Thesis for the Degree of Licentiate of Philosophy

Protein modelling by the zipping and assembly method with limited NMR-derived constraints

Maryana Wånggren

CHALMERS | GÖTEBORG UNIVERSITY

Department of Computer Science and Engineering
Chalmers University of Technology and Göteborg University
Göteborg, Sweden 2017

Protein modelling by the zipping and assembly method with limited NMR-derived constraints
Maryana Wånggren

Technical Report 167L
ISSN 1652-876X.
Department of Computer Science and Engineering
Research group: Computing Science

Department of Computer Science and Engineering
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31-772 1000

Printed at Chalmers
Göteborg, Sweden 2017

Abstract

Molecular dynamics simulations, often combined with simulated annealing, are commonly used when calculating structural models of proteins, e.g. based on NMR experiments. However, one is often faced with limited and, sometimes, insufficient information for determining a well-resolved 3D structure. In addition, the type of data available for different proteins may vary: ranges for torsion angles, distance approximations, relative orientation of different molecular parts etc. We are using whatever structural information is around, together with a dynamic programming approach (Zipping and Assembly) for searching the space of feasible conformations to rapidly determine 3D structures that are consistent with the input constraints. Time-efficiency is important for good sampling of the conformational space and necessary to replace expensive, complex and time consuming experiments. Our approach benefits from having both high level and low level descriptions of conformational features and constraints and the possibility to infer new constraints from those that are given.

Acknowledgments

Thanks to my supervisor, Graham J.L. Kemp.

Thanks to my co-supervisor, Martin Billeter.

This work is supported by a Project Research Grant from the Swedish Research Council (621-2011-6171).

Salient points

— A protein modelling program has been implemented in which the zipping and assembly method (a "bottom up" algorithm) is used to explore the large conformational space, guided by distance and angle constraints provided by straight-forward NMR experiments.

— This combination enables protein backbone models to be built using fewer constraints than are required using other constraint-based methods.

— When testing the protein modelling program, sets of angle and distance constraints can be generated automatically from high-level description of protein structural features using Prolog and then propagated through the zipping and assembly data structure.

— Additional distance constraints can be obtained from biological knowledge and can be expressed as logical rules using Prolog.

— These extra constraints can help to favour the search for protein models, consistent with all the requirements in the pool of solutions (conformational space).

— There is a potential to use additional rules for deriving further constraints and directing the computational effort on the most important parts of the search space in order to improve runtime performance and memory usage which is crucial for modelling longer proteins.

Contents

Chapter 1

Introduction

Proteins are important biological macromolecules that consist of chains of amino acid residues. Knowing the three-dimensional structures of proteins is important in fully understanding the molecular basis for their function, but experimental determination of protein structures can be difficult, costly and time-consuming. Therefore, there is a strong interest in using computational modelling methods to obtain models of protein structures (Baker and Sali 2001; Simons et al. 1997; Ozkan et al. 2007).

In different projects there could be accessible data from one or more kinds of experimental investigation, e.g. co-evolution information, disulphide linkage analysis results, secondary structure information, data from NMR experiments (Billeter et al. 2008). With NMR, one can carry out many kinds of experiment in order to collect more data that can be used in resolving a structure. However, doing this has a cost. For example, some information about main chain amide groups that are spatially close together can be obtained relatively easily, compared with obtaining a full list of all nuclear Overhauser effect (NOE) restraints that give distance estimates for pairs of atoms that are close together in three-dimensional space (Overhauser 1953).

In the current work we are developing a computational modelling method that can use whatever information is easy to obtain, which could be different from case to case.

1.1 Aims

We aim to develop a protein modelling program that can be used together with data from the relatively straightforward NMR experiments that are usually performed first in a study, to obtain accurate protein model structures quickly, reducing or even eliminating the need for more expensive and complex multidimensional NMR experiments that might require alternative labelling regimes or more complex experiments in order to further increase in the dimensionality of the spectra (Section 2.3.2).

The computational approach that has been developed will deepen the understanding of how nature folds and assembles proteins into larger molecular complexes and provide fast and economic access to structural information, for example on targets in drug discovery.

1.2 Contributions

A protein modelling program has been implemented in which the zipping and assembly method (see Chapter 3) is used to explore the large conformational space, guided by distance and angle constraints provided by straight-forward NMR experiments (see Chapter 2). Using this combination, protein backbone models can be built using fewer constraints than are required using other constraint-based methods. The number of constraints can vary from case to case (see Chapter 6) and even any subset of constraints that is available could be used and extended (see Chapter 5). When testing the protein modelling program, sets of angle and distance constraints can be generated automatically from high-level description of protein structural features using Prolog and then propagated through the zipping and assembly data structure. Moreover, additional distance constraints can be obtained from biological knowledge and can be expressed as logical rules using Prolog. These extra constraints can help to favour the search for protein models consistent with all the requirements in the large conformational space. There is a potential to use additional rules for deriving further constraints and directing the computational effort on the most important parts of the search space in order to improve runtime performance and memory usage which is crucial for modelling longer proteins (see Chapter 7).

1.3 Thesis overview

The rest of thesis is organised as follows: Chapter 2 describes aspects of protein structure that are essential for understanding the approaches described in the thesis.Chapter 3 describes zipping and assembly algorithm, the concept we are using in this research for exploring a large confomational space to simulate three dimentional structures of proteins, in details. Chapter 5 defines the kinds of data are being used to build protein main chains models. Chapter 6 illustrates the application of our method in attempt to reconstruct real proteins. We discuss our own system, related work and some future directions in Chapter 7. The main contributions of the work are summarised in Chapter 8.

Chapter 2

Background and Challenges

Proteins play significant roles in living organisms and are involved in all biological processes. Their activity is closely connected with their three-dimensional structure (Section 2.2.3). Ideally, protein structures are determined experimentally (Section 2.3), but this can be difficult and time-consuming. Therefore there is a strong interest in computational methods for modelling proteins. In this chapter, the basic features of protein structure and ways of modelling its three-dimensional structures are described.

2.1 Protein structure

Proteins consist of long linear chains of amino acid residues that are chemically bonded to each other: the carboxyl acid group (C(=O)OH)) of one amino acid reacts with the amino group (-NH2) of the next amino acid, forming a covalent bond called "peptide bond" (Pauling 1960). Formation of a peptide bond between two residues is shown in Figure 2.1:

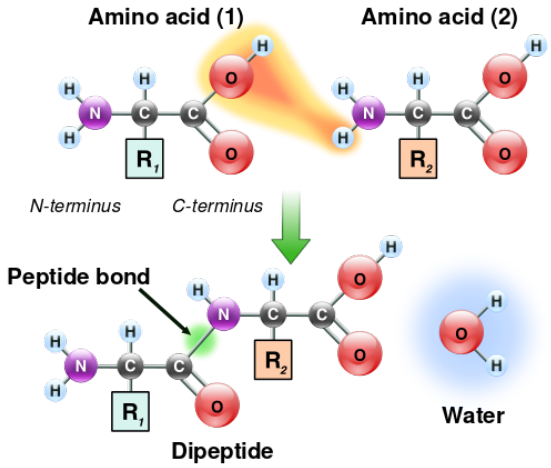

Figure 2.1: Peptide bond formed between two consecutive residues.[1]

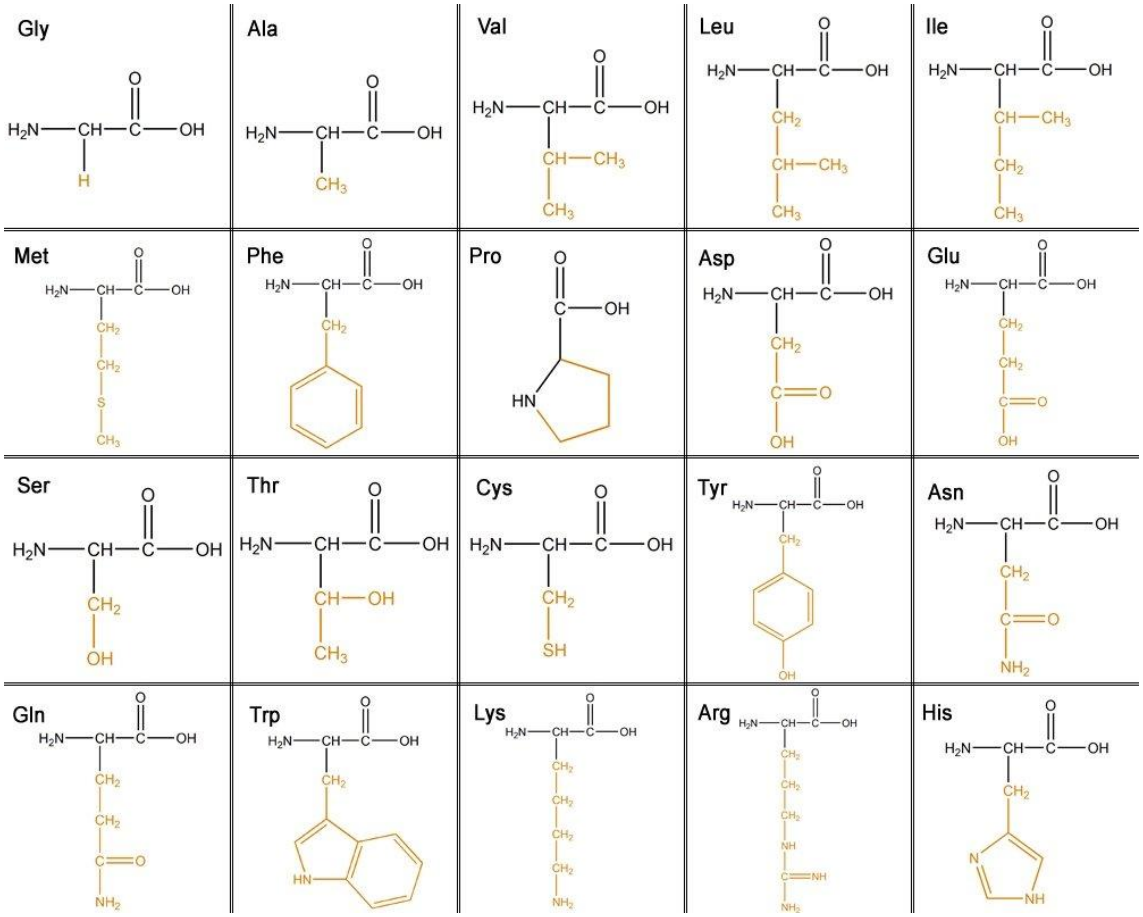

Figure 2.2: 20 common amino acids. Main chains are shown in black and side chains are shown in beige.[2]

There are 20 common amino acids that have same main chain (H2NCHCOOH) and a unique side chain (R) that determines an amino acid and its biophysical properties (Figure 2.2). Residues linked to each other by peptide bonds create a polypetide with an N-terminus and a C-terminus: the start of the polypeptide chain with a free amine group (-NH2) and the end with a free carboxyl group (-COOH).

A protein chain is able to fold into its native conformation by rotation around two of the bonds in the main chain, designated ϕ and ψ (Figure 2.3). ϕ describes the rotations of the polypeptide backbone around the bonds between N-C^{α} ($C_{(-1)}$-N-C^{α}-C) and ψ describes the rotations of the polypeptide backbone around the bonds between C^{α}-C (N-C^{α}-C-$N_{(+1)}$) Feasible ϕ and ψ combinations (i.e. those that do not result in *steric hindrance* — collisions between atoms) were calculated, and represented graphically as a two-dimensional plot (the *Ramachandran plot*

[1]Peptide bond formation, Yassine Mrabet, Wikimedia Commons: `https://commons.wikimedia.org/wiki/File:Peptidformationball.svg`, public domain.

[2]Tablica aminokiselina, Dalibor Bosits from hr, Wikimedia Commons: `https://commons.wikimedia.org/wiki/File:Tablica_aminokiselina.jpg`, used under Creative Commons Attribution-Share Alike 3.0 Unported (CC BY-SA 3.0).

(Ramachandran et al. 1963)) before the first atomic resolution protein structure (myoglobin) was obtained using X-ray crystalography (Kendrew et al. 1958). The third torsion angle within the protein backbone called ω (describe rotation about the $C_{(-1)}$-N bond and involves the $C^{\alpha}{}_{(-1)}$-$C_{(-1)}$(O)-N-C^{α} bond) is essentially flat and normally close to 180° (the *trans* configuration). More rarely, the ω angle can have a value close to 0° (the *cis* configuration). If we assume standard bond lengths and angles (Engh and Huber 1991), the task of predicting the conformation of a protein's main chain reduces to predicting values for all of the ϕ and ψ angles. Some ϕ and ψ combinations are energetically more favourable than others, and some combinations are not possible at all since these would result in atoms clashing into each other. Figure 2.4 shows the distribution of ϕ and ψ combinations taken from 20 protein structures from the Protein Data Bank (PDB) (Berman et al. 2003).

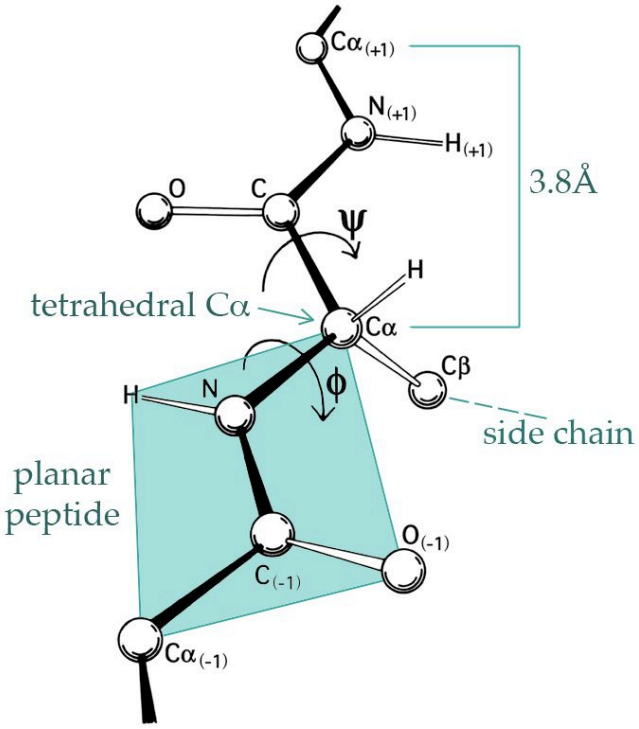

Figure 2.3: Protein main chain and side chain (shown as C^{β}). Six atoms from one C^{α} atom to the next lie in a plane ($C^{\alpha}{}_{(-1)}$-$C_{(-1)}$-NH-C^{α}) — the *peptide plane*. Peptide bond restricts the third torsion angle ω (lies in the peptide plane, not shown on the picture) to be close to 180° (the *trans* configuration) or, more rare, to be close to 0° (the *cys* configuration). The distance between $C^{\alpha}{}_{i}$ and $C^{\alpha}{}_{(i+1)}$ is 3.8 Å.[3]

Each of the 20 amino acids has its own very characteristic Ramachandran plot. Since glycine (Figure 2.2) does not have a side chain, it is less restricted than other residues in the ϕ and ψ that are possible; for other residues the side chain atoms can clash with other atoms. Considering the proline side chain bonds to the main chain

[3]"PhiPsi drawing with plane and labels", Jane S. Richardson, Wikimedia Commons, `https://commons.wikimedia.org/wiki/File:PhiPsi_drawing_with_plane_and_labels.jpg`, used under Creative Commons Attribution 3.0 Unported (CC BY 3.0).

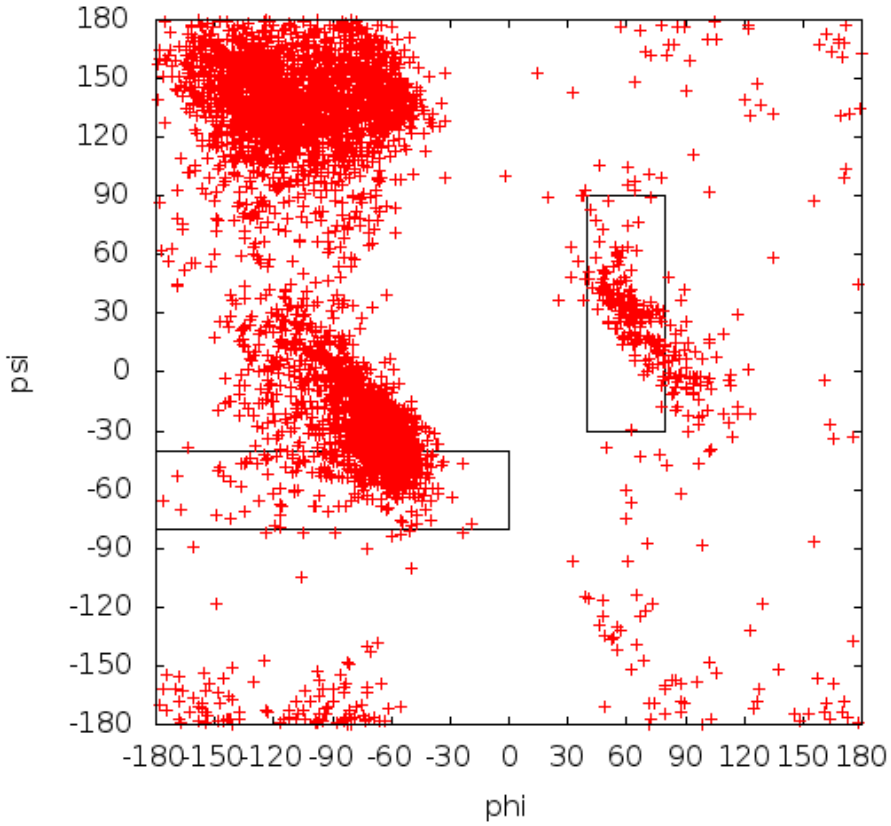

Figure 2.4: Combinations of values for ϕ and ψ torsion angles from a set of proteins from the Protein Data Bank (see Section 4.2.2). The values of the ϕ and ψ angles are in degrees. Two rectangular areas correspond to left-handed and right-handed α helices.

at both the C_α atom and the nitrogen, this limits the possible values for proline's ϕ angle to be (-63° ± -15°).

The Ramachandran plot is a powerful tool for checking the quality of a protein models achieved experimently or computationally. In our approach, the Ramachandran plot can be used to suggest possible conformations of the main chain, as is described in Chapter 4.

2.2 Levels of organization

2.2.1 Primary structure: protein sequence

The order in which amino acid residues are connected in sequence is determined by genes and is called *primary structure* (Figure 2.5). The number of known protein sequences is growing rapidly and is reflected in the size of the UniProt[4] — a protein sequence database which has over 80 million sequences (March 2017).

[4] www.uniprot.org

[5] Primary structure of fibroin (silk protein), Sponk, Wikimedia Commons: `https://commons.wikimedia.org/wiki/File:Silk_fibroin_primary_structure.svg`, public domain.

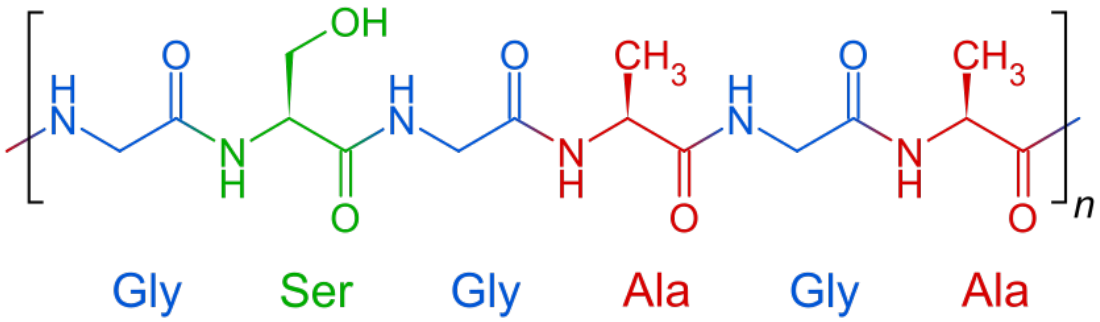

Figure 2.5: Example of primary structure[5].

2.2.2 Secondary structures: α helix and β sheet

A protein folds into its native conformation, dictated by its primary structure. Within a protein fold there are some substructures with a regular pattern of main chain hydrogen bonds (H–O bond) that occur frequently in different proteins. These recurring substructures are referred to as elements of *secondary structure* The two most common kinds of secondary structure, detected in proteins, are α helices and β sheets.

α helix (Pauling-Corey-Branson model)

α helix (Pauling, Corey, and Branson 1951) is a hydrogen-bonded helical configuration in the polypeptide chain and is the most common secondary structure element found in globular proteins. Mean helix length is approximately 10 residues, corresponding to a helix with 3 turns (approximately 15 Å) (Kabsch and Sander 1983). All the residues are equivalent exept for the difference in the side chain R.

Different authors have suggested different “ideal” torsion angles for residues in α helices. In 1988 it was shown that the conformations of observed helices are significantly different from the "ideal" linear structure (Barlow and Thornton 1988). All α were divided as linear, non-linear, irregular and curved (caused by Pro residue). Hovmöller *et al.* analyzed 1042 protein subunits taken from PDB in 2002 and estimated that the mean value of main chain torsion angles ϕ and ψ for α helix are (-63.8° ± 2°, -41.1° ± 2°) (Hovmöller et al. 2002).

β sheet

A β sheet (Pauling and Corey 1951) is a hydrogen-bonded layer structure of polypeptide and is a common motif of regular secondary structure in proteins. β sheets consist of strands connected by at least two or three main chain hydrogen bonds, forming a generally twisted, pleated sheet. Planar peptide groups lie in the plane of the sheet (Figure 2.8). There are two types of β sheets, consisting of either parallel or antiparallel extended strands A hydrogen-bonded layer structure of polypeptide chains with alternate chain oppositely oriented is called *antiparallel* β sheet (Figure

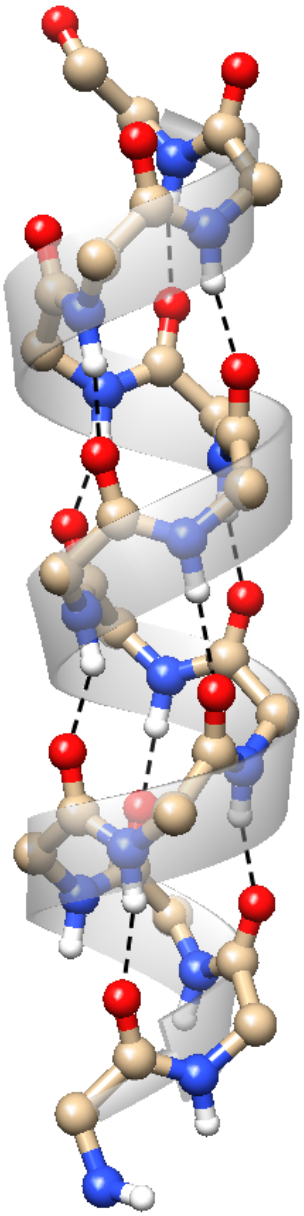

Figure 2.6: Protein secondary structures: right-handed α helix (most common). Left-handed α helix rotates around the same axis as the right-handed α helix, but in the opposite direction.

2.7) and a hydrogen-bonded layer structure of polypeptide chains with all chains similarly oriented (the pleated sheet) is called *parallel* β sheet (Figure 2.8)

2.2.3 Tertiary structure: three dimensional shape of a protein

Protein tertiary structure is the three dimensional shape of a protein, stabilized with hydrogen bonds, hydrophobic interactions and hydrophilic interactions. The way a

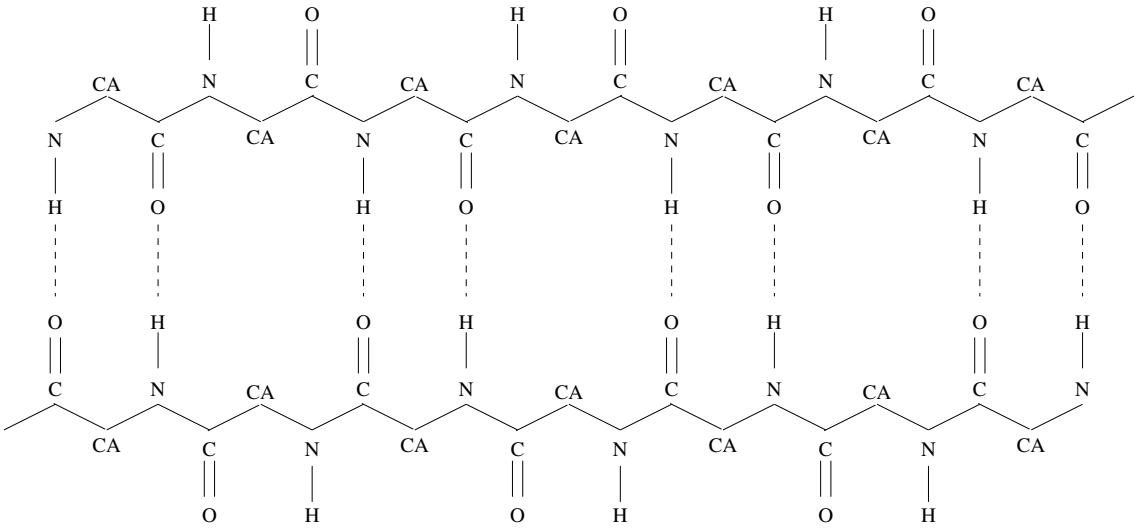

Figure 2.7: Protein secondary structures:antiparallel β strand. Hydrogen bond: N-H group donates a hydrogen bond to the backbone carbonyl C=O group of the amino acid.

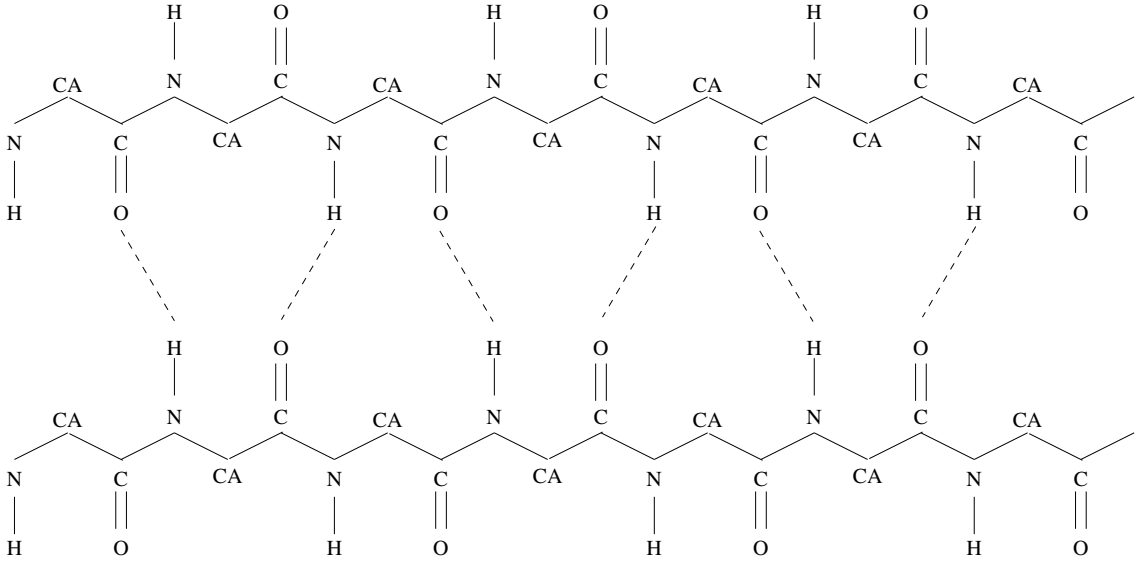

Figure 2.8: Protein secondary structures: parallel β strand.

protein folds into its native conformation is one of the most important problems in structural bioinformatics that still has not been solved (Dill, Ozkan, et al. 2007).

The protein folding dilemma arose from Anfinsen's experiment where proteins could spontaneously refold from their denaturated states (Anfinsen and Scheraga 1975). Thus the primary structure of a protein dictates its tertiary structure. The protein folding problem can be divided into two separate questions:

1. predicting 3D structure of a protein from its sequence;

2. predicting the pathways of folding by which an unfolded protein achieves its folded conformation.

In this project, we are concerned with solving the first question, connected with modelling protein's tertiary structure (Section 2.2.3) using information about its primary structure (Section 2.2.1) and some additional information that can be available (Chapter 5).

2.2.4 Quaternary structure: a protein complex

Quaternary structure refers of the assembly of several protein subunits into a larger complex. Such a composition is functional and very stable due to hydrophobic interactions between nonpolar side chains of subunits and hydrophilic interaction between polar groups.

2.3 Experimental determination of protein tertiary structures

Protein structure determination is a difficult and time-consuming process which is helpful for understanding how proteins interact and their role in the living processes. This kind of knowledge is important for drug design and biomedical research so a lot of techniques have been developed for this purpose.

The main experimental methods for determining the three-dimensional structure of a protein are X-ray crystallography and nuclear magnetic resonance (NMR). Both of these methods rely heavily on computational methods to help in determining a three-dimensional structure model based on the experimental data. Both methods have their particular strengths and weaknesses.

The structures obtained by X-ray crystallography (Section 2.3.1) and NMR spectroscopy (Section 2.3.2) are deposited in the Protein Data Bank (PDB) (Bernstein et al. 1977; Berman et al. 2003) — a resource that stores and distributes the three-dimensional structural data of macromolecules and can be accessed by researchers worldwide.

2.3.1 X-ray

X-ray crystallography is one of the most powerful techniques available for determining detailed protein structures. Most of protein molecules are very small units and cannot be seen under a microscope. For this reason using radiation (X-ray) with wavelength compatible with the size of an atom (0.1nm = 1 Å) is a solution.

Myoglobin, a protein with 152 residues, was the first protein whose structure was determined by X-ray crystallography (Kendrew et al. 1958). The interest in protein determination using this method has grown since then, and the majority of structures submitted to Protein Data Bank are obtained using this approach[6].

The method is based on the X-ray diffraction: an X-ray beam diffracts in multiple directions as it passes through the protein crystal under different angles. Positions and intensities of diffraction spots reflect the inner structure of a crystal.

[6] `http://www.rcsb.org/pdb/statistics/holdings.do`

An X-ray diffraction experiment is a long process that includes four independent time-consuming steps, such as:

1. Protein crystallization:

   In order to determine the protein structure by X-ray diffraction a protein must be first crystallized. X-ray scattering from a single molecule would be insignificant and extremely difficult to detect. A crystal contains millions of protein molecules arranged in a systematic way, packed together in a highly ordered structure.

   The crystallization process itself is a "trial-and-error process" that requires (Drenth 2007):

   (a) a protein has to be extemely pure: the purer the protein, the better the chance to grow a crystal;

   (b) there has to be enough protein to start crystallization.

   The amount of protein available for crystallization is often very small.

2. X-ray diffraction: reflection from the planes in the crystal

   The crystal is rotated in the X-ray beam and many diffraction images are collected. A diffraction pattern consists of an array of spots that reflects the inner structure of a crystal. During the experiment the sample is cooled to be able to withstand the radiation.

3. X-ray diffraction pattern reflections analysis: three-dimensional Fourier or electron-density maps.

   By measuring the angles and intensities of diffracted beams and applying mathematical analysis called “Fourier transformation” one can translate the diffraction pattern into an electron density map: a map of the distribution of electrons in the molecule. Since electrons are closely located to the nuclei, the electron density map can give us a picture of the whole molecule. The resolution of the maps depends on the quality of the crystal. In reality a crystal is not uniform — it can have disorders that affect final result in errors in protein structure determination.

4. The final step: protein structure prediction from electron density maps (T. Jones et al. 1991). The mean positions of the atoms in the crystal can be predicted from electron density (chemical bonds, angles). The quality of an atomic structure can be validated against the X-ray data by computing the free R factor, and comparing against expected stereochemical parameters using software tools such as MolProbity (Chen et al. 2010) or PROCHECK (Laskowski et al. 1993).

X-ray diffraction has proven to be a strong method for determining protein three-dimensional structure. The most difficult requirement is that the protein sample is crystallised. Using NMR structures, however, can be obtained from samples of protein in solution.

2.3.2 NMR

Nuclear magnetic resonance (NMR) is an experimental method for determining the three-dimensional structure of proteins based on magnetic properties of certain atomic nuclei (spin of H, C, N) (Perrakis et al. 1999). NMR is the only procedure that can deal with partly folded or disordered proteins in solution and solid state. A simplified version of the NMR experiment includes following steps (Billeter et al. 2008; Güntert 2003):

1. Sample preparation: a purified protein is placed in the water solvent;
2. NMR measurements: a sample is placed in a powerful magnet and radio frequency signals are directed through the sample. Each magnetically active atom absorbs a particular resonant radio frequency in a magnetic field — its *chemical shift*.
3. NMR data processing;
4. Chemical-shift assignment: assignment of a specific chemical shift value to an atom.

   Chemical shifts reflect on protein structure and this is how it can be recognized. But in protein that consists of thousands of atoms all the atoms get affected by their neighbours and change their electronic environment. Many peaks appear in the spectra, and it becomes necessary to perform multidimensional NMR experiments that would help to avoid peaks overlapping.
5. NOESY assignment: cross peaks assignment in NOESY spectra: the pairs of interacting hydrogen atoms have to be identified (to extract distance restraints from a NOESY spectrum).
6. NMR spectra analysis using computational methods:

   NMR experiments provide a variety of restraints that can be used when constructing a protein model structure. The most important NMR restraints for determining a protein structure are the nuclear Overhauser effect (NOE) (Overhauser 1953; Carver and Slichter 1956) restraints that give distance estimates for pairs of atoms that are close together in three-dimensional space. Other restraints relate to torsion angles and also to the orientation (in a fixed coordinate system) of selected bonds, often the N-H bonds of the backbone. These latter restraints can be derived from residual dipolar couplings (RDCs), which are considered the easiest data to obtain for large proteins (Tolman et al. 1995; Prestegard et al. 2004). Most of the method use solution to analyze a sample, but there are some experiments with solid state.

NMR tools

The term NMR covers a wide range of experiments that differ in their cost, time and difficulty. While some experiments are relatively straightforward and are routinely

performed when studying a new protein, there are other experiments that are very time-consuming and more expensive. Computational methods play an important role in NMR structure determination. Having obtained a set of distance, torsion angle and orientation restraints, the challenge is to find a three-dimensional structure, or more typically an ensemble of structures, compatible with those restraints. Traditionally, this has involved the use of distance geometry, but today molecular dynamics simulation methods and simulated annealing are most commonly used (Bowers et al. 2000; Güntert 2003; Linge et al. 2003; Brunger et al. 1998; Brunger 2007).

TALOS+ (Torsion Angle Likeliness Obtained from Shift and Sequence Similarity) is a hybrid method for predicting protein backbone torsion angles from NMR chemical shifts (Shen, Delaglio, et al. 2009) is widely used by NMR researchers. The original TALOS program establishes an empirical relation between 13 C, 15 N and 1 H chemical shifts and backbone torsion angles ϕ and ψ (Cornilescu et al. 1999). The predicting rate of TALOS+ is 88.5 %. TALOS+ exploits a large database of 200 proteins (the original TALOS had 20 proteins) originally taken from the BMRB - Biological Magnetic Resonance Data Bank (Ulrich et al. 2008). This database, extracted from the BMRB, contains proteins with nearly complete backbone NMR chemical shifts as well as PDB coordinates from high-resolution X-ray structures.

A number of automated NOESY assignment algorithms are like CYANA (Güntert 2003; Güntert 2004), ARIA (Linge et al. 2003), CNS (Brunger et al. 1998) are used to provide a flexible multi-level hierarchical approach for the most commonly used algorithms in macromolecular structure determination. CYANA uses the simulated annealing schedule with torsion angle dynamics. The scoring function is a sum of violations that is minimized by the iterations. CYANA assumes that each protein is unique and fully determined, so it requires a large set of information about the torsion angles and distances. If there is not enough information about a protein CYANA can become stuck in a local minimum. Our approach, discussed in Section 3, requires less information about a protein and does not allow any violations.

2.4 Computational protein structure prediction

Computational protein structure prediction methods are useful when an experimentally determined structure is not available for a particular protein of interest. The problem is important since the quantity of known protein sequences vastly exceeds the quantity of known protein structures, and the rate at which new sequences are determined vastly exceeds our capacity to determine their three-dimensional structures experimentally. Therefore, it is often necessary to rely on model three-dimensional protein structures that are built using computational methods.

In 1968 Cyrus Levinthal noted the paradox that proteins are able to fold into their native conformation in seconds despite having an astronomical number of possible conformations due to their many degrees of freedom (Levinthal 1968; Dill 1985; Karplus 1997).

When seeking to construct a three-dimensional model based on a target protein's sequence, there are various computational approaches that can be tried, depending on whether we expect the target to have a three-dimensional structure that is similar to

that of another protein whose structure has already been determined experimentally (Baker and Sali 2001). If there is significant sequence similarity between the target protein and a protein whose known structure is present in the Protein Data Bank (roughly more than 30% sequence identity between the proteins (Sander and Schneider 1991; Rost 1999)), then a *comparative modelling* approach can be used to produce a reasonably reliable model structure. In doing this, the known structure is used as a *template* structure, and minimal modifications are made to the template as the template structure is adjusted to match the sequence of the target protein. If no suitable template protein can be identified on the basis of sequence similarity with the target, then *fold recognition* methods (e.g. (Sippl and Weitckus 1992; D. Jones 1999)) can be used to identify a protein whose known three-dimensional structure is compatible with the sequence of the target protein.

Finally, we are left with the most general and most difficult case of the protein folding problem, where *de novo* (sometimes called *ab initio*) methods must be used (e.g. (Simons et al. 1997; Ozkan et al. 2007)). Some of the methods have been successful in predicting the conformations of relatively small proteins. Still the problem remains unsolved in the general case. Most of the methods require a lot of computational power, that is why now there exist some projects like Rosetta@home, Folding@home to be able to share the computation among many computers. Most *de novo* conformational prediction produce candidate structures that are thermodynamicly stable (with lower free energy and entropy). Constraint programming (Sections 2.4.2), Rosetta (Sections 2.4.1) and zipping and assembly (Chapter 3) are computational methods that have been proposed for addressing the protein folding problem. These approaches are further compared in Chapter 7.

2.4.1 Rosetta

Rosetta (Simons et al. 1997; Kaufmann et al. 2010; Das 2011) is perhaps the best known *de novo* protein structure prediction method. Rosetta proceeds by first building an entire extended protein chain. Then the conformation of the chain is repeatedly modified by replacing the conformation of a randomly chosen segment of fixed length (usually 9 residues or 3 residues) in the protein chain with the conformation of a fragment taken from another protein. Sets of potential replacement conformations are compiled in advance as described in (Simons et al. 1997). Thus Rosetta works with a model of the entire protein at all times, and the scoring function used to evaluate the generated models will evaluate the entire protein model. For this purpose Rosetta uses Metropolis Monte-Carlo sampling approaches together with energy functions.

While Rosetta can claim some successes, it can be seen from Figure 1 of (Raman et al. 2010) that Rosetta is often unsuccessful even with proteins that are much smaller than 100 residues, even when the scoring function is supplemented with experimental data from NMR chemical shifts.

While Rosetta works with a model of the entire protein from the start, a rather different approach is taken by the zipping and assembly method (see Chapter 3) where model structures are built incrementally, and the partial structures are all valid

substructures. As stated in (Dill, Lucas, et al. 2007), "the [zipping and assembly] search method is efficient because it never searches more than a few degrees of freedom at a time, and eliminates high energy conformations early in the search".

2.4.2 Constraint Programming

The protein modelling problem can be considered as a constraint satisfaction problem (Krippahl and Barahona 1999; Dal Palù, Dovier, and Fogolari 2004). Constraint Logic Programming is a declarative programming paradigm that is helpful for solving optimizational problems. This approach has shown to be a powerful method for protein structure determination (Backofen 1998). There are several examples of other work on protein modelling where constraint-based methods are used (Campeotto et al. 2013; Traoré et al. 2013).

Krippahl and Barahona suggest that rigid structure constraint enables the representation of known substructures (secondary structure components) that helps to cut down the search space drastically (Krippahl and Barahona 2002). This has been implemented in the PSICO system.

Backofen and Will (2006) use a constraint-based approach to a lattice model (HP model (Lau and Dill 1989)) of protein folding, where a sequence of hydrophobic and polar "amino acid residues" are folded onto a regular grid. While lattice models are a gross over-simplification of the real protein folding problem, they provide a convenient framework for experimenting with search strategies and simplified scoring functions.

Dal Palù et al. (2010) use a CLP-based method to model proteins using fragment assembly. The fragments are taken from known proteins are clustered and classified according to their frequency and similarity and then assembled into a complete conformation. In their work they use a reduced representation of amino acid residues in which each residue is represented by its Cα atom and the centroid of its side chain (face-centered cube lattice model).

Chapter 3

Zipping and assembly method

There is much evidence that protein folding is hierarchical: some small peptides fold into near-native structures independently first and only then fold into the native structure (Rose 1979; Crippen 1978). One hypothesis is that a protein can be folded by the zipping and assembly method (ZAM) which is hierarchical by its nature (Dill, Lucas, et al. 2007). Combined with data from NMR experiments, it can become a powerful tool for attempting to model proteins' three-dimensional conformations.

In many approaches to modelling protein chains an entire chain is first constructed, then its conformation is repeatedly adjusted and evaluated in an attempt to reach the protein's native conformation (see Section 2.4.1). An alternative approach to exploring the search space when constructing a model is to build models of short fragments of protein chain independently of each other, and then to combine these fragments into longer fragments. This is what is done in the Zipping and Assembly Mechanism by Dynamic Programming (Dill, Lucas, et al. 2007; Hockenmaier et al. 2007), which is a dynamic programming algorithm that constructs longer fragments from pairs of shorter ones. Local structuring happens first in independent peptide fragment sites along the chain, then those structures either grow (zip) or coalescence (assemble) with other structures, along pathways involving topologically local contacts (Voelz and Dill 2007). In this way, zipping and assembly takes a "divide and conquer" approach where there are:

- *zipping steps*, in which small, parallel, local and independent decisions are made about whether a short peptide fragment of a protein is "correct", i.e. compatible with the constraints;

- *assembly steps*, in which nonlocal, global, cooperative decisions are made, combining smaller solutions hierarchically until a final solution is found to the full problem.

The zipping and assembly method was first described in the context of lattice models together with the HP model (Dill, Lucas, et al. 2007; Hockenmaier et al. 2007). The chain was up to 20 hydrophobic ("H") and polar ("P") monomers on 2D lattice (Lau and Dill 1989). The conformational space grows exponentially with chain length. They were searching for the a globally optimal conformation, where

conformations are scored by the number of pairs of adjacent "H" monomers (HH contacts).

There are similarities between the zipping and assembly method and the Cocke-Kasami-Younger chart parsing algorithm (CKY) (Younger 1967) used for protein folding borrowed from computationals linguistics.

Dill et al. argue that the zipping and assembly method closely reflects the way that real proteins fold (Dill, Lucas, et al. 2007):

- ZAM identifies all direct folding routes that lead to the native state: hierarchical folding rates that lead directly to native state (Baldwin and Rose 1999);
- the problem is divided into small independent pieces that are solved separetely: its local-first-global-later search explains quick folding, and avoidance of vast stretches of conformational space ("local" here refers to local in sequence);
- the search happens in a small fraction of the search space: the search is efficient because it never searches more that a few degrees of freedom at a time;
- it reflects the parallel nature of physical kinetics;
- ZAM repersents folding as a tree (Hockenmaier et al. 2007) and has been shown to be more efficient that Monte-Carlo algorithms; ZAM is a hierachial dynamic programming that looks for the solution parallel while MC is sequential.
- folding is faster for proteins that have mostly the local contacts (near neighbours in the chain like in alpha helices) and slowest for proteins that have nonlocal contacts (like beta sheet proteins); it captures the relationship between contact order (a measure of the average separation along the chain of the contacting monomers) (Plaxco et al. 1998): whether pairs of amino acid residues that are close together in 3-D space also tend to be close to each other along the protein chain, or tend to be distant from each other along the protein chain) and folding rate;

These similarities with physical protein folding are not shared by other approaches (e.g. Rosetta, see Section 2.4.1) and they make the zipping and asssembly method well suited to the protein modelling task.

Ozkan et al.(Ozkan et al. 2007) used zipping and assembly method together with AMBER96 force field relying on purely physics-based approach (with no information taken from protein structure databases) and molecular dynamics for protein folding. They achieved near 2.2 Å RMSD (root mean square distance) by superposing C_α atoms to PDB native structures for eight proteins from 25 to 73 amino acid residues in length. The main bottleneck is that purely physics-based methods are too slow, but useful for modelling small proteins.

3.1 Zipping and assembly data structure

The results of the local search (fragments - part of protein chain model from 1 to N residues long) are stored in the zipping and assembly data structure (informally referred to here as "the pyramid"): a triangular matrix, consisting of cells — locations in data structure for storing fragments. Each cell contains sets of fragments that are candidates for modelling the conformation of part of the target protein. All cells keep a fixed number (defined by the user, usually 100-1000) of fragment conformations and the information about how fragments were created (Chapter 4). Some cells store distance and angle constraints provided for the target protein (see Chapter 5).

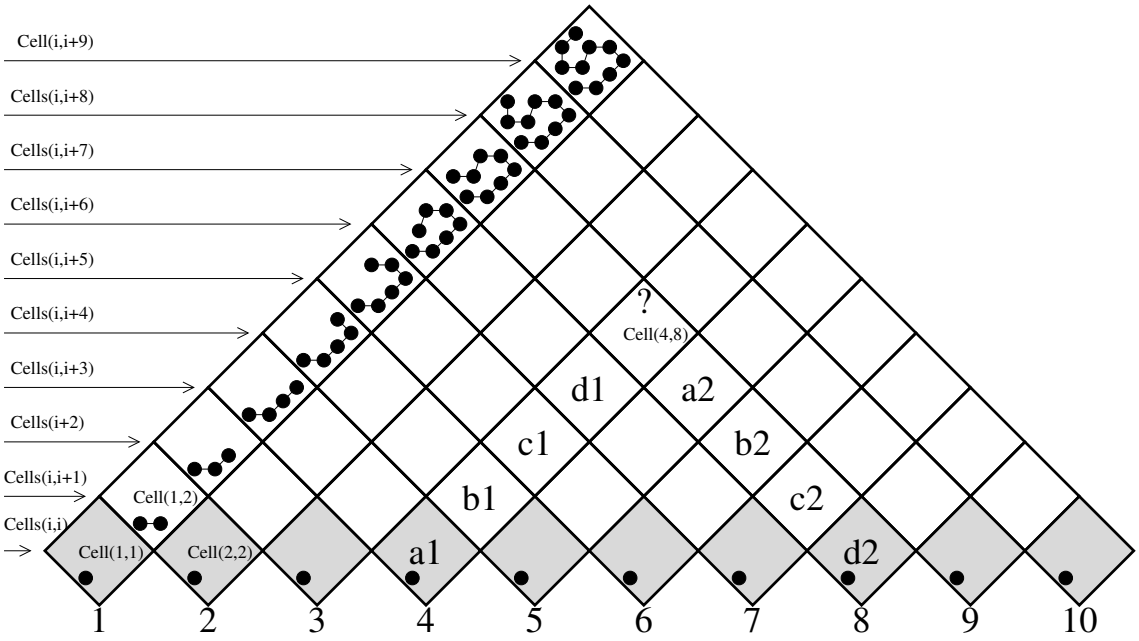

Figure 3.1: Zipping and assembly of a 10-residue protein (number of cell is equal to 55).

The numbers along the bottom of Figure 3.1 represent residue positions within the chain. $Cell_{i,i}$ (bottom row), where $i = 1, ..., N$, are positions in data structure, that store all conformations of a single residue in the protein sequence. For example, $Cell_{1,1}$ containts all conformations for the first residue in the sequence, $Cell_{2,2}$ — conformations for the second residue in the sequence, $Cell_{3,3}$ — conformations for the third residue in the sequence etc. The last residue's confomations are stored in $Cell_{N,N}$ ($Cell_{10,10}$ in Figure 3.1).

The cells on level two are cells with $Cell_{i,i+1}$, where $i = 1, ..., N-1$ ($Cell_{1,2}$, $Cell_{2,3}$, etc.), contain sets of two-residue fragments, each of which is formed by combining one random residue conformation from the cell to the lower left (from $Cell_{2,2}$) and one random residue conformation from the cell to the lower right (from $Cell_{3,3}$).

The cells on level three are defined as $Cell_{i,i+2}$, where $i = 1, ..., N-2$ For example, the fragment from positions 8 to 10 is represented by $Cell_{8,10}$ and is 3 residues long (residue 8, residue 9, residue 10). Number of cells depends on length of the target

protein:

$$number_{cells}(N) = \sum_{n=1}^{N} n$$

Possible conformations for longer fragments are constructed by combining different shorter fragments from lower cells . Consider the five-residue fragment from positions 4 to 8 (5 residues). Possible conformations for this fragment will be stored in $Cell_{4,8}$ cell, which contains a question mark. These can be constructed by combining a fragment chosen from the cell labelled a1 (one residue, $Cell_{4,4}$) with one from the cell labelled a2 (4 residues, $Cell_{5,8}$), or combining a fragment chosen from cell b1 (2 residues, $Cell_{4,5}$) with one chosen from cell b2 (3 residues, $Cell_{6,8}$), and so on. Similarly, all cells in the diagram can be filled with fragment conformations that are the result of combining a random fragment chosen from one of the cells to the lower left of that cell, with a random fragment from a cell to the lower right. The fragments at higher levels are successively longer than those at the levels below (this growth in chain length is illustrated in the cells on the left edge of Figure 3.1. Finally, $Cell_{1,10}$ at the apex will contain a set of possible conformations for the entire protein (consisting of 10 residues in this toy example).

3.2 ZAM supported by NMR data

For some time there has been interest in using *de novo* structure prediction methods in combination with NMR restraints to obtain three-dimensional structures that are compatible with those restraints (e.g. (Bowers et al. 2000; Rohl and Baker 2002; Shen, Lange, et al. 2008; Raman et al. 2010)). Existing work in this area is based on the Rosetta *de novo* structure prediction method (see Section 3.3.1), constraint programming (see Section 3.3.2).

The zipping and assembly method has not been previously used with NMR data. The bottom-up approach to structure generation used in zipping and assembly potentially allows it to scale better than Rosetta when applied to larger proteins. This should also make it better suited to modelling multi-domain proteins, since each domain can be modelled independently of the others. It is for these reasons that we are proposing here to use zipping and assembly together with NMR data.

3.3 Related work

The value of using fragments from known protein structures when building three-dimensional model structures based on NMR data has long been recognised. Kraulis and Jones (1987) derived distance matrices based on short-range NOEs (NOEs between amino acid residues that are relatively close to each other in sequence) and matched these with corresponding distance matrices derived from fragments from known protein structures. Those fragments that matched were clustered, and the fragment closest to the centre of the largest cluster was selected as the most likely conformation for part of the protein being modelled. Delaglio et al. (2000) use the additional information available from RDCs in their molecular fragment replacement

method (MFR), identifying 7-residue fragments in the Protein Data Bank (Berman et al. 2003) whose backbone ϕ and ψ torsion angles are compatible with the measured dipolar couplings.

More recently, there has been interest in using *de novo* structure prediction methods in combination with NMR restraints to obtain three-dimensional structures that are compatible with those restraints (e.g. (Bowers et al. 2000; Shen, Lange, et al. 2008; Raman et al. 2010)). Existing work in this area is based on the Rosetta *de novo* structure prediction method (see Section 3.3.1).

3.3.1 Rosetta supported by NMR data

The Rosetta *de novo* protein structure prediction method has been used to support structure determination by NMR. Bowers *et al.* (Bowers et al. 2000) use a variant of the method described in (Simons et al. 1997) in which the three- and nine-residue fragments used for modelling segments of the model protein are scored according to agreement with a multiple sequence alignment, but also compatibility with ϕ and ψ backbone torsion angle restraints derived from NMR chemical shift assignments. Further, those fragments that are incompatible with short-range NOE distance restraints are discarded. That approach is extended and evaluated in chemical-shift-Rosetta (CS-Rosetta) (Shen, Lange, et al. 2008). The resulting protocol, CS-RDC-Rosetta, uses both backbone NOEs and backbone RDCs, and is found to give improved convergence over CS-Rosetta for some test cases. This protocol does not always produce accurate models, and sometimes substantial parts of the model do not converge to an unambiguous conformation. Further, large proteins are difficult to model using this approach: "for proteins with over 120 residues, conformational sampling becomes limiting" (Raman et al. 2010). This limitation is due to the conformational search strategy used by Rosetta, and is not a limitation of the NMR experiment or the RDC data.

3.3.2 CP supported by NMR data

The constraint programming approach can be used to resolve a protein structure using NMR data. The PSICO system (Krippahl and Barahona 2002) constructs protein models that minimise constraint violations, where many thousands (5-10) of distance constraints obtained from NMR spectroscopy experiments are taken into consideration. All the constraints, including the bond angle constraints (represented in terms of distance constraints) are divided into two types: In constraints and Out constraints. The approach has shown to be able to produce protein models much faster (a few minutes) than other algorithms used by NMR (a few hours).

Chapter 4

ZAM implementation

The approach proposed in this project is based on a few main components:

- representation of the protein chain;
- a source of alternative local conformations;
- representation of angles;
- method for combining fragments;
- a small number of easy-to-obtain distance and torsion angles constraints provided by NMR;
- additional inferred knowledge-based distance and torsion angles constraints.

There are 2 main versions (see Section 4.1 and Section 4.2) of the protein modelling program that have been developed until now. The main difference between the versions is the way a protein main chain is represented and how different cells in the "pyramid" are being used: we start with a simple main chain representation, where a chain is a string of C^{α} atoms, and develop the system to the stage where we can simulate all heavy atoms (non-hydrogen) of the main chain and use longer fragments from known structres for protein modelling.

4.1 Cα version

In the C^{α} implementation a simple and easy prototype of the protein main chain is combined with zipping and assembly algorithm for modelling protein three-dimensional structure. The "building blocks" for zipping and assembly are the three-C^{α} units that are placed in the cells in "level 3" of the pyramid. One "pseudo angle" is defined by three consecutive C^{α} atoms. The oversimplification that has been made here is very useful for testing ability of the approach to model protein-like structures. Using this version we were able to model all PDB files stored in the PDB angle library with RMSD < 5 Å over the core (constrained regions in protein). All the results are gathered in Section 6.3.

4.1.1 Protein main chain model

A protein main chain is represented as a string of C^{α} atoms separated with a distance of 3.8 Å (assuming a trans peptide plane). No other atoms or side chains are being modelled at this stage. This simplification is good enough for performing the tests in reasonable time (a few minutes).

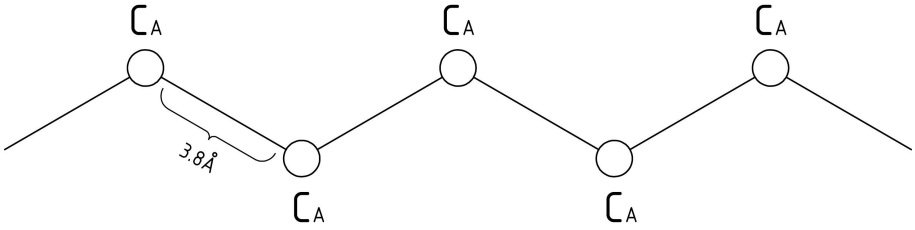

Figure 4.1: C^{α} representation of protein main chain.

4.1.2 PDB angle library

A source of alternative local conformations is represented by PDB angle library — a library of PDB files chosen according these criteria:

- A PDB file contains a single chain (usually A);
- A protein structure has from 1 or more disulphide bonds;
- Chain length should be less than 150 residues;

18 PDB files were chosen after filtering on the PDB website, so that there would be more variations in structures, no metal ions present and good resolution.

4.1.3 Angle representation

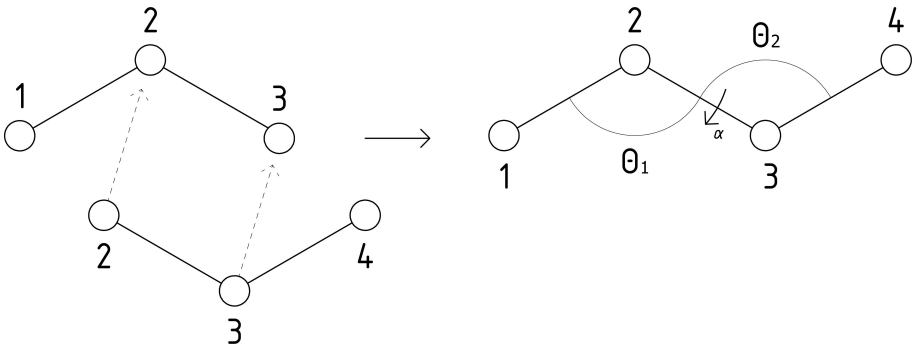

Figure 4.2: Assembly of two shorter fragments to form a longer fragment.

C^{α} version of protein modelling program uses two types of angles: "pseudo angles" and one "pseudo torsion angle" (Figure 4.2): θ_1, θ_2 and α. θ angles are selected

Table 4.1: PDB angle library

Name	Number of residues
1BOS	69
1CRN	46
1EI0	38
1EIG	73
1FD3	41
1HRP	86
1IJV	36
2B88	58
2BRZ	53
2HP8	68
2K5W	111
2LRD	62
2LWL	45
2PSM	117
4PTI	58
5B1F	129
5CKA	99
5CN2	114

when we construct the fragments in the base level cells. α angles are selected when two fragments are combined together (Section 4.1.4). For α-helical regions, the α torsion angle is equal to 50° and θ is 89° as standard values. In other cases α is taken randomly from the table of angles provided by angle library.

4.1.4 Zipping and assembly method implementation

The principle of how two fragments are combined to form a longer fragment in shown in Figure 4.3. In level 3 (base level) residues are connected with each other using theta θ_1. Starting from level 4 the algorithm combines one 3-residue fragment with another 3-residue fragment. The transformation that will splice fragment 2 onto the end of fragment 1 as shown in Figure 4.2. If A, B and C are the last three atoms in fragment 1, and let B', C' and D' be the first three atoms in fragment 2, both fragments will be brought into the same frame of reference by placing C (and C') at the origin, and B (and B') on the negative z-axis (Figure 4.4). When fragment 2 is spliced onto the end of fragment 1, the α angle at the join has an arbitrary value. The rotation will be around the z-axis (recall that B and C lie on the z-axis). The fragments being joined do not have internal clashes, so there is only a need to look for clashes between fragment 1 and the transformed fragment 2.

The number of final solutions in the top cell is defined by user and usually is chosen to be between 100 and 1000. For each fragment in the top cell of the pyramid, the coordinates of the atoms in that fragment will be written in PDB format to a

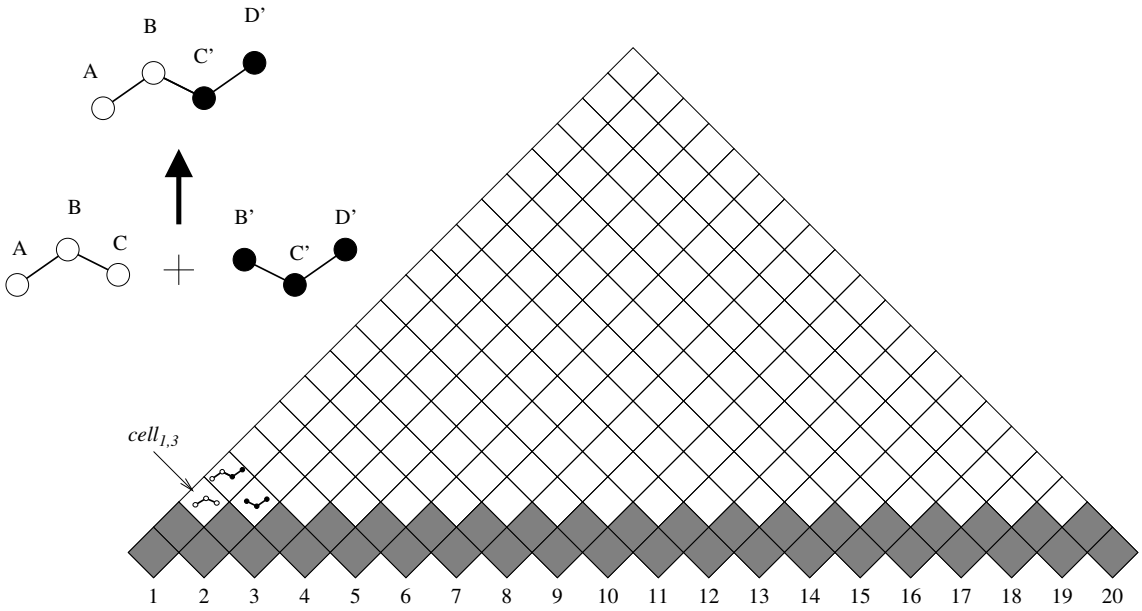

Figure 4.3: Zipping and assembly data structure for Cα version. Level 1: single residues; Level 2: 2 residues connected with each other (3,8 Å); Level 3 (base level): choice of θ_1 (3 residues connected with each other); Level 4 and above: choice of α (two 3-residue fragments are combined together).

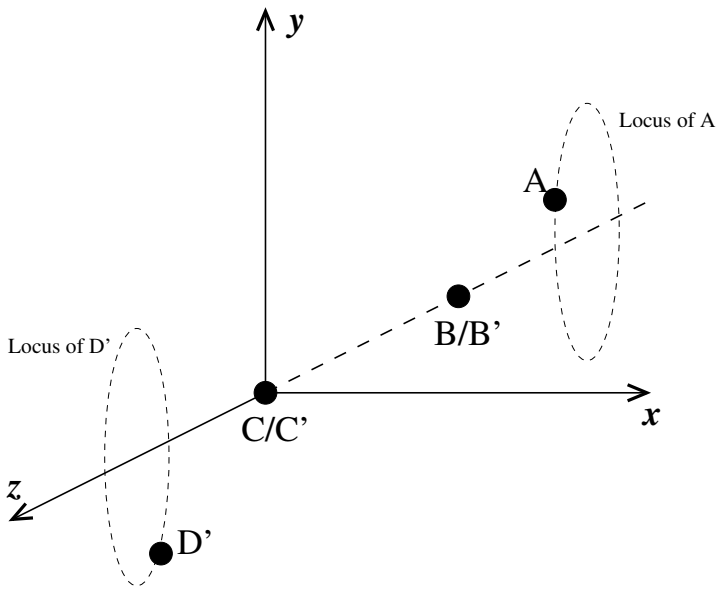

Figure 4.4: Frame of reference for combining two fragments.

file and can be viewed using molecular graphics applications like RasMol (Sayle and Milner-White 1995). All the results are discussed in details in Chapter 6.

4.2 All heavy atom version (non-hydrogen)

In this version an attempt is made to models proteins that have all heavy (non-hydrogen) atoms. The "building blocks" for zipping and assembly are amino acid residues and all ϕ and ψ combinations (that are consistent with angle ranges specified in a file, see Section 5.3) that present in the library of provided PDB files are placed in the cells in "level 1".

4.2.1 All heavy atom version: protein main chain model

Due to the complexity of the folding problem a simplified model of proteins is being used. This design reflects all main features that are essential for protein folding modelling. We are mainly concerned with modelling the main chain of a protein to reduce the number of degrees of freedom caused by the side chains (Figure 4.5). Side chains are not being prioritized, because the folding of a protein could be mainly described with proteins main chain. Side chains are discussed in Section 7.2.1.

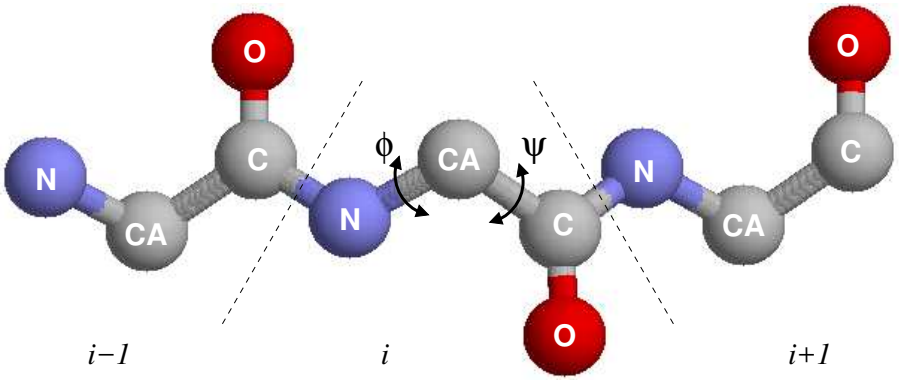

Figure 4.5: Protein main chain

The atoms included in the model are non-hydrogen atoms of the protein (Figure 4.5). This simplification can be applied due to the fact that hydrogen atom can bind to only one atom (oxygen, nitrogen or carbon).

4.2.2 PDB angle library

An angle library provides possible ϕ and ψ angle combinations, derived from residues in the proteins mentioned in Section 4.1.2. A Ramachandran plot for all proteins is shown in Figure 2.4. We concatenate all of the chains from the library of PDB files, and mutate all residues to alanine.This information will be used when constructing our own models. Suppose there are four chains in our PDB library, and that these have 12, 12, 12 and 46 residues. The first and the last residue in a sequence never has torsion angle predictions (ϕ and ψ respectively):

$$numAngles = N - 2$$

where N is a number of residues. The angle table will contain 74 entries (10+10+10+44). The polyalanine chain constructed here will have 76 residues in total. Construction

of a polyalanine has been made as a pre-step for using longer fragments (Section 7.2.4). The first residue is placed with respect to the first phi angle in the angle table. The last residue is placed with respect to the last psi angle in the angle table. We put into each cell amino acid residues having all of ϕ and ψ conbinations satisfying the ϕ and ψ range contraints that are observed in the library of PDB (templates) structures that are read when the program is run. This set gives a variety of different structures that help to reconstruct a protein's three-dimensional structure from a given sequence. When modelling a protein its original PDB file is never included.

4.2.3 Zipping and assembly method implementation

This version of the program constructs fragments in each cell of the pyramid starting with single residues placed at level 1:

$$Cell_{i,i} \quad i = 1, ..., N$$

where N is number of residues in the target sequence (see Figure 4.6).

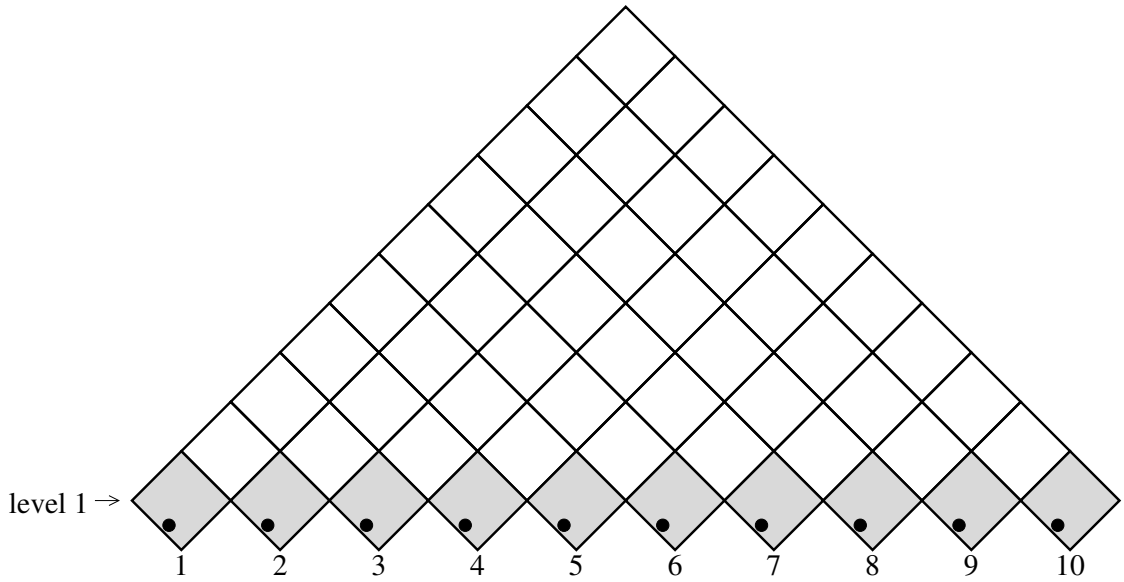

Figure 4.6: The first level of the zipping and assembly data structure (the pyramid). N is number of residues in the target sequence and is equal to 10 in this case.

There are two alternatives for choosing ϕ and ψ pair for each of the residues on the first level:

1. There is no torsion angle prediction for this residue. In this case all possible ϕ and ψ combinations taken from the PDB angle library (Section 4.2.2) are being used for creating conformations for this residue.

2. There is a torsion angle prediction made by TALOS+ for this residue. The program first generates conformations for this residue with all ϕ and ψ combinations taken from the PDB library and then selects only those compatible with the prediction made for this residue. In the end, the number of alternative

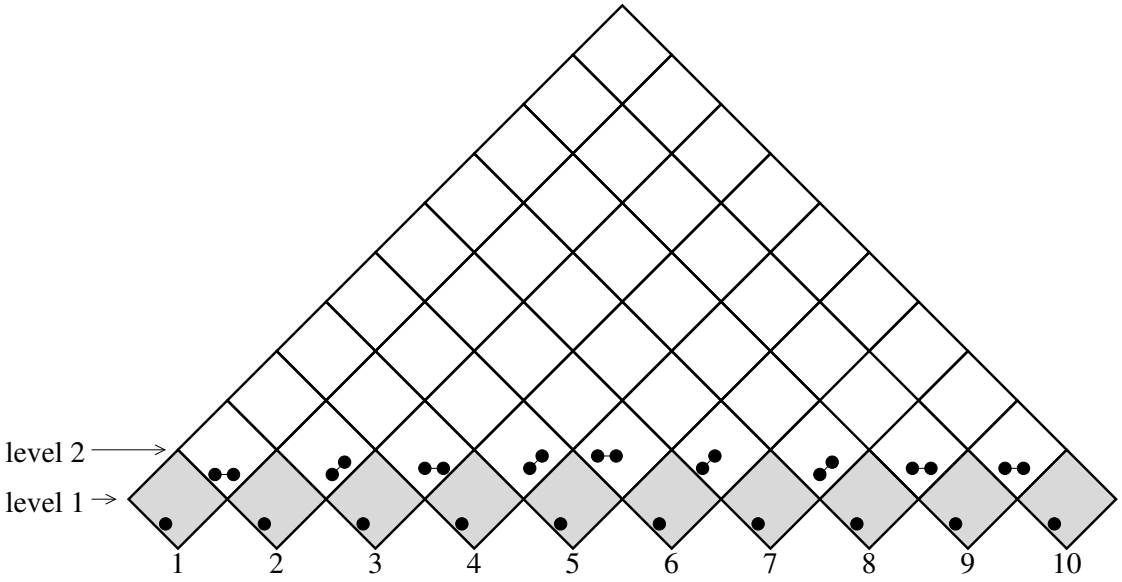

Figure 4.7: The second level of the zipping and assembly data structure (the pyramid) containts two neighbour residues connected with a peptide bond.

conformations for a residue with TALOS+ prediction on ϕ and ψ will be lower than for the residues with no constraints. That means that we have finite domains for each variable.

After filling the first level, the program continues with the second level that stores fragments with two residues in each cell (see Figure 4.7):

$$Cell_{i,i+1} \quad i = 1, ..., N-1$$

On the second level of the pyramid single neighbour residues from level 1 ($residue_i$ ($Cell_{i,i}$) and $residue_{i+1}$ ($Cell_{i+1,i+1}$), $i = 1, ..., N-1$, where N is a number of residues in a sequence) are combined to create a fragment that contains two residues connected with a peptide bond.

Steps for combining 2 residues together:

1. Step 1 (Figure 4.8): N and C^{α} atoms. Find the transformation that fits a peptide plane onto residue i-1, then apply that transformation to atoms N and C^{α} of the peptide plane to bring these into the frame of reference of residue i-1. If the positions of the main chain atoms of residue i-1 are known, then the positions of the N and C^{α} atoms are determined (assuming that the peptide plane has a rigid trans conformation).

2. Step 2 (Figure 4.9): C and C^{β}. If the positions of the main chain atoms of residue i-1 and the N and C^{α} atoms of residue i are known, and the phi angle of residue i is known, then the positions of atoms C and C^{β} are determined.

   If the positions of the main chain atoms of residue i-1 and the N and C^{α} atoms of residue i are known, and the phi angle of residue i is known, then the positions of atoms C and C^{β} are determined.

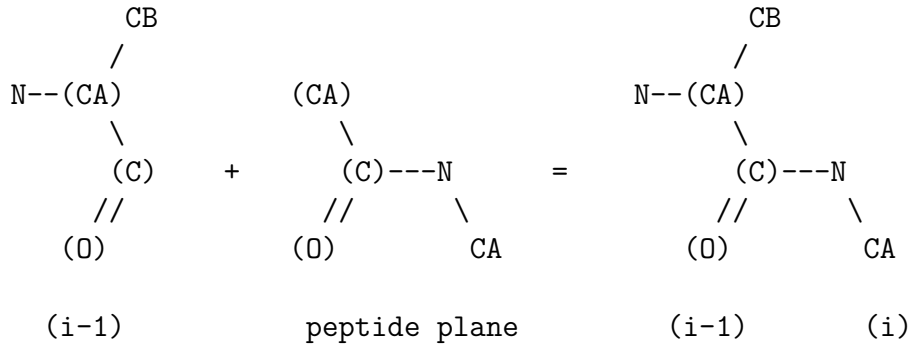

Figure 4.8: Joining two residues together: step 1. Atoms in brackets are those atoms that are being superposed.

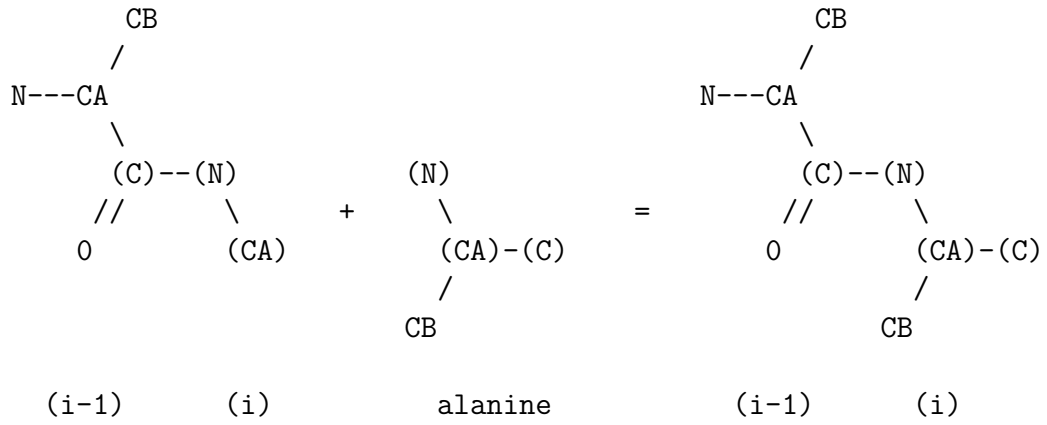

Figure 4.9: Joining two residues together: step 2. The 4 atoms in brackets are the ones that define ϕ torsion angle.

3. Step 3 (Figure 4.10): O.

   If the positions of the N, C^{α} and C atoms of residue i are known, and the psi angle of residue i is known, then the position of atom O is determined.

   Assuming a rigid trans peptide plane, the dihedral angle defined by atoms N, C^{α}, C, O has the value (ψ - 180°).

   The positions of atoms N and (assuming that the peptide plane has a rigid trans conformation) C^{α} of residue i+1 are also determined. Those atoms will be added to the polypeptide chain at the start of the next interation.

The third level is made by combining two fragments ($Cell_{i,i+2}$) to create a new fragment that is three residues long (Figure 4.11).

The program accepts only those fragments that do not have internal clashes and compatible with the distance constaints (Chapter 5). When checking for internal

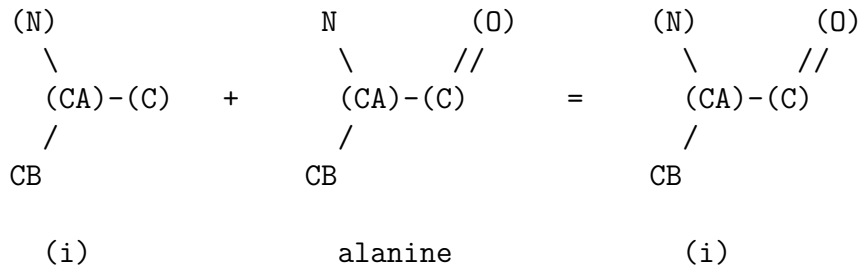

Figure 4.10: Joining two residues together: step 3. Atoms in brackets are those that define the ψ torsion angle.

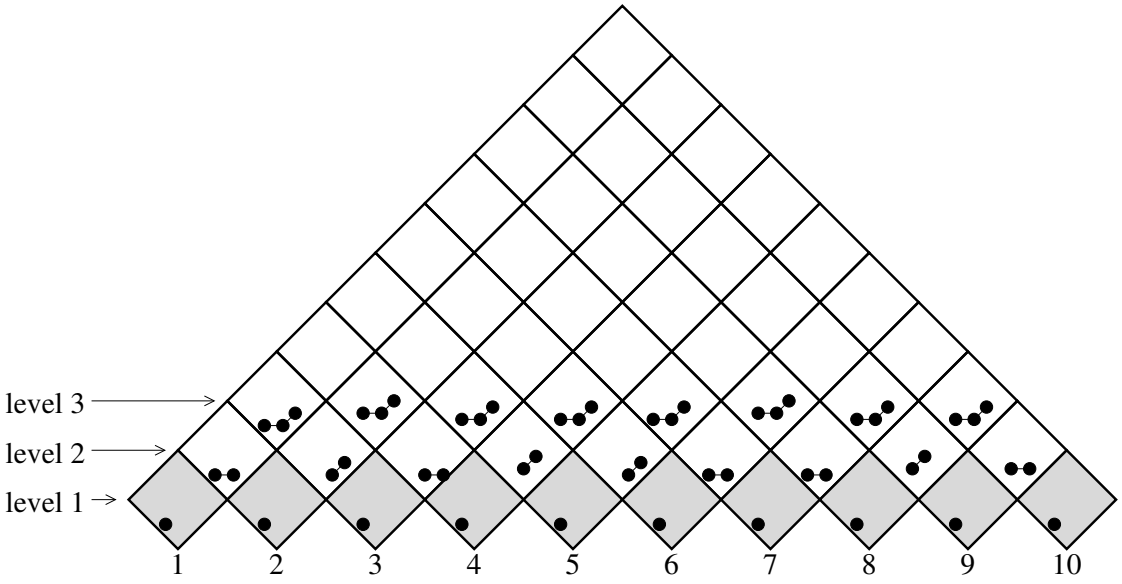

Figure 4.11: The third level of the zipping and assembly data structure (the pyramid). To create fragments in $Cell_{i,i+2}$ that are 3 residues long, the program can pick a random fragment from $Cell_{i,i}$ (one residue long) from the left and a random fragment from $Cell_{i,i+1}$ (two residue long). An alternative way is to pick $Cell_{i,i+1}$ (two residue long) from the left and $Cell_{i,i}$ (one residue long) from the right. All these combinations will contribute in a final number of solutions (defined by the user).

clashes, only main chain atoms, C^{β} atoms and the atoms in proline side chains are considered; possible clashes involving atoms beyond the C^{β} of other residues are not considered. Thus conformations with feasible main chains are not rejected due to side chain clashes, since it might be possible to remove these clashes in a subsequent modelling step (see Section 7.2.1). Assuming that feasible ϕ and ψ torsion angles have been achieved, there is no need to check for steric clashes between adjacent residues. Further, there will be some acceptable close contacts between adjacent residues, e.g. the peptide bond between the C atom of one residue and the N atom of the next residue. If there are any distance constraints for a particular cell, only

the fragments that are compatiable with these will be saved.

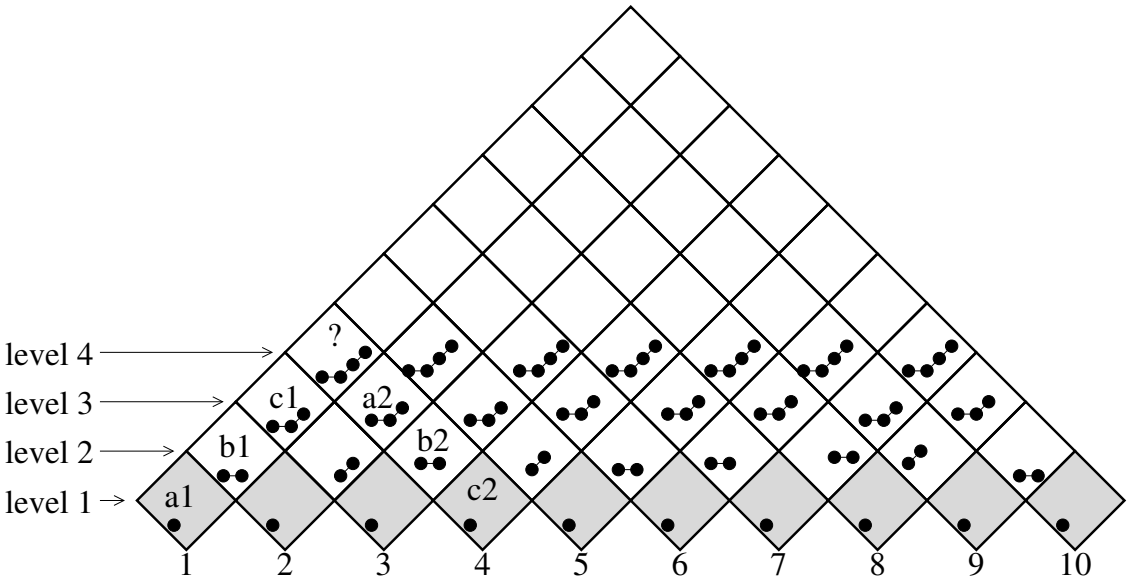

Figure 4.12: The fourth level of the zipping and assembly data structure (the pyramid) The cell with a question mark can be formed combining a1+a2, b1+b2, c1+c2 each of which contain a pool of different fragments.

On the fourth level and upwards all the fragments will be combined the same way it is done on the level three. Fragments generated in the upper cells are only those fragments that are compatiable with all distance constraints and do not have any violations. In the top $Cell_{i,N}$ the program writes out M protein structures saved in PDB format that are compatiable with all constraints given, where M can be specified by the user (Figure 4.13). These protein models are then superposed on each other to see clustering and calculate the RMSD with the original PDB file. The quality of the PDB file provided can vary, that is why the RMSD value cannot always be considered to be a quality measure.

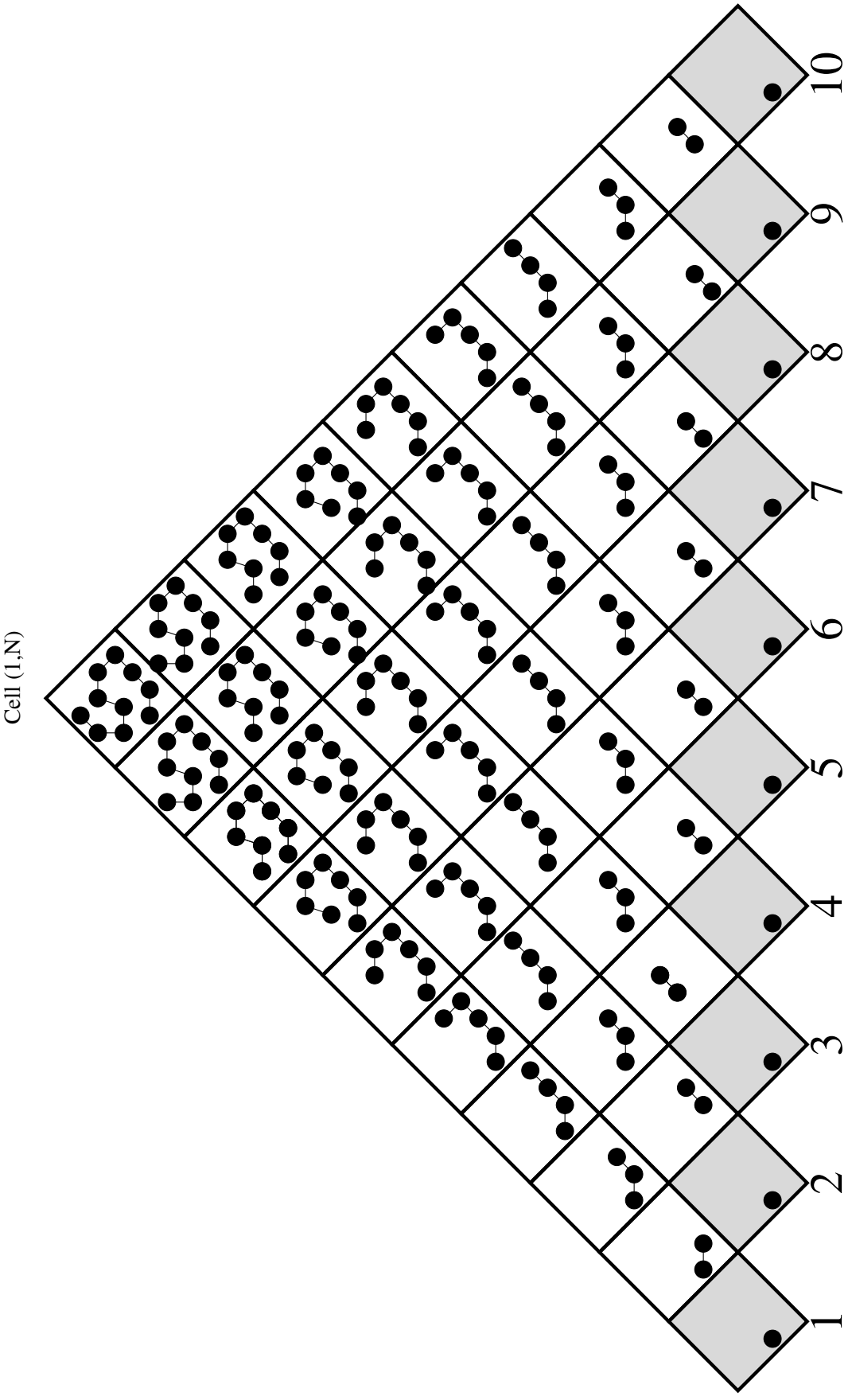

Figure 4.13: Zipping and assembly data structure after running the program for a protein with 10 residues.

Chapter 5

Constraints

In the work, we attempt to build protein models using much smaller sets of constraints than is required by other computational methods. This approach will allow us to obtain informative three-dimensional protein models using only those restraints that are obtained from relatively quick NMR experiments. In addition to distance constraints, we make use of constraints on torsion angle values and knowledge-based constraints. The protein native conformation search is biased and is guided by a small number (10-20) of constraint obtained by straight-forward NMR experiments and constraints based on general knowledge of protein structures. The information about protein structure can vary from case to case (see Chapter 6). In many cases we will have access to constraints that will be discussed in this chapter:

1. distance constraints (Section 5.1) derived from experimental data: distances between different pairs of atoms;

2. ϕ and ψ torsion angle prediction made with TALOS+ (Section 5.3);

3. inferred constraints based on knowledge about proteins and particular geometrical features.

For converting the high-level representation of a protein into distance and angle constraints a logic programing language Prolog is used (Kowalski 1988). The declarative nature of this language allows to express all rules in terms of relations and facts.

5.1 Distance constraints

The distance constraints used in this work include upper and lower distance bounds between pairs of atoms. This information can be obtained from a variety of sources:

1. **Elements of protein secondary structure** (α helices and strands in a β-sheet) are associated with tight distance constraints between main chain nitrogen (N) and oxygen (O) atoms that are involved in hydrogen bonds.From NMR experiments, the HN-HN distance constraints from NOEs determine the

extent of the α helices as well as the relative position and orientation of strands in a β-sheet. Almost no other short HN-HN distances are expected in a protein; these NOEs therefore provide constraints with a very high information content. This allows, in contrast to regular structure calculations from NMR data, to rely on **a very low number of constraints**.

(a) α **helices**: For alpha helical regions the distances between i and [i+4] residues could be inferred due to the fact that these distances define alpha helix (Barlow and Thornton 1988). Main chain nitrogen and oxygen atoms that are involved in hydrogen bonds should be within [2.5, 3.5] Å. Suppose that information about the extents of α-helices are represented as a set of Prolog facts: `alpha_helix(Start, End)`. Upper and lower distance constraints can be asserted using the code in Figure 5.1:

```
helix_distance_constraints :-
   alpha_helix(Start, End),
   X is Start - 1,
   Y is End - 3,
   between(I,X,Y),
   N is I + 4,
   assert_upper_distance_bound((I,'O'),(N,'N'),3.5,helix),
   assert_lower_distance_bound((I,'O'),(N,'N'),2.5,helix),
   fail.

helix_distance_constraints.
```

Figure 5.1: Prolog code for angle constraints for alpha helices.

(b) **Antiparallel bridges (strands in a β-sheet)**: If two residues are involved in an antiparallel bridge, then it places tight upper and lower constraints on the distances between oxygen and nitrogen and corresponding C^{α} atoms (Figure 2.7). Main chain nitrogen and oxygen atoms that are involved in hydrogen bonds should be within [2.5, 3.5] Å and corresponding C^{α} atoms have to have separation of 6 Å. In case if it is known that residues A and B are involved in antiparallel bridge. This information ican be represented using Prolog: `antiparallel_bridge(A, B)`. Upper and lower distance constraints can be asserted using the code in Figure 5.2.

2. **Disulphide bonds**: A disulphide bond can be formed between the sulphur atoms of a pair of spatially adjacent cysteine residues. If it is know that two cysteine residues form a disulphide bond, then their atoms must be sufficiently close. This places upper and lower distance constraints on their C^{α} atoms (Thornton 1981). This information can be represented using Prolog: `disulphide_bond(A,B)`. Upper and lower distance constraints can be asserted using the code in Figure 5.3.

```
antiparallel_bridge_distance_constraints :-
   antiparallel_bridge(A,B),
   assert_upper_distance_bound((A,'O'),(B,'N'),3.5,antiparallel_bridge),
   assert_upper_distance_bound((A,'N'),(B,'O'),3.5,antiparallel_bridge),
   assert_lower_distance_bound((A,'O'),(B,'N'),2.5,antiparallel_bridge),
   assert_lower_distance_bound((A,'N'),(B,'O'),2.5,antiparallel_bridge),
   assert_upper_distance_bound((A,'CA'),(B,'CA'),6.0,antiparallel_bridge),
   fail.

antiparallel_bridge_distance_constraints.
```

Figure 5.2: Prolog code for creating lower and upper distance constraints due to antiparallel bridges.

```
disulphide_distance_constraints :-
    disulphide_bond(A,B),
    assert_lower_distance_bound((A,'CA'),(B,'CA'),4.0,disulphide),
    assert_upper_distance_bound((A,'CA'),(B,'CA'),7.0,disulphide),
    fail.

disulphide_distance_constraints.
```

Figure 5.3: Prolog code for creating distance constraints in case of disulphide bonds.

5.2 Inferred distance constraints

There are cases when we can infer extra distance constraint due to:

1. If we have a constraint that C^{α} atoms of residues i and j must be within distance d from each other, this places upper distance bounds on atoms in the residues between i and j. The expected separation between the C^{α} atoms of two consecutive residues is 3.8 Å (Engh and Huber 1991; Laskowski et al. 1993) and the triangle inequality can be used to infer additional distance constraints involving the residues between i and j. Prolog code for propagating distance constraints to lower cells in the zipping and assembly data structure is shown in Figure 5.4.

2. Additional constraints are inferred from information about the position of disulphide bridges and general knowledge of protein conformation. Inferred upper distances are calculated as a triangle inequality for neighbour residues to those involved in disulphide bridges. If two cystein residues are involved in a disulphide bridge (residues A and B), Prolog code for inferring distance constraints between disulphide bond partners and adjacent residues can be implemened as it is shown in Figure 5.5:

   The example of when adjacent residues happen to be in disulphide bonds is shown and discussed in Section 6.2.

```
infer_upper_bounds :-
    upper_distance_bound((A,'CA'),(B,'CA'),Distance,Reason),
    between(I,A,B),
    between(J,I,B),
    not(upper_distance_bound((I,_),(J,_),_,_)),
    MustNotBeMoreThan is (((B - A) - (J - I)) * 3.8) + Distance,
    CannotBeMoreThan is (J - I) * 3.8,
    MustNotBeMoreThan < CannotBeMoreThan,
    assert_inferred_upper_distance_bound((I,'CA'),(J,'CA'),
                                    MustNotBeMoreThan,Reason),
 fail.

infer_upper_bounds.
```

Figure 5.4: Prolog code for propagating extra distance constraints to lower cells (inferred constraints).

```
in_antiparallel_bridge(X) :- antiparallel_bridge(X,_).
in_antiparallel_bridge(X) :- antiparallel_bridge(_,X).

disulphide(A,B) :- disulphide_bond(A,B).
disulphide(A,B) :- disulphide_bond(B,A).


disulphide_distance_constraints :-
    disulphide_bond(A,B),
    assert_lower_distance_bound((A,'CA'),(B,'CA'),4.0,disulphide),
    assert_upper_distance_bound((A,'CA'),(B,'CA'),7.0,disulphide),
    fail.

disulphide_distance_constraints :-
    disulphide(A,B),
    disulphide(C,D),
    1 is C-B,
    in_antiparallel_bridge(B),
    in_antiparallel_bridge(C),
    assert_lower_distance_bound((A,'CA'),(D,'CA'),13.0,from_disulphide),
    assert_upper_distance_bound((A,'CA'),(D,'CA'),15.0,from_disulphide),
    fail.

disulphide_distance_constraints.
```

Figure 5.5: Prolog code for inferring distance constraints between disulphide bond partners and adjacent residues.The first clause asserts upper and lower distance constraints between the C^{α} atoms of the two residues that are involved in the disulphide bridge (extension of Figure 5.3). The second clause tests whether residues B and C are adjacent in the protein chain, are both present in a strand and asserts upper and lower constraints on the distance between the C^{α} atoms of the disulphide bond partners of residues B and C.

```
write_helix_angle_constraints :-
    alpha_helix(Start, End),
    EndMinusOne is End - 1,
    between(X, Start, EndMinusOne),
    residue(X, XName),
    format('~t~p~4+   ~p  PHI     -71.0    -57.0~n', [X, XName]),
    format('~t~p~4+   ~p  PSI     -48.0    -34.0~n', [X, XName]),
    fail.

write_helix_angle_constraints.
```

Figure 5.6: Prolog code for writing angle constraints for α helices (inferred constraints).

```
write_proline_angle_constraints :-
    residue(X,'PRO'),
    format('~t~p~4+   ~p  PHI     -78.0    -48.0~n', [X, 'PRO']),
    fail.

write_proline_angle_constraints.
```

Figure 5.7: Prolog code for angle constraints for proline residues.

5.3 Torsion angles constraints

There can be place upper and lower bounds on the values of main chain torsion angles ϕ and ψ. These can come from knowledge about the protein's secondary structure (e.g. based on HN-HN NOEs from NMR experiments) or torsion angle ranges predicted by TALOS+ from chemical shifts data (Shen, Lange, et al. 2008) and secondary structure information based on HN-HN NOEs.

5.4 Inferred angle constraints

There are cases when we can infer extra angle constraint due to:

1. α helices

   Helical regions have constraints on possible ϕ and ψ angle combinations (Thornton 1981). Average dihedral angles (ϕ, ψ) have values (-64° ± 7°, -41° ± 7°). That allows us to infer extra constaints on ϕ and ψ angles combinations in the alpha helical regions using Prolog 5.6.

2. PRO (proline) Special case are PRO (proline) residues due to its rigid structure (Figure 5.7). The limits on ϕ value of about (-63° ± 15°).

Chapter 6

Results

To demonstrate the use of zipping and assembly with constraints, we have attempted to reconstruct the structure of proteins using only their amino acid residue sequences, secondary structure information (the extents of α-helix regions and information about antiparallel bridges), and disulphide bond pairings as starting information.

The length of the test proteins vary from 35-80 residues. The resulting models are compared with experimentally determined structures from the Protein Data Bank by superposing corresponding C^{α} atoms on each other. Most of the models that we have generated have RMSD < 3 Å in the core.

The modelling process and results are presented in detail in this chapter for two proteins: human p8MTCP1 (Section 6.1) and human β-defensin 6 (Section 6.2). Both of these proteins have three disulphide bonds. Human p8MTCP1 (68 amino acid residues) has three α-helix regions, while human β-defensin 6 (45 amino acid residues) has a mainly β-sheet structure with several antiparallel bridges. These proteins were chosen as a good examples to demonstrate what kind of structural information can be derived from existing constraints. Some other tests are also presented and discussed in this chapter (Section 6.3). All proteins from the PDB angle library (Section 4.2.2) were modelled, using C_{α} version of the program.

6.1 Human p8MTCP1 [PDB entry: 2HP8]

The structure of human p8MTCP1 (PDB entry: 2HP8) has been determined experimentally by NMR (Barthe et al. 1997). The protein was chosen as an easy example to demonstrate the use of zipping and assembly method for modelling proteins because:

— the protein is short: it has 68 residues;

— there are 3 regions of α-helix;

— there are 3 disulphide bonds.

High level information about this protein is encoded as Prolog facts and is shown in Figure 6.1. These structural features are illustrated in Figure 6.1 and are shown schematically in Figure 6.2. From these facts, a large set of upper and lower distance

```
residue(1,'MET').
residue(2,'PRO').
residue(3,'GLN').
residue(4,'LYS').
residue(5,'ASP').  % etc.

disulphide_bond(7,38).
disulphide_bond(17,28).
disulphide_bond(39,50).

alpha_helix(8,20).
alpha_helix(29,39).
alpha_helix(48,62).
```

Figure 6.1: Prolog facts describing structural features of human p8MTCP1: amino acid sequence, disulphide bonds and α-helices.

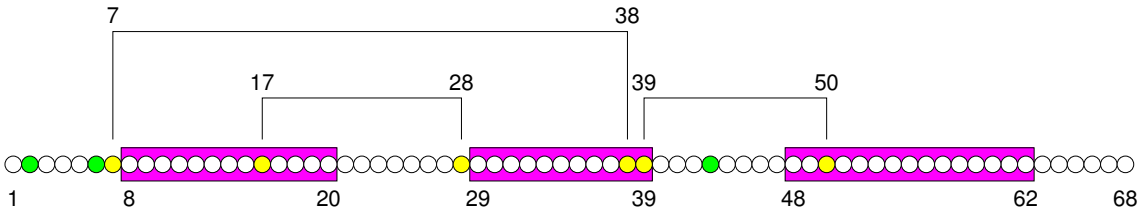

Figure 6.2: 2HP constaints: alpha helices: 8-20, 29-39, 48-62; disulphide bonds: 7-38, 17-28, 39-50. All the colours correspond to those, shown in Figure 6.3.

and angle constraints can be derived. Figure 6.3 shows all the constraints available for this protein, mapped onto zipping and assembly data structure. The yellow cells with letter "S" represent three disulphide bonds that provide tight distance constraints between pairs of residues at positions (7, 38), (17, 28) and (39, 50), and weaker distance constraints between pairs of residues represented by the other yellow cells. Proline residues at positions 2, 6 and 43 have tight constraints on the range of possible values for their ϕ torsion angle (green cells). It can be noticed that 2HP8 has unconstrained regions at the beginning and at the end of the chain. 2 α-helices are connected to each other with the disulphide bonds. This represenatation is a useful tool for predicting the outcome of the program in advanced.

The zipping and assembly method was run generating 1000 fragments in each cell in the data structure. In the current implementation we keep the first 1000 generated fragments that satisfy the distance constraints associated with the cell. Of the 1000 models built for the entire chain, the most similar model to the experimentally determined structure had a root mean square distance of 2.6 Å over all Cα atoms (Figure 6.5). Regions of greatest difference are close to the unconstrained ends of the chain. Comparing the 1000 models with each other the main differences are the orientation of the third helix which is only anchored to the second helix by a disulphide bond near one end (Figure 6.2).The program took a few minutes to run on an ordinary laptop.

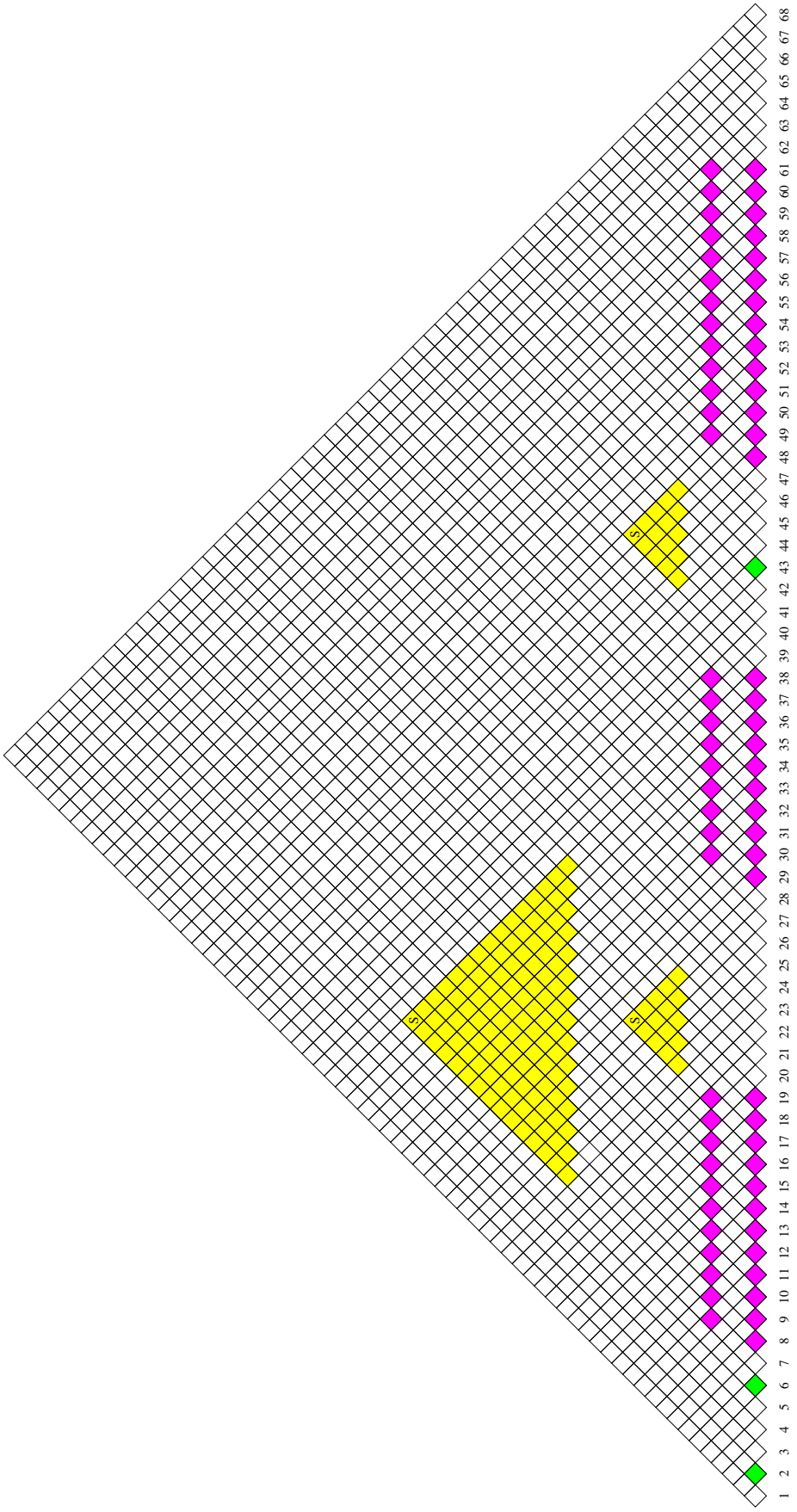

Figure 6.3: Constraints used in modelling human p8MTCP.

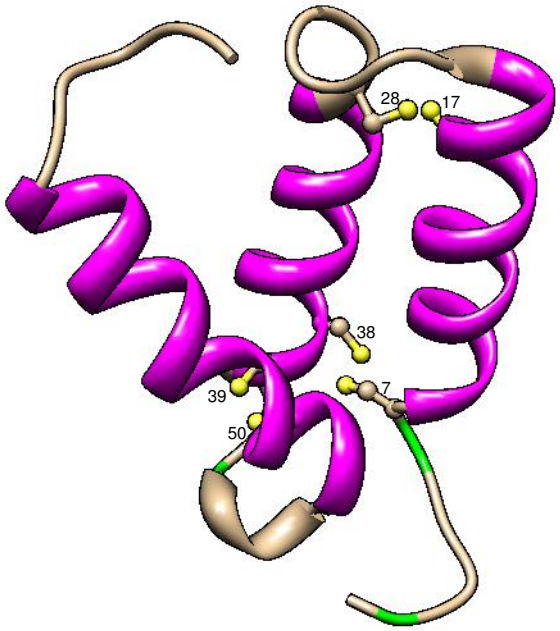

Figure 6.4: Human p8MTCP1 (Protein Data Bank entry 2HP8). The ribbon cartoon represents the main chain of human p8MTCP1. There are 7 Cys residues (side chains shown as ball-and-stick), 6 of which form three disulphide bonds (Cys7-Cys38, Cys17-Cys28, Cys39-Cys50).

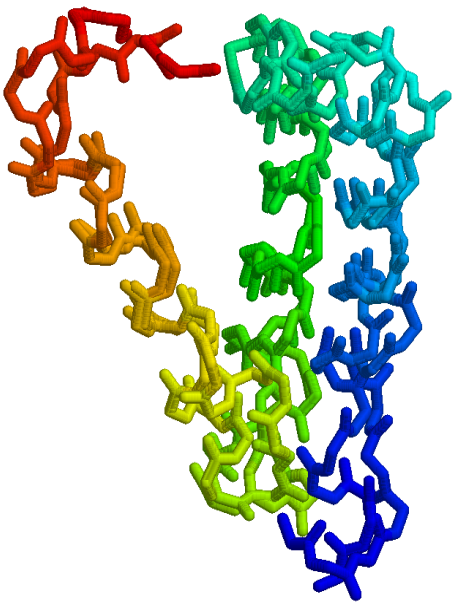

Figure 6.5: Main chain of modelled human p8MTCP, built using knowledge of the extents helical regions and its disulphide bridges superposed on the experimentally determined structure [PDB: 2HP8](Cα RMSD is 2.6 Å). The colour gradient allows the chains to be followed easily from the N-terminal (residue 1, blue) to the C-terminal (residue 68, red).

6.2 Human β-Defensin 6 [PDB entry: 2LWL]

The structure of human β-defensin 6 has been determined experimentally by NMR (PDB entry: 2LWL) (De Paula et al. 2013). This structure contains four antiparallel bridges (Figure 6.6). First and last residues do not have any constraints because of their conformational freedom (De Paula et al. 2013). A high level description of structural features of 2LWL is presented as Prolog facts in Figure 6.7.

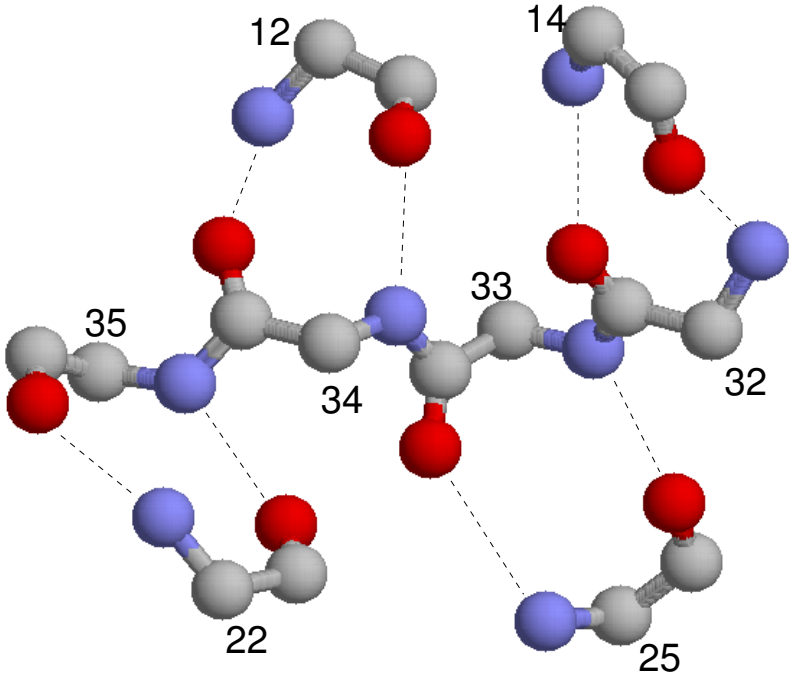

Figure 6.6: Antiparallel bridges in human β-defensin 6 inferred from HN-HN NOEs.

Initially we were unable to obtain models using only information in Figure 6.7, and two additional constraints were needed. The first of these could be inferred from the facts in Figure 6.7 using knowledge about protein structure, and the second was additional information provided by the NMR experiment.

— Within a strand in a β-sheet the side chains of consecutive residues are on alternate faces of the sheet, i.e. side chains of adjacent residues are oriented in opposite directions. In the case of human β-defensin 6, the residues at positions 33 and 34 are both cysteine residues involved in disulphide bonds. Since the side chains of residues 33 and 34 are oriented away from each other, the C^{α} atoms of their disulphide bond partners (residues 6 and 17) are expected to be between 13 Å and 15 Å apart. This situation is illustrated in Figure 6.8 and Figure 6.9. A rule for implementing this has been used in Prolog (Figure 5.5). Additional strong constraints found to be at the lower cells [in the pyramid], help to guide search towards feasible solutions early in the search.

— NH-NH distances for residues 29 and 30 hinted at a turn with these residues having main chain torsion angles that are similar to those in helices. Interesting

```
residue(1,'PHE').
residue(2,'PHE').
residue(3,'ASP').
residue(4,'GLU').
residue(5,'LYS').  % etc.

disulphide_bond(6,33).
disulphide_bond(13,27).
disulphide_bond(17,34).

alpha_helix(4,8).

antiparallel_bridge(12,34).
antiparallel_bridge(14,32).
antiparallel_bridge(22,35).
antiparallel_bridge(25,33).
```

Figure 6.7: Prolog facts describing structural features of human β-defensin 6.

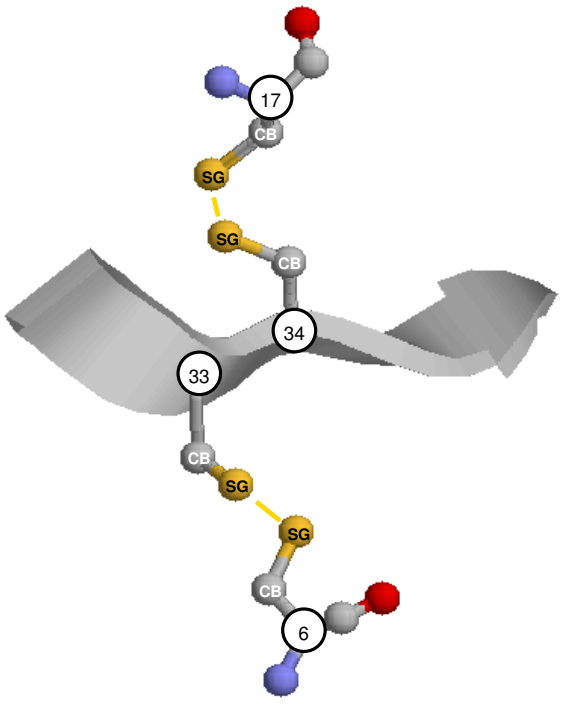

Figure 6.8: The large grey arrow represents a strand in a β-sheet. C^{α} atoms are represented by numbered circles. Consecutive residues 33 and 34 lie within the strand, and both are involved in disulphide bonds (shown in yellow) with residues 6 and 17, respectively. This situation places upper and lower distance constraints between residues 6 and 17.

that it was not clear whether it was a right-handed or left-handed helix. Potential ϕ and ϕ values for these residues are:

$$helix_{right}: \quad \phi\ [-180°, 0°], \quad \psi\ [-80°, -40°]$$
$$helix_{left}: \quad \phi\ [40°, 80°], \quad \psi\ [-30°, 90°]$$

All the constraints derived for defensin are "mapped" on to the zipping and assembly data structure in Figure 6.9. This representation helps to visualize all the constraints available for this protein and "predict" the outcome of the program. We made an attempt to build this model where these residues had a right-handed conformation and left-handed. The zipping and assembly method was only able to build models with left-handed conformation for these residues in the experimentally determined human β-defensin 6 structure. The most similar model to the experimentally determined structure had a root mean square distance of 3 Å over C^{α} atoms in the core region spanning residues 6 to 35 (Figure 6.10). Superposed C^{α} traces of 50 model structures show that the models agree well in core, but vary considerably in their unconstrained regions near the ends of the chain which experimentally are shown to be dynamic (Figure 6.11).

6.3 Other tests

There have been more tests performed for protein modelling, using zipping and assembly algorithm, shown in Table 6.1. Some proteins were more challenging than others. Mapping sets of available constraints (which can vary from case to case) helps to visualize and estimate difficulty in modelling a protein. Constraints for the α-helical hairpin of P8MTCP1 (PDB entry: 1EI0) are shown in Figure 6.12. This protein has two α helices (4-16 and 25-37) connected to each other with two disulphide bridges (3-34 and 13-24).

There are still proteins that we have not yet succeeded. Figure 6.13 and Figure 6.14 show constraints for 2LRD and 1EIG — proteins with 61 and 73 residues respectively. Mapping constraints for the monomeric Acanthaporin (PDB entry: 2LRD) on the zipping and assembly data structure makes it visually more obvious that this protein can be difficult to model. 2LRD is a protein with 61 amino acids, 5 α helical regions and 5 disulphide bonds. The left and the right sides of the pyramid are not connected with each other (roughly residues 5-45 and 48-61). These two could be modelled separetly, but still there is no rule that would bring those two parts together. Another example of a protein with a "difficult" structure is the human chemokine eotaxin-2 (PDB entry: 1EIG). Its constraints are shown in Figure 6.14. This protein consists of 73 amino acid residues, has one α helix, 4 antiparallel bridges and 1 disulphide bond. It can be seen from the "pyramid" (as in example discussed above) that 1EIG could be potentially challenging to model. The left part is well defined with the constraints (residues 6-51), while the right part (α helix, residues 53-68) is also well-defined, but there are no constraints that can anchor these parts together.

Figure 6.9: Constraints used in modelling 2LWL: yellow: distance constraints due to disulphide bonds; cyan: antiparallel bridges; green: inferred constaints from adjacent CYS).

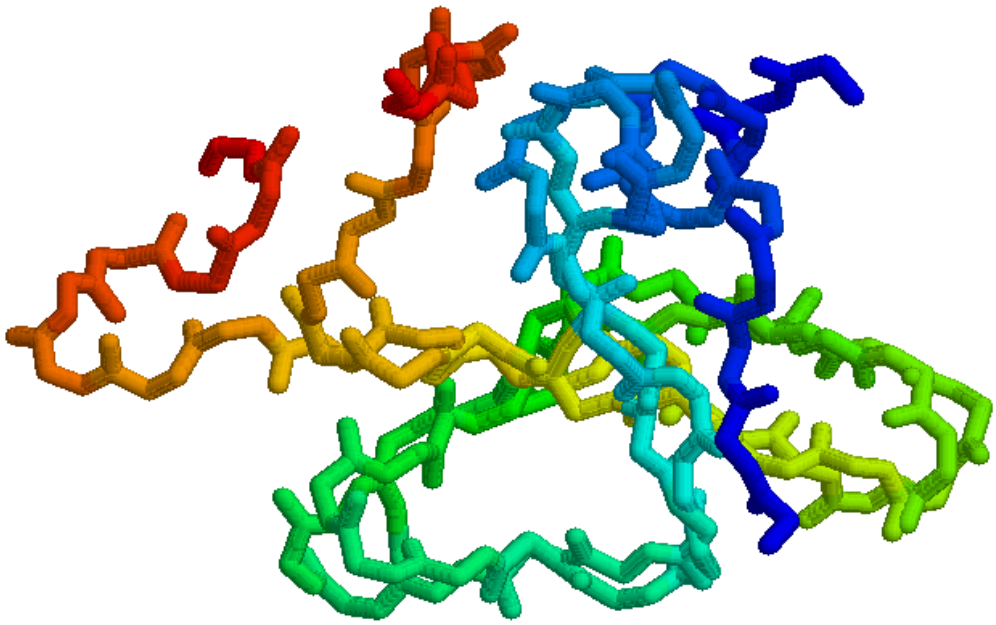

Figure 6.10: Main chain model of human β-defensin 6 superposed on the main chain of experimentally determined structure from PDB entry 2LWL.

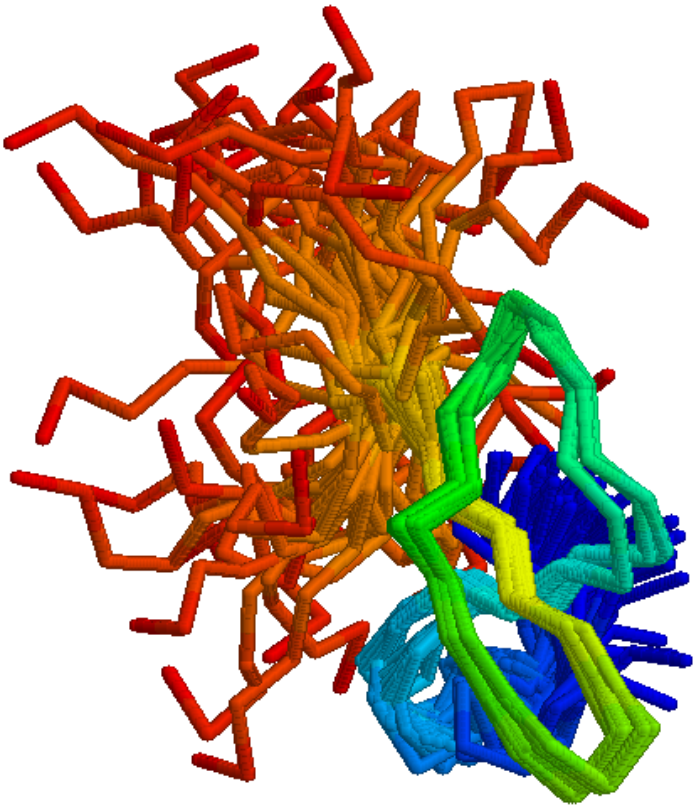

Figure 6.11: C^{α} traces of 50 models of human β-defensin 6. The structures of the core are in good agreement, There is good agreement in the core, but the terminal regions are dynamic.

Table 6.1: Modelled proteins

Name	Number of residues	RMSD (CA version), Å	RMSD (All heavy atom version), Å
1BOS	69	3.8	-
1CRN	46	2,6	-
1EI0	38	2.0	2.0
1EIG	73	4.4	-
1FD3	41	2.2	-
1HRP	86	8.4	-
1IJV	36	2.5	-
2B88	58	2.6	2.7
2BRZ	53	4.5	-
2HP8	68	2.6	2.3
2K5W	111	5.8	-
2LRD	61	3.3	-
2LWL	45	3.0	2.6
2PSM	117	6.7	-
4PTI	58	3.2	-
5B1F	129	7.0	-
5CKA	99	6.6	-
5CN2	114	7.3	-

Figure 6.12: Constraints "mapped" on the zipping and assembly data structure for 1EI0. α helices are shown in margenta and disulphide bridges are shown in gold. This structue is well-defined what makes it "easy" to model.

Figure 6.13: Constraints "mapped" on the zipping and assembly data structure for 2LRD. α helices are shown in margenta and disulphide bridges are shown in gold. This structue is well-defined, but different parts of the protein are not connected with each other, which makes it more difficult to model.

Figure 6.14: Constraints "mapped" on the zipping and assembly data structure for 1EIG. α helices are shown in margenta and disulphide bridges are shown in gold, antiparallel bridges are shown in blue.

Chapter 7

Discussion

A prototype system that attempts to model all heavy atoms in a protein's main chain in a way that is compatible with a given set of constraints has been implemented (Wånggren et al. 2016). The combination of zipping and assembly with NMR-derived constraints is novel. This chapter discusses some of the benefits and limitations of the system (Section 7.1), and some of the directions for extending the work (Section 7.2).

7.1 Benefits and limitations

Using the approach, described in the thesis, we are able to construct model protein structures that are compatible with given constraints. The source of the consatraints and the number of constraints that are available can vary from case to case.

Starting with very few constraints, it is possible to infer many more constraints, based on the geometrical properties of a protein or basic knowledge of protein structure.

While we use relatively fewer initial constraints that other methods (Section 3.3), it is difficult to compare how many fewer constraints are used. Some groups' models include acceptable bond lengths and angles as constraints. In contrast, we use standard protein geometry or conformations taken directly from a library of known protein structures. Thus we do not have constraints that model acceptable bond lengths and angles.

Some proteins are easier to model than others using our system, and some cannot currently be modelled satisfactorily. Those that cannot be modelled satisfactorily generally fall into two categories: (i) those where the ZAM algorithm is unable to find any feasible solutions and (ii) those where the ZAM algorithm finds many feasible solutions including many that do not resemble the target protein's native conformation. The first case may be due to the problem being "over-constrained" or the conformational space not being explored adequately. Possible approaches to overcoming these problematic cases are suggested in Section 7.2.3. The second case is due to the problem being "under-constrained" and many alternative solutions are found that are compatible with the given constraint set. Possible approaches to overcoming this problem are to seek additional constriants or to employ a scoring

function to help select among fragments and models (Section 7.2.5). There are possibilities for inferring extra constraints from the given set. For example, if it is known that in a protein, a metal ion is stabilized by cysteine residues, it places strict distance constraints between atoms of these groups.

To make it easy for the user to understand the program's operation for a given set of constraints, tools have been implemented that automatically generate visual representations of the constraints mapped onto the zipping and assembly data structure (as shown in Figure 6.9) and that generate a visual representation of the cells that contribute to a model or to a set of models (as shown in Figures 7.1 and 7.2). This is where mapping the constraints onto zipping and assembly data structure can be a useful tool for analyzing the outcome of the program, as has been shown in Chapter 6. Observed deficiencies of the current prototype provide the motivation for futher research.

7.2 Future work

The zipping and assembly method has shown (Chapter 6) to be a promising approach for 3D protein modelling based on a small number of constraints provided by straightforward NMR experiment (see Figure 6.11, 6.5). However, there are a few tasks that need to be accomplished and several problems need to be solved for better performance and result.

7.2.1 Modelling side chains

So far our focus has been on modelling protein main chains, and we have not placed side chains carefully. Side chain modelling could be carried out within the zipping and assembly framework or in a separate modelling step that follows construction of the main chain.

The most straightforward way to perform side chain modelling within the zipping and assembly framework would be to expand the set of residue conformations in the lowest cells in the zipping and assembly data structure by adding alternative side chain rotamer conformations (Ponder and Richards 1987; Dunbrack and Karplus 1993) to each main conformation. The ZAM algorithm would then make selections from among these alternative residue conformations when building fragments at higher levels.

An alternative approach would be to keep separate the tasks of modelling the main chain and modelling side chains. In this way we could continue to focus on only modelling the protein's main chain using the zipping and assembly method, then add side chains onto the complete main chain (e,g, (Swain and Kemp 2001; Traoré et al. 2013)).

7.2.2 Some cells are more important than others

In our current work we have found that some cells in the zipping and assembly data structure are more important than others; when modelling a particular protein we

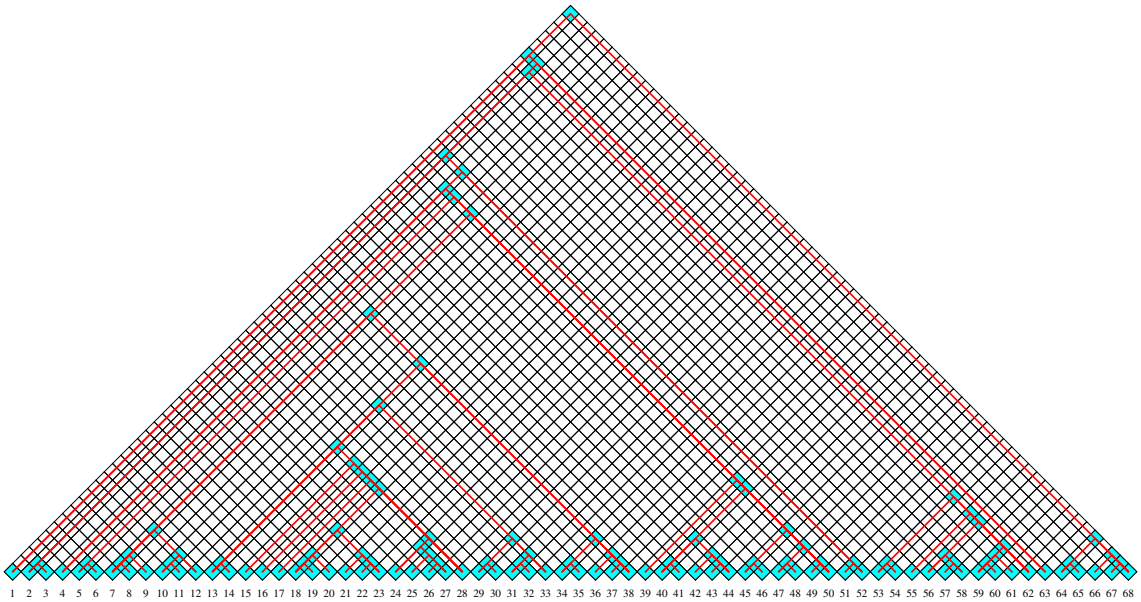

Figure 7.1: Cells that contribute fragments to one model of a 68-residue protein (cyan). The cyan cell at the apex represents a model spanning all 68 residues. The red lines form a binary tree; from each cyan cell above level 1 there are two red lines (one left and one right) leading to the lower cyan cells from which fragments were selected and combined to form a fragment in the higher cell.

generate the same number of fragment conformations for each intermediate cell, but some cells contribute fragments to many of the final models while other cells contribute rarely or not at all. This is illustrated in Figures 7.1 and 7.2. Figure 7.1 shows the cells that contributed fragments in building one model of a 68-residue protein. When using the zipping and assembly method to build a protein of length N residues starting with only individual residue conformations from the lowest level, fragments in $N - 2$ intermediate cells are used in building a model in the entire protein; the vast majority of cells do not contribute to that model. If we go on to build 50 or 100 or 1000 models, other cells might contribute fragments to those models (Figure 7.2), with some cells contributing fragments to many models, while other cells are rarely or never used. This observation suggests that it is wasteful in computational time (the time taken to compute many fragment conformations for many cells) and memory (needed to store the conformations of many fragments in cells that are unlikely to contribute to the final models). Reducing memory requirements would enable more models to be built for each target protein, and would enable longer proteins to be modelled.

7.2.3 Overcoming bottlenecks

When our current program succeeds in generating models, a few hundred models can typically be built in a few minutes on a laptop. However, in some modelling runs one or more constraints can create a bottleneck in generating feasible conformations. This can either slow the execution, or even block the method from finding a feasible conformation altogether. At present these problems are detected by the

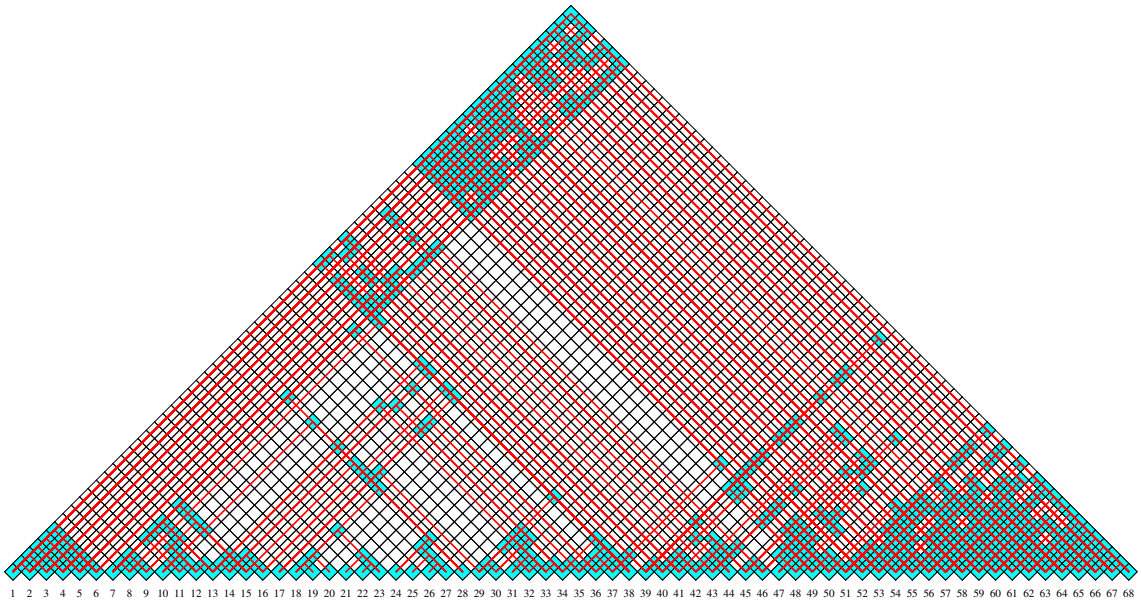

Figure 7.2: Cells that contribute fragments to fifty models of a 68-residue protein.

user monitoring the program's diagnostic output, and s/he must decide whether to abort the program and restart it with a different set-up. Strategies to overcome the problem causing the bottleneck include generating more conformations in each cell (increasing the size of the search space), or relaxing some constraints (to "widen the bottlenecks") or adding/tightening other constraints (to guide the search towards feasible solutions). The question that needs to be explored is how to detect and overcome bottlenecks automatically.

7.2.4 Fragment-based approach

The computational framework for zipping and assembly can be adapted to allow fragments with "known" conformation to be incorporated into the model, thus taking advantage of structural knowledge.

A recent article (Wang et al. 2017) discusses the utility of using peptide fragments from the Protein Data Bank in protein modelling. Wang et al. state, however, that the zipping and assembly method is a non-fragment-based technique. While zipping and assembly can build models starting with conformations for individual residues (from the lowest level in the zipping and assembly data structure), it is possible to place conformations of longer peptide fragments directly into higher cells. We have implemented code that searches through a library of protein chains for long fragments ("long" here is a fragment of at least two residues) that satisfy all of the constraints over a range of residue positions in the protein being modelled. Thus, our implementation already accommodates the use of peptide fragments of arbitrary length. This feature has not yet been thoroughly evaluated, but the work is in progress.

Using fragments from known structures (either based on compatibility with RDCs , or local sequence-structure preferences) would be different to previous off-lattice work

with zipping and assembly, which uses molecular mechanics (rather than database search, as proposed here) to generate fragment conformations. That might increase the chance to build a good model and save some computational time when modelling larger proteins. At the same time this approach would let us save the computational time and focus only on those cells that actually are more important in the building models.

7.2.5 Scoring function

Our current implementation focuses on obtaining a set of model structures that are compatible with the given set of constraints. All fitting conformations are considered equally in the present implementation. Scoring becomes a natural step for a sucessful protein modelling. We shall investigate the utility of various scoring functions for evaluating and selecting fragments, e.g. to select those fragments from a cell that will be used to create longer fragments (in higher cells in the zipping and assembly data structure). It is anticipated that different scoring functions will be most useful at different stages as modelling proceeds. For example, favourable atom contacts could be more important when evaluating short fragments, whereas more global features like compactness of the modelled protein chain is important when evaluating longer fragments.

A scoring function could be based on physics of molecular interactions or statistics and knowledge of the protein conformation. An electrostatic potential might be the most relevant "physical" contribution during folding. Maybe a similar, *ad hoc* hydrophobic potential could be added in order to give points to early forming hydrophobic cores or filtering based on hydrophibic contacts. Finding a suitable scoring function is one of the challenges in any protein modelling program. This is even a weakness in Rosetta, the leading program in the protein folding area — a study of four modelling cases highlights “the poor discrimination of the Rosetta all-atom energy function” (Das 2011).

Chapter 8

Conclusion

We have implemented a method for modelling protein main chains that uses high level declarative descriptions of protein features as its starting point. Lower level distance and angle constraints can be generated automatically from these. Expert structural knowledge that can be crucial in finding satisfactory solutions is expressed as declarative Prolog rules that are used to infer additional constraints. Declarative Prolog rules are used to propagate distance constraints, so that unpromising solutions are pruned early.

The zipping and assembly method for exploring the vast conformational space has several benefits (Dill, Lucas, et al. 2007): it offers a competitive alternative to other computational methods. The “divide-and-conquer” nature of the dynamic programming algorithm used for zipping and assembly should allow larger proteins to be tackled; today this is still a big problem. Our future plan is to continue to develop the method (zipping and assembly algorithm), apply a scoring function based on biophysical knowledge, start using "ready" fragments for known structures to decrees the number of possible solutions and optimize the runtime and memory use.

The approach proposed here is targeted towards improving our understanding of protein folding and how nature (almost) always achieves the correct fold. The project enables us to obtain informative three-dimensional protein models using only those restraints that are obtained from relatively quick NMR experiments. This will help to reduce or even eliminate the need for more expensive multidimensional NMR experiments that require complex isotope labelling to be done and take longer to perform.

The software developed in this research will be of use to NMR groups in academia and in the pharmaceutical industry. The aim is that by combining a zipping and assembly modelling approach, together with relatively easily obtained data from NMR experiments, experimental groups will be able to obtain useful models without the need to invest time and money in performing additional experiments.

Bibliography

Anfinsen, C.B. and H.A. Scheraga (1975). "Experimental and theoretical aspects of protein folding". In: *Advances in protein chemistry* 29, pp. 205–300 (cit. on p. 9).

Backofen, R. (1998). "Constraint techniques for solving the protein structure prediction problem". In: *International Conference on Principles and Practice of Constraint Programming*. Springer, pp. 72–86 (cit. on p. 15).

Backofen, R. and S. Will (2006). "A constraint-based approach to fast and exact structure prediction in three-dimensional protein models". In: *Constraints* 11.1, pp. 5–30.

Baker, D. and A. Sali (2001). "Protein Structure Prediction and Structural Genomics". In: *Science* 294, pp. 93–96 (cit. on pp. 1, 14).

Baldwin, R.L. and G.D. Rose (1999). "Is protein folding hierarchic? I. Local structure and peptide folding". In: *Trends in biochemical sciences* 24.1, pp. 26–33 (cit. on p. 18).

Barlow, D. J. and J.M. Thornton (1988). "Helix geometry in proteins". In: *J. Mol. Biol.* 201, pp. 601–619 (cit. on pp. 7, 36).

Barthe, P. et al. (1997). "Solution structure of human p8 MTCP1, a cysteine-rich protein encoded by the MTCP1 oncogene, reveals a new α-helical assembly motif". In: *J. Mol. Biol.* 274, pp. 801–815 (cit. on p. 41).

Berman, H., K. Henrick, and H. Nakamura (2003). "Announcing the worldwide Protein Data Bank". In: *Nat. Struct. Biol.* 10, p. 980 (cit. on pp. 5, 10, 21).

Bernstein, F. C., T. F. Koetzle, G. J. B. Williams, E. F. Mayer, M. D. Bruce, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi (1977). "The Protein Data Bank: a Computer-based Archival File for Macromolecular Structures". In: *J. Mol. Biol.* 112, pp. 535–542 (cit. on p. 10).

Billeter, M., G. Wagner, and K. Wüthrich (2008). "Solution NMR structure determination of proteins revisited". In: *Journal of biomolecular NMR* 42.3, pp. 155–158 (cit. on pp. 1, 12).

Bowers, P. M., C. E. M. Strauss, and D. Baker (2000). "De novo protein structure determination using sparse NMR data". In: *J. Biomol. NMR* 18, pp. 311–318 (cit. on pp. 13, 20, 21).

Brunger, A.T. (2007). " Version 1.2 of the Crystallography and NMR System". In: *Nature Protocols* 2, pp. 2728–2733 (cit. on p. 13).

Brunger, A.T. et al. (1998). "Crystallography and NMR System (CNS): A new software suite for macromolecular structure determination". In: *Acta Cryst.D* 54, pp. 905–921 (cit. on p. 13).

Campeotto, F., A. Dal Palù, A. Dovier, F. Fioretto, and E. Pontelli (2013). "A constraint solver for flexible protein model". In: *Journal of Artificial Intelligence Research* 48, pp. 958–1000 (cit. on p. 15).

Carver, T. R. and C.P. Slichter (1956). "Experimental verification of the Overhauser nuclear polarization effect". In: *Physical Review* 102.4, p. 975 (cit. on p. 12).

Chen, V.B., W.B. Arendall, J.J. Headd, D.A. Keedy, R.M. Immormino, G.J. Kapral, L.W. Murray, J.S. Richardson, and D.C. Richardson (2010). "MolProbity: all-atom structure validation for macromolecular crystallography". In: *Acta Crystallographica Section D: Biological Crystallography* 66.1, pp. 12–21 (cit. on p. 11).

Cornilescu, G., F. Delaglio, and A. Bax (1999). "Protein backbone angle restraints from searching a database for chemical shift and sequence homology". In: *Journal of biomolecular NMR* 13.3, pp. 289–302 (cit. on p. 13).

Crippen, G.M. (1978). "Are there pathways for protein folding?" In: *J. Mol. Biol.* 126, pp. 315–332 (cit. on p. 17).

Dal Palù, A., A. Dovier, and F. Fogolari (2004). "Constraint logic programming approach to protein structure prediction". In: *BMC bioinformatics* 5.1, p. 186 (cit. on p. 15).

Dal Palù, A., A. Dovier, F. Fogolari, and E. Pontelli (2010). "CLP-based protein fragment assembly". In: *Theory and Practice of Logic Programming* 10, pp. 709–724.

Das, R. (2011). "Four small puzzles that Rosetta doesn't solve". In: *PLoS One* 6, e20044 (cit. on pp. 14, 59).

De Paula, V. S., N. S. F. Gomes, L. G. Lima, C. A. Miyamoto, R. Q. Monteiro, F. C. L. Almeida, and A. P. Valente (2013). "Structural Basis for the Interaction of Human β-Defensin 6 and Its Putative Chemokine Receptor {CCR2} and Breast Cancer Microvesicles". In: *J. Mol. Biol.* 425, pp. 4479–4495 (cit. on p. 45).

Delaglio, F., G. Kontaxis, and A. Bax (2000). "Protein Structure Determination Using Molecular Fragment Replacement and NMR Dipolar Couplings". In: *J. Am. Chem. Soc.* 122, pp. 2142–2143.

Dill, K. A. (1985). "Theory for the folding and stability of globular proteins". In: *Biochemistry* 24.6, pp. 1501–1509 (cit. on p. 13).

Dill, K. A., A. Lucas, J. Hockenmaier, L. Huang, D. Chiang, and A. K. Joshi (2007). "Computational linguistics: A new tool for exploring biopolymer structures and statistical mechanics". In: *Polymer* 48, pp. 4289–4300 (cit. on pp. 15, 17, 18, 61).

Dill, K. A., S. B. Ozkan, T. R. Weikl, J. D. Chodera, and V. A. Voelz (2007). "The protein folding problem: when will it be solved?" In: *Current Opinion in Structural Biology* 17, pp. 342–346 (cit. on p. 9).

Drenth, J. (2007). *Principles of protein X-ray crystallography*. Springer Science & Business Media (cit. on p. 11).

Dunbrack, R. L. and M. Karplus (1993). "Backbone dependent rotamer library for proteins. Application to side chain prediction". In: *J. Mol. Biol.* 230, pp. 543–574 (cit. on p. 56).

Engh, R.A. and R. Huber (1991). "Accurate bond and angle parameters for X-ray protein structure refinement". In: *Acta Crystallographica Section A: Foundations of Crystallography* 47, pp. 392–400 (cit. on pp. 5, 37).

Güntert, P. (2003). "Automated NMR protein structure calculation". In: *Progress in Nuclear Magnetic Resonance Spectroscopy* 43.3, pp. 105–125 (cit. on pp. 12, 13).

Güntert, P. (2004). "Automated NMR structure calculation with CYANA". In: *Protein NMR Techniques*, pp. 353–378 (cit. on p. 13).

Hockenmaier, J., A. K. Joshi, and K. A. Dill (2007). "Routes Are Trees: The Parsing Perspective on Protein Folding". In: *Proteins: Structure, Function and Bioinformatics* 66, pp. 1–15 (cit. on pp. 17, 18).

Hovmöller, S., T. Zhou, and T. Ohlson (2002). "Conformations of amino acids in proteins". In: *Acta Crystallographica Section D: Biological Crystallography* 58.5, pp. 768–776 (cit. on p. 7).

Jones, D.T. (1999). "GenTHREADER: An Efficient and Reliable Protein Fold Recognition Method for Genomic Sequences". In: *J. Mol. Biol.* 287, pp. 797–815 (cit. on p. 14).

Jones, T.A., J.-Y. Zou, S.W. Cowan, and M. Kjeldgaard (1991). "Improved methods for building protein models in electron density maps and the location of errors in these models". In: *Acta Crystallographica Section A: Foundations of Crystallography* 47.2, pp. 110–119 (cit. on p. 11).

Kabsch, W. and C. Sander (1983). "Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features". In: *Biopolymers* 22, pp. 2577–2637 (cit. on p. 7).

Karplus, M. (1997). "The Levinthal paradox: yesterday and today". In: *Folding and design* 2, S69–S75 (cit. on p. 13).

Kaufmann, K. W., G. H. Lemmon, S. L. DeLuca, J. H. Sheehan, and J. Meiler (2010). "Practically Useful: What the Rosetta Protein Modeling Suite Can Do for You". In: *Biochemistry* 49, pp. 2987–2998 (cit. on p. 14).

Kendrew, J.C., G. Bodo, H.M. Dintzis, R.G. Parrish, H. Wyckoff, and D.C. Phillips (1958). "A three-dimensional model of the myoglobin molecule obtained by x-ray analysis". In: *Nature* 181.4610, pp. 662–666 (cit. on pp. 5, 10).

Kowalski, R.A. (1988). "The Early Years of Logic Programming". In: *Communications of the ACM* 31, pp. 38–43 (cit. on p. 35).

Kraulis, P.J. and T. A. Jones (1987). "Determination of Three-Dimensional Protein Structures From Nuclear Magnetic Resonance Data Using Fragments of Known Structures". In: *Proteins: Structure, Function and Genetics* 2, pp. 188–201.

Krippahl, L. and P. Barahona (1999). "Applying constraint programming to protein structure determination". In: *International Conference on Principles and Practice of Constraint Programming*. Springer, pp. 289–302 (cit. on p. 15).

Krippahl, L. and P. Barahona (2002). "PSICO: Solving Protein Structures with Constraint Programming and Optimization". In: *Constraints* 7, pp. 317–331 (cit. on pp. 15, 21).

Laskowski, R.A., M.W. MacArthur, D.S. Moss, and J.M. Thornton (1993). "PROCHECK: a program to check the stereochemical quality of protein structures". In: *Journal of applied crystallography* 26.2, pp. 283–291 (cit. on pp. 11, 37).

Lau, K.F. and K. A. Dill (1989). "A Lattice Statistical Mechanics Model of the Conformational and Sequence Spaces of Proteins". In: *Macromolecules* 22, pp. 3986–3997 (cit. on pp. 15, 17).

Levinthal, C. (1968). "Are there pathways for protein folding?" In: *Journal de Chimie Physique* 65, pp. 44–45 (cit. on p. 13).

Linge, J.P., M. Habeck, W. Rieping, and M. Nilges (2003). "ARIA: automated NOE assignment and NMR structure calculation". In: *Bioinformatics* 19, pp. 315–316 (cit. on p. 13).

Overhauser, A.W. (1953). "Polarization of nuclei in metals". In: *Physical Review* 92.2, p. 411 (cit. on pp. 1, 12).

Ozkan, S. B., G. A. Wu, J. D. Chodera, and K. A. Dill (2007). "Protein folding by zipping and assembly". In: *Proc. Natl. Acad. Sci. USA* 104, pp. 11987–11992 (cit. on pp. 1, 14, 18).

Pauling, L. (1960). *The nature of the chemical bond and the structure of molecules and crystals: an introduction to modern structural chemistry*. Vol. 18. Cornell university press (cit. on p. 3).

Pauling, L. and R.B. Corey (1951). "The pleated sheet, a new layer configuration of polypeptide chains". In: *Proceedings of the National Academy of Sciences* 37.5, pp. 251–256 (cit. on p. 7).

Pauling, L., R.B. Corey, and H.R. Branson (1951). "The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain". In: *Proceedings of the National Academy of Sciences* 37.4, pp. 205–211 (cit. on p. 7).

Perrakis, A., R. Morris, and V.S. Lamzin (1999). "Automated protein model building combined with iterative structure refinement". In: *Nature structural & molecular biology* 6.5, pp. 458–463 (cit. on p. 12).

Plaxco, K.W., K.T. Simons, and D. Baker (1998). "Contact order, transition state placement and the refolding rates of single domain proteins". In: *J. Mol. Biol.* 277, pp. 985–994 (cit. on p. 18).

Ponder, J. W. and F. M. Richards (1987). "Tertiary templates for proteins. Use of packing criteria in the enumeration of allowed sequences for different structural classes". In: *J. Mol. Biol.* 193, pp. 775–791 (cit. on p. 56).

Prestegard, J.H., C.M. Bougault, and A.I. Kishore (2004). "Residual dipolar couplings in structure determination of biomolecules". In: *Chemical reviews* 104.8, pp. 3519–3540 (cit. on p. 12).

Ramachandran, G.N., C. Ramakrishnan, and V. Sasisekharan (1963). "Stereochemistry of polypeptide chain configurations". In: *J. Mol. Biol.* 7, pp. 95–99 (cit. on p. 5).

Raman, S. et al. (2010). "NMR Structure Determination for Larger Proteins Using Backbone-Only Data". In: *Science* 327, pp. 1014–1018 (cit. on pp. 14, 20, 21).

Rohl, C.A. and D. Baker (2002). "De novo determination of protein backbone structure from residual dipolar couplings using Rosetta". In: *Journal of the American Chemical Society* 124.11, pp. 2723–2729 (cit. on p. 20).

Rose, G.D. (1979). "Hierarchic organization of domains in globular proteins". In: *J. Mol. Biol.* 134.3, pp. 447–470 (cit. on p. 17).

Rost, B. (1999). "Twilight zone of protein sequence alignments". In: *Protein Engineering* 12, pp. 85–94 (cit. on p. 14).

Sander, C. and R. Schneider (1991). "Database of Homology-Derived Protein Structures and the Structural Meaning of Sequence Alignment". In: *Proteins: Structure, Function and Genetics* 9, pp. 56–68 (cit. on p. 14).

Sayle, R.A. and E.J. Milner-White (1995). "RASMOL: biomolecular graphics for all". In: *Trends in biochemical sciences* 20.9, pp. 374–376 (cit. on p. 26).

Shen, Y., F. Delaglio, G. Cornilescu, and A. Bax (2009). "TALOS+: a hybrid method for predicting protein backbone torsion angles from NMR chemical shifts". In: *Journal of biomolecular NMR* 44.4, pp. 213–223 (cit. on p. 13).

Shen, Y., O. Lange, et al. (2008). "Consistent blind protein structure generation from NMR chemical shift data". In: *Proc. Natl. Acad. Sci. USA* 105, pp. 4685–4690 (cit. on pp. 20, 21, 39).

Simons, K. T., C. Kooperberg, E. Huang, and D. Baker (1997). "Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences using Simulated Annealing and Bayesian Scoring Functions". In: *J. Mol. Biol.* 268, pp. 209–225 (cit. on pp. 1, 14, 21).

Sippl, M. J. and S. Weitckus (1992). "Detection of Native-Like Models for Amino Acid Sequences of Unknown Three-Dimensional Structure in a Data Base of Known Protein Conformations". In: *Proteins: Structure, Function and Genetics* 13, pp. 258–271 (cit. on p. 14).

Swain, M. T. and G. J. L. Kemp (2001). "A CLP approach to the protein side-chain placement problem". In: *Principles and Practice of Constraint Programming — CP2001*. Ed. by T. Walsh. Springer-Verlag, pp. 479–493 (cit. on p. 56).

Thornton, J.M. (1981). "Disulphide bridges in globular proteins". In: *J. Mol. Biol.* 151, pp. 261–287 (cit. on pp. 36, 39).

Tolman, J.R., J.M. Flanagan, M.A. Kennedy, and J.H. Prestegard (1995). "Nuclear magnetic dipole interactions in field-oriented proteins: information for structure determination in solution". In: *Proceedings of the National Academy of Sciences* 92.20, pp. 9279–9283 (cit. on p. 12).

Traoré, S., D. Allouche, I. André, S. de Givry, G. Katsirelos, T. Schiex, and S. Barbe (2013). "A new framework for computational protein design through cost function network optimization". In: *Bioinformatics* 29.17, pp. 2129–2136 (cit. on pp. 15, 56).

Ulrich, E.L., H. Akutsu, J.F. Doreleijers, Y. Harano, Y.E. Ioannidis, J. Lin, M. Livny, S. Mading, D. Maziuk, Z. Miller, et al. (2008). "BioMagResBank". In: *Nucleic acids research* 36.suppl 1, pp. D402–D408 (cit. on p. 13).

Voelz, V. A. and K. A. Dill (2007). "Exploring zipping and assembly as a protein folding principle". In: *Proteins: Structure, Function and Bioinformatics* 66, pp. 877–888 (cit. on p. 17).

Wang, T., Y. Yang, Y. Zhou, and H. Gong (2017). "LRFragLib: an effective algorithm to identify fragments for de novo protein structure prediction". In: *Bioinformatics* 33, pp. 6775–684 (cit. on p. 58).

Wånggren, M., M. Billeter, and G. J. L. Kemp (2016). "Computational protein modelling based on limited sets of constraints". In: *Proceedings of the 12th International Workshop on Constraint-Based Methods for Bioinformatics*. Ed. by A. Dal Palù, A. Dovier, and S. de Givry, pp. 99–113 (cit. on p. 55).

Younger, D.H. (1967). “Recognition and parsing of context-free languages in time n3”. In: *Information and Control* 10, pp. 189–208 (cit. on p. 18).