re-arrangement of the rows of the Cost Matrix $M(i,j)$. The index pair $(i, j)$ represents the particular row $i$ and the particular column $j$ of the cost matrix, whose individual element shall be denoted by $m_{i,j}$. Two rows $i$ and $i'$ are identified as adjacent rows in the final arrangement of the matrix, where row $i'$ lies below row $i$. This would
mean that in the final arrangement, a binary variable \( y_{i,i'} \) can be defined as:

\[
y_{i,i'} = \begin{cases} 
1 & \text{if row } i' \text{ immediately precedes row } i \\
0 & \text{otherwise}
\end{cases}
\]  

(5.1)

In order to finally place a particular row next to another one, the objective function is to minimize the dissimilarity between the two rows. A number of metrics of similarity can be used to define the objective function. The most commonly used objective functions are symmetric in nature [48]. However, for specific data sets, the objective function can be tailored according to the nature of the problem. For example, if it is known apriori that the neighboring rows of the final matrix would be such that for one row, the trend is monotonic, and the objective function can be forced to penalize only those cases when this trend is violated.

For the problem here, the following objective function is introduced:

\[
\text{OBJECTIVE} = \sum_i \sum_{i'} y_{i,i'} \phi(m_{i,j}, m_{i',j})
\]

(5.2)

where \( \phi(m_{i,j}, m_{i',j}) \) is given by:

\[
\phi(m_{i,j}, m_{i',j}) = \sum_j \min(m_{i,j} - m_{i',j}, 360 - (m_{i,j} - m_{i',j}))^2
\]

(5.3)

The objective function should reflect the circular nature of the dihedral angles. In particular, a squared difference potential cannot be used, because it would ensure that a dihedral angle of \(+180^\circ\) and \(-180^\circ\) are furthest away from each other, when in fact they are identical. Hence, the objective function in Equation (5.2) selects between the minimum of the difference in dihedral angles and their difference from \(360^\circ\). This way, if the difference between the dihedral angles is greater than \(180^\circ\), the value chosen in the objective function is the correct one.
The traveling salesman problem (TSP) \cite{10} is one of the most well-studied problems in combinatorial optimization. The main objective is to visit a list of N cities and return to the starting city via the lowest cost route. In the TSP formulation, each row of our cost matrix represents a node (or a city). If an edge connects two such nodes, then the two rows (i.e., the two conformers of the protein) are placed next to each other in the final arrangement. Therefore, the objective of the TSP can be re-formulated as visiting each conformer of the working set exactly once via these edges, while incurring the minimum cost, and to return to the first conformer. The cost of traveling from one node to the next is the objective function as expressed above.

Since the problem definition requires a circular tour which starts and ends at the same conformer, we introduce a dummy conformer to connect the first and the last structures. The cost of traveling from this dummy conformer to the top one is zero. The TSP formulation of this problem can be expressed mathematically, using a series of constraints to ensure that each row has exactly one neighbor above and below it. It is represented as:

\[
\min \sum_{i,i'} c_{i,i'} y_{i,i'} 
\]

\[
\sum_{i'} y_{i,i'} = 1 \quad \forall i 
\]

\[
\sum_i y_{i,i'} = 1 \quad \forall i' 
\]

Here $c_{i,i'}$ represents the cost of creating a final ordered list such that rows $i$ and $i'$ are placed adjacent to each other. For ICON, the cost function, and hence the objective function is given as in Equation \ref{5.2}. It should be noted that cyclic tours satisfy the constraints above. Hence, additional constraints are implemented to eliminate these subtours. Such constraints are very efficiently incorporated into TSP solvers such as Concorde, via cutting plane methods.
5.2.2 Cluster boundary definition and analysis

Once the final order of rows is determined, we have a path from the first conformer of the final matrix to the last one. In order to identify the densest clusters in the given ensemble, the optimal traveling salesman path would have to be divided into clusters in the most optimal manner. To address this, two approaches have been developed. The first approach identifies cluster boundaries using an Expectation-Maximization procedure, while the second method uses an integer linear optimization approach to determine the cluster boundaries.

In the first approach, after optimally re-ordering the set of structures, the clusters are determined in a hierarchical manner. Let the final ordering to range over the index \( i = 1, \ldots, |I| \). First, the pairwise distances, \( d(i, i + 1) \forall i < |I| \), between all neighboring elements in the final ordering are computed and stored on a sorted list, which we will define as \( SL \), from lowest to highest. The most similar pair of elements (i.e., \( \{(i, i + 1) : d(i, i + 1) \leq d(i', i' + 1) \forall (i, i' \neq i)\} \)) is merged to form the first cluster, \( c_{i,i+1} \), and the distances \( d(i - 1, i), d(i, i + 1), \) and \( d(i + 1, i + 2) \) are removed from \( SL \). The distances between this new cluster, \( c_{i,i+1} \), and the elements immediately below and above it in the final ordering are then computed and these distances are added to the sorted list (i.e, \( d(i - 1, c_{i,i+1}) \) and \( d(c_{i,i+1}, i + 2) \) are computed and added to \( SL \)). One should note that the distance between an element and a cluster, \( d(i, c) \), is based upon the average distance of the element \( i \) to all members of the cluster \( c \). The merging of two elements, an element and a cluster, or two clusters decrements the size of \( SL \) by one, and this process is repeated until \( SL \) reaches some specified value.

The minimum distance found in \( SL \) is generally an increasing function of \(|I| - |SL|\). When initially creating new clusters, the most similar elements in the final ordering are merged and the corresponding minimum distance found in \( SL \), say \( d^{\text{min}} \), would increase slowly as the number of elements in \( SL \) decreases. After all the most similar elements have been properly grouped into clusters, we inevitably encounter the situation where only dissimilar clusters and/or elements are
candidates for forming a new cluster, which should result in a noticeable increase in $d_{min}$. Thus, if we can confidently determine where this distance begins to change substantially, we can quantify when to terminate the merging of clusters.

Conceptually, this amounts to finding the “knee” in curve of $d_{min}$ as a function of $|SL|$. To illustrate, consider the black circles in Figure 5.2 which represents the values of $d_{min}$ as $|SL|$ decreases. It is easy to geometrically approximate where the knee in this plot occurs. As shown by the dashed green and blue lines in the figure, this curve can be represented by two distinct linear segments, and where these two segments intersect identifies the knee in the curve. This amounts to solving two separate linear regression problems, where the slope and intercept of each line segment, say line 1 and line 2, is a function of the points used to fit that line segment. Therefore, we need a robust way of determining which points belong to which line segment. Previous attempts [173] employed a complete enumeration approach for assigning points to the two line segments, and the assignment which resulted in the minimum weighted RSMD was selected. In this section, we present an efficient and automated strategy for assigning the points to the two line segments and determining the knee in the distance curve as a function of the number of clusters.

The algorithm begins by selecting 1/5 of the points of highest $|SL|$ value (e.g., the points on the right in the figure) and fitting a line segment through these points by solving a least squares regression problem (note that this corresponds to line segment 1). We then compute the vertical distances from all the points to this line segment and examine the corresponding distribution of these distances. The resulting distances assume a bimodal distribution, with a large, narrow distribution shouldering zero which corresponds to the points that are close to this line segment (which we will denote as class 1 since they belong to line segment 1) and a smaller, broad distribution extending to larger distances (which we will denote as class 2 as they belong to the second line segment). The average, variance, and mixture proportions of these two distributions can be computed by solving a mixture model, where here we assume that these distances approximately follow Gaussian distributions. To solve this Gaussian mixture model, we use the method of expectation
maximization (EM) to maximize the log likelihood function \[22\] that each of the points belongs to the first line segment (by default, the remaining points will belong to the other line segment). This provides us with the posterior probability distribution of the points belonging to either line segment, and we re-fit the line segment using those points that have a posterior probability greater than 0.5 for belonging to class 1. This procedure is iterated until the slope of this line segment converges to some value.

We also implement convergence strategies to avoid singularities and pathological behavior \[22\]. For instance, from a statistical point of view, the standard deviation of the first line segment should be at least one third of the largest distance for any outlier (i.e., \(3\sigma_1 = d_{\text{max outlier}}\), where an outlier is a point assigned to class 2 whose neighboring points on either side belong to class 1). Also, if the class 2 distribution begins to collapse (that is, less than 90% of the points belong to class 2), then we restart the EM algorithm with a lower standard deviation for class 1. It was also observed in previous work \[173\] that sometimes the initial points on the far right-hand side can skew the fit of the line segment. To address this issue, we check that the majority of the points in class 2 (e.g., at least 75% of the points in class 2) lie above the hyperplane defined by the first line segment. If they do not, then we eliminate the 10% of points with the largest \(|SN|\) value and reiterate the aforementioned procedure until the above criterion is satisfied.

The second approach for cluster boundary analysis is using integer linear optimization. First, we identify a set of “cluster seeds” by the set \(Seeds\), which consists of neighboring elements in the final ordering that are locally most similar. This has been illustrated in Figure 5.3. We also denote the set of elements that are outliers, or elements that are not cluster seeds, by the set \(Outliers\). The following notation is introduced: \(\bar{c}\) denotes the global average of \(c(i, i + 1)\) over all \(i\), \(\sigma_e\) is the corresponding standard deviation of \(c(i, i + 1)\) over all \(i\), and \(\hat{c}_{i,X}\) denotes the local average of \(c(i', i' + 1)\) for all \(i'\) within a neighborhood of \(\pm X\) around element \(i\). The sets \(Seeds\) and \(Outliers\) are constructed using the following algorithm:
Figure 5.2: Illustrative example of $d_{\text{min}}$ as a function of $SL$
1. Set $Seeds = \emptyset$ and $Outliers = \emptyset$.

2. Find the $i \notin Outliers \cup Seeds$ with the minimum $c(i, i + 1)$ in the optimal reordering.

3. If $\hat{c}_{i,X} \leq \bar{c} - \sigma_\bar{c}$, then add $i$ to $Seeds$ and all other elements $i'$ to $Outliers$ within the range of $\pm X$ elements of $i$. Else add $i$ to $Outliers$.

4. Return to step 2 and repeat until all elements $i$ are examined.

![Illustrative example of determination of “seeds” and “outliers”](image)

Figure 5.3: Illustrative example of determination of “seeds” and “outliers”

Given the set of cluster seeds, $Seeds$, we will formulate an ILP model to assign all other elements to one of these initial clusters. We introduce binary variables $z_i$ which are equal to 1 if the element is assigned to the cluster immediately preceding it in the final ordering, and 0 if it is
assigned to the cluster immediately after it in the final ordering.

\[ z_i = \begin{cases} 
1, & \text{if element } i \text{ is assigned to the cluster seed immediately before it} \\
0, & \text{if element } i \text{ is assigned to the cluster seed immediate after it}
\end{cases} \]

We define the sets $\text{Behind}(i)$ and $\text{InFront}(i)$ to denote the cluster seeds, represented by the index $k$, that are behind and in front of the element $i$, respectively. Finally, for every cluster $k$, we denote the set of elements that are fixed to belong to this cluster seed \textit{a priori} by the set $\text{Fixed}(k)$. For instance, if the first cluster seed contains the elements 2, 3, and 4, then $\text{Fixed}(1) = 2, 3, 4$.

The cost associated with the assignment of any element $i$ into the cluster preceding or following it can be dissected into several terms:

1. The fixed cost associated with assigning element $i$ to the cluster preceding it, which are the distances between element $i$ and all elements initially belonging this cluster.

\[
\text{FixedCost}1(i) = \sum_{i' \in \text{Fixed(\text{Behind}(i))}} c(i, i') z_i
\] (5.7)

2. If element $i$ is assigned to cluster $k \in \text{Behind}(i)$ and element $i' < i$ is assigned to the same cluster $k \in \text{InFront}(i')$, then we need to include the cost associated with placing these two elements in the same cluster.

\[
\text{VarCost}1(i) = \sum_{i' : \text{InFront}(i') = \text{Behind}(i)} c(i, i')(1 - z_{i'}) z_i
\] (5.8)

3. We also need to consider the contributions between element $i$ and elements $i' < i$ if they are assigned to the same cluster $k$, which precedes these elements.

\[
\text{VarCost}2(i) = \sum_{i' : \text{Behind}(i') = \text{Behind}(i)} c(i, i') z_{i'} z_i
\] (5.9)
4. Analogous expressions are derived for assigning elements to the clusters succeeding them in the final ordering. The fixed cost associated with assigning element $i$ to the cluster after it is given by:

$$\text{FixedCost}_2(i) = \sum_{i' \in \text{Fixed(InFront}(i))} c(i, i')(1 - z_i) \quad (5.10)$$

5. Lastly, we need to include the cost associated with placing elements $i$ and $i' > i$ in the same cluster $k$ that is after these elements in the final ordering.

$$\text{VarCost}_3(i) = \sum_{i' : \text{InFront}(i') = \text{InFront}(i)} c(i, i')(1 - z_{i'})(1 - z_i) \quad (5.11)$$

The objective function is then given by minimizing the summation of these individual contributions:

$$\min \sum_i \text{FixedCost}_1(i) + \text{FixedCost}_2(i) + \text{VarCost}_1(i) + \text{VarCost}_2(i) + \text{VarCost}_3(i) \quad (5.12)$$

Note that we must constrain the feasible cluster assignments to prevent the cross-assignment of elements. In other words, if element $i + 1$ is assigned to the cluster before it, then element $i$ cannot be assigned to the cluster after it. The following constraint enforces this restriction:

$$z_i \geq z_{i+1} \quad (5.13)$$

The nonlinearity associated with bilinear terms in the objective function can be alleviated by defining the following binary variable:

$$w_{i,i'} = z_i z_{i'} \quad (5.14)$$
and incorporating the following constraints [58] into the model:

\[ w_{i,i'} \leq z_i \]  
\[ w_{i,i'} \leq z_{i'} \]  
\[ z_i + z_{i'} - 1 \leq w_{i,i'} \]

Minimizing Equation 5.12 subject to constraint Equations 5.13, 5.15-5.17 provides the resulting cluster assignments for a given optimal ordering and set of cluster seeds (Seeds). The initial membership of the set Seeds is a function of the exclusion window \( X \). We vary the value of \( X \) and select the one which results in the minimum total cluster error, which is the sum of the intra and inter cluster errors [203, 204].

### 5.2.3 Evaluating cluster medoids and average spherical radii

Once we have the set of conformers partitioned into the individual clusters, we would like to eliminate all the clusters which are sparse and/or include structures which are outliers. This is done by evaluating the cluster centers for each of the clusters. The cluster average radius (modeling the cluster as a hypersphere) is then calculated by averaging the pairwise rmsd of the cluster medoid with all other conformers of that cluster. The cluster medoid is the closest node to the cluster centroid. The evaluation of the cluster medoid can be modeled as an integer linear optimization (ILP) problem. The objective is to minimize the distance of the cluster medoid to each of the elements of the cluster, while making sure that only one such point exists. Let us define binary variables \( y(i) \) which are assigned to 1, if conformer \( i \) is the medoid of its cluster and zero otherwise. We define parameters exist(\( i \)) to have the value 1, if conformer \( i \) lies in the current cluster and 0
otherwise. The model can be formulated as:

$$\min \sum_i \sum_{i'} b(i, i') exist(i') y(i)$$  \hspace{1cm} (5.18)

such that

$$y(i) \leq exist(i)$$  \hspace{1cm} (5.19)

$$\sum_i y(i) = 1$$  \hspace{1cm} (5.20)

$$y(i) = 0 - 1$$

where $b(i, i')$ is an element of the matrix $B$, a square matrix of dimension $N \times N$, which represents the pairwise rmsds between each pair of conformers in the working set. Once the cluster medoid is identified, the average cluster radius is calculated by evaluating the average pairwise rmsd of the conformers of the cluster to the medoid. The aforementioned model is implemented individually for each cluster.

### 5.2.4 Eliminating clusters based on cluster properties

Based on the number of elements in a cluster, $N_j$, and its cluster average rmsd, denoted as $\overline{RMSD}$, we define the Cluster Concentration $CC_j$ as:

$$CC_j = \frac{N_j}{\overline{RMSD}_j}$$  \hspace{1cm} (5.21)

Based on the definition of the cluster concentration, we would like this term to be as large as possible. Larger number of elements in the cluster shows the possibility of multiple local minima surrounding this region in the energy landscape, while a low average rmsd shows the tightness of the cluster (based on the very similar backbone dihedral angles of the conformers in the cluster). This is highly desirable, as it is likely that the clusters which contain outlier structures would have lower number of conformers and/or a high average RMSD.
The value of cluster concentrations for each of the clusters define a distribution, which can be modeled as a gaussian distribution. It is desirable to have a metric, which would ensure that we end up selecting as many of the good clusters from this distribution as possible, while ensuring that we do our best to eliminate the poorer ones. To achieve this, we propose the following iterative sequence of steps. We first check whether, for this given distribution, there are any clusters which have their concentration value greater than three standard deviations above mean. If there are clusters which satisfy this condition, then we remove them from the current list and store them for the next stage of clustering. These clusters are the ones which are the most likely to contain structures closest to the native. For the remainder of the list, we re-evaluate the mean and the standard deviation. It is then again checked if there are clusters greater than 3 standard deviations above mean. This procedure is carried out till there are no more clusters greater than 3 standard deviations of the mean of the existing distribution.

If initially, we do not find any clusters above three standard deviations of the mean, we check whether there are clusters which are above two standard deviations, and if not, then one standard deviation above the mean of this distribution. Once these are found, a similar procedure of removal and analysis of the shortened list of clusters is done till there are no more clusters above this new threshold.

Finally, we select all clusters which were previously removed and stored, along with all clusters which are greater than or equal to the median of this modified distribution, as these would correspond to the clusters with the highest concentration. The median is selected because it is more immune to the existence of outliers in the distribution of cluster concentrations. All the conformers in the selected clusters form the working set of the next iteration.

This entire procedure is carried out for ten iterations, or till the number of conformers in the working set is not below 50% of the number of conformers in the original ensemble.
5.2.5 Selection of near-native structures

At the end of the iterative clustering procedure, we select the most likely structures which are going to be closest the putative native structure. In order to do this, we collect the medoids of all the clusters at the end of the final stage, which have their concentration value above or equal to the median of this distribution. For each of these cluster medoids, we implement the novel $C_\alpha - C_\alpha$ and Centroid - Centroid distance dependent high resolution force fields, proposed by [156, 157]. These force fields aim to isolate native and near-native folds of a protein as lower energy structures, compared to structures further away from the native structure. Finally, the five cluster medoids with the lowest energies are picked as the selected structures.

5.2.6 Automated Implementation

ICON has been implemented as an automated procedure, which collects the dihedral angle matrix and pairwise RMSD data and implements the entire iterative algorithm. The TSP implementation of the OREO Clustering Approach has been implemented as a C++ program in interaction with CPLEX 11.0 and Concorde. Steps which comprise of collection of dihedral angle data, evaluation of cluster boundaries, evaluation of cluster medoids and the elimination of loosely bound clusters, are implemented in C. Finally, the high resolution force fields have been implemented in C++, and are available for download [156, 157]. ICON is freely available for use to the scientific community at http://helios.princeton.edu/ICON.

ICON has been tested for computational time on a single processor machine using a single processor CPLEX implementation, as well as with a multi-processor CPLEX implementation on a computer cluster. The parallel version of ICON was run on Quad core Intel Xeon 2.83 GHz processors. For a run on a test protein (PDB: 1elr chain A) with 762 conformers, the parallel version of ICON took 5 CPU minutes for a run. ICON was also tested on a single Intel Pentium(4) 3.2 GHz processor. The same run is completed in about 25 minutes.
### 5.3 Computational Results

ICON was tested on a large number of proteins. The test sets were divided into 3 distinct categories: High Resolution data set, Medium to Low resolution data set and CASP8 and CASP9 targets. The Medium to low resolution data sets is further sub-categorized into ensembles from Decoys ‘R’ Us [174] and ASTROFOLD [104]. Fig 5.4 shows the brief results of the application of the method. As can be seen from the histogram plot, the method performs consistently above 90% in almost all cases, by using both the Cα-Cα and Centroid-Centroid force fields.

![Histogram showing overall results of ICON algorithm on individual test sets](image)

Figure 5.4: Overall results of ICON algorithm on individual test sets

The following subsections present details on the generation of the datasets, and the results generated from the application of the novel clustering method.

#### 5.3.1 High Resolution Data Set

For generating the decoys for the high resolution data set, a well represented collection of 1400 proteins developed by Zhang and Skolnick [228] was used. All the proteins of this set are non-homologous single domain proteins with a maximum pairwise sequence similarity of 35%. The length of the proteins varies from 41 to 200 amino acids. It also has a mixed representation of α, β and α/β proteins.
The generation of decoys for each of these proteins was carried out using a torsion angle dynamics approach, DYANA [75]. The main premise of the decoy generation framework is the idea of retaining distance information among the residues within the hydrophobic core of the protein. Once the hydrophobic core has been defined, distance bounds are introduced among the hydrophobic residues to relax the native distance between them. Further details on the generation and quality of the decoys generated can be found elsewhere [157].

For all decoys of the 1400 proteins, the proposed novel iterative clustering method, ICON, was applied. At the final stage, in order to select the final 5 conformers from the list, both the Cα - Cα [156] and the Centroid - Centroid [157] distance dependent force fields were applied, and the results compared to the state of the art SPICKER method [229]. Figure 5.5 shows the rank of the selected conformer using both the criteria. The rank reflects the number of structures in the individual ensemble which are ahead of the picked structure. In order to give a better representation of how many conformers are better or worse than the selected structure, Figure 5.7 presents a percentile graph of the number of structures that are worse than the selected structure for each of the force fields. As shown in the figure, for 84.7% proteins out of the high resolution data set, ICON selects structures which are above the 95 percentile in terms of quality of the structure.

As a comparison to the SPICKER method, we ran SPICKER on the same dataset. Figure 5.5 presents a graph where the proteins have been sorted based on the ranks of the structures selected by SPICKER method. Further, ranks that were selected by ICON are also presented there. As can be seen, for a large number of cases, ICON performs better than the SPICKER method. Furthermore, Figure 5.6 presents a graph showing the difference in RMSDs between the best structures selected by ICON, and the corresponding structures picked by SPICKER. In 81.5% of the cases, ICON performs better in selecting near-native structures from the ensembles, while in 86.2% of the cases, ICON performs at least as well as SPICKER. This suggests that ICON can select near-native structures from a given high resolution ensemble of protein structures. A detailed presentation of the results for this dataset is in the Supplementary Material S1.
Figure 5.5: Graph representing ranks of structures selected by SPICKER and ICON. About 83% of points from the ICON algorithm fall below the monotonic curve represented by SPICKER.

Figure 5.6: Graph representing difference in rmsd values of structures selected by ICON and SPICKER, $\Delta RMSD = \text{rmsd}_{ICON} - \text{rmsd}_{SPICKER}$. 1193 points fall below zero, representing the number of cases where the structure selected by ICON has a lower RMSD than the one selected by SPICKER.
In order to compare the relative contributions of the clustering algorithm and the force field towards the performance of the ICON algorithm, the Cα-Cα and Centroid-Centroid energies of all decoys of a set of 150 proteins from the high resolution data set were evaluated. These 150 proteins were not used in the generation of either of these force fields. We compiled the five lowest energy structures in these ensembles, and the lowest RMSD structures among these were analyzed. The average percentile of the selected structures by using just the Cα-Cα and Centroid-Centroid force fields were 92.7% and 91.9% respectively, while the use of the proposed approach resulted in the average percentiles to be 97.1% and 96.6%, respectively. Comparisons were also made with the statistical full atom Rosetta potential. By using just the Rosetta potential, the average percentile of best selected structure was 70.8%. This shows that the algorithm improves the structures selected by the force fields. The force fields themselves perform quite well on this test set, since they
Chapter 5. Near-native structure identification

were trained on a large, high resolution training set of 1250 proteins, and are likely to handle high resolution ensembles well.

5.3.2 Medium to low resolution data set: Decoys ’R’ Us

In order to generate the medium to low resolution data set, two distinct procedures were used. Firstly, the algorithm was tested on five decoy sets of the Decoys-‘R’-Us database [174], which are identified as the most challenging datasets based on the results of Rajgaria et al. [157]. These included the test sets fisa and fisa-casp3 [186], lmds [94], lattice-ssfit [176, 221] and semfold [175]. The fisa and fisa-casp3 datasets were generated by a fragment insertion simulated annealing procedure, where the procedure was used to assemble native like fragments from unrelated proteins using Bayesian scoring functions. Each conformer of the lmds dataset is a local minima structure obtained using the ENCAD function, which contains a penalty term for steric clashes and a favorable contribution term for compactness and native similarity. The lattice dataset was generated by firstly generating all possible conformations using a tetrahedral lattice. A scoring function was used to rank these structures. Some of the best structures were minimized locally using a different energy function, while maintaining the secondary structure features.

As can be seen from Figure 5.8, ICON performs well in selecting structures in the top tenth of the ensemble of structures in most cases. It is of particular importance to note that the sets of decoys used in this test set have been generated by different methods. Further, the RMSD ranges of the individual decoys for a significant number of proteins have a majority of their ensembles in the medium or low resolution decoy range. For this data set, using the $C_\alpha - C_\alpha$ force field, the average percentile of structures selected by ICON is 91.2%. Using the Centroid - Centroid force field, ICON selects structures with an average percentile of 92.0%.

Comparisons were also made between the use of the proposed clustering algorithm, and the use of the force fields directly. By directly using the $C_\alpha-C_\alpha$ and Centroid-Centroid force fields, we
get an average percentile of best structure to be 80.2% and 82.1% respectively. By using the all atom Rosetta potential, we get an average selected percentile of 83.8%. Note that ICON exhibits a superior performance (i.e. 91.2% and 92.0%) when compared to the exclusive use of Cα-Cα, Centroid-Centroid and the all atom Rosetta potentials (80.2%, 82.1% and 83.8% respectively).

A disparity is seen in the results obtained using either the different force fields. This may be attributed to the nature of the ensembles of structures produced. Since the Cα-Cα force field does not account for side chain information, a protein structure where side chains are too close or too far will not be accounted for. On the other hand, the Centroid-Centroid force field accounts for the side chains of the structures. Hence, misplaced side chains would cause an increase in the Centroid-Centroid energy of the structure, while keeping the backbone based energy constant.

Figure 5.8: Performance of ICON algorithm on the Decoys ‘R’ Us dataset
Table 5.1: Results for Medium Resolution Data Set generated by ATRO-FOLD first principles framework

<table>
<thead>
<tr>
<th>Protein</th>
<th>Number of conformers</th>
<th>RMSD range</th>
<th>RMSD selected</th>
<th>Percentiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1R69</td>
<td>5000</td>
<td>3.74-13.91</td>
<td>5.60,5.60</td>
<td>93.50,93.50</td>
</tr>
<tr>
<td>1ELR</td>
<td>2622</td>
<td>4.93-17.41</td>
<td>5.60,5.60</td>
<td>93.50,93.50</td>
</tr>
<tr>
<td>1GB1</td>
<td>5000</td>
<td>3.26- 8.99</td>
<td>5.60,5.60</td>
<td>93.50,93.50</td>
</tr>
<tr>
<td>1C75</td>
<td>5000</td>
<td>5.86-12.62</td>
<td>5.60,5.60</td>
<td>93.50,93.50</td>
</tr>
<tr>
<td>1E29</td>
<td>1885</td>
<td>9.72-21.13</td>
<td>5.60,5.60</td>
<td>93.50,93.50</td>
</tr>
<tr>
<td>1ECA</td>
<td>1816</td>
<td>11.02-21.49</td>
<td>5.60,5.60</td>
<td>93.50,93.50</td>
</tr>
<tr>
<td>1HCR</td>
<td>5000</td>
<td>4.85-12.74</td>
<td>5.60,5.60</td>
<td>93.50,93.50</td>
</tr>
<tr>
<td>1ROP</td>
<td>3124</td>
<td>1.25-6.53</td>
<td>5.60,5.60</td>
<td>93.50,93.50</td>
</tr>
</tbody>
</table>

### 5.3.3 Medium to low resolution data set: ASTROFOLD

A first principles based method was also used for the generation of medium resolution ensembles of a small set of proteins. The fourth stage of the ASTRO-FOLD algorithm uses torsion angle dynamics, deterministic global optimization and a stochastic computational space annealing procedure to predict the tertiary structure of a protein given its amino acid sequence [100, 101, 102, 103, 104, 105, 106, 131, 199]. As can be seen from the results shown in Table 5.1, the average percentiles of the best structure selected by ICON are 87.1% and 91.5%, using the C\textsubscript{α}-C\textsubscript{α} and Centroid-Centroid force fields, respectively. Further, the method is seen to be reasonably independent of the quality of the ensemble provided to it. This is particularly important, since for a protein with a completely new fold, it is possible that structure prediction techniques may not be able to produce structures of high resolution quality.

### 5.3.4 Selected CASP8 and CASP9 Targets

A selection of the CASP targets was also used as an additional test set for evaluating the effectiveness of the proposed iterative clustering method, ICON. The results for the CASP targets are presented in Table 5.2. As can be seen from the results, for the range of RMSDs of the structures
in the ensemble, the structures selected by ICON are of good quality. This is particularly important to note in this case, as the range of RMSDs lies within the medium to low resolution regime.

The average percentile of selected structures using the Calpha-Calpha force field is 92.9%, while it is 92.6% when the Centroid-Centroid force field is used. The CASP data set provides the most realistic, and up-to-date test set of the ICON formulation. Based on its nature, the CASP target structures are not known apriori. This is especially relevant for target structures with low sequence and structural homology to databases. As can be seen, ICON performs very well by selecting structures which are on an average, within the top 7.5% of predicted structures in the respective ensembles.

### 5.3.5 Stage-wise Enrichment of ICON

In order to demonstrate the benefit of the proposed iterative clustering method ICON, we present a histogram for each individual stage of the iterative procedure, which shows the distribution of the conformers in the working set of the method at the particular stage for an example protein (PDB: 2d6w).
l1elrA). As can be seen from Figure 5.9, at each stage, the better structures are retained more than the comparatively worse structures. Table 5.3 shows the enrichment factor for each bin in the histogram for the various stages. The enrichment factor for a bin at a stage is given by:

\[
\text{Enrichment} = \frac{N_{\text{bin}, \text{stage}}}{N_{\text{bin}, \text{start}}} \times \frac{N_{\text{total}, \text{start}}}{N_{\text{total}, \text{stage}}}
\]

(5.22)

Figure 5.9: Graph showing the change in distribution of conformer RMSDs over iterations

As can be seen from this table, the RMSD regions which we are interested in (the ones closer to the native structure) are enriched in a favorable manner (> 1). The top two regions are enriched significantly. Clearly, the conformers which are most likely to be falsely selected would lie in the middle of this table. As can be seen, a very large number of them are eliminated at the individual stages. Furthermore, for RMSD ranges 1.5 to 4.0, the trend is monotonic which is highly favorable.
5.4 Discussion and Conclusions

A novel iterative clustering method, ICON, is introduced to identify the near-native folds for a protein from an ensemble of given structures. The method uses a clustering in the dihedral angle space via a TSP based framework and eliminates loose and widely spread clusters at each iteration to reach the final solution. ICON was tested on a set of 1400 non-homologous proteins. The method identified structures within the top 10% of conformers in 97% of cases. The average percentile of the selected conformer was 2.9%, that is, on average the selected conformer was in the top 2.82% of the conformers in the ensemble.

The method was also tested on medium resolution data sets taken from external sources, and performed very well. The fact that the accuracy of the method does not deteriorate significantly when considering a variety of high resolution, medium resolution and low resolution datasets suggests that the method is robust to diverse conformational ensembles.

As was discussed in the introduction section of the chapter, the overall effectiveness of the method will depend to a certain degree on the “prior“ distribution of a given ensemble. For a given target protein, if the ensemble of predicted structures are of a different topology or fold to the native structure, using clustering techniques to identify the “near-native“ structure can be limited. In such cases, even the correct selection of structures with lowest RMSDs to the native would hold

Table 5.3: Table showing enrichment factor for different RMSD regions over number of stages

<table>
<thead>
<tr>
<th>RMSD Range</th>
<th>N</th>
<th>Stage 1</th>
<th>Stage 2</th>
<th>Stage 3</th>
<th>Stage 4</th>
<th>Stage 5</th>
<th>Stage 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5-1.0</td>
<td>49</td>
<td>1.1091</td>
<td>1.1905</td>
<td>1.2843</td>
<td>1.3412</td>
<td>1.4547</td>
<td>1.6518</td>
</tr>
<tr>
<td>1.0-1.5</td>
<td>236</td>
<td>1.0843</td>
<td>1.1020</td>
<td>1.171</td>
<td>1.2531</td>
<td>1.3232</td>
<td>1.3969</td>
</tr>
<tr>
<td>1.5-2.0</td>
<td>206</td>
<td>1.0113</td>
<td>0.9734</td>
<td>0.9497</td>
<td>0.9950</td>
<td>1.0216</td>
<td>0.9870</td>
</tr>
<tr>
<td>2.0-2.5</td>
<td>103</td>
<td>0.9674</td>
<td>0.9675</td>
<td>0.9032</td>
<td>0.7596</td>
<td>0.6261</td>
<td>0.6325</td>
</tr>
<tr>
<td>2.5-3.0</td>
<td>104</td>
<td>0.8492</td>
<td>0.8647</td>
<td>0.8024</td>
<td>0.7071</td>
<td>0.6527</td>
<td>0.5125</td>
</tr>
<tr>
<td>3.0-3.5</td>
<td>54</td>
<td>0.8387</td>
<td>0.7877</td>
<td>0.7347</td>
<td>0.6085</td>
<td>0.5343</td>
<td>0.4752</td>
</tr>
<tr>
<td>3.5-4.0</td>
<td>9</td>
<td>1.0064</td>
<td>1.0803</td>
<td>1.064</td>
<td>1.0431</td>
<td>0.5657</td>
<td>0.2193</td>
</tr>
</tbody>
</table>
little significance. Such situations could introduce sources of errors in the iterative process of the algorithm, and the algorithm may result in a direction that is not favorable.
Chapter 6

Tertiary structure prediction:
Computational Studies

As mentioned previously, the protein structure prediction problem continues to represent a proverbial *holy grail* in computational chemistry and structural biology communities. Stated simply, the problem can be described as an attempt to elucidate the three dimensional structure of a protein, given its amino acid sequence.

### 6.1 Introduction

In order to do this, most algorithms base their approach on Anfinsen’s thermodynamic hypothesis [9]. According to the hypothesis, the native structure of a protein in a given environment corresponds to the global minimum free energy of the system.

Given the expanding collection of proteins in the Protein Data Bank (PDB) [20], along with the inherently difficult task of ab-initio protein structure prediction, a number of database-driven methods have been developed which exploit information from the experimentally-determined structures of proteins in the PDB. While clear classification between approaches towards protein structure
prediction has become difficult, most approaches towards protein folding can be classified into the following three categories: (a) homology based methods, (b) fold recognition based techniques and (c) first principles based methods. A detailed review of various protein folding approaches has been presented in literature [60, 62, 226]. Homology, or Comparative, modeling methods aim at directly identifying the template of a homologous protein, which can then be used as a starting point for refining the structure to suit the target protein better. Typically, these methods are most successful when there is a high degree of similarity between the target and parent structures which exist in the database. Many methods are based on sequence alignment methods like BLAST or PSI-BLAST [6]. While the implementation of sequence alignment or machine learning methods provides a good starting point for these algorithms, the challenge of loop building and side chain modeling persists. In order to address these issues, many methods employ well-known algorithms for model building [54] and side chain placement [107]. A detailed review of approaches and drawbacks in comparative protein modeling can be found elsewhere [118, 140].

Fold recognition approaches aim at identifying distant homologs to a target sequence by working in the structural space instead of the sequence space. The premise for these methods lies in the observation that the fold space is much more limited than the sequence space. Based on the evaluation of researchers that the PDB [20] is almost complete in terms of observed folds [230], a number of “threading” algorithms have been developed. Threading algorithms aim to find the best fit for a target protein sequence onto the structure of a template in the database. For distantly related proteins, a successful prediction would require accurate atomic prediction of the dissimilar regions, along with the aligned regions. A number of successful methods have been proposed which use fold recognition, including dynamic programming methods [90], iterative methods [187] and optimization-based methods [162, 170].

An intermediate category between pure fold-recognition techniques and pure first principles based methods can be considered to be the fragment-assembly based methods. Here, a target protein is divided into short oligopeptides, which are used to identify fragments of structures from
various templates in the database. This way, separate fragments can be selected from unique template structures. The individual fragments can then be brought together using statistical potentials and optimization based algorithms. Optimization algorithms like simulated annealing \[166, 186\] have been successfully used to bring fragments together to identify a fold, before using all-atom scoring functions to refine the model further. Skolnick and co-workers have aimed at combining multiple sequence alignment and threading with a unified atom lattice model to generate initial folds, before refining and clustering structures to identify the best structures from an ensemble \[152\].

The final category of methods for protein structure prediction, the first principles based methods, avoid the direct use of homology or structural alignment information. These algorithms work with a much larger search space, and base their search algorithms on Anfinsen’s hypothesis \[9\], i.e. the structure sought would lie at the global free energy minimum of the system. Despite the increased computational complexity of first principles based methods, the primary advantage of these methods is the ability to predict the structure of a protein in the absence of a good structural or sequence homolog. Furthermore, the use of physics-based scoring or energy functions would allow the extension of the protein structure prediction algorithms to various environments. Physics-based energy functions would also provide an insight into the protein folding process, along with creating a picture of the energy landscape of a protein \[125\].

A number of first principles-based structure prediction algorithms aim at using a hierarchical process to protein folding. The use of a hierarchical process sequentially reduces the search space for the search algorithms. Dill and co-workers use replica exchange molecular dynamics for the search algorithm, in conjunction with a zipping and assembly model to bring distant parts of the sequence together to form the protein fold \[151\]. Rose and co-workers have proposed a Metropolis Monte Carlo based search algorithm for the structure prediction problem, and identify conformational biases based on discrete moves selected using a physics-based force field \[190, 191\]. Using a distributed grid computation algorithm, Folding@Home, Pande and co-workers have used carte-
sian molecular dynamics to fold protein villin [224]. A number of methods have aimed to use coarse grained potential at an early stage to determine the fold of a protein, followed by a model refinement procedure with a more detailed atomistic force field. One of the more popular force fields in this regard is the united-residue (UNRES) force field introduced by Scheraga and co-workers [120, 122, 126]. By representing each amino acid to two interaction sites and using a stochastic conformational space annealing [111], the conformational space is reduced to the low energy regions. Recently, researchers have successfully managed to fold small proteins like WW protein domain to high resolution structures using molecular dynamics and improved all atom force fields [114, 179, 188].

Another first principles-based approach to protein folding is the ASTRO-FOLD approach developed by Floudas and co-workers [100, 101, 102, 103, 104, 105, 130, 131]. The framework follows a hierarchical approach to the protein structure prediction problem, by combining the all-atom physics-based ECEPP/3 energy function [145], deterministic $\alpha$BB global optimization algorithm, stochastic conformational space annealing algorithm and a molecular dynamics approach in the torsion angle space. The deterministic global minimization algorithm, $\alpha$BB [113], guarantees convergence to the global optimum for a problem with twice differentiable objective function and constraints, by creating a converging series of lower and upper bounds. Given the highly nonlinear nature of the force field, and the complex terrain defining the conformational search space, the deterministic global optimization algorithm is supported by torsion angle dynamics and conformational search annealing procedures. The torsion angle dynamics routine is used to generate low energy, feasible solutions, by employing a simple steric-based energy function in a molecular dynamics routine. The conformational search annealing procedure, which is based on a combination of genetic algorithms and simulated annealing methods, is used for enhanced searching of the conformational space to identify better upper bounding function values.
This chapter presents an overview of the tertiary structure prediction step of the ASTRO-FOLD 2.0 [199] framework, followed by a presentation of blind target prediction results in the recently concluded CASP9 experiment.

6.2 ASTRO-FOLD 2.0 framework

Given an amino acid sequence for a protein, the aim, in a blind prediction, is to identify a small set of proteins which are likely to be closest to the native structure. The ASTRO-FOLD 2.0 approach is presented in flowchart form in Figure 6.1. In this chapter, the focus is on the tertiary structure prediction component of ASTRO-FOLD 2.0. An overview of the algorithms and approaches that support the tertiary structure prediction algorithm, both novel and from the previous version of ASTRO-FOLD is also presented.

As shown in the figure, a number of precursor steps are used to generate constraints which are used in the tertiary structure prediction stage. The following present a broad overview of the various stages leading up to the tertiary structure prediction algorithm.

6.2.1 Secondary structure prediction

Given the amino acid sequence of a target protein, the first stage is the prediction of the location of the α-helices and β-strands. There are a number of advantages of carrying out this initial step. The hierarchical process of protein folding permits us to tackle the local problem of secondary structure prediction, without any significant loss caused due to the neglect of long range interactions. Secondly, amino acids in the α-helices and β-strands have backbone dihedral angles found in a very defined subregion of the Ramachandran plot, thus reducing the search space for the tertiary structure prediction algorithm. Finally, the prediction of secondary structure permits the implementation of other intermediate stages which generate additional distance and dihedral angle constraints for the final three dimensional structure prediction algorithm.
CHAPTER 6. TERTIARY STRUCTURE PREDICTION: COMPUTATIONAL STUDIES

6. TERTIARY STRUCTURE PREDICTION: COMPUTATIONAL STUDIES

High Resolution Force Field
- Novel linear program scheme
- Distinguishes high resolution
  (Large-scale linear programming)

Novel linear program scheme
Distinguishes high resolution
(Large-scale linear programming)

Derivation of Restraints
- Dihedral angle restraints
- Ca-Ca distance restraints
  (Reduce Search Space)

Secondary Structure Prediction
- Helix Prediction
  - Detailed atomistic modeling and free energy calculations
- β strand Prediction
  - Novel hydrophobic modeling with a ILP model

Simultaneous Helix-Strand Prediction
- MILP Optimization
  - Helix propensity via regional classification & strand prob. using Bayesian-Markov model
- CONCORD
  - Consensus methods based on 7 locally installed programs
- Contact Prediction
  - Minimize pairwise contact energy with a MILP model
- Loop Prediction
  - Use flexible stems and dihedral angle sampling

β-sheet Topology Prediction
- Ab-initio method with ILP formulation
- Statistical force field with MILP model

3D Structure Prediction
- Use a combinatorial and global optimization framework
  (combining aBB with conformational space annealing)

Input Sequence

Figure 6.1: Flowsheet representing the ASTRO-FOLD 2.0 Approach
The prediction of α-helical regions of the protein was carried out using the single sequence helix prediction algorithm presented in Chapter 2. Given an amino acid sequence, the algorithm divides the protein into overlapping nonapeptides. For each nonapeptide, the propensity of the central amino acid to be in a helix is evaluating by evaluating the likelihood of the surrounding amino acids to form a network of hydrogen bonds representing the fingerprint of an α-helix. Further, the hydrophobicity and sidechain-to-backbone hydrogen bonds at the ends of helices, which are present to compensate for the lack of backbone hydrogen bonds [15], are modeled by predicting the propensity for any helical residue to be the beginning or end of the helix.

For the prediction of secondary structure in β and mixed α/β proteins, a novel integer optimization based consensus approach CONCORD [213] is used. The algorithm combines the secondary structure predictions of seven methods, SSpro [31], DSC [96], PROF [150], PROFphd [169], PSIPRED [91], Predator [67] and GorIV [68] in an MILP-based consensus formulation. The method succeeds in identifying the strengths of the different methods, and results in an average prediction accuracy which is superior than any of the individual methods.

At the end of the algorithm, we obtain the location of the secondary structure elements of the protein. This information is transformed into two types of constraints. The backbone dihedral angles of all α-helical residues is restricted to \([-90, -35]\) and \([-75, -15]\) for \(\phi\) and \(\psi\), respectively. The backbone dihedral angles of all residues in β-strands are restricted to \([-180, -135]\) and \([135, 180]\) for \(\phi\) and \(\psi\), respectively. In addition, distance constraints on all residue pairs separated by four amino amino acids restrict these pairs so that hydrogen bonding is permitted.

6.2.2 β-sheet topology prediction

Given the location of the β-strands in a β or mixed α/β proteins, the next stage predicts their arrangement in three dimensional space. The detailed β-sheet topology prediction algorithm was presented in Chapter 3. Using strand-to-strand contact potentials derived from a previously known
force field [30], the objective is to maximize the total contact potential of the protein, thus maximizing the area of the hydrophobic core of the protein. A large number of physical, biological, relational, structural and steric based constraints have been applied to ensure the prediction of physically meaningful structures. At the end of the algorithm, a detailed atomistic-potential based re-ranking algorithm is used to identify the most likely arrangement of the $\beta$-strands of the protein.

The implementation of the $\beta$-sheet algorithm provides a large number of non-local distance contacts for the tertiary structure prediction algorithm. All amino acid pairs which are predicted to be contacting in the sheet topology prediction algorithm are restricted such that the distance between them enables the formation of hydrogen bonds between donor and acceptor pairs on the amino acids. A large number of such restrictions prove very vital towards the reduction of the tertiary structure search space.

6.2.3 Contact Prediction

Given the protein sequence and secondary structure information, residue contacts and protein topology (relative positions of various secondary structure elements) can be predicted. This is done in ASTRO-FOLD 2.0 through an integer linear optimization model. This model predicts residue contacts in $\alpha$, $\beta$, $\alpha + \beta$, and $\alpha/\beta$ proteins [158, 160]. The total energy of a protein in this model is expressed as sum of a $C\alpha - C\alpha$ distance dependent contact energy contribution and a hydrophobic contribution. Contacts are predicted by minimizing the total energy while satisfying a set of constraints that are included in the model to enforce certain physically observed topological information.

The Contact prediction algorithm provides a significant number of non-local distance constraints which would be very useful in the tertiary structure prediction stage. The increased utility of the contact prediction algorithm is observed in $\alpha$ and mixed $\alpha/\beta$ proteins, since the $\beta$-sheet topology prediction algorithm only targets the relative orientations of $\beta$-strands. Results from the
contact prediction algorithm are used to derive distance constraints between pairs of amino acids in \( \alpha \)-helices, and with their counterparts in \( \beta \)-strands and loops.

### 6.2.4 Loop Structure Prediction

The last major algorithm preceding the tertiary structure prediction algorithm is the loop structure prediction algorithm presented in Chapter 4. The algorithm derives initial structures by building the target loop from a probability distribution created out of known loop structures. Unique probability distributions are used, depending on the type of secondary structure flanking the loop on either side. A number of fast rotamer optimization steps are used to remove steric clashes in the initial structures. The structures are then subjected to full atom energy minimization using the ECEPP/3 potential, defined further in this chapter. Clustering techniques are used to identify a subset of representative loop structures for each target loop sequence.

The main aim of the loop structure prediction algorithm is the derivation of backbone dihedral angle bounds for all amino acids in the loop regions of the protein. Thus far, all algorithms have targeted amino acids in the secondary structure regions of the protein. Tight bounds on the dihedral angles of the loop region reduce the search space for the tertiary structure prediction algorithm drastically. Further, unlike distance constraints, dihedral angle bounds can be included into the model as linear constraints, as discussed further in the chapter.

### 6.3 Tertiary structure prediction

Using the outputs from each of these steps, the tertiary structure prediction algorithm has been implemented as a combination of the deterministic global optimization (\( \alpha \)BB algorithm), stochastic global optimization (Conformational Space Annealing) and molecular dynamics in the torsion angle space (Torsion Angle Dynamics).
6.3.1 Energy function

According to Anfinsen’s hypothesis [9], the native state of the protein lies at the global free energy minimum of the system. Hence, while solving for the protein structure, it is imperative that the energy function be a reflection of the free energy landscape of the protein. Common physics-based energy functions include energetic contributions from terms based on atomic bonds, atomic angles, torsional angles, van der Waals interactions and electrostatics. Some examples of force fields which include all of such terms are AMBER [39] and CHARMM [124]. A modified approach is to assume the covalent bond lengths and bond angles to lie at their mean values. This way, one can ignore the energetic contributions which arise from the first two terms previously mentioned. By fixing these parameters to their mean values, force fields such as ECEPP [137], ECEPP/3 [145] and ECEPP-05 [11] can define their energy expressions purely based on the protein dihedral angles. Modeling the protein using only its dihedral angles significantly reduces the variable space when compared to the cartesian representation that one would have to adopt if bond lengths and angles were included. For ASTRO-FOLD 2.0, we use the ECEPP/3 potential function, which contains terms representing electrostatic, van der Waals, hydrogen bonding and torsion angle contributions given by:

\[
E_{\text{ECEPP/3}} = \sum_{(i,j)\in ES} \frac{q_i q_j}{r_{ij}} + \sum_{(i,j)\in NB} F_{ij} \frac{A_{ij}}{r_{ij}^{12}} - \frac{C_{ij}}{r_{ij}^{6}} + \sum_{(i,j)\in HB} \frac{A'_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{10}} + \sum_{(k)\in \text{TOR}} \frac{E_{0,k}}{2} \left(1 + c_k \cos n_k \theta_k \right) \tag{6.1}
\]

In Equation 6.1, \(r_{ij}\) represents the distance between a pair of atoms \(i\) and \(j\), given that both the atoms fall into the set of atoms over which the summation is carried out. The parameter \(F_{ij}\), which represents the relative impact of the repulsive part of the Lennard-Jones expression, is taken as 0.5 for \(1 - 4\) interactions, and 1.0 for \(1 - 5\) interactions. Non-bonding parameters such as \(A_{ij}, A'_{ij}, B_{ij}\) and \(C_{ij}\) are atom pair dependent. The sets \(ES, NB\) and \(HB\) are defined over the set of
pairs of atoms $i$ and $j$ that can have electrostatic, non-bonded and hydrogen bonding interactions, respectively. The set $TOR$ runs over all torsion angles of the protein that can contribute to the last term of the expression.

### 6.3.2 Problem Formulation

The problem of finding the global energy minimum of a protein can be formulated as:

$$\min_\theta E_{ECEP}^{P/3}(\theta)$$

$$E_{\text{dist}}(\theta) \leq E_{\text{ref}}$$

$$\theta_L^k \leq \theta_k \leq \theta_U^k$$

(6.2)

In Equation 6.2, $E_{ECEP}^{P/3}(\theta)$ is the ECEPP/3 energy of the protein described previously. $\theta_L^k$ and $\theta_U^k$ represent the lower and upper bounds on any dihedral angle $\theta_k$. The distance penalty term $E_{\text{dist}}^l$, for a given conformation $\theta$ can be written out as a combination of lower and upper distance penalty terms given by:

$$E_{\text{dist}}^L = \sum_j \left\{ \begin{array}{ll}
A_L^j (d_j - d_L^j)^2 & \text{if } d_j < d_L^j \\
0 & \text{otherwise}
\end{array} \right. \quad (6.3)$$

$$E_{\text{dist}}^U = \sum_j \left\{ \begin{array}{ll}
A_U^j (d_j - d_U^j)^2 & \text{if } d_j < d_U^j \\
0 & \text{otherwise}
\end{array} \right. \quad (6.4)$$

### 6.3.3 Deterministic Global Optimization using $\alpha$BB

In order to solve the constrained minimization problem given by Equation 6.2, we need to employ global optimization based search techniques. One such global optimization technique, which
avoids dependence on initial conditions and search heuristics, is the $\alpha$BB global optimization approach [1, 2, 3, 4, 7]. The algorithm is a deterministic global optimization approach, which provides theoretical guarantee of convergence to the global optimum solution for problems with twice-continuously differentiable objective functions and constraints. The $\alpha$BB global optimization approach guarantees convergence to an $\epsilon$-global optimum solution by creating a sequence of non-decreasing lower bounds, along with a sequence of non-increasing upper bounds on the optimum value. Eventual convergence of these sequences lead to the identification of the global optimum. The method has been applied successfully to the problem of protein structure prediction previously [104, 131], and only the key aspects of the algorithm are highlighted.

The lower bounding problems are constructed by augmenting the objective function and constraints with separable quadratic functions. Mathematically, the lower bounding function is represented as:

$$
\min_{\theta} \quad L_{ECEP/3}(\theta) \\
L_{\text{dist}}(\theta) \leq E_{\text{ref}} \\
\theta^L_k \leq \theta_k \leq \theta^U_k
$$

The term $L_{ECEP/3}(\theta)$ refers to the convex lower bounding function representation of the objective function, and is expressed as shown in Equation (6.6). The $\alpha$-parameters represent non-negative parameters which must be greater than or equal to negative one-half of the minimum eigenvalue of the hessian of the original energy function over the defined domain [59]. Mathematically, the aim of the additional quadratic terms is to overpower the non-convexities of the original terms by adding a value of $2\alpha$ to the eigenvalues of the hessian of the original energy function. Given solutions to the lower and upper bounding problems, the algorithm branches on the subproblem which holds the infimum of all the lower bounding function values. This ensures
that we get a series of non-decreasing lower bounds. The series of non-decreasing upper bounds is determined by identifying the protein structure with the minimum energy value. Any region where the lower bounding energy value exceeds the best current upper bound can be safely fathomed, as the global minimum would definitely not be present in this region.

\[ L_{ECEPP/3}(\theta) = E_{ECEPP/3}(\theta) + \sum_{i=1}^{N_g} \alpha_{\theta_i}(\theta_{L_i}^i - \theta_i)(\theta_{U_i}^i - \theta_i) \]  

(6.6)

### 6.3.4 Torsion Angle Dynamics

Prior to the implementation of the deterministic global optimization, it is vital to get initial structures which fall into the feasible space of the optimization problem. Here, the feasible space of the problem is defined by the dihedral angle and distance bounds generated in the previous sections. Various algorithms have been used for the problem of identifying structures which satisfy a sparse set of distance and dihedral angle constraints. For protein structure prediction problems, distance geometry algorithms like EMBED [41] and dgsol [139] have been used to produce feasible initial structures. In addition to distance geometry methods, a number of other algorithms like variable target methods [76] and molecular dynamics [5] have also been employed for this problem. A detailed review on algorithms for constrained protein structure determination is available elsewhere [74].

The ASTRO-FOLD framework involves an interface with the torsion angle dynamics package CYANA [75]. By fixing the covalent bonds and bond angles to their mean values, the torsion angle dynamics package works in the dihedral angle space, thus reducing the number of variables drastically. Further, unlike target minimization, molecular dynamics allows itself the possibility of overcoming energy barriers, due to the presence of kinetic energy. Unlike classical molecular dynamics simulations, the torsion angle dynamics algorithms combine steric clashes-based energy terms and constraint-based penalties in a simplified target function. This allows for faster calcula-
tions, and results in the algorithm aiming to identify structures which are fairly low in energy, but are more importantly, feasible. Algorithmic implementation details of the initial point selection can be found elsewhere [131].

6.3.5 Conformational Space Annealing

In conjunction with the αBB deterministic global optimization approach presented, we can use stochastic or heuristic search techniques in order to improve the search process for the identification of low energy conformations. While the lower bounding problem, as presented in the αBB algorithm provides the theoretical guarantee to create a non-decreasing sequence of lower bounds which would approach the global optimum solution from below, we can integrate heuristic search techniques into the process of creating the sequence of non-increasing upper bounds. This would provide multiple advantages. Firstly, a faster, albeit stochastic, method would identify new regions of the search space which may hold the global minimum solution. In addition, by identifying such regions and their corresponding upper bounds, one can fathom other regions of the space where the minimum structure is such that the lower bounding function has an energy value greater than the best current upper bound.

One such algorithm is Conformational Space Annealing (CSA) [42, 110, 111], proposed by Scheraga and co-workers. While the CSA approach lacks theoretical guarantees, given its high efficiency, a hybrid implementation of the αBB and CSA would be highly favorable.

Starting with a bank ($N_{\text{bank}}$) of conformers generated by the αBB global optimization algorithm, a distance metric for evaluating separation between structures $i$ and $j$ in the bank is given by:

$$D_{ij} = \sum_{k=1}^{N_{\phi}} |\phi_{i}^k - \phi_{j}^k|$$  \hspace{1cm} (6.7)
where $N_\phi$ is the number of dihedral angles of the protein, and $\phi_{i}^{k}, \phi_{j}^{k}$ are the $k^{th}$ dihedral angle of conformers $i$ and $j$, respectively. Given this definition of distance, the average distance between structures in the bank ($D_{\text{avg}}$) can be evaluated by:

$$D_{\text{avg}} = \frac{1}{\binom{N_{\text{bank}}}{2} \cdot (N_{\text{bank}} - 1)} \sum_{i} \sum_{j > i} D_{ij}$$  \hspace{1cm} (6.8)

The conformational space is searched using heuristics based on genetic algorithms. This involves alteration of conformers which exist in the bank using two heuristic modifications: mutations and crossover operations. The mutation operation identifies between one and four $\phi, \psi$ and $\xi_1$ dihedral angles, and changes their values to the ones held by another conformer in the bank. Simultaneously, the crossover operation replaces a randomly selected continuous range of dihedral angles (between 1/8 and 1/4 of the total number of dihedral angles) from a given conformer with the values from a second conformer. Once a mutation or crossover operation is carried out, the new conformer is subjected to local minimization. In order to ensure that any new structures identified by the genetic algorithm does not fall too close to a structure already existing in the bank, a “radius of influence”, $D_{\text{cut}}$, is determined. This ensures that the conformer bank does not become too biased towards a specific region in the search space too early. Further implementation details of the hybrid $\alpha$BB/CSA algorithm can be found elsewhere \cite{131}.

At the end of the tertiary structure prediction stage, a large ensemble of predicted structures is generated. In order to identify the structures nearest to the native from the given ensemble, we use ICON, a near-native structure identification algorithm based on the traveling salesman problem \cite{198}. The details of this algorithm have been presented in Chapter 5.
6.3.6 Improved distance and dihedral bound generation using chemical shifts

Chemical shift information is widely used for protein structure prediction [40, 115, 133, 144, 181, 182, 183, 216]. This is based on the fact that chemical shifts of protein backbone atoms are very sensitive to the local structures of the protein, thus, chemical shifts can help protein structure prediction in various ways. Shen et al. developed a protocol Chemical-Shift-Rosetta (CS-Rosetta), to study the influence of the completeness of chemical shifts on protein structure prediction [183].

TALOS [40] is an algorithm for backbone dihedral angle prediction using chemical shift database information. The database consists of amino acid triplets with corresponding secondary chemical shift and sequence information. By searching the best match of triplet of the query protein against the database, the dihedral angles can be predicted using a consensus scheme of the top 10 matches. The test shows that the predicted dihedral angles are on average 15 degree from the X-ray derived backbone angles. TOUCHSTONEX [115] predicts protein structure by using some long-range distance restraints derived from NMR experimental data (including chemical shifts, NOE contacts, slow amide protein exchange etc.). A NOE-specific pairwise potential is incorporated to tackle the NMR experimental data-derived constraints. 108 of 125 test proteins are folded below 6.5 Å of Cα RMSD.

In ASTRO-FOLD 2.0, the structures resulting from the clustering are subject to the SPARTA [181] algorithm which predicts the backbone chemical shifts from tertiary structure. The predicted chemical shifts are then used by CS23D (Chemical Shift to 3D structure) [216] to predict the protein 3D structure. SPARTA [181] predicts backbone chemical shifts for a given protein structure by searching a database of amino acid triplets with chemical shift data of $^{15}$N, $^1$H$^N$, $^1$H$^\alpha$, $^{13}$C$^\alpha$, $^{13}$C$^b$ and $^{13}$C$'\alpha$ atoms. The triplet database of SPARTA is expanded by adding more proteins from Biological Magnetic Resonance Data Bank (BMRB). The same procedure is used for adding triplet into the database as in Ref [181]. For example, completeness of chemical shifts data for $^1$H$^\alpha$, $^{13}$C$^\alpha$,
\[^{13}\text{C}^b\] and \[^{13}\text{C}'\] is checked by ensuring at least 4 of 5 chemical shifts for these five atoms should exist. For Glycine and Proline, 3 out of 4 chemical shifts should exist in order to be added into the database; Only PDBs with 2.4 \(\text{Å}\) resolution or less are selected for analysis from the BMRB.

CS23D predicts the protein structure given the protein backbone chemical shift information and sequence information. No NOE or J-coupling information is needed for CS23D to predict the protein structure. It consists of several steps involving maximal sub-fragment assembly, chemical shift threading or chemical shift based de novo structure prediction, chemical shift refinement. The performance of CS23D is dependent on the completeness and correctness of the chemical shift data and the sequence similarity between the query protein and the proteins in the database.

Both SPARTA and CS23D are installed locally [181, 216]. By generating the modified protein structure from CS23D, we identify improved distance constraints between pairs of amino acids. In addition, consensus methods between the original dihedral angle constraints and the angles of structures generated by CS23D are taken to generate improved dihedral angle bounds. These improved constraints are used to re-run the tertiary structure prediction algorithm discussed previously. The final ensemble of collected structures is compiled and clustered using the novel traveling salesman problem-based clustering algorithm ICON, shown in the previous section. From this stage, the final subset of structures closest to the native are identified.

### 6.3.7 Parallel implementation of ASTROFOLD 2.0 framework

The entire ASTROFOLD 2.0 framework has been implemented using an automated approach. The entire algorithm has been written out using a Python 2.3 script. Each stage of the ASTROFOLD 2.0 framework, presented in Figure 6.1 has been implemented as a callable function in this script, thus permitting the addition of new intermediate stages into the ASTROFOLD framework. An overview of the implementation details of the tertiary structure prediction algorithm is presented below.
The transformation between cartesian coordinate representation and dihedral angle representation of a protein conformer is carried out using the PACK package. The package also provides energy, gradient and hessian calculations using the ECEPP/3 force field. The torsion angle dynamics routine has been implemented using the CYANA package [75], which was directly interfaced with the implementation of the tertiary structure algorithm. The local minimization at each $\alpha$BB and CSA stage was carried out using the NPSOL package [69], which is a sequential quadratic programming implementation suitable to the protein structure prediction problem. Each individual stage of the ASTROFOLD 2.0 framework has been written using C++ and the standard template library (STL). This alleviates the instances of memory tampering due to user interference or memory corruption, while providing a number of standard algorithms and storage containers for efficient memory usage and recovery. The entire ASTROFOLD framework has been implemented in parallel on a number of distributed memory computer clusters available. The parallel implementation uses the message passing interface (MPI) protocol to communicate between processors. The parent node maintains a register of all active jobs at any instance. It collects outputs from all worker nodes, which are divided between $\alpha$BB and CSA jobs. New jobs are assigned to the worker nodes depending on the status of the $\alpha$BB and CSA queues.

### 6.4 Computational Studies

A major test of protein structure prediction methods is done through a biennial world wide competition, Critical Assessment of Techniques for Protein Structure Prediction (CASP). In this section, several examples of CASP9 prediction (http://predictioncenter.org/casp9/) using the ASTRO-FOLD 2.0 framework will be presented. Detailed analysis of secondary structure prediction is presented in terms of the three-state (Q3) prediction accuracy, contact prediction is evaluated by its prediction accuracy, and tertiary structure prediction is evaluated by
Chapter 6. Tertiary structure prediction: Computational Studies

Root Mean Square Deviation (RMSD), Template Modeling (TM) score, and Global Distance Test (GDT) score of the Cα atoms from the native structure.

Forty seven tertiary structure predictions made by ASTRO-FOLD 2.0 during CASP9 are presented in this section. The identification of domains for analysis was carried out by the organizers of CASP9. The detailed structure evaluations are listed in Table 6.1. In this table, RMSD, GDT and TM scores are listed first for the top submitted prediction of the ASTRO-FOLD 2.0 models, followed by the same information for the best of the five ASTRO-FOLD 2.0 submitted models. As it is shown in this section, ASTRO-FOLD 2.0 successfully predicts good quality structures and substructures for a number of proteins. In this section, a few of the protein structures predicted by ASTRO-FOLD 2.0 are selected for detailed analysis. We present the results from the secondary structure and contact prediction algorithms, along with the tertiary structure prediction results, thus representing the impact and importance of these intermediate algorithms towards the final 3D prediction algorithm.

Table 6.1: Structure evaluations of CASP9 targets for ASTRO-FOLD 2.0.

<table>
<thead>
<tr>
<th>Prot</th>
<th>Top 1 model</th>
<th></th>
<th>Best model</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GDT</td>
<td>TM</td>
<td>RMSD</td>
<td>GDT</td>
</tr>
<tr>
<td>T531-D1</td>
<td>0.25</td>
<td>0.19</td>
<td>11.94</td>
<td>0.28</td>
</tr>
<tr>
<td>T544-D1</td>
<td>0.19</td>
<td>0.23</td>
<td>15.14</td>
<td>0.26</td>
</tr>
<tr>
<td>T553-D1</td>
<td>0.40</td>
<td>0.32</td>
<td>7.53</td>
<td>0.44</td>
</tr>
<tr>
<td>T553-D2</td>
<td>0.32</td>
<td>0.29</td>
<td>12.11</td>
<td>0.38</td>
</tr>
<tr>
<td>T561-D1</td>
<td>0.31</td>
<td>0.39</td>
<td>11.66</td>
<td>0.31</td>
</tr>
<tr>
<td>T578-D1</td>
<td>0.23</td>
<td>0.24</td>
<td>17.66</td>
<td>0.23</td>
</tr>
<tr>
<td>T581-D1</td>
<td>0.36</td>
<td>0.37</td>
<td>12.08</td>
<td>0.36</td>
</tr>
<tr>
<td>T616-D1</td>
<td>0.35</td>
<td>0.31</td>
<td>14.75</td>
<td>0.36</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>----------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>T618-D1</td>
<td>0.21</td>
<td>0.26</td>
<td>18.66</td>
<td>0.26</td>
</tr>
<tr>
<td>621-D1</td>
<td>0.13</td>
<td>0.21</td>
<td>19.08</td>
<td>0.18</td>
</tr>
<tr>
<td>T624-D1</td>
<td>0.24</td>
<td>0.19</td>
<td>12.08</td>
<td>0.34</td>
</tr>
<tr>
<td>T517-D1</td>
<td>0.35</td>
<td>0.46</td>
<td>14.48</td>
<td>0.35</td>
</tr>
<tr>
<td>T520-D1</td>
<td>0.60</td>
<td>0.72</td>
<td>3.98</td>
<td>0.60</td>
</tr>
<tr>
<td>T523-D1</td>
<td>0.62</td>
<td>0.67</td>
<td>5.16</td>
<td>0.62</td>
</tr>
<tr>
<td>T540-D1</td>
<td>0.27</td>
<td>0.27</td>
<td>12.68</td>
<td>0.36</td>
</tr>
<tr>
<td>T562-D1</td>
<td>0.35</td>
<td>0.41</td>
<td>10.80</td>
<td>0.35</td>
</tr>
<tr>
<td>T564-D1</td>
<td>0.43</td>
<td>0.40</td>
<td>13.40</td>
<td>0.43</td>
</tr>
<tr>
<td>T566-D1</td>
<td>0.34</td>
<td>0.43</td>
<td>7.90</td>
<td>0.50</td>
</tr>
<tr>
<td>T568-D1</td>
<td>0.25</td>
<td>0.28</td>
<td>12.31</td>
<td>0.28</td>
</tr>
<tr>
<td>T569-D1</td>
<td>0.22</td>
<td>0.20</td>
<td>12.89</td>
<td>0.73</td>
</tr>
<tr>
<td>T574-D1</td>
<td>0.40</td>
<td>0.41</td>
<td>7.54</td>
<td>0.40</td>
</tr>
<tr>
<td>T576-D1</td>
<td>0.16</td>
<td>0.20</td>
<td>17.59</td>
<td>0.35</td>
</tr>
<tr>
<td>T579-D1</td>
<td>0.53</td>
<td>0.40</td>
<td>4.94</td>
<td>0.53</td>
</tr>
<tr>
<td>T579-D2</td>
<td>0.39</td>
<td>0.36</td>
<td>10.09</td>
<td>0.39</td>
</tr>
<tr>
<td>T580-D1</td>
<td>0.89</td>
<td>0.91</td>
<td>1.37</td>
<td>0.89</td>
</tr>
<tr>
<td>T582-D1</td>
<td>0.23</td>
<td>0.28</td>
<td>10.00</td>
<td>0.35</td>
</tr>
<tr>
<td>T584-D1</td>
<td>0.28</td>
<td>0.43</td>
<td>23.07</td>
<td>0.28</td>
</tr>
<tr>
<td>T586-D1</td>
<td>0.79</td>
<td>0.79</td>
<td>2.18</td>
<td>0.79</td>
</tr>
<tr>
<td>T586-D2</td>
<td>0.81</td>
<td>0.64</td>
<td>2.87</td>
<td>0.81</td>
</tr>
<tr>
<td>T590-D1</td>
<td>0.74</td>
<td>0.70</td>
<td>2.59</td>
<td>0.74</td>
</tr>
<tr>
<td>T592-D1</td>
<td>0.56</td>
<td>0.64</td>
<td>8.69</td>
<td>0.60</td>
</tr>
<tr>
<td>T594-D1</td>
<td>0.40</td>
<td>0.48</td>
<td>7.00</td>
<td>0.53</td>
</tr>
</tbody>
</table>
6.1 T581:3NPD

Target T581 has been categorized as a free modeling target (PDB code: 3NPD). 3NPD is a one-chain protein with 112 amino acids having specified native coordinates (amino acids 20 to 131). Out of the 136 amino acids of T581, only amino acids 20 to 131 are used for evaluation.

As shown in Figure 6.2, there are five strands in T581 forming a beta sheet with all neighboring strands contacting in anti-parallel fashion. In the native structure, the positions of the five $\beta$ strands are: strand 1 from amino acid 44 to 45, strand 2 from amino acid 50 to 58, strand 3 from amino

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T596-D1</td>
<td>0.83</td>
<td>0.72</td>
<td>1.60</td>
<td>0.85</td>
<td>0.74</td>
</tr>
<tr>
<td>T596-D2</td>
<td>0.51</td>
<td>0.54</td>
<td>4.92</td>
<td>0.66</td>
<td>0.72</td>
</tr>
<tr>
<td>T598-D1</td>
<td>0.50</td>
<td>0.55</td>
<td>5.81</td>
<td>0.50</td>
<td>0.55</td>
</tr>
<tr>
<td>T602-D1</td>
<td>0.72</td>
<td>0.57</td>
<td>2.87</td>
<td>0.72</td>
<td>0.61</td>
</tr>
<tr>
<td>T605-D1</td>
<td>0.85</td>
<td>0.73</td>
<td>1.61</td>
<td>0.87</td>
<td>0.77</td>
</tr>
<tr>
<td>T606-D1</td>
<td>0.57</td>
<td>0.62</td>
<td>7.91</td>
<td>0.57</td>
<td>0.62</td>
</tr>
<tr>
<td>T610-D1</td>
<td>0.62</td>
<td>0.71</td>
<td>6.43</td>
<td>0.62</td>
<td>0.71</td>
</tr>
<tr>
<td>T612-D1</td>
<td>0.40</td>
<td>0.39</td>
<td>8.36</td>
<td>0.40</td>
<td>0.42</td>
</tr>
<tr>
<td>T614-D1</td>
<td>0.30</td>
<td>0.29</td>
<td>12.47</td>
<td>0.46</td>
<td>0.41</td>
</tr>
<tr>
<td>T619-D1</td>
<td>0.73</td>
<td>0.78</td>
<td>2.22</td>
<td>0.77</td>
<td>0.83</td>
</tr>
<tr>
<td>T622-D1</td>
<td>0.51</td>
<td>0.55</td>
<td>7.32</td>
<td>0.53</td>
<td>0.62</td>
</tr>
<tr>
<td>T625-D1</td>
<td>0.29</td>
<td>0.40</td>
<td>13.66</td>
<td>0.29</td>
<td>0.40</td>
</tr>
<tr>
<td>T627-D1</td>
<td>0.40</td>
<td>0.60</td>
<td>7.58</td>
<td>0.40</td>
<td>0.60</td>
</tr>
<tr>
<td>T629-D1</td>
<td>0.27</td>
<td>0.21</td>
<td>10.55</td>
<td>0.44</td>
<td>0.34</td>
</tr>
<tr>
<td>T630-D1</td>
<td>0.51</td>
<td>0.55</td>
<td>5.45</td>
<td>0.51</td>
<td>0.55</td>
</tr>
</tbody>
</table>
acid 61 to 68, strand 4 from amino acid 106 to 113 and strand 5 from amino acid 119 to 125. In addition, the protein contains five helices located at: helix 1 from amino acid 22 to 39, helix 2 from amino acid 70 to 78, helix 3 from amino acid 80 to 92, helix 4 from amino acid 95 to 101 and helix 5 from amino acid 127 to 130. These five helices are on one side of the 5-strand sheet, thus forming a hydrophobic-core like region between the two secondary structure layers. The interacting pairs of the helices are between helix 1 and helix 2 (N-terminal to C-terminal), between helix 2 and helix 3 (C-terminal to N-terminal), between helix 3 and helix 4 (C-terminal to the middle), between helix 3 and helix 5 (the middle to C-terminal).

The secondary structure prediction for T581 has a relatively low prediction accuracy (59.8%). This can be attributed to the fact that the protein is categorized as a free modeling target, thus having a low sequence similarity to the protein database. This would result in varied predictions of secondary structures by multiple methods, thus resulting in an unsatisfactory consensus result. The predicted secondary structure information is listed in Table 6.2.

As for the helical prediction, predicted helix 1 can not be evaluated because the first 19 amino acids of T581 do not have coordinates in native structure. The prediction of the two helices at the
Table 6.2: Predicted secondary structure information for T581 by ASTRO-FOLD 2.0.

<table>
<thead>
<tr>
<th>Helix</th>
<th>Strand</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-12</td>
<td>106-111</td>
</tr>
<tr>
<td>15-35</td>
<td>120-125</td>
</tr>
<tr>
<td>44-52</td>
<td>129-132</td>
</tr>
<tr>
<td>62-77</td>
<td></td>
</tr>
<tr>
<td>81-89</td>
<td></td>
</tr>
<tr>
<td>97-102</td>
<td></td>
</tr>
</tbody>
</table>

C-terminal is accurate, while the predicted helix 3 contains strand 1 of the native structure, and the predicted helix 4 corresponds to strand 3 of the native structure.

While evaluating the contact prediction results (Table 6.3), we observe that ASTRO-FOLD 2.0 has an accuracy of 40% for amino acid pairs that are at least 6 amino acids apart, with average true prediction distance 5.9 Å and average false prediction distance 15 Å. A maximum distance cutoff of 12 Å between two Cα atoms has been used for defining a contact between two amino acids. The distance cutoff of 12 Å has been used to match the distances predicted in the contact prediction model, which uses a high resolution Cα-Cα distance dependent force field for its predictions. The lower accuracy of the contact prediction algorithm can be attributed to inconsistencies in the secondary structure prediction algorithm.

It is worthy to note that the above accuracy is calculated based on the lower and upper distance bounds used by the contact prediction model in ASTRO-FOLD 2.0. If the real distance falls in the range between lower and upper distance bounds, the contact is taken as a true contact, otherwise it is a false contact. In ASTRO-FOLD 2.0, lower and upper distance bounds are used as constraints for tertiary structure prediction. ASTRO-FOLD 2.0 uses 4.5 Å to 6.5 Å for the vertically contacted strand amino acid pairs (if two amino acids on two different interacting strands are closest in space, these two amino acids are denoted as “being vertically contacted”). If lower and upper bounds are set to 0 Å and 12 Å for all the predicted contacts, the contact prediction accuracy is 62.5%.
Table 6.3: Predicted amino acid contacts for T581 of CASP9. Data is shown for amino acid pairs that are at least 6 amino acids apart. First 3 columns show the true predicted contacts whose distances fall between the predicted lower and upper distance bounds; while the last 3 columns show the false predicted contacts where the real distance falls outside the predicted distance range.

<table>
<thead>
<tr>
<th>AA1</th>
<th>AA2</th>
<th>Distance</th>
<th>AA1</th>
<th>AA2</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>106</td>
<td>125</td>
<td>5.458</td>
<td>107</td>
<td>123</td>
<td>8.238</td>
</tr>
<tr>
<td>107</td>
<td>124</td>
<td>5.646</td>
<td>108</td>
<td>122</td>
<td>7.05</td>
</tr>
<tr>
<td>108</td>
<td>123</td>
<td>5.642</td>
<td>109</td>
<td>121</td>
<td>7.79</td>
</tr>
<tr>
<td>109</td>
<td>122</td>
<td>5.57</td>
<td>110</td>
<td>120</td>
<td>7.933</td>
</tr>
<tr>
<td>110</td>
<td>121</td>
<td>6.215</td>
<td>108</td>
<td>124</td>
<td>4.303</td>
</tr>
<tr>
<td>111</td>
<td>120</td>
<td>5.377</td>
<td>106</td>
<td>123</td>
<td>10.02</td>
</tr>
<tr>
<td>106</td>
<td>124</td>
<td>6.445</td>
<td>107</td>
<td>122</td>
<td>10.58</td>
</tr>
<tr>
<td>107</td>
<td>125</td>
<td>5.154</td>
<td>108</td>
<td>121</td>
<td>10.35</td>
</tr>
<tr>
<td>109</td>
<td>123</td>
<td>5.171</td>
<td>109</td>
<td>120</td>
<td>10.47</td>
</tr>
<tr>
<td>110</td>
<td>122</td>
<td>4.634</td>
<td>124</td>
<td>131</td>
<td>13.86</td>
</tr>
<tr>
<td>111</td>
<td>121</td>
<td>5.17</td>
<td>123</td>
<td>131</td>
<td>15.97</td>
</tr>
<tr>
<td>108</td>
<td>125</td>
<td>6.379</td>
<td>124</td>
<td>130</td>
<td>10.83</td>
</tr>
<tr>
<td>109</td>
<td>124</td>
<td>6.421</td>
<td>125</td>
<td>131</td>
<td>10.4</td>
</tr>
<tr>
<td>110</td>
<td>123</td>
<td>6.765</td>
<td>122</td>
<td>131</td>
<td>19.75</td>
</tr>
<tr>
<td>111</td>
<td>122</td>
<td>6.171</td>
<td>123</td>
<td>130</td>
<td>12.88</td>
</tr>
<tr>
<td>27</td>
<td>62</td>
<td>10.33</td>
<td>27</td>
<td>46</td>
<td>28.12</td>
</tr>
<tr>
<td>50</td>
<td>68</td>
<td>4.65</td>
<td>30</td>
<td>46</td>
<td>25.47</td>
</tr>
<tr>
<td>51</td>
<td>66</td>
<td>5.443</td>
<td>31</td>
<td>46</td>
<td>22.3</td>
</tr>
<tr>
<td>76</td>
<td>83</td>
<td>6.429</td>
<td>27</td>
<td>63</td>
<td>13.93</td>
</tr>
</tbody>
</table>
The top submitted structure from ASTRO-FOLD 2.0 predictions has a RMSD value of 12.08 Å, a GDT score of 36%, and a TM score of 37% from the native. The overlay between the top structure of ASTRO-FOLD 2.0 and the native does not fit well between these two structures due to the high RMSD and low GDT, TM scores. However, when compared to all submitted structures to the CASP9 organizers (including predictions by servers and human expert groups), the ranking of the top submitted structure by ASTRO-FOLD 2.0 is 6, 5 and 16 based on TM, GDT and RMSD, respectively.

A closer analysis shows that for a segment of T581 (amino acids 70 to 128), its RMSD value is 3.9 Å, and its GDT score is 61%. Figure 6.3 shows the overlay of this segment with the corresponding native segment. The overall fit between the predicted segment and the native is much better than the whole protein. This indicates that ASTRO-FOLD 2.0 is able to predict parts of the protein very well even when the overall topology prediction is incorrect. In a number of cases, there are parts of crystallographic structures that are missing due to inconsistencies in experimen-
tal results. The role of ASTRO-FOLD in modeling small subsections accurately could be utilized for identification of these regions. This is particularly relevant for free modeling targets where we observe that most methods which depend on the sequence or structural database (for alignment, threading or fragment assembly) are unable to provide structures of very high quality.

![Native structure of T581 segment (70-128 amino acids) shown in gray. Top 1 model segment (70-128 amino acids) from ASTRO-FOLD 2.0 is colored in rainbow](image)

Figure 6.3: Native structure of T581 segment (70-128 amino acids) shown in gray. Top 1 model segment (70-128 amino acids) from ASTRO-FOLD 2.0 is colored in rainbow

### 6.4.2 T602:3NKZ

Target T602 of CASP9 corresponds to protein 3NKZ. 3NKZ has four identical chains with 3 helices on each chain. T602 sequence has 123 amino acids while each chain of protein 3NKZ has only the first 97 amino acids available. Thus, the evaluations of secondary structure prediction, contact prediction and tertiary structure prediction are based on the 97 amino acids only. The 3 helices of 3NKZA form a plane with helices 1 and 2 anti-parallel, helices 2 and 3 anti-parallel (see Figure 6.4).

The secondary structure prediction resulted in 4 helices for T602 (123 amino acids), and they are: helix 1 from amino acid 4 to amino acid 30; helix 2 from amino acid 34 to amino acid 51; helix 3 from amino acid 63 to amino acid 96; helix 4 from amino acid 100 to amino acid 110.
Figure 6.4: Native structure of T602 (3NKZA) shown in gray. Helix 1: Amino acid 3 to 28; Helix 2: Amino acid 35 to 56; Helix 3: Amino acid 61 to 96. Top 1 model from ASTRO-FOLD 2.0 is colored in rainbow.

By excluding the amino acids beyond amino acid 97, the secondary structure prediction has a Q3 prediction accuracy 88.7%.

Contact prediction shows an average accuracy of 42.9% for amino acid pairs that are at least 6 apart in protein T602. The detailed list of predicted contacts for T602 is presented in Table 6.4.

As can be seen from Table 6.4, there are five false and seven true predicted contacts between helix 1 and helix 2, while for helix 2 and helix 3, there are only 2 true contact predictions, and 7 false predicted contacts. One thing to note from Figure 6.4 is the 3 helices of T602 are not compact and it is an open structure. The top submitted structure of ASTRO-FOLD 2.0 has a RMSD value of 2.18 Å from the native, a GDT score of 72% from the native, and a TM score of 57% from the native. ASTRO-FOLD 2.0 was able to predict the structure for T602 with 2.18 Å RMSD to the native. The overlay between the top 1 model and the native is shown in Figure 6.4. In this figure, native structure is presented in gray and top 1 model is colored in rainbow. As can be seen from Figure 6.4, the overall topology of the first submitted structure is the same as the native, which is also indicated by its above 70% GDT score to the native.
Table 6.4: Predicted amino acid contacts for T602 of CASP9. Data is shown for amino acid pairs that are at least 6 amino acids apart. First 3 columns show the true predicted contacts whose distances fall between the predicted lower and upper distance bounds; while the last 3 columns show the false predicted contacts where the real distance falls outside the predicted distance range.

<table>
<thead>
<tr>
<th>AA1</th>
<th>AA2</th>
<th>Distance</th>
<th>AA1</th>
<th>AA2</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>39</td>
<td>10.64</td>
<td>17</td>
<td>36</td>
<td>14.14</td>
</tr>
<tr>
<td>20</td>
<td>36</td>
<td>9.97</td>
<td>18</td>
<td>36</td>
<td>14.55</td>
</tr>
<tr>
<td>20</td>
<td>39</td>
<td>7.09</td>
<td>18</td>
<td>39</td>
<td>12.19</td>
</tr>
<tr>
<td>24</td>
<td>37</td>
<td>8.65</td>
<td>25</td>
<td>41</td>
<td>14.80</td>
</tr>
<tr>
<td>24</td>
<td>41</td>
<td>11.55</td>
<td>27</td>
<td>41</td>
<td>15.11</td>
</tr>
<tr>
<td>25</td>
<td>37</td>
<td>11.63</td>
<td>43</td>
<td>66</td>
<td>22.92</td>
</tr>
<tr>
<td>27</td>
<td>37</td>
<td>9.95</td>
<td>43</td>
<td>67</td>
<td>19.89</td>
</tr>
<tr>
<td>44</td>
<td>71</td>
<td>11.11</td>
<td>44</td>
<td>66</td>
<td>19.46</td>
</tr>
<tr>
<td>47</td>
<td>71</td>
<td>10.13</td>
<td>44</td>
<td>67</td>
<td>16.60</td>
</tr>
<tr>
<td>47</td>
<td>67</td>
<td>17.80</td>
<td>47</td>
<td>67</td>
<td>14.44</td>
</tr>
</tbody>
</table>

6.4.3 T562:2KZX

Target T562 has 123 amino acids and the corresponding PDB code is 2KZX. 2KZX is a NMR structure and has only 1 chain (A). 2KZX is a mixed $\alpha/\beta$ protein with 3 strands in the first 45 amino acids forming a beta sheet, and 3 helices in the rest part of protein. The 3 helices surround the beta sheet, see the gray structure in Figure 6.5.

The secondary structure prediction has a Q3 prediction accuracy of 70%. The detailed secondary structure prediction, together with the native secondary structure information are listed in Table 6.5.

As shown in the table, the second and third helices are correctly predicted, while the first helix is mispredicted. For the beta strand prediction, all the three native strands are correctly predicted.

Contact prediction for T562 has an accuracy of 35.7% for predicted contacts of amino acids that are at least 6 amino acids apart using predicted lower and upper distance bounds as distance cutoff. If 0 Å and 12 Å are used as lower and upper bounds, the prediction accuracy increases
Figure 6.5: Native structure of T562 shown in gray. Top 1 model from ASTRO-FOLD 2.0 is colored in rainbow.

Table 6.5: Predicted secondary structure information for T562 by ASTRO-FOLD 2.0 and the native secondary structure information are shown in this table.

<table>
<thead>
<tr>
<th>Helix (Predicted)</th>
<th>Helix (Native)</th>
<th>Strand (Predicted)</th>
<th>Strand (Native)</th>
</tr>
</thead>
<tbody>
<tr>
<td>47-53</td>
<td>5-10</td>
<td>19-26</td>
<td>18-27</td>
</tr>
<tr>
<td>67-81</td>
<td>72-82</td>
<td>30-38</td>
<td>30-40</td>
</tr>
<tr>
<td>85-89</td>
<td>30-40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>107-121</td>
<td>108-117</td>
<td>56-61</td>
<td></td>
</tr>
</tbody>
</table>

to 62%. A total of 98 contacts are predicted. The anti-parallel beta sheet topology is correctly predicted for T562, as can be seen from the true contacts of Table 6.6. Strand 1 (amino acids 5-9) and strand 2 (amino acids 18-27) interact in an anti-parallel orientation, and the corresponding amino acids pairs from these two strands are predicted to form contacts. For example, amino acids 5 and 26 form a contact with a distance of 6.01 Å; amino acids 9 and 22 form a contact with a distance of 6.92 Å; Since the beginning of strand 1 (amino acid 5) contacts with the end of strand 2 (amino acid 26), and the beginning of strand 2 (amino acid 22) contacts with the end of strand 1 (amino acid 9), these two strands form an anti-parallel contact. A similar observation can be made to strands 2 and 3 as well.
Table 6.6: Predicted amino acid contacts for T562 of CASP9. Data is shown for amino acid pairs that are at least 6 amino acids apart. First 3 columns show the true predicted contacts whose distances fall between the predicted lower and upper distance bounds; while the last 6 columns show the false predicted contacts where the real distance falls outside the predicted distance range.

<table>
<thead>
<tr>
<th>AA1</th>
<th>AA2</th>
<th>Distance</th>
<th>AA1</th>
<th>AA2</th>
<th>Distance</th>
<th>AA1</th>
<th>AA2</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>26</td>
<td>6.01</td>
<td>6</td>
<td>26</td>
<td>7.83</td>
<td>30</td>
<td>61</td>
<td>37.83</td>
</tr>
<tr>
<td>6</td>
<td>25</td>
<td>6.63</td>
<td>7</td>
<td>25</td>
<td>8.71</td>
<td>31</td>
<td>60</td>
<td>33.43</td>
</tr>
<tr>
<td>7</td>
<td>24</td>
<td>6.09</td>
<td>8</td>
<td>24</td>
<td>9.01</td>
<td>32</td>
<td>59</td>
<td>32.82</td>
</tr>
<tr>
<td>9</td>
<td>22</td>
<td>6.92</td>
<td>9</td>
<td>23</td>
<td>8.94</td>
<td>33</td>
<td>58</td>
<td>26.50</td>
</tr>
<tr>
<td>10</td>
<td>21</td>
<td>6.06</td>
<td>10</td>
<td>22</td>
<td>9.02</td>
<td>34</td>
<td>57</td>
<td>23.27</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>4.67</td>
<td>8</td>
<td>23</td>
<td>6.90</td>
<td>35</td>
<td>56</td>
<td>17.39</td>
</tr>
<tr>
<td>6</td>
<td>24</td>
<td>5.05</td>
<td>7</td>
<td>26</td>
<td>10.91</td>
<td>32</td>
<td>61</td>
<td>33.67</td>
</tr>
<tr>
<td>8</td>
<td>22</td>
<td>5.80</td>
<td>8</td>
<td>25</td>
<td>12.17</td>
<td>33</td>
<td>60</td>
<td>29.69</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>5.46</td>
<td>9</td>
<td>24</td>
<td>11.86</td>
<td>34</td>
<td>59</td>
<td>25.70</td>
</tr>
<tr>
<td>19</td>
<td>38</td>
<td>5.17</td>
<td>10</td>
<td>23</td>
<td>11.92</td>
<td>35</td>
<td>58</td>
<td>19.69</td>
</tr>
<tr>
<td>22</td>
<td>35</td>
<td>6.49</td>
<td>7</td>
<td>23</td>
<td>4.23</td>
<td>36</td>
<td>57</td>
<td>16.69</td>
</tr>
<tr>
<td>23</td>
<td>34</td>
<td>5.62</td>
<td>9</td>
<td>21</td>
<td>4.26</td>
<td>37</td>
<td>56</td>
<td>12.41</td>
</tr>
<tr>
<td>24</td>
<td>33</td>
<td>6.36</td>
<td>8</td>
<td>26</td>
<td>14.14</td>
<td>30</td>
<td>60</td>
<td>36.98</td>
</tr>
<tr>
<td>25</td>
<td>32</td>
<td>6.15</td>
<td>9</td>
<td>25</td>
<td>15.03</td>
<td>31</td>
<td>59</td>
<td>32.81</td>
</tr>
<tr>
<td>25</td>
<td>31</td>
<td>5.95</td>
<td>10</td>
<td>24</td>
<td>14.72</td>
<td>32</td>
<td>58</td>
<td>30.23</td>
</tr>
<tr>
<td>21</td>
<td>37</td>
<td>5.34</td>
<td>20</td>
<td>37</td>
<td>6.51</td>
<td>33</td>
<td>57</td>
<td>26.30</td>
</tr>
<tr>
<td>22</td>
<td>36</td>
<td>5.80</td>
<td>21</td>
<td>36</td>
<td>6.80</td>
<td>34</td>
<td>56</td>
<td>21.16</td>
</tr>
<tr>
<td>23</td>
<td>35</td>
<td>5.00</td>
<td>19</td>
<td>37</td>
<td>8.68</td>
<td>33</td>
<td>61</td>
<td>29.95</td>
</tr>
<tr>
<td>24</td>
<td>34</td>
<td>4.59</td>
<td>20</td>
<td>36</td>
<td>8.05</td>
<td>34</td>
<td>60</td>
<td>26.21</td>
</tr>
</tbody>
</table>
The 3D structure prediction of ASTRO-FOLD 2.0 generates a top 1 model with a RMSD value of 10.8 Å, a TM score of 40.9%, and a GDT score of 34.8% from the native structure. The corresponding rankings of the top 1 model compared with all other human and server predictions are 19, 3, and 4, respectively. As shown from Figure 6.5, the overall topology of the prediction is fairly similar to the native, but not close enough to generate RMSD values lower than 6 Å. This can be attributed to the inconsistencies in secondary structure prediction, which led to some incorrect contact predictions.
The segment analysis shows that for a segment of T562 (first 43 amino acids), the top 1 predicted model of ASTRO-FOLD 2.0 has a RMSD value of 4.9 Å with a GDT score of 54.7%, and a TM score of 37.2%. The overlay structure between the segment of the top 1 model of ASTRO-FOLD 2.0 predictions and the corresponding segment of the native structure is displayed in Figure 6.6. As shown in this figure, the two structures are well fitted to each other. This segment is a 3-strand-beta sheet structure. As described earlier, these 3 strands form anti-parallel contacts with each other and the contact predictions between these 3 strands are correct. This is why ASTRO-FOLD 2.0 predicted this segment with a low RMSD value (below 5 Å), thus reflecting the importance of the intermediate contact prediction algorithm in the larger picture of tertiary structure prediction.

Figure 6.6: Native structure of T562 segment (first 43 amino acids) shown in gray. Top 1 model from ASTRO-FOLD 2.0 is colored in rainbow

6.4.4 T580:3NBM

T580 has 105 amino acids, and it is a mixed α/β protein with 4 strands and 5 helices. Its PDB code is 3NBM. 3NBM has only one chain (A). Only the first amino acid does not have coordinates in the PDB structure, thus total 104 amino acids are used for evaluation. The four strands form one sheet in the center of the protein with all pairs parallel, and five helices lie on both sides of the
sheet in an alternative fashion, see Figure 6.7. The four strands of T580 are: strand 1 from amino acid 4 to 10; strand 2 from amino acid 34 to 40; strand 3 from amino acid 53 to 56 and strand 4 from amino acid 77 to 80. The five helices are: helix 1 from amino acid 15 to 30; helix 2 from amino acid 48 to 50; helix 3 from amino acid 64 to 71; helix 4 from amino acid 83 to 90 and helix 5 from amino acid 93 to 102.

Figure 6.7: Native structure of T580 shown in gray. Top 1 model from ASTRO-FOLD 2.0 is colored in rainbow.

The contact prediction of T580 has an accuracy of 51.4% for amino acid pairs that are at least 6 amino acids apart. If lower and upper distance bounds are set to 0 Å and 12 Å, respectively, the contact prediction accuracy for T580 increases to 70.8%. The analysis of the predicted contacts (see Table 6.7) shows that the parallel sheet topology is correctly predicted.

Table 6.7: Predicted amino acid contacts for T580 of CASP9. Data is shown for amino acid pairs that are at least 6 amino acids apart. First 3 columns show the true predicted contacts whose distances fall between the predicted lower and upper distance bounds; while the last 3 columns show the false predicted contacts where the real distance falls outside the predicted distance range.

<table>
<thead>
<tr>
<th>AA1</th>
<th>AA2</th>
<th>Distance</th>
<th>AA1</th>
<th>AA2</th>
<th>Distance</th>
</tr>
</thead>
</table>


<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>34</td>
<td>6.04</td>
<td>6</td>
<td>34</td>
<td>7.38</td>
</tr>
<tr>
<td>6</td>
<td>35</td>
<td>6.08</td>
<td>7</td>
<td>35</td>
<td>8.57</td>
</tr>
<tr>
<td>7</td>
<td>36</td>
<td>5.62</td>
<td>8</td>
<td>36</td>
<td>7.12</td>
</tr>
<tr>
<td>8</td>
<td>37</td>
<td>6.15</td>
<td>9</td>
<td>37</td>
<td>8.63</td>
</tr>
<tr>
<td>4</td>
<td>34</td>
<td>4.59</td>
<td>7</td>
<td>34</td>
<td>10.81</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>5.25</td>
<td>8</td>
<td>35</td>
<td>10.75</td>
</tr>
<tr>
<td>7</td>
<td>37</td>
<td>4.89</td>
<td>9</td>
<td>36</td>
<td>10.60</td>
</tr>
<tr>
<td>6</td>
<td>53</td>
<td>4.88</td>
<td>10</td>
<td>37</td>
<td>11.35</td>
</tr>
<tr>
<td>7</td>
<td>54</td>
<td>4.78</td>
<td>8</td>
<td>34</td>
<td>12.90</td>
</tr>
<tr>
<td>8</td>
<td>55</td>
<td>4.73</td>
<td>9</td>
<td>35</td>
<td>14.24</td>
</tr>
<tr>
<td>9</td>
<td>56</td>
<td>5.13</td>
<td>10</td>
<td>36</td>
<td>13.47</td>
</tr>
<tr>
<td>7</td>
<td>53</td>
<td>6.46</td>
<td>6</td>
<td>36</td>
<td>4.46</td>
</tr>
<tr>
<td>8</td>
<td>54</td>
<td>5.80</td>
<td>53</td>
<td>79</td>
<td>7.70</td>
</tr>
<tr>
<td>9</td>
<td>55</td>
<td>6.30</td>
<td>54</td>
<td>80</td>
<td>8.60</td>
</tr>
<tr>
<td>10</td>
<td>56</td>
<td>6.40</td>
<td>55</td>
<td>81</td>
<td>7.40</td>
</tr>
<tr>
<td>5</td>
<td>53</td>
<td>6.60</td>
<td>53</td>
<td>80</td>
<td>10.90</td>
</tr>
<tr>
<td>6</td>
<td>54</td>
<td>5.75</td>
<td>54</td>
<td>81</td>
<td>10.94</td>
</tr>
<tr>
<td>7</td>
<td>55</td>
<td>6.25</td>
<td>19</td>
<td>47</td>
<td>18.84</td>
</tr>
<tr>
<td>8</td>
<td>56</td>
<td>5.97</td>
<td>23</td>
<td>45</td>
<td>22.44</td>
</tr>
<tr>
<td>8</td>
<td>53</td>
<td>8.54</td>
<td>23</td>
<td>47</td>
<td>19.54</td>
</tr>
<tr>
<td>9</td>
<td>54</td>
<td>7.38</td>
<td>23</td>
<td>48</td>
<td>18.17</td>
</tr>
<tr>
<td>10</td>
<td>55</td>
<td>8.93</td>
<td>29</td>
<td>45</td>
<td>31.16</td>
</tr>
<tr>
<td>4</td>
<td>53</td>
<td>7.96</td>
<td>29</td>
<td>48</td>
<td>26.37</td>
</tr>
<tr>
<td>5</td>
<td>54</td>
<td>8.68</td>
<td>60</td>
<td>99</td>
<td>18.25</td>
</tr>
</tbody>
</table>
6 55 7.49 63 97 23.39
7 56 8.59 63 99 20.18
53 78 5.80 63 101 22.33
54 79 6.41 64 99 17.57
55 80 5.86 64 100 16.96
56 81 6.11 67 97 22.31
53 77 4.64 67 100 17.75
54 78 4.94 67 101 20.00
55 79 4.93 69 97 25.58
56 80 4.73 69 100 20.61
54 77 6.32 69 101 22.10
55 78 5.98
56 79 6.23

The top 1 prediction of ASTRO-FOLD 2.0 agrees well with the native structure, see Figure 6.7. This prediction has a RMSD value of 1.37 Å, a TM score of 91% and a GDT score of 88%. The overall rankings of this structure compared with other methods are 2 according to GDT score, 3 according to TM score, and 3 according to RMSD score.

6.4.5 T596:3NI7

T596 is a pure α protein with 213 amino acids, and its PDB code is 3NI7. Protein 3NI7 has two chains and out of which, only one chain is sequence-unique. 3NI7A has amino acids 6 to 188 in its PDB structure, thus only 183 amino acids are used for evaluation of structure prediction, secondary structure prediction and contact prediction. See Figure 6.8 for the native structure of T596.
Figure 6.8: Native structure of T596 (3NI7A) shown in gray (Only amino acids 6 to 188 are shown). Top 1 model from ASTRO-FOLD 2.0 is colored in rainbow

The secondary structure prediction of T596 has a Q3 accuracy of 82.7%. The two small 3-10 helices are not predicted, and helices 6 and 7 are predicted as one helix (helix 5 of the prediction result). All other helices are correctly predicted and a high prediction accuracy indicates the overall secondary structure prediction for T596 is successful.

Table 6.8: Predicted secondary structure information for T596 by ASTRO-FOLD 2.0 and the native secondary structure information are shown in this table. Note helix 5 and helix 10 of the native structure are 3-10 helices.
The high secondary structure prediction accuracy in part contributes to the high contact prediction accuracy (60.5%). This contact accuracy is for amino acid pairs that are at least 6 amino acids apart. The detailed true and false contact predictions are listed in Table 6.9.

Table 6.9: Predicted amino acid contacts for T596 of CASP9. Data is shown for amino acid pairs that are at least 6 amino acids apart. First 3 columns show the true predicted contacts whose distances fall between the predicted lower and upper distance bounds; while the last 3 columns show the false predicted contacts where the real distance falls outside the predicted distance range.

<table>
<thead>
<tr>
<th>AA1</th>
<th>AA2</th>
<th>Distance</th>
<th>AA1</th>
<th>AA2</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>25</td>
<td>7.79</td>
<td>67</td>
<td>92</td>
<td>13.12</td>
</tr>
<tr>
<td>17</td>
<td>28</td>
<td>8.70</td>
<td>59</td>
<td>123</td>
<td>12.97</td>
</tr>
<tr>
<td>17</td>
<td>30</td>
<td>11.49</td>
<td>95</td>
<td>125</td>
<td>14.97</td>
</tr>
<tr>
<td>19</td>
<td>25</td>
<td>7.80</td>
<td>97</td>
<td>125</td>
<td>20.35</td>
</tr>
<tr>
<td>19</td>
<td>30</td>
<td>10.46</td>
<td>101</td>
<td>119</td>
<td>20.96</td>
</tr>
<tr>
<td>19</td>
<td>31</td>
<td>11.58</td>
<td>101</td>
<td>122</td>
<td>20.07</td>
</tr>
<tr>
<td>31</td>
<td>41</td>
<td>5.39</td>
<td>105</td>
<td>119</td>
<td>15.16</td>
</tr>
<tr>
<td>31</td>
<td>44</td>
<td>8.13</td>
<td>105</td>
<td>122</td>
<td>15.08</td>
</tr>
<tr>
<td>33</td>
<td>41</td>
<td>8.12</td>
<td>105</td>
<td>123</td>
<td>17.04</td>
</tr>
<tr>
<td>37</td>
<td>44</td>
<td>9.46</td>
<td>105</td>
<td>123</td>
<td>13.13</td>
</tr>
<tr>
<td>66</td>
<td>88</td>
<td>7.60</td>
<td>106</td>
<td>119</td>
<td>14.94</td>
</tr>
<tr>
<td>66</td>
<td>89</td>
<td>8.61</td>
<td>107</td>
<td>122</td>
<td>14.78</td>
</tr>
<tr>
<td>66</td>
<td>92</td>
<td>9.41</td>
<td>131</td>
<td>147</td>
<td>13.17</td>
</tr>
<tr>
<td>67</td>
<td>87</td>
<td>8.35</td>
<td>164</td>
<td>177</td>
<td>12.76</td>
</tr>
<tr>
<td>67</td>
<td>88</td>
<td>10.90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>147</td>
<td>11.08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>147</td>
<td>11.36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>178</td>
<td>7.52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>157</td>
<td>178</td>
<td>10.57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>177</td>
<td>9.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>177</td>
<td>10.46</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>178</td>
<td>11.59</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The best tertiary structure of the ASTRO-FOLD 2.0 predictions has a GDT score of 66%, a TM score of 72% and a RMSD value of 3.3 Å to the native. The overlay between the best model of ASTRO-FOLD 2.0 and the native is displayed in Figure 6.8. The overall topology as well as
the local structures are well fitted to each other. The rankings compared with the best model from other methods of CASP9 are 5, 7 and 7 according to GDT, TM and RMSD scores, respectively.

### 6.5 Conclusions

This chapter presented an overview of the tertiary structure prediction component of ASTRO-FOLD 2.0. All of the constraints mentioned in the previous chapters were introduced into the final three dimensional structure prediction algorithm, which combines deterministic global optimization ($\alpha$BB), stochastic conformational space annealing (CSA) and torsion angle dynamics (TAD). Predicted structures were clustered using a novel traveling salesman problem based iterative clustering algorithm, ICON. The selected structures from the ICON clustering procedure were used to derive improved structures using chemical shift data. These improved structures were used to derive improved dihedral angle and distance bounds, which were used to run a second iteration of the tertiary structure prediction algorithm. All of the novel components of ASTRO-FOLD 2.0 were integrated into the existing ASTRO-FOLD framework. The performance of the improved ASTRO-FOLD 2.0 framework was demonstrated for a number of blind targets from the recently concluded CASP9 community-wide experiment.
Chapter 7

Conclusions and Future work

In this chapter, the contributions of the thesis work presented in Chapters 2-6 are summarized in Section 7.1. An overview of promising areas of future research applications and investigations is presented in Section 7.2.

7.1 Conclusions

The contributions of this thesis can be divided into five categories, (i) prediction of $\alpha$-helices, (ii) $\beta$-sheet topology prediction, (iii) Flexible stem based loop structure prediction, (iv) near-native structure identification and (v) Tertiary structure prediction. The importance and the unique components of each algorithm presented in the thesis are summarized below.

7.1.1 Prediction of $\alpha$-helical regions

A linear programming and a mixed-integer linear programming model were presented in Chapter 2 to address the problem of prediction of helical regions in globular proteins. The primary aim of the models was the development of an $\alpha$-helix prediction algorithm which does not require the knowledge of the similarity of a sequence to the protein data bank. The method is rendered independent
of the ability of sequence similarity algorithms like BLAST and PSI-BLAST to identify database structures whose sequences resemble the target, thus removing any bias during the prediction of helical segments in proteins which are unique to all known structures. The key contributions of the approach were: (i) development of pairwise amino-acid hydrogen bonding probabilities, (ii) defining helical propensity of an amino acid based on the pairwise hydrogen bonding probability of surrounding neighbors and (iii) the application of a linear and mixed-integer linear programming formulations to train helical propensity parameters and predict $\alpha$-helices in target proteins, respectively. Constraints for eliminating specific amino acid from $\alpha$-helices are implemented using an extensive analysis of a chemical shift database. An important contribution of the method is its use in effectively narrowing the conformational search space that must be explored by tertiary structure prediction algorithms, through the prediction of strict distance and dihedral angle bounds for amino acids predicted to be in $\alpha$-helices.

7.1.2 $\beta$-sheet topology prediction

In Chapter 3, a mixed integer-linear optimization based approach was presented to predict the $\beta$-sheet topology in $\beta$ and mixed $\alpha/\beta$ proteins. The main challenge to overcome is the exponential possibilities of arrangements of $\beta$-strands in a protein. The key contributions of the developed approach are: (i) mixed integer-linear optimization model which maximizes the total contact potential of a protein, (ii) large number of biological, relational, physical and steric based constraints to provide biologically meaningful predictions, (iii) integer cut constraints to create a rank-ordered list of predicted $\beta$-sheet topologies and (iv) re-ranking of predicted topologies using torsion angle dynamics and clustering. $\beta$-sheet topology prediction is considered one of the biggest bottlenecks in tertiary structure prediction of $\beta$ and mixed $\alpha/\beta$ proteins. Accurate $\beta$-sheet topology prediction can narrow the tertiary structure search space drastically, by providing tight distance bounds between amino acids separated in the primary sequence.
7.1.3 Loop Structure Prediction

In Chapter 4, a non-linear constrained optimization based approach was developed for the prediction of dihedral angle bounds on amino acids in the loop regions of proteins. This step is an important intermediate stage of the tertiary structure prediction of proteins, as it provides tight dihedral angle bounds on the amino acids which are not targeted by any other intermediate step involving secondary structure elements. The key aspects of the approach are (i) initial loop structure generation using a probability distribution generated from existing databases, (ii) fast rotamer optimization steps to alleviate steric clashes, (iii) non-linear constrained optimization using the all atom force field ECEPP/3 and (iv) clustering of predicted structures to develop tighter bounds on backbone dihedral angles of loop amino acids. The method was shown to be independent of the similarity of target sequences in the CASP9 event, thus demonstrating its ability to be used for proteins with low sequence homology to existing proteins in the Protein Data Bank.

7.1.4 Near-native structure identification

In Chapter 5, the problem of identification of structures nearest to the native in a predicted conformational ensemble has been addressed. The aim of the algorithm is to develop a method which identifies near-native structures from a predicted ensemble of structures, irrespective of the quality of the predicted ensemble. The key aspects of the approach are (i) traveling salesman problem approach to identify optimal path through each predicted structure, (ii) optimal cluster assignment using expectation maximization and an integer optimization approach, (iii) iterative elimination of sparse clusters to increase quality of predicted ensemble and (iv) re-ranking of selected structures using all-atom Cα-Cα and Centroid-Centroid force fields. This algorithm, known as ICON, is very important in blind target predictions where energy functions have been shown to be poor choices for the identification of near-native structures. All-atom and database derived energy functions depend on the training set used to develop their parameter values. At the same time,
given that the native structure itself is very rarely identified, a monotonic relationship between native-similarity and energy of the conformation cannot be assured. The algorithm is shown to be effective in identifying high quality structures from diverse conformational ensembles.

### 7.1.5 Tertiary Structure Prediction

In Chapter 6, a parallel implementation of the hybrid global optimization algorithm for tertiary structure prediction in the ASTROFOLD framework was introduced. The key aspects of the approach are (i) deterministic global optimization approach $\alpha$BB, which provides lower bounds for the feasible conformational space; (ii) stochastic conformational space annealing (CSA) approach, which provides upper bounds for the feasible conformational space, and works as an effective means of identifying low energy structures; (iii) identification of feasible initial protein structure configurations using torsion angle dynamics; (iv) fast rotamer optimization algorithms to alleviate steric clashes, thus acting as effective local minimizers; and (v) constrained non-linear optimization using sequential quadratic programming techniques. The main contributions of the thesis work are (i) streamlined implementation of all components of the tertiary structure prediction algorithm in a hybrid parallel algorithm and (ii) unification of the tertiary structure prediction algorithm with all preceding stages of ASTROFOLD presented in the previous chapters.

The algorithm was tested on a large number of proteins during the recently concluded community-wide experiment, CASP9. All targets provided during the CASP9 experiment have unknown structures at the time of the prediction, thus eliminating any bias in the structure prediction algorithm. The proposed ASTROFOLD framework was shown to predict high quality conformers for a large number of blind targets. The accuracy and importance of a number of the intermediate steps leading to the tertiary structure prediction stage is also demonstrated.


7.2 Future Work

A number of areas of future work based on the work in this thesis can be divided into five categories, (i) β-strand and β-sheet topology prediction, (ii) super-secondary structure prediction, (iii) improvement in the expression of energetic terms in tertiary structure prediction, (iv) improved conformational search strategies and (v) blind and double-blind target studies.

7.2.1 β-strand and β-sheet topology prediction

The research on the prediction of α-helices presented in Chapter 2 can prove to be groundwork for the prediction all of secondary structure elements in globular proteins. One of the major disadvantages with the use of a similar approach for the prediction of β-strand locations is the lack of global information. By dividing a protein into overlapping oligopeptides, local information can be captured successfully. While local interactions are vital for the formation of α-helices, the formation of β-strands also depend on the interaction of sequentially separated β-strands with each other. A framework for the simultaneous prediction of all secondary structure elements and the β-sheet topology of a protein could include (i) dividing a protein into overlapping oligopeptides, perhaps nonapeptides for helices and pentapeptides for strands; (ii) evaluating the propensity for the central amino acid to be in α-helices and β-strands, as presented in Chapter 2; (iii) evaluating pair probability of amino acid pairs in β-strands to form contacts with each other in a manner similar to the approach presented in Chapter 3 and in literature 103; (iv) training parameters to evaluate α-helix propensity, β-strand propensity and the pair propensity of amino acid pairs to form contacts in β-strands and (v) development of a mixed-integer linear optimization problem for the simultaneous prediction of α-helices, β-strands and β-sheet topology in globular proteins. The complete prediction of secondary structure elements in proteins without sequence alignment information is a challenging problem in protein structure prediction. In the presence of sequence similarity information derived from PSI-BLAST, average secondary structure prediction accuracy
has been observed to tail off between 82-85%. A new approach combining local secondary structure and tertiary contacts in the form of $\beta$-sheet topology prediction would be a possible way to overcome this observed barrier.

### 7.2.2 Supersecondary structure prediction

One of the extensions of the loop structure prediction algorithm presented in Chapter 4 is the prediction of supersecondary structures in globular proteins. Supersecondary structures are compact, three-dimensional components of a protein formed by two or more neighboring secondary structure elements. Supersecondary structure prediction allows the introduction of constraints on the relative orientation of secondary structure elements. While the relative orientation of $\beta$-strands in a sheet can be known using $\beta$-sheet topology prediction presented in Chapter 3, the orientation of $\beta$-strands in different $\beta$-sheets, or their interaction to $\alpha$-helices can be derived out of supersecondary structure prediction. A possible framework for the prediction of supersecondary structures could include (i) generating initial backbone structures for supersecondary structure using torsion angle dynamics, as presented in Chapter 6, (ii) rotamer optimization to ensure steric clashes are avoided, (iii) free energy minimization calculations using all atom potential energy terms, solvation terms and statistical expressions to mimic entropy, (iv) clustering predicted conformers to identify representative structures and (v) re-introducing additional penalty terms for differences in dihedral angles found for similar secondary structure elements which are a part of different supersecondary structure units. The problem of supersecondary structure prediction has been targeted previously in literature \textsuperscript{108}. While this problem has primarily been tackled by approaches which rely on fragment assembly, distance and angle constraints derived out of the solution to this problem can reduce conformational search space for tertiary structure prediction drastically \textsuperscript{89}.
7.2.3 Conformational Search Strategies

Most protein structure prediction algorithms are limited by their ability to search the protein conformational space. This is especially true for first principles based tertiary structure prediction algorithms like ASTROFOLD, presented in Chapter 6. As shown in the performance of the tertiary structure prediction algorithm in the blind target set provided during CASP9, there are cases where the algorithm was unable to locate the native basin of the target protein. A few possible areas of future work aimed at improving conformational search space reduction and navigation include (i) the development of improved lower bounding strategies, by deriving precise values of the lower bounding coefficient $\alpha$ in the $\alpha$BB algorithm; (ii) development of an iterative approach which predicts the fold of the protein using combination of a unified atom representation of the protein, a simplified force field based on steric clashes and hydrogen bonding and dividing the conformational space using an atom grid; (iii) an improved representation of the extended regions of a protein, by incorporating the well known twist in $\beta$-strands which would lead to a more compact hydrophobic core prediction for $\beta$ and mixed $\alpha/\beta$ proteins; (iv) the study of introducing directionality in local and non-local hydrogen bonds, ensuring that donor and acceptor atoms lie approximately in a straight line and (v) the study of protein length and protein class based parameter choices for the conformational space annealing and torsion angle dynamics stages.

7.2.4 Energy Function

A number of additions and improvements to the energy evaluations in the tertiary structure prediction algorithm presented in Chapter 6 can be suggested. A major challenge for protein structure prediction algorithms is the identification of a useful force field for the purposes of protein structure prediction. One of the limitations of the current implementation of ASTROFOLD is the restricted correspondence between the minimum energy structure and the native structure of the protein. According to Anfinsen’s hypothesis, the native structure of a protein corresponds to the
global free energy minimum of the system. Currently, the objective function presented in Equation 6.1 only accounts for the ECEPP/3 potential energy of the protein. The two additional terms to factor into the objective function include the entropic and solvation contributions. Approximate calculations to incorporate entropic effects have been introduced previously using statistical approximations [101]. Separately, clustering techniques [185, 229] have been used to represent the depth of a local energy basin by the population of the local minima in the neighborhood, which can be used to introduce knowledge-based expressions to account for entropic effects. Solvation effects can be modeled with rigorous calculations using the Generalized Born [197] model or the poisson-boltzmann equation. Approximate calculations for solvation calculations include solvent accessible surface area [153] and volume based [14, 104] calculations. Another area of future study is the comparison of the use of the ECEPP/3 force field with that of the use of AMBER [39] and CHARMM [124] force fields. Both of these force fields involve energy terms associated with bond length and bond angle violations, which are assumed to be zero in the ECEPP/3 force field.

7.2.5 Hybrid approach and Future Blind Studies

The recently concluded community-wide CASP9 experiment demonstrated a number of positive developments in the structure prediction ability of the ASTROFOLD framework. The tenth community wide experiment would present an ideal opportunity to test a hybrid approach to protein structure prediction. ASTROFOLD is seen to predict good quality structures for proteins with low sequence homology to the protein data bank, given that initial structures for the tertiary structure prediction stage are not derived based on sequence alignment algorithms like PSI-BLAST [6]. However, it has been widely accepted that the structure space is more conserved than the sequence space. For proteins with a high degree of sequence similarity, a hybrid approach using fold recognition and global optimization would be expected to produce high quality conformers for a target protein. Moreover, the biggest challenge in tertiary structure prediction is the identification of the
correct fold for a protein. For proteins where the fold can be determined with a high degree of confidence, an approach using the output of fold recognition algorithms like MODELLER [54] to generate tight dihedral angle bounds on the amino acids of the protein would help narrow the conformational search space drastically.
Bibliography


Chapter 7. Conclusions and Future Work


CHAPTER 7. CONCLUSIONS AND FUTURE WORK


CHAPTER 7. CONCLUSIONS AND FUTURE WORK


