Protein Structure Prediction: Improving and Automating Knowledge-based Approaches

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von
Dipl.-Wirtsch.-Inf. Silvio Carlo Ermanno Tosatto
aus Bressanone
Italien

Mannheim, 2002

Dekan:	Professor Dr. Herbert Popp, Universität Mannheim
Referent:	Professor Dr. Reinhard Männer, Universität Mannheim
Korreferent:	Professor Dr. Jeremy Smith, Universität Heidelberg

Tag der mündlichen Prüfung: 19. April 2002

Contents

List of Figures

List of Tables

Acknowledgements

I thank Prof. Männer for his support and for providing our research group with a good working environment. Prof. Smith for his willingness to be a referee for this thesis. The Land Baden-Württemberg is acknowledged for awarding me a 2-year scholarship in the LGFG program.

I appreciate the research group in which I spent these years. I thank Dr. Jürgen Hesser for the lively discussions, support and ideas. Dr. Eckart Bindewald will always be remembered for the exceptional experience of participating in CASP-4 and for teaching me how to work efficiently with limited resources, as well as being a good friend. For enduring my supervision of their diploma theses Andreas Kindler and Achim Trabold. Marcus Prümmer and Jochen Maydt for doing a good job while working as "Hiwi". Michael Reuß for lively discussions.

I thank the staff at the Lehrstuhl für Informatik V, in particular Andrea Seeger and Christiane Glasbrenner, for making my life as easy as possible. I appreciated the discussions with several colleagues (in no particular order): Harald Simmler, Holger Singpiel, Martijn de Boer, Clemens Wagner, Karsten Mühlmann and his group, and the guys from our 11:30 lunch break to name just a few.

Several good friends I met on summer schools and conferences (in particular: Elide Formentin, Stefano Toppo, Mario Albrecht, Gianluca Pollastri, Emiliano Biasini, Stefan Wuchty, Claudio Anselmi and Timm Essigke) have given me the strength to hold on.

I am very grateful to the proof-readers of this thesis: Harald Simmler and Timm Essigke. Mario Albrecht deserves a special mention for delivering exaggeratedly long lists of errors, taking me a lot of time to correct but no doubt significantly improving the overall quality.

This thesis is dedicated to my family. My parents, and my sister, have always believed in me and supported me. Words cannot express how thankful I am.

Cum tibi contigerit studio cognoscere multa,
fac discas multa, vita nescire doceri
— **Catonis**

1
Introduction

1.1 Motivation

Proteins are fundamental to life. If DNA can be considered the organic "data storage", containing blueprints and control information for the cell, then proteins are the "molecular machines" used to perform all the vital functions. They have a remarkable variety of functions. For instance, they act as enzymes, catalyzing most biochemical reactions. They also have structural and functional roles on the cellular level and above. For example, the protein collagen is a major component of human skin and gives it stability. The actin-myosin protein complex is responsible for muscle contraction and thus macroscopic movement in living organisms. Proteins can function as signal transducers or "molecular switches", changing the state of other proteins or regulating DNA expression. In short, proteins are of central importance to almost every biological process.

Deciphering the human genome is a major scientific breakthrough and certainly the most important example of a genome project. Knowledge on the genome of a species, that is the sequence of base pairs in the DNA, is radically changing molecular biology. This information is, in theory at least, sufficient to understand the way an organism functions. However, the genome projects produce only the raw data that needs to be analyzed. Many questions are raised by the genomic data and interpreting it is the major task in molecular biology for years or decades to come.

The parts of the DNA sequence, which are today best understood, are the genes. These DNA segments that are translated into protein sequences by the cell, as shown in Figure 1.1. Apart from some recurrent sequence motifs, little

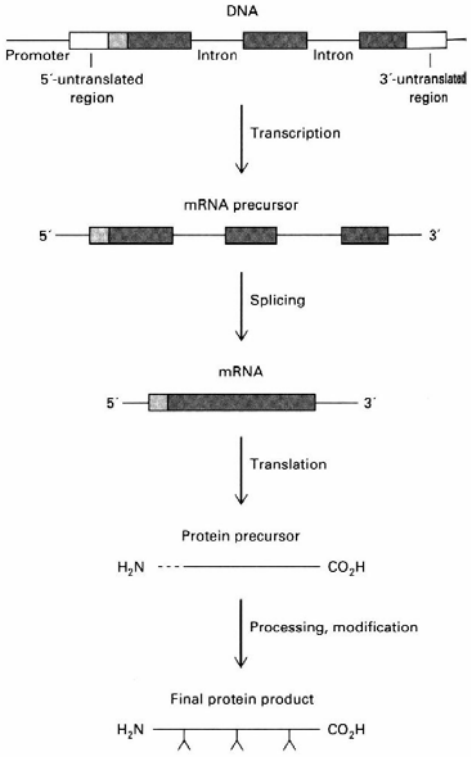

FIGURE 1.1. Protein synthesis from DNA.

is known about the function of non-coding DNA sequence and most of it is believed to be "garbage" introduced into the DNA during evolution and no longer needed.

The next logical step after the genome projects is to understand the function of the genes. This means to understand the function of proteins. Indirect experimental methods like knock-out experiments, where a particular gene is "turned off" and the changes in an organism are studied, provide only limited information on gene function. A better way to establish protein function passes through protein structure.

Proteins adopt specific structures to fulfill their different roles. Structural knowledge on a protein structure allows a researcher to establish hypotheses for its function. Most proteins interact with small molecules, called ligands, or other proteins to perform their function. These biochemical interactions are induced by the geometry of the protein and highly specific. They can be compared to a lock and its key.

Drug design is enormously facilitated by knowing the structure of the target protein. Specific molecules that dock into a groove on the protein surface

of a key protein can be designed to inhibit its function. Diseases caused by viruses and bacteria can be cured by selectively blocking a protein involved in cell reproduction [93]. Since protein sequences between bacteria and man are frequently related, a detailed knowledge about the structure is required to select drugs that do not affect humans.

Protein structure prediction has been long considered the "Holy Grail" of structural biology as it would greatly improve our understanding of life at the molecular level. The long time, over 30 years, it is being studied has produced a somewhat pessimistic view among some long-time researchers. However, over the last five to ten years things have constantly improved, thanks to better computer performance and the introduction of bioinformatics techniques. Based on recent results, the problem can be tackled for at least half of the known protein sequences. Methodological improvements are already raising the standards for what is considered a "good" protein structure prediction.

1.2 Objective

The objective of this thesis is two-fold. First of all, it aims to improve the performance of existing knowledge-based methods for protein structure prediction. Accurate modeling of protein structures goes beyond the analysis of its sequence. Techniques using structural informations are only being developed over the last years and are by far not as well-studied as sequence similarity in all of its guises. Good modeling methods would not only improve the biochemical understanding of structure-function relationships at the base of the structure prediction problem, but would in the future allow to tackle the inverse problem: Given a desired protein fold, which sequence of amino acids would most likely be able to adopt it? Closer to the current possibilities is the question of structural flexibility of the protein chain and the prediction of mutations occurring in the sequence. A good loop modeling algorithm is capable of predicting short protein segments and therefore central to answering these problems. Developing it was one of the main goals of this thesis.

The second, and not less important, aim of this thesis was to automate the process of predicting protein structures. With the increasing number of completed genomes the gap between known sequences and structures is widening. In order to manage this flood of new sequences it is necessary to implement methods capable to process them and produce good structural models in little time. This is a central question to structure-based approaches to functional genomics [299], i.e. the understanding of protein function from sequence. The best way to produce reliable structural models is to extract the knowledge available from databases. Homology modeling is the method of choice whenever a sufficient sequence similarity is encountered. Only if this is not the case

more complex methods are necessary. It has been estimated that 30-40% of all protein structures can be built by homology modeling methods. Automating the process will take the interpretation of genomic data a large step forward towards the elucidation of the functions of proteins in cells.

A more practical aim of the work in this thesis was to measure the performance of the implemented methods in a worldwide structure prediction "competition", called CASP. This allows to establish the state of the art and determine the quality of one's work. Since the field of protein structure prediction is rapidly expanding, it is no longer feasible or even desirable to invent new methods for each subproblem. The most promising strategy consists in implementing state of the art methods for most aspects and focusing on some subproblems where new methods are expected to perform better than the existing ones. For the present thesis, this philosophy consisted in assembling knowledge-based prediction method from state of the art components and focusing on loop modeling as the most innovative part of the work. As we shall see, the results support this kind of approach.

1.3 Outline

This thesis is divided into three parts. The first part, chapters 2 through 5, covers the basic concepts pertinent to protein structure prediction. Starting with the description of proteins, the reader will find an overview of experimental methods, a description of recurrent concepts and how they are employed in computational methods.

The second part describes in detail the problems encountered in modeling the structure of proteins. Starting with a description of the state of the art (Chapter 6), it continues elucidating the steps required to assemble the full model (chapters 7 to 9). Finally, it describes the implementation of the homology modeling server (Chapter 10) and presents and discusses the results achieved during this thesis (Chapter 11).

The third part covers the most innovative aspect of the present work. As will be seen, loop modeling may be the least well-understood problem in modeling protein structures. After describing the state of the art (Chapter 12), a new algorithm and its implementation (Chapters 13 and 14) will be presented. This algorithm will be compared to existing methods and the results will be discussed in Chapter 15.

Some conclusions from this work are drawn in Chapter 16, where an outlook of future research opportunities will also be given. The summary and glossary form the last two chapters. An appendix comprising the numerical evaluation of the models presented in the CASP4 contest is given as well.

Part I

Protein Structure & Prediction

2
Proteins

Proteins are regular, linear polymers composed of amino acids. They share a set of precise rules in their composition. As will be described later in this chapter the sequence of amino acids is usually sufficient to produce a well-defined three-dimensional structure. This, in turn, serves to perform the most diverse functions in living organisms.

Protein structures are defined in the following manner: The primary structure is the amino acid sequence forming the polypeptide chain. Local structural patterns define the secondary structure. The 3D conformation of the protein is the tertiary structure, whereas the aggregation and complex formation of different proteins defines the quaternary structure.

2.1 Amino Acids

Twenty amino acids form the basis for every natural protein. An amino acid, also called residue in proteins, is composed of a carboxyl group, an amino group and a side chain. The geometry of amino acids has been studied extensively on small peptides. Bond lengths and bond angles between atoms are fixed, except for small variations. Figure 2.1 shows this standard geometry.

The carbon atom carrying the side chain is usually referred to as C_α. The atoms of the side chain are commonly designated as β, γ, δ, ϵ and ζ in order away from the α carbon atom. The amino acids are linked in proteins by the formation of peptide bonds between amino and carboxyl groups of two adjacent residues. The polypeptide chain forming the protein is also referred to as backbone.

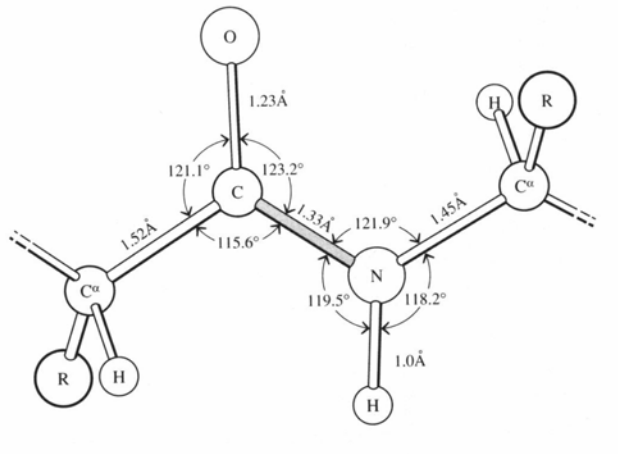

FIGURE 2.1. The geometry of an amino acid.

The peptide bond is forced to remain planar, due to its partial double bond character. The corresponding torsion angle ω is usually found in the trans-configuration (180°). The cis-configuration (0°) is only rarely found in proline residues. The two remaining torsion angles are ϕ (between N and C_α atoms) and ψ (between C_α and C atoms) as shown in Figure 2.2. These form the only free parameters for a protein to fold into a specific structure.

The twenty amino acids differ only in the side chain atoms. In addition to having full names, two abbreviations are commonly used: the one-letter and three-letter code. These are shown with some additional characteristics in Figure 2.3. The different side chains are depicted in Figure 2.4.

The chemical properties of amino acids differ considerably. Some are hydrophobic (leucine, isoleucine, tryptophan, phenyl alanine) or aromatic (tyrosine, phenyl alanine), others are acidic (glutamic acid, aspartic acid), basic (arginine, lysine) or alcoholic (serine, threonine). Some have unique properties. Two cysteine residues can form a covalent bond between their S_γ atoms, called disulfide bridge. Histidine can function both as hydrogen donor and acceptor, the chemical equivalent of being ambidextrous. Due to its side chain being connected to the backbone N atom, proline is indeed an imino acid. Its structure increases the rigidity of the backbone. Glycine, on the contrary, lacking a side chain, serves to increase the flexibility of the backbone and is frequently found in loop regions.

The properties of amino acids can be visualized using a Venn diagram as shown in Figure 2.5. The chemical diversity among the amino acids is probably the reason why proteins can fulfill so many different functions in biological organisms.

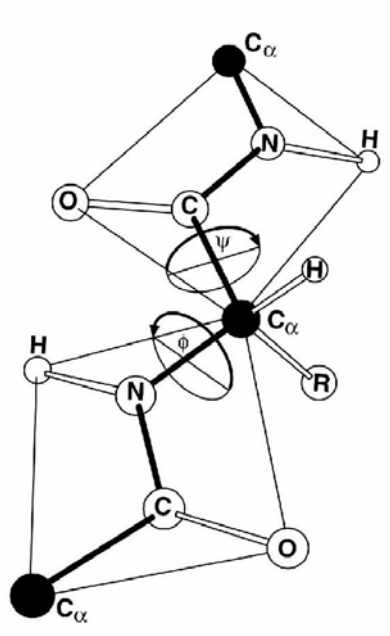

FIGURE 2.2. A schematic definition of the φ and ψ torsion angles.

Amino acid or residue thereof	Three-letter symbol	One letter symbol	Mnemonic help for one-letter symbol	Relative abundance in *E. coli* proteins (19) (%)	M.W. of residue at pH7.0 (daltons)	pK value of side chain (19)	ΔG values for transfer of side chain from water to ethanol at 25°C (16) (kcal/mol)
Alanine	Ala	A	Alanine	13.0	71		−0.5
Glutamate	Glu	E	gluEtamic acid	10.8	128	4.3	
Glutamine	Gln	Q	Q-tamine		128		
Aspartate	Asp	D	asparDic acid	9.9	114	3.9	
Asparagine	Asn	N	asparagiNe		114		
Leucine	Leu	L	Leucine	7.8	113		−1.8
Glycine	Gly	G	Glycine	7.8	57		
Lysine	Lys	K	before L	7.0	129	10.5	
Serine	Ser	S	Serine	6.0	87		+0.3
Valine	Val	V	Valine	6.0	99		−1.5
Arginine	Arg	R	aRginine	5.3	157	12.5	
Threonine	Thr	T	Threonine	4.6	101		−0.4
Proline	Pro	P	Proline	4.6	97		
Isoleucine	Ile	I	Isoleucine	4.4	113		
Methionine	Met	M	Methionine	3.8	131		−1.3
Phenylalanine	Phe	F	Fenylalanine	3.3	147		−2.5
Tyrosine	Tyr	Y	tYrosine	2.2	163	10.1	−2.3
Cysteine	Cys	C	Cysteine	1.8	103		
Tryptophan	Trp	W	tWo rings	1.0	186		−3.4
Histidine	His	H	Histidine	0.7	137	6.0	−0.5
					Weighted mean 108.7		

FIGURE 2.3. Some characteristics of the 20 natural amino acids.

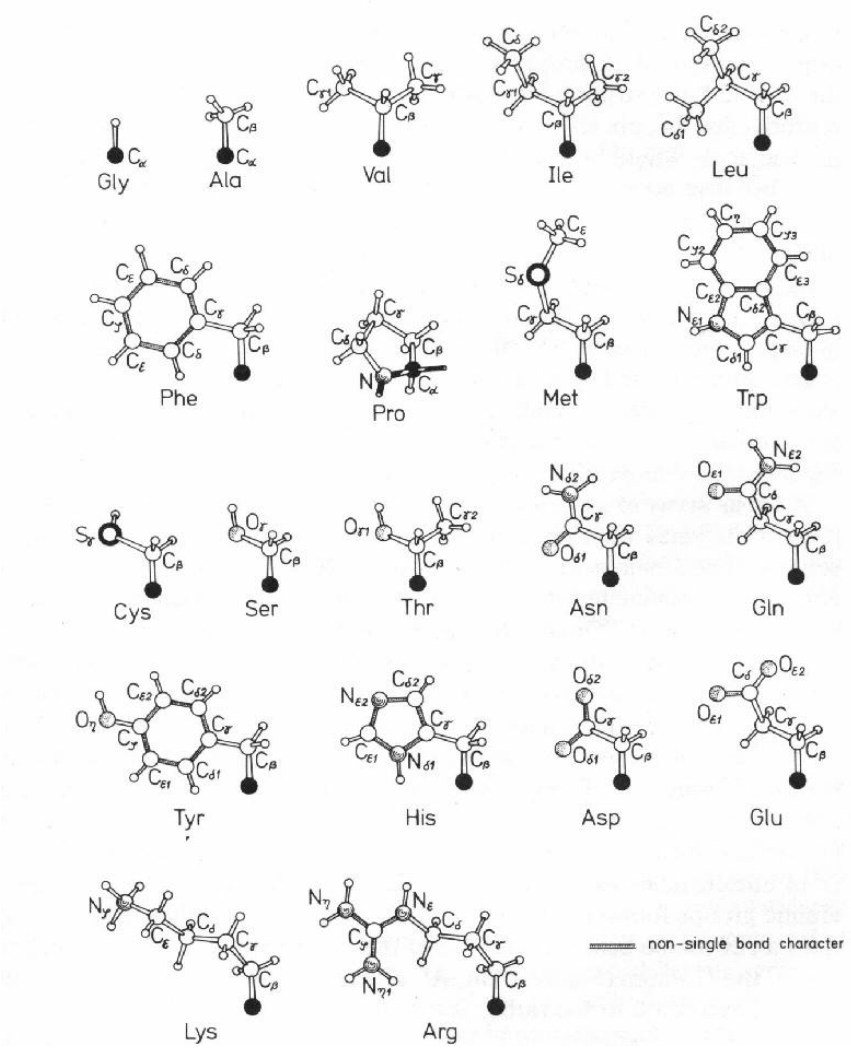

FIGURE 2.4. Side chain structure of the 20 natural amino acids.

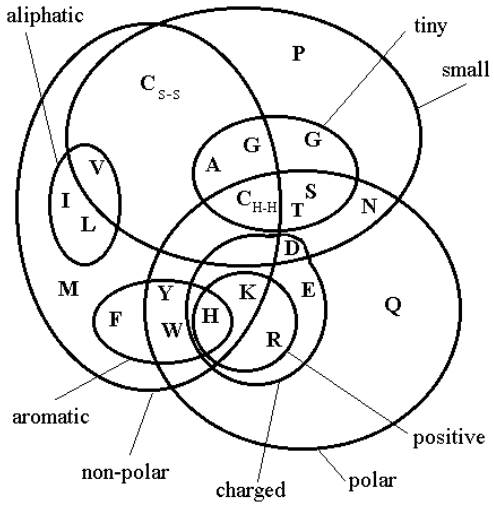

FIGURE 2.5. A Venn diagram of the properties of amino acids.

2.2 Secondary Structure

There is a distinct correlation between the potential ϕ and ψ torsion angle combinations. This has been first studied by Ramachandran et al. [73], who plotted a two-dimensional diagram of the two torsion angles. A sample Ramachandran map is shown in Figure 2.6. Due to steric hindrance of the protein backbone, only certain regions of the map are allowed. Since side chains may also create steric clashes the plot can be further subdivided based on the amino acid type. Typically three different classes are examined.

Glycine residues, lacking a C_β atom, are much more flexible and can therefore cover a wider area of the Ramachandran map. Proline, having its side chain connected to the N atom, has fewer allowed torsion angles. The remaining 18 amino acids are usually placed in the same map as the differences in backbone conformation are relatively small. Figure 2.6 shows the most common conformations in a typical Ramachandran map.

The two most populated areas of the map, around $(-62°, -41°)$ and $(-120°, +120°)$, correspond to the two most frequent secondary structure elements α-helix and β-sheet. These areas of the Ramachandran map are favored because the side chains are relatively freely orientable and the backbone can form hydrogen bonds.

The most abundant secondary structure element is the α-helix. It is very stable and also the most easily recognizable regular structure, shown in Figure 2.7. It corresponds to the area around $(-62°, -41°)$. At these torsion angles, the peptide nitrogen of the ith residue forms a hydrogen bond with the car-

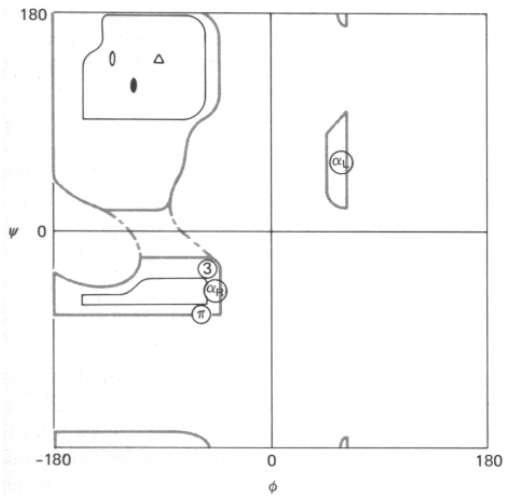

FIGURE 2.6. Ramachandran map description of the φ/ψ angle combinations for secondary structure elements. The α-helix is represented by α_R. Variations of the helix are the 3 and π symbols. The hypothetical left-handed helix is represented by α_L. The β-strand is represented by the filled and open circles and triangle.

bonyl oxygen of the $i + 4$th residue. This rigid rod structure is very compact and makes favorable van der Waals interactions. All hydrogen bonds and peptide groups point in the same direction, giving the structure a cumulative dipole moment.

The side chains project outward into the solution. Many α-helices are amphipatic, having predominantly nonpolar side chains along one side and polar residues along the remaining surface. Such helices often aggregate with each other to form tertiary structures.

The second frequently encountered secondary structure is the β-sheet. It is composed of single polypeptide fragments in extended conformation, called β-strands, with torsion angles around $(-120°, +120°)$. This is shown in Figure 2.8. Every residue of two β-strands forms a hydrogen bond between the peptide nitrogen of the first and the carbonyl oxygen of the second strand. More β-strands can be assembled to form larger β-sheets. Depending on the relative orientation of the strands, β-sheets can be either parallel or anti-parallel. Parallel β-sheets have a slight right-handed twist due to the somewhat distorted hydrogen bond geometry. Anti-parallel β-sheets in contrast are more planar, due to the perpendicular hydrogen bond geometry. There are also mixed forms of β-sheets where parallel and anti-parallel strands are in the same sheet.

The side chains of adjacent residues in a strand protrude from different sides and do not interact with each other. Instead they interact with side chains of neighboring strands. β-sheets can also have a very hydrophobic surface on one

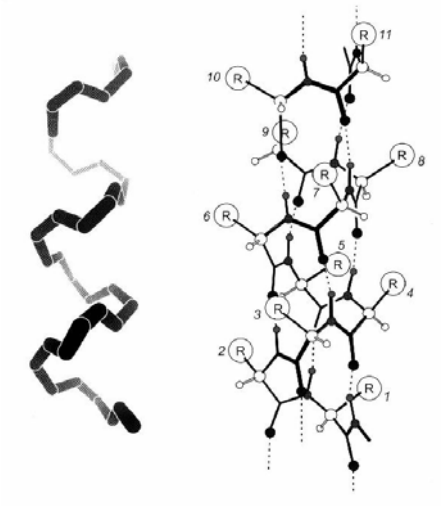

FIGURE 2.7. The structure of the α-helix.

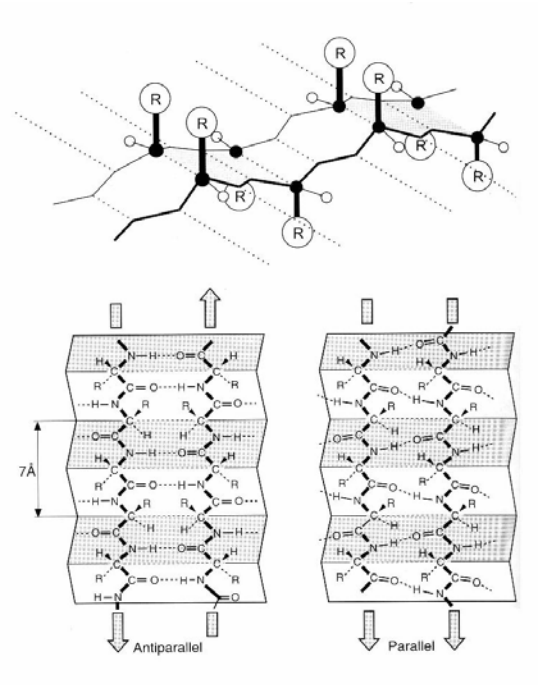

FIGURE 2.8. The structure of the β-sheet.

side and a polar surface on the opposite side. This facilitates the formation of a hydrophobic core in the tertiary structure.

Other regular secondary structure elements exist, but are usually quite rare and will not be covered here. See Schulz and Schirmer [1] for more details concerning them. Parts of the protein backbone which do not form regular secondary structure are referred to as coil, random coil or loop. Strictly speaking the latter is not totally equivalent to the former two, but the terms are nevertheless used interchangeably.

2.3 Tertiary & Quaternary Structure

Proteins tend to organize their secondary structure segments in motifs. E.g. the β-α-β motif, where the two β-strands are parallel and the α-helix serves to bridge the space between both segments. Several motifs typically combine to form domains.

A domain is a part of the protein that forms a compact, independently folded unit. Frequently, it has an autonomous biochemical function. It can be collocated somewhere between the secondary and quaternary structure, as shown in Figure 2.9. Large proteins with more than 300 residues usually contain several domains. For the study of 3D structures the domain concept is advantageous.

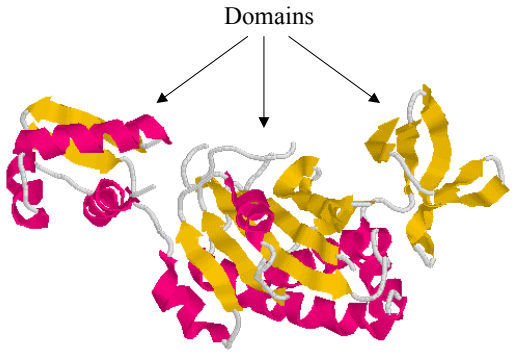

FIGURE 2.9. Sample Domain Structure. The domains are easily identified as the compact subunits of this protein.

Experimental methods are generally more successful at determining the structure of single domains, because these are easier to crystallize (X-ray) and smaller (NMR). Since most computational methods rely on database information, single domains are usually predicted. Assembling domains into the full protein structure can be considered one form of quaternary structure. The

assembly of multiple proteins into supramolecular machines as quaternary structure will not be dealt with in the present thesis.

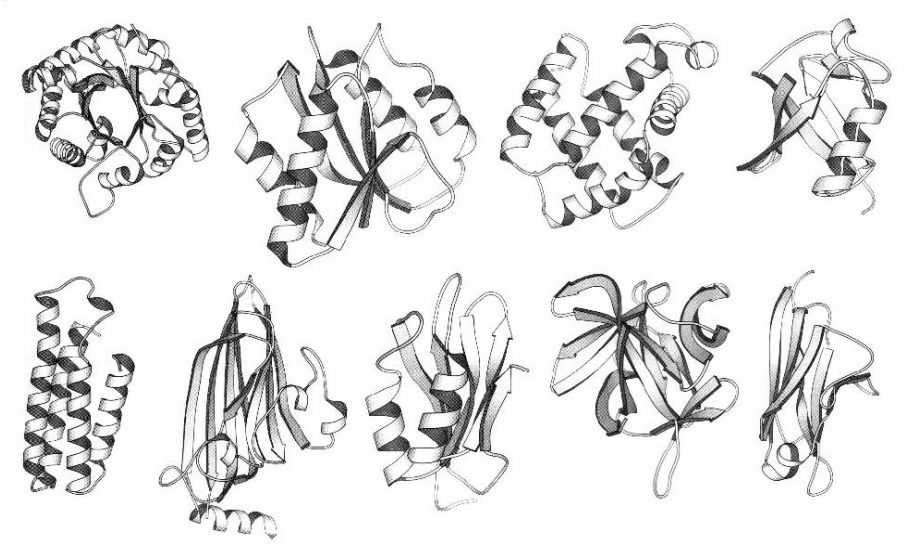

FIGURE 2.10. About 30% of the known protein structures belong to one of these 9 fold classes.

The number of naturally occurring protein folds appears to be limited. It has been found by Chothia and Lesk [61] that protein sequence evolves much faster than structure, giving rise to a large number of proteins sharing similar structures. Whether structural similarity among sequences with low similarity is simply due to divergent evolution is still open to debate. It has also been hypothesized that evolution would select the most stable protein folds for unrelated proteins, leading to convergent evolution. The concept of evolution and the limitedness of fold space will be addressed in more detail in Section 4.2. In any case, the total number of protein folds has been estimated to be about 1,000 [135]. The most representative protein folds are shown in Figure 2.10.

2.4 Folding

Under physiological conditions a protein always folds into the same 3D structure, called native structure. Anfinsen and co-workers [228] first demonstrated that it is possible to denaturate a protein in solution and then have it re-fold into its native structure. The structure is therefore entirely determined by the sequence. But how does the protein recognize how to fold into its native structure?

Two opposite ideas exist of what the native structure really is. In the thermodynamic hypothesis the native structure is simply the global energetic minimum of all conformational states of the polypeptide chain. This would imply that the native structure does not depend on denaturing conditions. The kinetic hypothesis states that the native structure is rather to be seen as the energetic minimum attainable under given circumstances (e.g. limited time frame). This would imply that the native structure may not represent the global optimum, but merely a meta stable structure which could suffer alterations. Both hypotheses are still being debated.

Levinthal has stated the following paradox [330]: Let us assume that a 100 residue protein can assume 10 conformations per residue. Let us further assume that it takes 10^{-13} seconds per transition. It would then take the protein 10^{87}seconds, or 10^{79} years, to visit all conformations. This is in sharp contrast to observed folding times of 10^{-3} to 10^{1} seconds. Levinthal concluded that proteins must fold by specific "folding pathways", which directly lead from a starting conformation to the native state, without the need to visit all conformations.

Different models for protein folding, using both the thermodynamic and kinetic hypotheses, have been proposed. In the framework model, the secondary structure elements are formed first. These find then together to form the correct 3D structure. Figure 2.11 shows different beta sheet formation paths. In the diffusion-collision model, "microdomains" are formed from local interactions, which arrange themselves into the 3D structure.

The "new view" of protein folding states that an unfolded polypeptide chain rapidly collapses, due to hydrophobic interactions, into a "molten globule". This intermediate state allows a reduced set of states in which the secondary structure elements are formed. In contrast to previous theories the protein folds along "funnels", which allow an ensemble of possible paths. From this intermediate "molten globule", the protein is then able to find its native structure. [326][327]

The "new view" is not without scientific dispute, as it depends on the underlying model [328]. However it allows to describe some interesting phenomena, e.g. the folding of the lysozyme. Experimental data suggest that lysozyme has a fast-folding population and a "fast α-domain, then slow β-domain" folding population. An idealized energy funnel for this is shown in Figure 2.12. Depending on the starting conformation it is possible to explain the different folding rates with the "new view" [327]. The "new view" is therefore the best currently available method to describe the reality of protein folding.

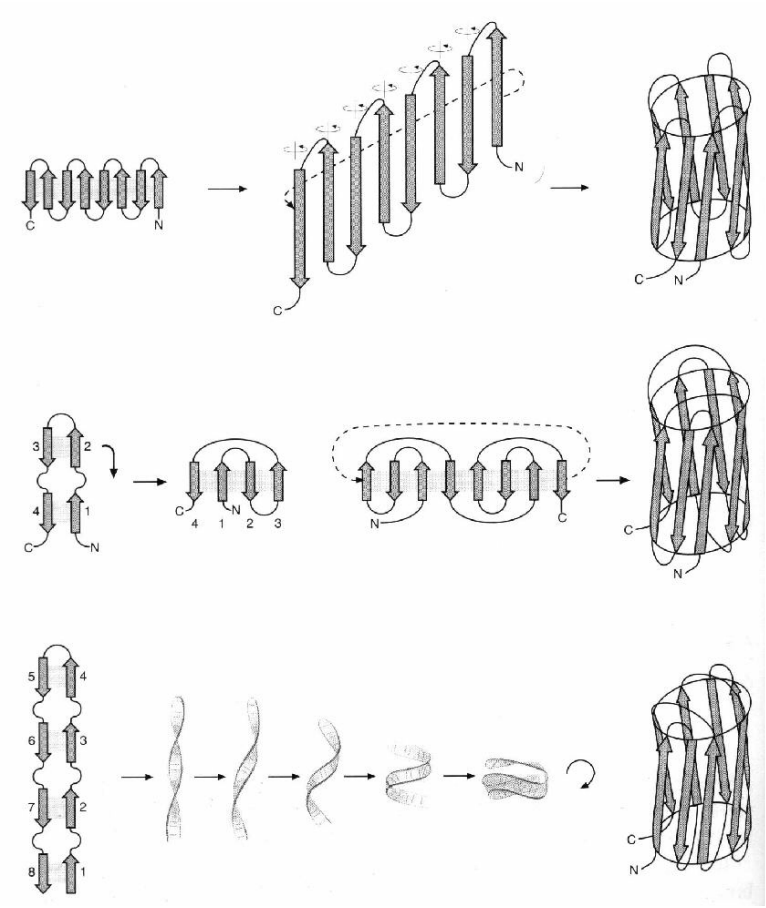

FIGURE 2.11. Different β barrel formation paths form unique structural patterns: up-and-down barrel (*top*), greek-key barrel (*middle*), jelly-roll (*bottom*).

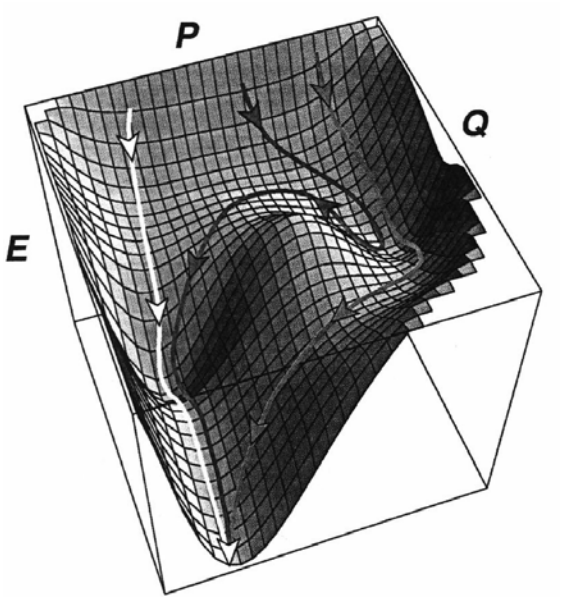

FIGURE 2.12. Funnels describing the folding of lysozyme. Two alternative folding patterns are suggested by experimental data.

3
Experimental Methods

In order to understand the importance of protein structure prediction it is first necessary to describe the experimental methods used to measure protein structures. These methods and their main bottlenecks will be described in the following. The depository of all publicly available experimentally solved structures, the PDB, will also be introduced.

3.1 X-ray Crystallography

X-ray crystallography is the most frequently used experimental method to determine protein structures. It is also the most accurate, being able to determine structures at a resolution of less than 2 Å[1].

The first step in X-ray crystallography is the crystallization of the protein. A large, individual and well-ordered crystal is required. The production of the crystal consumes the most time of the process, and is not always successful. Little is known about the mechanism of protein crystallization. Some general techniques facilitating the process are known, but are not guaranteed to work in any particular case. A strictly empirical approach is generally taken, searching as systematically as possible the many parameters (e.g. pH value of solvent, concentration of additional substances) affecting crystal formation. Crystallization is still considered something of an art and may take anything from weeks to years (or decades) for a particular protein.

Protein crystals irradiated with monochromatic x-rays produce a diffraction pattern. X-rays have a wave-like character and behave analogously to visible

[1] An Ångström (Å) is defined as 10^{-10} m. A typical bond length ranges between 1.1 and 1.5 Å.

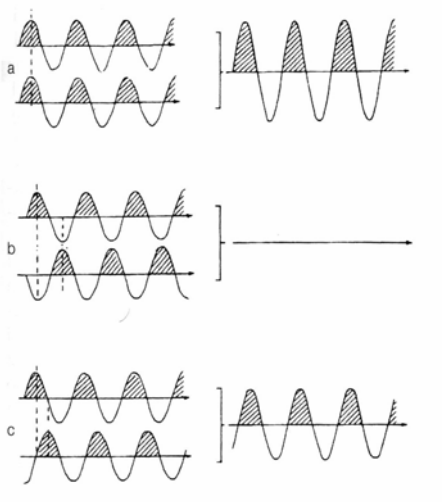

FIGURE 3.1. X-rays interfere with each other. Depending on the phase, the result can either (a) double in intensity, (b) cancel out or (c) retain the same amplitude.

light on a larger scale, producing interference when scattered at the crystal. This is shown in Figure 3.1. X-rays irradiating a crystal from a single direction are diffracted, which results in an interference pattern. The directions of these scattered x-rays, designated reflections, depend only on the crystal lattice and not on the structure of the molecule. Using a computer-guided diffractometer it is possible to collect reflections from all planes in the crystal.

The intensity of the collected reflections correspond to the amplitudes of the molecular shape in Fourier space. Using a Fourier transform it is possible to reconstruct the structure of the protein from the amplitude of the reflections plus the phase information. Unfortunately the diffractometer only detects the amplitude of the reflections. The phase information is lost. This is the phase problem of X-ray crystallography. It is overcome for example by isomorphous replacement, that is heavy atoms that strongly diffract X-rays are bound to the structure. From the difference in the measured intensities with and without the additional heavy atoms it is possible to approximately reconstruct the phase information.

A successful Fourier transform contains an image of the protein crystal in form of an electron density map. This map has to be interpreted in order to place the atoms of the protein structure. If the resolution is good enough, the peaks of the electron density map correspond to the atomic nuclei. Unfortunately this is usually not the case for proteins. The atoms have to be fitted to the electron density map using information about standard geometries for the protein backbone. This produces an approximation of the structure which may take several iterations of refinement to complete. The whole process is summarized in Figure 3.2.

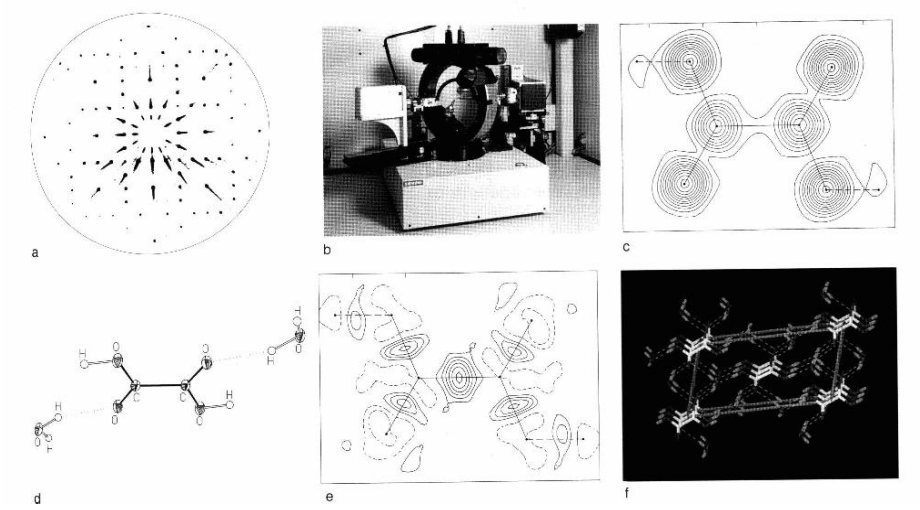

FIGURE 3.2. Various steps of X-ray crystallography. (*a*) X-ray reflection pattern; (*b*) diffractometer; (*c*) electron density map of a sample; (*d*) real structure of the same molecule; (*e*) high-resolution crystals allow the determination of electron densities between atoms; (*f*) molecular packing in the crystal lattice.

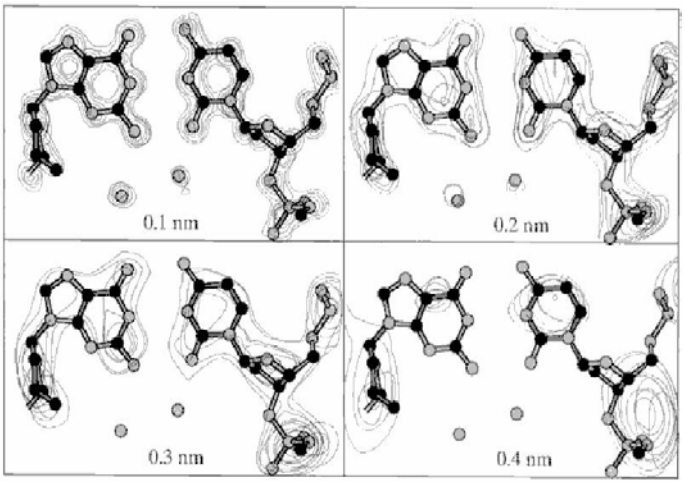

FIGURE 3.3. Influence of X-ray resolution on quality of structure. The electron density map (contour diagram) of the same structure is shown with an X-ray resolution of 1, 2, 3 and 4 Å. Atoms start to get missed between 2 and 3 Å resolution.

The resolution at which the protein was solved is a very important measure of its quality. The lower it is, the more errors the structure will contain. In general it can be said that over 3-4 Å the position of the backbone can only be guessed and side chains are not correctly reproduced. Indeed, crystallographers nowadays refuse to submit structures with less than 3 Å resolution. Only at resolution of at least 2.5 Å are the flexible parts of the protein reasonably well-defined. This difference in quality, shown in Figure 3.3, is important to keep in mind whenever looking at a particular structure.

3.2 NMR Spectroscopy

An alternative method for determining the general topology of a protein in solution is nuclear magnetic resonance (NMR). The structure obtained in this way is not as accurate as that obtained by crystallography, but it has the advantage of being in solution, which is the natural environment for proteins in contrast to crystals.

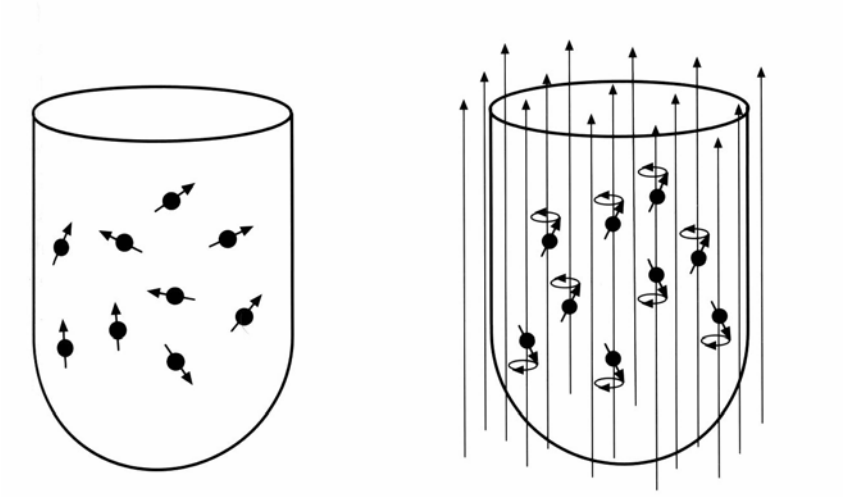

FIGURE 3.4. The atomic spin is forced to change when a strong external electromagnetic field is applied.

Many atomic nuclei possess an angular momentum, called spin. For biological systems this is the case for ^{1}H. The isotopes ^{13}C, ^{15}N and ^{31}P also share this characteristic, but are present only in low natural abundance. When a strong external magnetic field is applied, their spins are oriented in a discrete manner. The above mentioned atoms have two possible states, "up" (i.e. parallel to the external magnetic field) and "down" (i.e. antiparallel), with a slight energetic preference for the former. This is shown in Figure 3.4. Using an

additional electromagnetic field it is possible to influence the nuclei, turning additional spin states from "up" to "down". This happens when the frequency of the field matches the energy difference between the spin states, the resonance frequency. After a relaxation time the spins return to their original state. The resonance frequency differs for atoms depending on their chemical environment, giving rise to the chemical shift.

In order to produce a NMR spectrum, it is necessary to subject the solution containing the protein to a strong magnetic field. With an additional modulated electromagnetic field it is possible to record the resonance frequencies. Multi-dimensional techniques selectively use sequences of electromagnetic impulses to separate the information on interactions between atoms. It is nowadays possible to interpret the spectra of proteins up to 300 amino acids in length.

Taking advantage of the nuclear Overhauser effect (NOE), resulting from dipole interactions between nuclear spins, distances between atoms that are close in space but not covalently bound can be measured. Given a sufficient number of such distance constraints, these can be used to perform distance geometry calculations to define the spatial arrangement of the polypeptide chain. This is shown in Figure 3.5. For complex proteins it can be difficult to find solutions fulfilling all distance constraints. Model construction is therefore usually coupled to molecular dynamics simulations, which produce energetically favorable structures. Segments of the structure with few distance constraints give rise to several slightly different structures. This, together with the more flexible character of proteins in solution, is the reason why NMR spectroscopists generally submit up to 20 or more models to the protein data bank.

3.3 Protein Data Bank (*PDB*)

All structures which have been solved experimentally and are available to the public are deposited in the so-called Protein Data Bank (*PDB*). [47] At present 13,960 structures are available (July 2001). The full statistics are given in Table 3.1. A large portion of proteins in the *PDB* have been solved with different resolution, with or without specific ligands and cofactors, etc. It has been estimated that almost three quarters of the structures deposited are very similar to each other [135], reducing the number of "unique" structures to less than 4,000.

In recent years, the *PDB*, like all biological databases, has seen an exponential growth, as shown in Figure 3.6. For comparison, Swiss-Prot, the manually edited protein sequence database [132], has grown from 87,397 entries (July 2000) to 99,134 entries (July 2001). A total of 471,191 sequences are

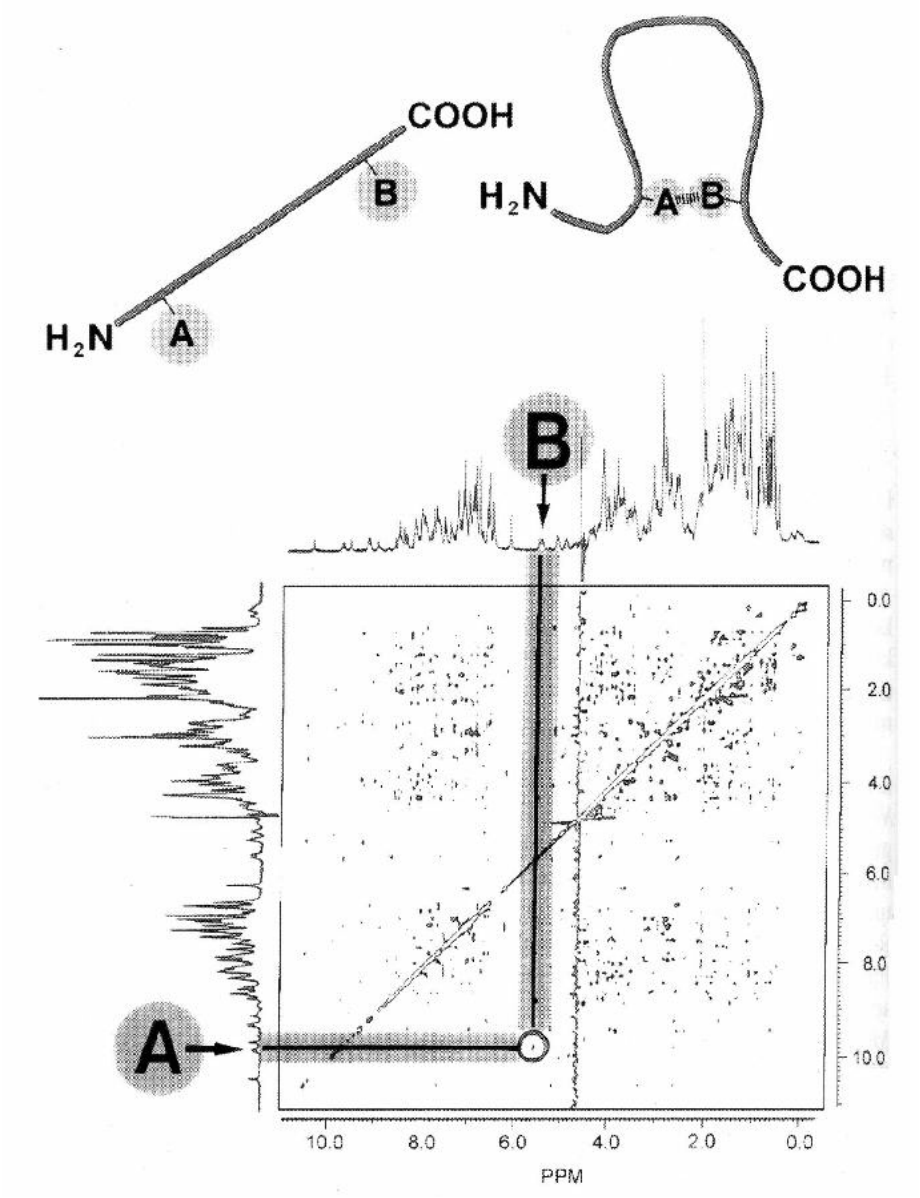

FIGURE 3.5. Distance constraints derived from NMR spectra. The two atoms A and B are separated along the protein backbone. Presence of a peak in the NMR spectra reveals their close spatial proximity in the folded protein.

Exp. Method	Proteins	Protein/Nucleic Acid	Nucleic Acid
X-ray Diffraction	11,733	562	580
NMR	1,939	73	385
Theo. Model	288	20	23
Total	13,960	655	988

TABLE 3.1. Statistics for the *PDB* as of July 2001.

deposited in the TrEMBL database [132] of translated protein coding DNA regions coding regions. A further explosion of known sequences is resulting from the genome sequencing projects being completed. Despite an increase in solved structures per year, the gap between known sequences and known structures is widening at a rapid rate.

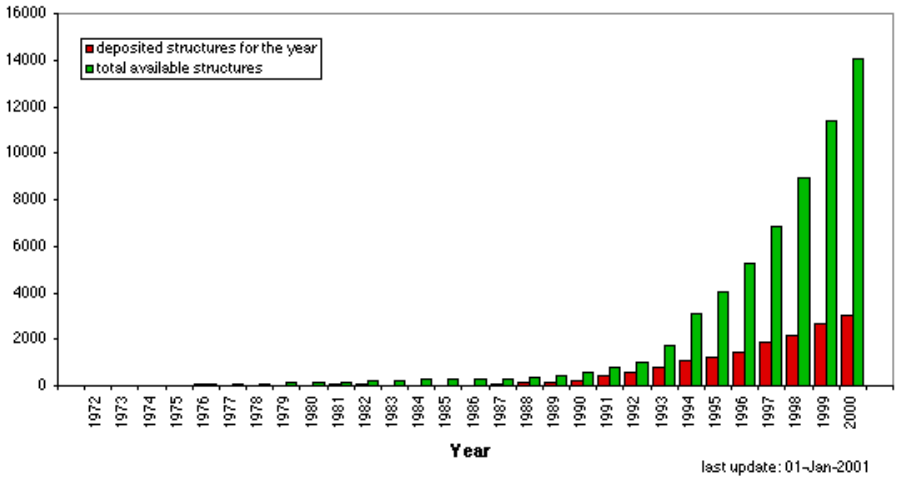

FIGURE 3.6. Growth of the PDB.

4
Structural and Sequence Similarity

This Chapter describes the recurring concepts of sequence and structural similarity. The definitions for some commonly used keywords will be used throughout the rest of this thesis. The central concept of "homology" for protein structures will be illustrated and structural classifications presented.

4.1 Alignments & Similarity Measures

A number of definitions have been used in the literature to describe similarity measures for both protein sequences and structures. Some commonly accepted definitions are reported below and aim to clarify the terminology used throughout the present thesis.

Definition 1 *An **alignment** M is a mapping between residues of two sequences. Two residues r_1 of sequence A and r_2 of sequence B are said to be **aligned** with respect to an alignment M, if the alignment of A and B maps r_1 and r_2 onto each other.*

The alignment is a central concept throughout computational biology. It is not restricted to amino acid sequences, but can be used for DNA as well. In amino acid sequences, each residue is represented with its one-letter code (refer to Figure 2.3 for the one-letter codes).

Definition 2 *Given an alignment M, a **gap** ("−") replaces amino acids of sequence A not aligned with any of the sequence B (or vice versa). Relative to sequence A, an **insertion** is a contiguous stretch of residues from A aligned with the gap character. A **deletion** is a stretch of gaps aligned with residues from B.*

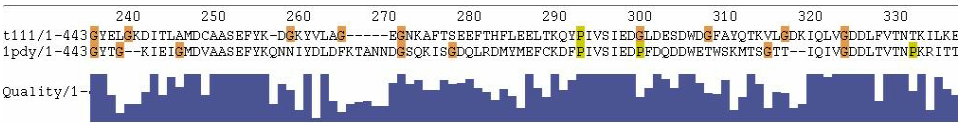

FIGURE 4.1. Sample Alignment between t0111 (*top row*) and PDB `1pdz` (*middle row*). The bottom row shows a similarity value for each aligned position.

Insertions and deletions are common in biological sequences due to the effects of evolution. Figure 4.1 shows a typical alignment. It is further possible to distinguish two alternative types of alignments: **global** and **local**.

A **global** alignment will always cover the entire input sequences, no matter how different these may be. Unrelated sequences will still be "aligned". **Local** alignments on the other hand contain only contiguous parts of the sequence that are "similar". When considering multi-domain sequences it is not uncommon to have a segment of the sequence representing a single domain aligned, with the remaining sequence missing.

Definition 3 *A **profile** is an alignment between several closely related protein sequences, usually representing a single protein family.*

Profiles are used by computer programs to improve the detection of more distantly related protein sequences. They can be used to extend pairwise sequence alignment methods to simulate multiple sequence alignments. A similarity measure for amino acid sequences can now be defined as follows:

Definition 4 *Two amino acid sequences have a **pairwise sequence identity** of x, if an alignment between the two sequences can be found, such that the number of aligned residues, which are identical, divided by the length of the shorter sequence is x. Two amino acid sequences are said to have a **similar sequence**, if they have a pairwise sequence identity equal or higher than 25%.*

The cutoff value of 25% can vary somewhat, since the "twilight zone" for comparative modelling reaches from 20% to 35%, depending on alignment length [202]. See also Chapter 6 for a detailed explanation.

Definition 5 *Let a system have N elements that can be in any of k states. z_i is the correct prediction of element z being in state i. The prediction accuracy Q_n is:*

$$Q_k = 100 * \frac{\sum_i z_i}{N} \tag{4.1}$$

Q_k can be used to measure the accuracy of multi-class predictions. Its most frequently used form is the Q_3 used for secondary structure prediction. For structures the most important measure is the RMSD.

Definition 6 *Let r_{ai} and r_{bi} be the coordinates of atom i of structure a and structure b. The root mean square deviation (**RMSD**) between the two structures is:*

$$RMSD = \sqrt{\frac{\sum (r_{ai} - r_{bi})^2}{n}} \tag{4.2}$$

The RMSD is the most common similarity measure for protein structures. Simple as it may seem, there are some difficulties in comparing results from different authors. When two entire structures are being compared, these will be optimally superimposed to yield the lowest possible RMSD. Problems arise when the RMSD calculation includes only parts of a structure. Let us consider the case of two structures that contain identical and modified parts. In this case two alternative, and equally legitimate, ways to calculate RMSD exist. The first one, sometimes called "global" RMSD, is to superimpose only the fixed part of the structures. The second possibility, "local" RMSD, is to superimpose the modified part of the structure, disregarding its orientation to the rest of the structure. The latter approach obviously yields lower RMSD values, as it does not take into consideration the position relative to the fixed structure. The results calculated with differing methods cannot be compared. An example is shown in Figure 4.2.

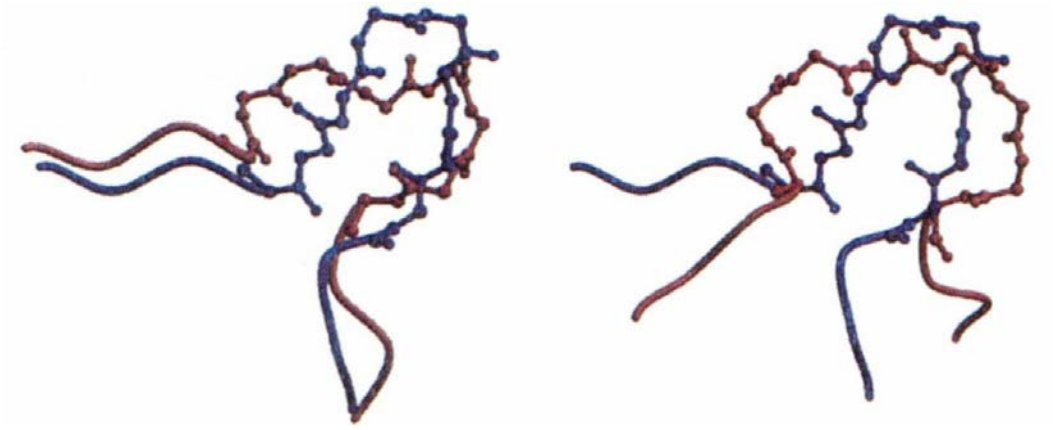

FIGURE 4.2. Comparison of global and local RMSD. Global RMSD (*left*) is 7.63 Å, whereas local RMSD (*right*) is 2.83 Å.

The other difficulty in comparing RMSD values is to know which atoms have been included in the calculation. When examining protein backbones, a number of possibilities exist: C_α only; backbone heavy atoms (N, C_α and C) with or without the carbonyl O; all backbone atoms (i.e. including hydrogens). The same distinction can be repeated with side chain atoms. As a rule, the more atoms are being included in RMSD calculations, the higher the value

will be. Again, fair comparisons are only possible when the same atoms are used in RMSD calculation.

4.2 Homology

The term "homology" is frequently used to describe relationships of genes and proteins. Two sequences are said to be homologous if a common ancestor is assumed, from which both have originated by means of divergent evolution. For proteins this means that their structures are likely to be very similar.

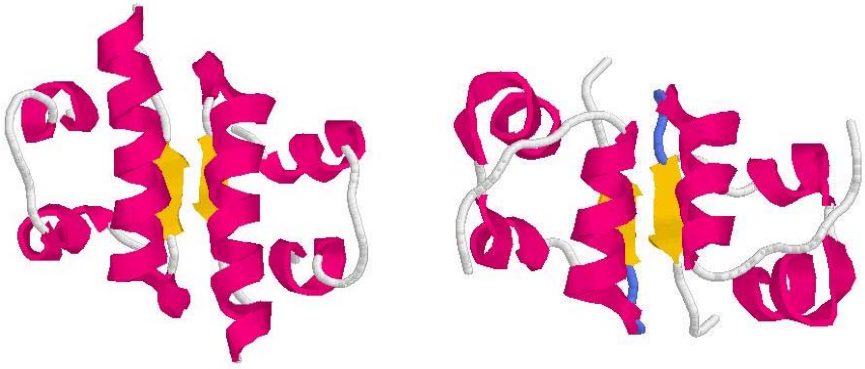

FIGURE 4.3. Divergent evolution exemplified by the structures of human (*left*) and pig insulin (*right*).

An example for this is insulin. Human and pig insulin are 91% identical in amino acid sequence. The corresponding structure fragments are shown in Figure 4.3. Two points related to protein structure prediction are worth mentioning. Two sequences with such a level of identity are mutually almost identical in structure. Some differences in structure remain, as shown in insulin, but these relate only to the local structure. Indeed, two structures that still have some differences were selected in order to make a point. (The two structures have been crystallized under different conditions)

Unfortunately, the term "homology" is not always used consistently in the literature. Statements like "protein A and B are 50% homologous" are erroneous, unless both proteins share a homologous domain (out of two). Attempts have been made to separate the term homology from sequence similarity [334].

Due to the limited number of folds apparently present in natural proteins, the term "analogy" has been reserved for apparently unrelated protein sequences sharing a similar structure. It is assumed that convergent evolution has selected some folds as being particularly stable. Proteins with different, and sometimes even opposite, functions can have a very similar structure.

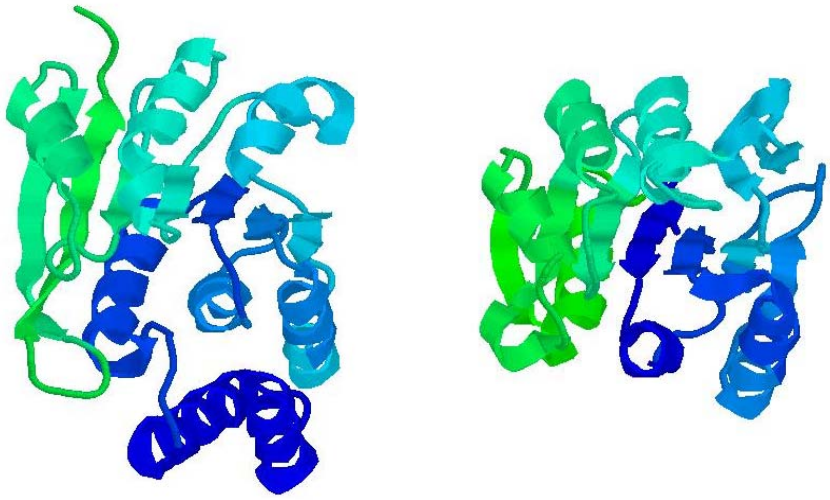

FIGURE 4.4. Convergent Evolution? Two structures with less than 20% sequence identity sharing the Rossmann fold.

An example for this is the so-called Rossmann fold. It is composed of a central β-sheet, with a particular connectivity. A number of α-helices are located above and below the β-sheet. This fold is particularly common and can have a variety of functions. Figure 4.4 shows two proteins sharing the Rossmann fold. Sequence identity is about 10%, yet the two structures can be superimposed with a RMSD of 3.0 Å for 104 out of 198 residues. Protein `1vid` is a transferase (i.e. moves a chemical group between two molecules) responsible for the inactivation of neurotransmitters in *Rattus norvegicus*. Protein `1chd` is a methylesterase (i.e. cleaves a methyl group through hydrolysis) responsible for sensory responses of the cell in *Salmonella typhimurium*.

4.3 Tertiary Structure Classification

As has been stated before, the number of natural protein folds appears to be limited to less than 1,000 structural families [135]. Structural similarity is obvious in a large number of cases, so it is important to have a simplified hierarchical way for classification of structural features. Different approaches have been reported. The three most common ones are: SCOP [134], CATH [200] and FSSP [133]. The methods differ in a number of issues. Classification ranges from almost entirely manual (SCOP) to entirely automatic (FSSP). SCOP and CATH use domains as the unit of classification, whereas FSSP uses protein chains. Since about 30% of non-identical protein structures contain two or more domains, these have a unique FSSP classifier and several SCOP and

CATH classifiers. The salient features of the three classification systems will now be addressed.

A manual classification system, Structural Classification Of Proteins (SCOP) [134], was designed by Alexey Murzin et al. in Cambridge, UK. In the SCOP database the classification is on hierarchical levels, with the principal levels as follows:

Class: Secondary structure content and/or special features

For convenience of users, all proteins have been assigned to one of several classes based on secondary structure content. The main classes represent: all alpha, all beta, alpha and beta (for proteins where α-helices and β-sheets are largely interspersed), alpha plus beta (where α-helices and β-sheets are largely segregated) and multi-domain (for unique domains from multi-domain proteins, not found in other proteins). Additional classes have been added for special features, e.g. designed proteins, theoretical models, etc.

Common Fold: Major structural similarity

If proteins have major secondary structures in the same arrangement with the same topological connections, they are defined as having a common fold whether or not they have a common evolutionary origin. In these cases, the structural similarities may have developed as a result of physical principles that favor particular packing arrangements and fold topologies.

Superfamily: Probable common evolutionary origin

Proteins with low sequence identities, whose structural and functional features suggest that a common evolutionary origin is probable, are placed together in superfamilies. For example, actin, the ATPase domain of the heat shock protein, and hexokinase form a superfamily.

Family: Clear evolutionary relationship

Proteins are clustered together into families on the basis of one of two criteria that imply a common evolutionary origin: first, all proteins that have residue identities of 30% and greater; second, proteins with lower sequence identities but whose functions and structures are very similar; for example, globins with sequence identities of 15%.

The statistics for SCOP release 1.55 (1 March 2001), including 13,220 *PDB* entries, 31,474 domains and 39 literature references (excluding nucleic acids and theoretical models) are given in Table 4.1.

Class	Folds	Superfamilies	Families
All alpha	138	224	337
All beta	93	171	276
Alpha and beta (α/β)	97	167	374
Alpha and beta ($\alpha+\beta$)	184	263	391
Multi-domain	28	28	35
Membrane and cell surface	11	17	28
Small	54	77	116
Total	605	947	1,557

TABLE 4.1. Statistics for SCOP release 1.55 (1 March 2001).

A similar approach is used by the CATH classification of C. Orengo and J. Thornton [200]. CATH stands for Class, Architecture, Topology, and Homologous superfamily, which describe the different hierarchical levels used. Four classes are defined at the top of the hierarchy, in analogy to SCOP:

1. Mainly α-helix
2. Mainly β-sheet
3. α/β proteins
4. Few secondary structures

No distinction is made between α/β and $\alpha+\beta$ structures. Each class is divided into a number of architectures describing the gross arrangement of secondary structures, independent of connectivity. The architectural groupings can sometimes be rather broad as they describe the general features of protein-fold shape. The topology level describes fold families. Structures in the same topology have a similar number and arrangement of secondary structures. The connectivity linking their secondary structure elements is also the same. A schematic representation of the first three levels is shown in Figure 4.5.

Highly similar structures and those with similar function, suggesting evolution from a common ancestor, are grouped in the same homologous superfamily. Extremely similar structures with sequence identities $> 35\%$, which may just be different examples of the same protein, are clustered at the sequence family level. Of the five hierarchical levels described above, only the architecture is assigned manually. The other four levels are assigned automatically based on structural or sequence similarity. It is possible to plot the distribution of folds among different architectures and topologies in the so-called "CATHerine wheel", illustrated in Figure 4.6. 30 out of 35 existing architectures are shown in figures 4.7 and 4.8. Current statistics (June 2001) for CATH follow:

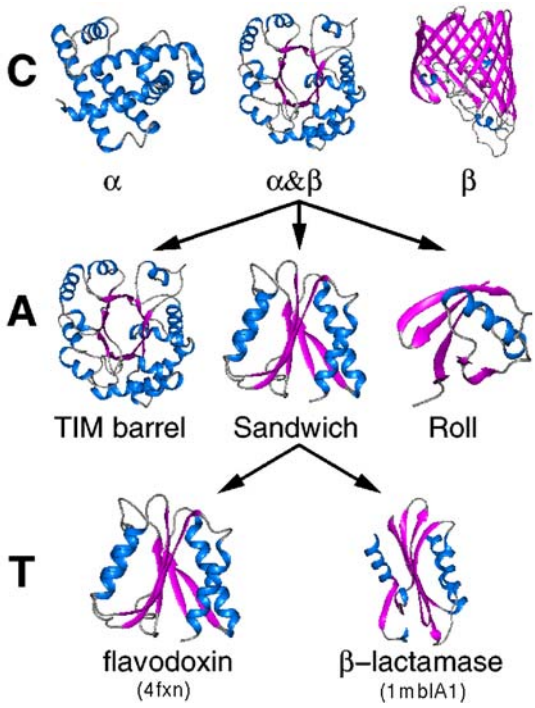

FIGURE 4.5. Schematic representation of the first three levels of CATH classifications.

Class	4	
Architecture	35	
Topology	580	
Homologous Superfamily	900	
Sequence Family	1,846	(>35% sequence identity)
Near Identical Structures	3,864	(>95% sequence identity)
Identical Structures	7,690	(100% sequence identity)

A different approach has been followed in FSSP, Families of Structurally Similar Proteins, of L. Holm and C. Sander [133]. This database of structural alignments is derived from all-against-all structural comparisons using the program DALI [246]. It defines a six level hierarchy and uses polypeptide chains rather than domains for classification. Very close homologs (> 70% sequence identity) are represented by a single structure. Medium homologs (> 30% sequence identity) are grouped in structural families. This reduced set of families is then compared on an all-against-all basis, in order to find remote homologs. In this way the computational task is reduced from comparing over 20,000 chains to about 2,200 representative chains. The main advantage of FSSP is the possibility to immediately view the structural alignments and RMSD between different chains. On the downside, being totally automated,

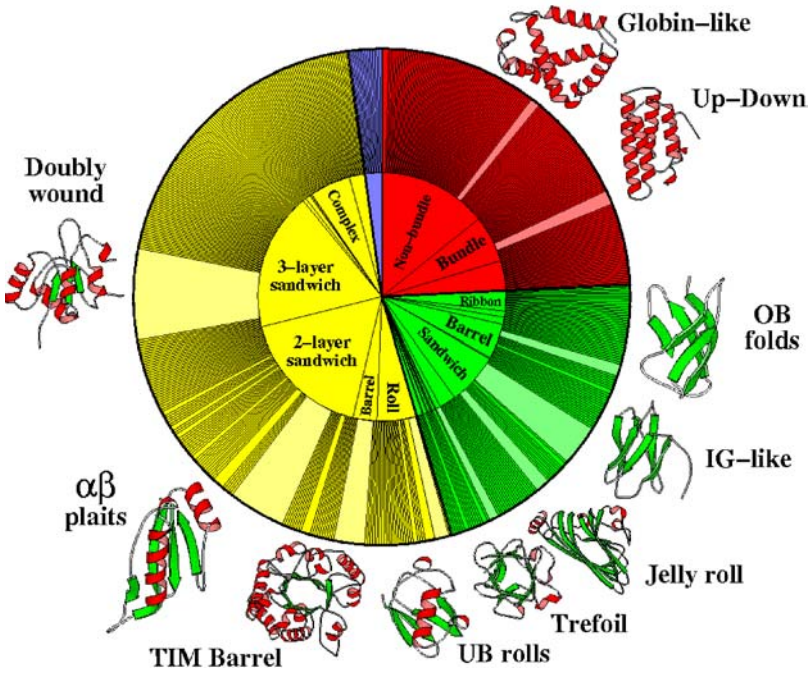

FIGURE 4.6. CATHerine wheel. Distribution of the fold classes among architectures and topologies.

it does not offer information about domains in proteins. This may sometimes produce misleading results, especially in multi-domain proteins.

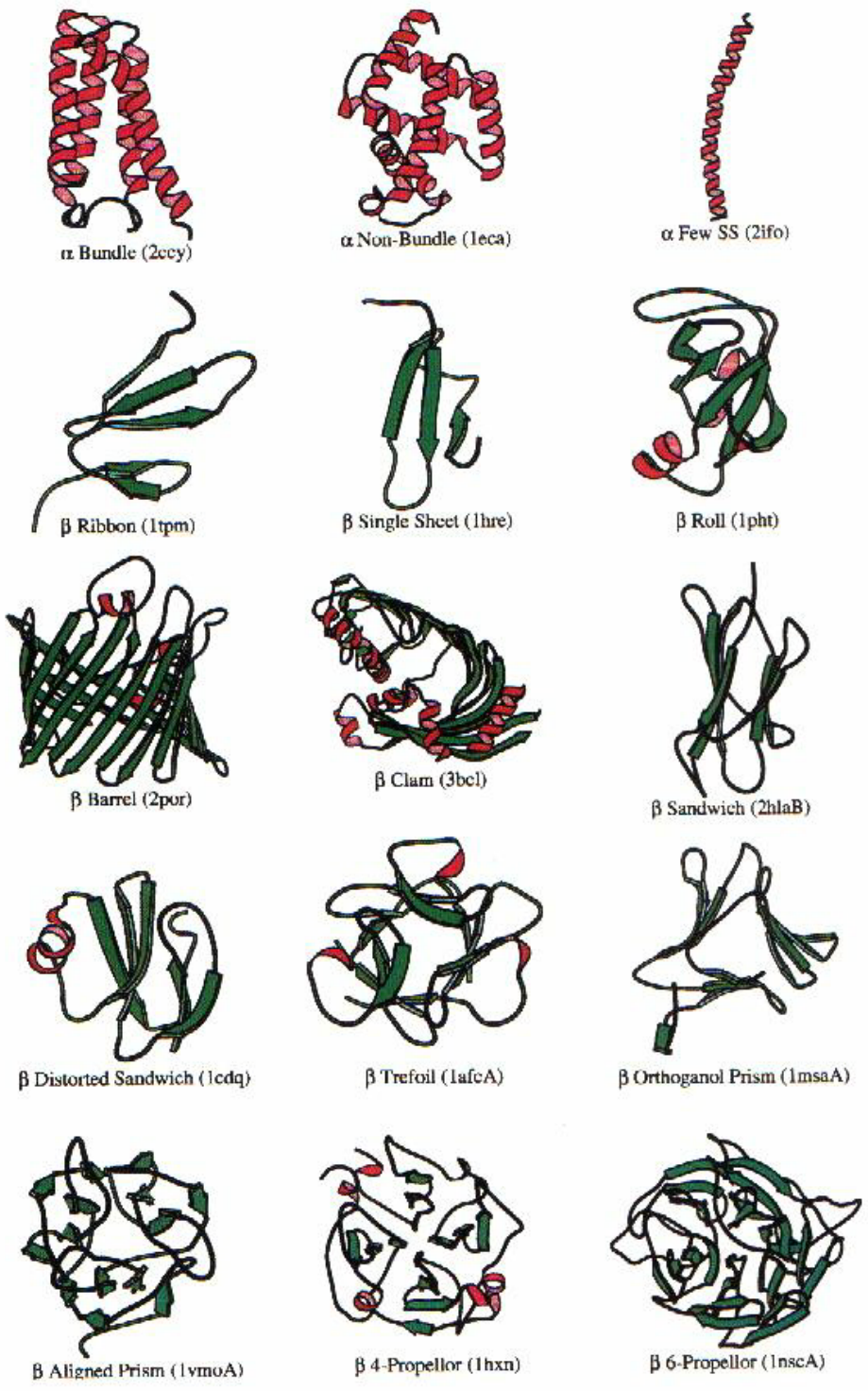

FIGURE 4.7. CATH architectures, part I.

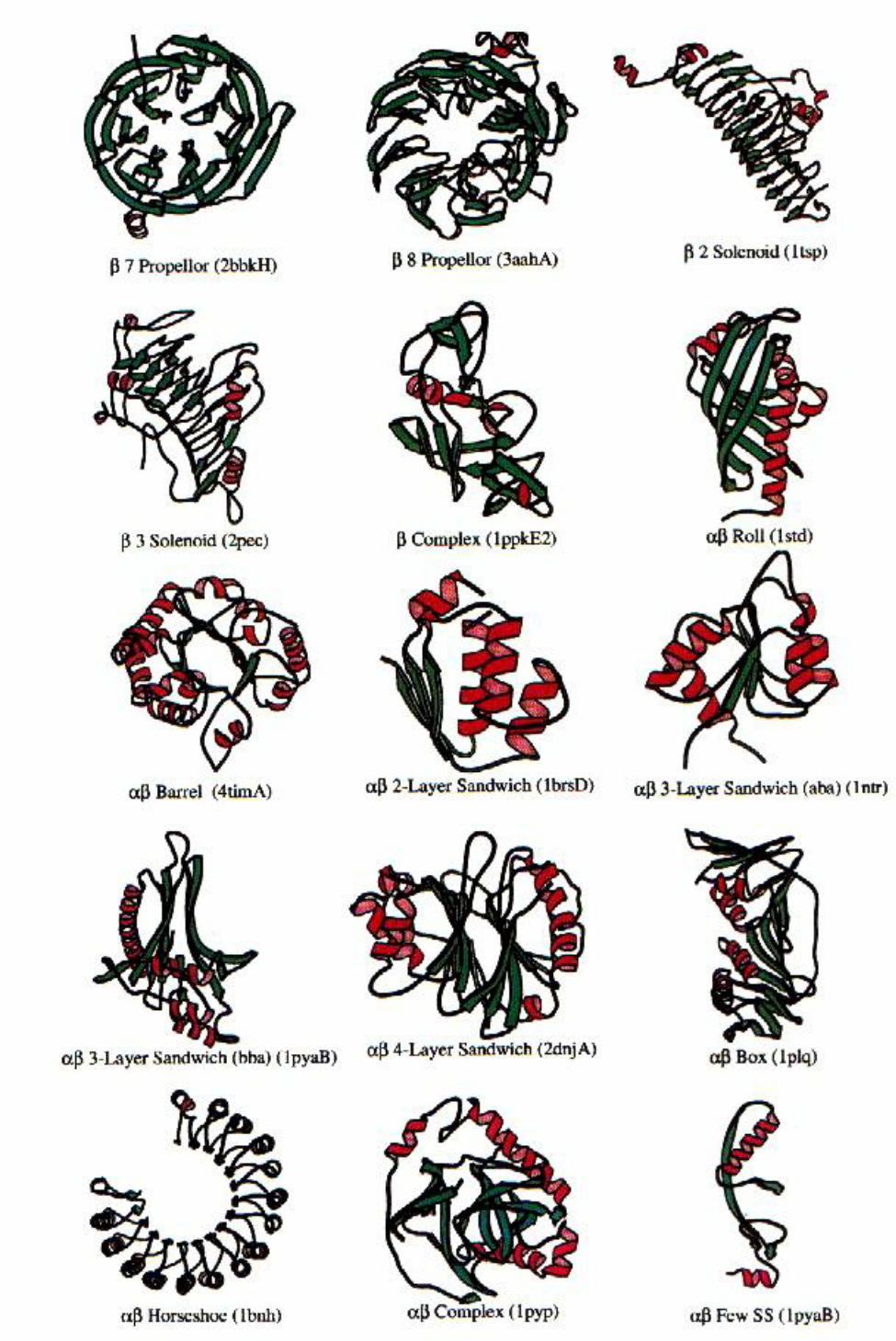

FIGURE 4.8. CATH architectures, part II.

5
Computational Methods

An overview of all computational methods used for protein structure prediction will be given in this Chapter. This will form the basis for a more thorough description of the methods related to this thesis in Part II. The limits for homology modeling will be discussed to motivate the use of more complex, and less reliable, methods. Predictions of certain features of the protein structure, e.g. secondary structure or solvent accessibility, will be introduced as these can be used to aid in tertiary structure prediction. *Ab initio* methods and common optimization methods will be finally addressed to complete the overview.

5.1 Overview of Methods

A number of different approaches for protein structure prediction have been developed over the last 30 years. These range in scope from adapting a solved structure to match the sequence of an unknown one, to the attempt to fold a protein from first principles. Success varies and different methods should be used for different proteins. Based on the degree of similarity between the unknown structure (*target*) and structures from the database (*templates*), one can broadly distinguish three approaches (also shown in Figure 5.1):

- comparative (or homology) modeling
- fold recognition
- *ab initio*

Comparative modeling builds the target structure based upon a homologous structure. Stretches of the polypeptide chain presumably differing between the

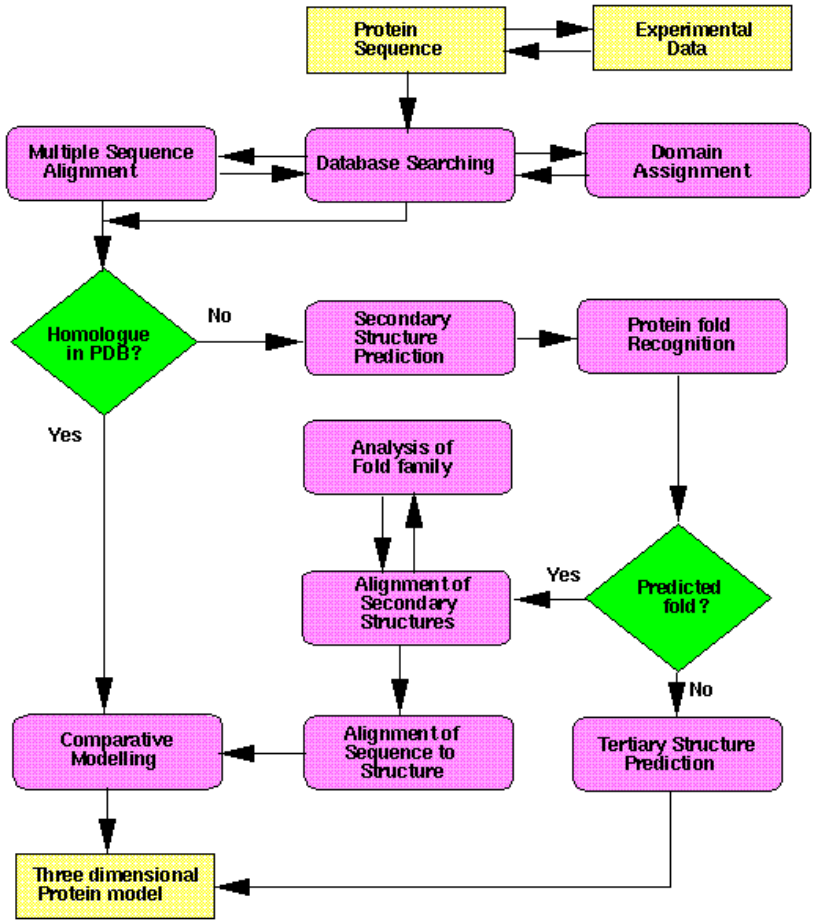

FIGURE 5.1. Overview of approaches to protein structure prediction.

two structures have to be edited, but otherwise the structure can be more or less copied. For this approach to work, there needs to be a significant sequence similarity detected between the target and one or more sequences in the database. This approach relies heavily on good sequence comparison and alignment methods. The state of the art will be described in Section 6.2. Under the circumstances described above it is the method of choice, as it will produce fewer errors than the other methods.

Fold recognition exploits knowledge from the fact that the number of naturally occurring protein folds is limited. It is therefore likely that a sequence with no significant sequence similarity may still be similar to structures in the database. Two different sub-categories exist in fold recognition.

Threading is closest to *ab initio* methods. The target sequence is placed on the 3D coordinates of protein structures in a fold library ("threaded"). The

folds with the substituted amino acid sequence are then evaluated with an energy function.

Profile or mapping methods instead try to extend the capabilities of sequence comparison algorithms to detect weak sequence homologies. In addition, and depending on the method used, information like predicted secondary structure and/or predicted surface accessibility is incorporated to improve the results. The state of the art of fold recognition will be described in Section 6.3. Once an alignment with a known structure is found, the same methods as in comparative modeling may be used to produce a model.

Ab initio methods attempt to construct a model structure based on the physico-chemical properties of the amino acid chain. No knowledge on known structures is required. Calculations are based on complex energy functions, which encapsulate the information about atomic forces. An optimization method is used to guide the process of selecting promising structures throughout construction. In addition, secondary structure predictions can be incorporated to produce better structures. Despite recent improvements, the *ab initio* methods still produce more erroneous models for all but the most difficult structures. Section 5.5 gives a more detailed overview.

5.2 Limits for Homology Modeling

In the previous section, it has been stated that homology modeling produces the most accurate results but requires a significant sequence similarity between the target and one or more sequences in the database to work. So the question is: what level of sequence similarity is "significant" enough for two sequences to share a similar structure?

The first tentative analysis was made by Chothia in 1986 [61] who analyzed 32 pairs of homologous structures and found out that they had a RMSD $\leq$ 3.0 Å for as low as 20% sequence identity. A more thorough investigation was carried out by Schneider and Sander in 1991 [182] who tried to define a threshold of sequence identity versus alignment length to discriminate between sequence pairs sharing the same structure from those with different structures. This concept was later elaborated by Rost in 1999 [202] with a much larger structural database, so we will focus on his results.

Rost selected two sets of data. The first one contained true positives, i.e. pairs of sequences sharing similar structures. Percentage of identical residues was then plotted against alignment length, as shown in Figure 5.2, A. The second data set contained false positives, i.e. pairs of sequences with dissimilar structures. Again, these were plotted as percentage of identical residues against alignment length. This is shown in Figure 5.2, B. Comparing both plots shows that for reasonably long alignments, above 50 residues, there are practically no

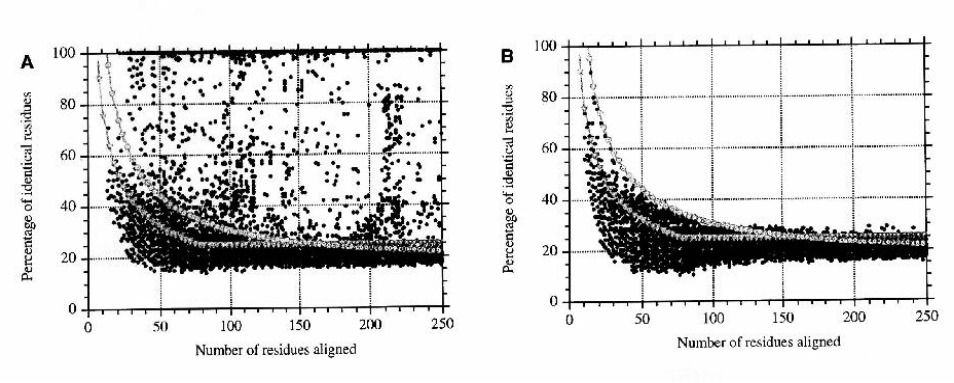

FIGURE 5.2. Pairs of sequences with similar (*left*) and dissimilar structure (*right*) elaborated by Rost. The two curves try to discriminate between the two classes.

dissimilar structures (false positives) above 40% sequence identity. At about 100 aligned residues this value drops to about 30% and converges to about 20-25% for very long alignments ($\geq$ 250 residues)[1]. Rost also derives an empirical curve to approximate the observed figures.

Below these cutoffs there is still a significant number of similar structures (true positives), but these cannot be identified among a vast number of false positives. Rost calls this the "twilight zone" of homology modeling. In theory, it is possible to model these proteins with structures from the data bank, but aligning them is not enough to discriminate true from false positives. It is in this area where fold recognition methods operate, trying to exploit other sources of information to detect remote homologues.

5.3 Secondary Structure Prediction

The single characteristic of the polypeptide chain that can today be predicted reasonably well is the secondary structure. As we have seen in Section 2.2, there are two frequently recurring secondary structure elements: α-helix and β-strand. The remainder is easily defined as loop or (random) coil. The most intuitive secondary structure classification is therefore to distinguish between three classes: H, E and C. (β-strands being termed "extended" in this context) From this classification we can derive a simple accuracy measure, the Q_3 value. This value indicates the percentage of amino acids correctly predicted as belonging into their relative state (H, E or C). A random prediction would yield a Q_3 value of 35.4% [311].

[1]This statement is true for naturally evolved proteins. Some studies suggest that it may not be the case for artificial ones.

		Blout *et al.*, 1960 (328)	Kotelchuck and Scheraga, 1968 (363)	Lewis *et al.*, 1970 (368)	Robson and Pain, 1971 (346)	Chou and Fasman, 1974 (340)	Finkelstein and Ptitsyn, 1976 (371)
A	Ala	(H)	H	I	+0.09	1.45	1.08
C	Cys	C	H	I	+0.03	0.77	0.95
D	Asp	H	C	B	−0.02	0.98	0.85
E	Glu	H	H	H	−0.12	1.53	1.15
F	Phe	(H)	H	H	+0.03	1.12	1.10
G	Gly	—	Indifferent	B	−0.05	0.53	0.55
H	His	(H)	H	I	+0.08	1.24	1.00
I	Ile	(C)	H	H	+0.07	1.00	1.05
K	Lys	(H)	C	I	−0.03	1.07	1.15
L	Leu	H	H	H	−0.11	1.34	1.25
M	Met	H	H	H	+0.10	1.20	1.15
N	Asn	(C)	C	I	−0.04	0.73	0.85
P	Pro	—	Special	B	—	0.59	—
Q	Gln	(H)	H	I	+0.07	1.17	0.95
R	Arg	(H)	H	I	+0.02	0.79	1.05
S	Ser	C	C	B	−0.07	0.79	0.75
T	Thr	(C)	H	I	−0.01	0.82	0.75
V	Val	C	H	I	+0.04	1.14	0.95
W	Trp	(H)	C	H	+0.10	1.14	1.10
Y	Tyr	(H)	C	H	−0.02	0.61	1.10

FIGURE 5.3. α-helix formation propensities.

A simple approach to secondary structure prediction is to use statistics of how often different amino acid types appear in the secondary structure types. Such an analysis yields a propensity for an amino acid type to form or break α-helices and β-strands. Figure 5.3 shows such a statistic. Analyzing doublet or triplets of consecutive amino acids was found to further improve the results. The first to use such an approach to predict secondary structure were Chou and Fasman [214]. In their algorithm α-helices are predicted to start at a position where four out of six adjacent residues are helix formers. The helix is extended until four out of six residues are helix breakers. The same is repeated with β-strands. Such a simple algorithm already yields a Q_3 value of 50-60%.

The two major improvements have found to raise the Q_3 value above 70%. First, instead of using a purely statistical approach, results were improved by using neural networks. The neural networks are trained on a large dataset and appear to better capture the non-linear details causing sequences to adopt particular secondary structures. The second improvement derives from feeding the neural network not just with the query sequence but rather a profile of homologous sequences generated with programs such as PSI-BLAST [128]. In this way, the details about mutations occurring at any position in the sequence

are captured, allowing the neural network to exploit more information during training.

The first method to incorporate these improvements is PHD by B. Rost [311]. It uses a three layered neural network. In the first sequence to structure layer, a 13 residue window taking the sequence and its homologues as input is used to predict the structure of the central residue. The second structure to structure layer again uses a 13 residue window and takes as input the predictions from the first layer. This serves to allow the network to consistently predict one type of structure for a longer segment, avoiding single mispredicted residues (e.g. a single E inside a stretch of H's). The third layer builds a jury decision from three differently trained neural networks for the central amino acid of the 13 residue window. The system also outputs a confidence value, showing how sure the algorithm is about any single predicted residue. A schematic representation of this process is shown in Figure 5.4. The authors report an overall Q_3 value of 71.6%.

Since different prediction methods have unique strengths and weaknesses, a consensus method has been implemented in JPRED [312]. This was shown to perform better than the single methods it uses to compute the consensus (including PHD). Another improvement was reached by using more sensitive homology search tools, like PSI-BLAST or hidden Markov models. Two methods which have raised the Q_3 value to about 76%. The first is PSI-PRED by D. Jones. It uses PSI-BLAST to compute a list of homologues and uses a two-layer feed-forward neural network, trained on a large set of data. SAM-T98 by K. Karplus uses hidden Markov models and performs similarly well. Perhaps the only recent algorithmic improvement is the development of bidirectional recurrent neural networks in the SSpro algorithm by G. Pollastri. From a theoretical point of view, these should be able to outperform the more classic feed-forward networks.

Nevertheless, secondary structure prediction appears to be stuck in the 76-77% Q_3 value region for the last two or three years. The best methods remain within $\pm$ 0.5% Q_3 of one another. It is assumed that 80-85% is the maximum Q_3 because agreement of secondary structure assignment from 3D coordinates using different standard programs is limited to this value.

5.4 Contact & Accessibility Prediction

In addition to the prediction of secondary structures, it is possible to build predictors for other features of protein structures. These can then be used to restrict the number of possible folds in tertiary structure prediction. In recent years, there has been an interest in predicting the number of residue contacts and solvent accessibility of single amino acids of a protein.

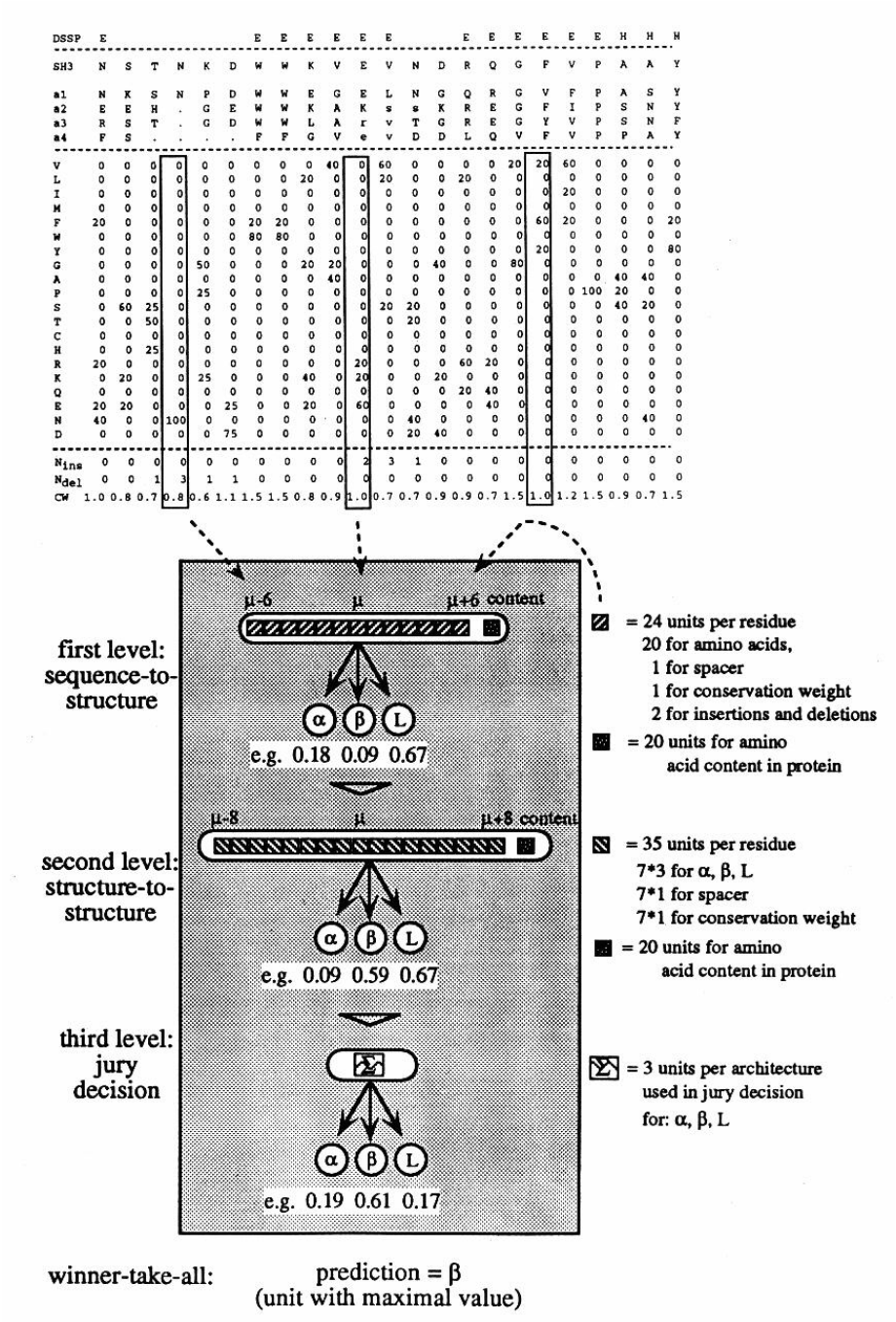

FIGURE 5.4. Schematic representation of PHD secondary structure prediction method.

These two features are correlated. A residue with a high number of inter-residue contacts will be located in the hydrophobic core of the protein, being shielded from the solvent. Conversely, a solvent exposed residue will have a low number of contacts. It has been argued that the two features are not identical and a partial separation is still required.[231]

The number of contacts for each residue is computed inside a spherical distance cut-off centered on each residue and by counting the number of residues falling inside a defined volume [268]. This can be expressed as a probability distribution. For contact prediction, this distribution is typically separated in two classes: higher or lower than the average value. In analogy to secondary structure prediction, the quality measure Q_2 is used to assess the two-state performance of the predictions. A base line classifier always outputs the most frequent category for each amino acid independently of its environment. It has a Q_2 value of 57% correct predictions [232].

In the last few years, different attempts to predict contacts [232] and distances between residues [233] have been made with some degree of success. These methods typically train neural network classifiers to predict inter-residue contacts from sequence, using amino acid properties. Of the newer methods, Fariselli & Casadio [233] use a feed forward neural network with a local window to predict a contact radius of 6.5 Å, yielding a Q_2 value of 69%.

A more complex approach is the one of Pollastri et al. [231]. Here a bidirectional recurrent neural network (BRNN) is used to predict four radius categories: 6, 8, 10 and 12 Å. A multiple sequence alignment is produced from the query sequence using the BLAST [131] program with standard parameters. The BRNN architecture is shown to expand the sequence window used for prediction, improving the results. They report an average Q_2 value ranging from 70% to 73%.

Predicting the map of inter-residue contacts may become an alternative way to predict tertiary structure. This would mean extending the current approaches to answer the question "is residue A less than X Å (e.g. 8 Å) away from residue B?". Given enough predicted distance restraints, it would be possible to treat such information like NOE constraints in NMR, generating compatible 3D structures. It is estimated that 40% correct predictions on a contact map should be sufficient to reconstruct a protein. The best method evaluated in CASP-4, from Fariselli & Casadio [313], achieved a rate of 10% correct predictions [310]. Figure 5.5 shows a contact map for a fold recognition target.

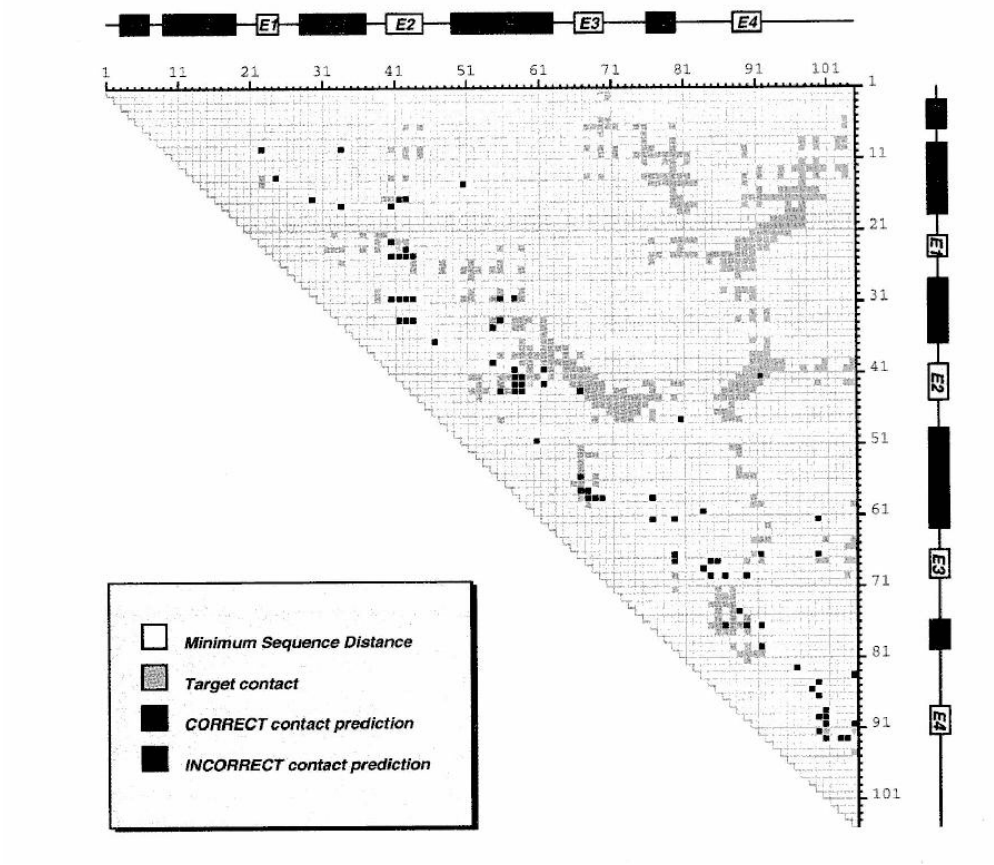

FIGURE 5.5. Sample contact map. Secondary structure elements for the two sequences are printed along the border. Helices in black, strands in white.

5.5 Ab Initio

Computing the tertiary structure of a protein from its sequence alone has been considered the "Holy Grail" of structural biology over the last 30 or so years. Anfinsen [228] showed that folding of most globular proteins is a purely physical phenomenon. Hence it should be possible to define a force-field based on the physics of the interactions among atoms, including the solvent, and to use a search method to determine the most stable structure of the protein at a given temperature and solvent conditions. This should be an ideal task for modern computers. Unfortunately, protein folding is still incompletely understood. Different *ab initio* methods have been implemented, but with limited success. All *ab initio* methods consist of three parts: a representation of the protein geometry, an energy function and a search method.

The geometry representation is important in defining the computational effort to calculate the potential energy for the model. Considering that the

number of atoms in a protein and the number of possible states in a polypeptide chain is prohibitive, some simplification is required. Two aspects can be simplified. The first is the number of atoms used to represent the protein structure. This can be reduced from the united-atom model (no hydrogens) to the virtual-atom models with one or more atoms per residue and further to models that use one atom to represent more than one residue. The second aspect is the nature of the search space, which is either continuous or discrete (so-called lattice models).

Most *ab initio* methods use a simplified continuous geometry model, in which virtual atoms represent several atoms. Lattice models are also used, with the advantage of allowing full enumeration of all possible states. The major problem here is that it is not possible to fit the native structure into the simplified lattice. Recently there has been a trend to use fragments of the protein chain, typically between three and nine residues long, to use as fixed "building blocks" for the structure. These combine the simplicity of lattice models with the flexibility of using continuous geometry models. Indeed, flexibility is scalable and proportional to the number of fragments to use in model building. An additional advantage is the possibility to extract the fragments from the *PDB*, capturing local interactions better than any force-field would allow to select good local structures.

Energy functions used for *ab initio* methods follow the same distinction between physical and statistical (or knowledge-based) potentials. These will be fully explained in Section 6.4. Most *ab initio* methods, due to the simplified geometry model, use some sort of statistical potential. The problem here is the parameter optimization, i.e. producing potentials that are able to assign a better energy value to the native structure than to wrong models. A step in the direction of solving this problem has been made with the creation of decoy sets for compact misfolded structures. These can be used to adjust the parameters of the energy potential to discriminate the native structure among erroneous alternatives.

The last element in an *ab initio* method is the search algorithm. From a computational point of view, almost any search algorithm for complex systems can be adapted. In practice, due partly to the biophysical background of many authors, mostly variations of Monte Carlo and/or simulated annealing methods are used to sample the search space. An exception to this is the limited usage of genetic algorithms [103][19]. Recently, a couple of methods using discrete search methods based on the branch & bound paradigm have also been proposed [85][274].

The methods applied by different research groups vary greatly. These range from using only free energy minimization approaches to the incorporation

of statistical knowledge of protein folding, both in the energy function and additional constraints. Examples for the latter are average α-helix and β-sheet lengths, fragments from *PDB* and predicted secondary structure. The best way to assess the state of *ab initio* methods is to refer to the CASP experiments. (see Section 6.1 for a detailed explanation of CASP) These give the most accurate picture of the current status in the field.

The first case where *ab initio* methods performed better, i.e. predicting a higher percentage of a structure correctly, than fold recognition methods on a novel fold classified as "hard" appeared in CASP-3 (in 1998). [2] The Baker group [321][322] in particular performed consistently well and demonstrated the potential of *ab initio* methods. This trend was confirmed in CASP-4 (in 2000), where the results of the Baker group became outstanding. Using a mixture of *ab initio* (for non-homologous structures/fragments) and homology modeling (for homologous structures/fragments) they even managed to reach the top post in the fold recognition ranking. Since in comparison the other *ab initio* methods were far less successful [310], the description will mainly focus on the Baker group, with additional descriptions of interesting alternative methods.

The central concept of Baker's method is the assumption that a library of structural fragments, three or nine residues long, samples the distribution of local conformations adopted by the sequence of the fragment in known proteins. The conformations of polypeptide segments are biased, but not restricted, towards those from secondary structure predictions. A Monte Carlo simulated annealing protocol is used to rapidly sample the conformational space. Either fragments are exchanged and displacement of the flanking regions minimized or (ϕ, ψ) angles of single residues are slightly modified. The number of steps is limited to about one minute of computation time and the process repeated with as many as 200,000 independent simulations. The potential function includes a hydrogen bonding term, a solvation term based on solvent accessible surface area scaled using atomic solvation parameters and a packing term using a modified Lennard-Jones potential. A set of filters is used to remove structures with low contact order (i.e. poorly packed) and those with many unpaired β-strands prior to refinement. The remaining structures are clustered. Clusters are compared with known protein folds and a frequently occurring structure with low energy is selected as the submitted model.

Results for the Baker group in CASP-4 were outstanding. For domains as large as 242 residues they were able to correctly identify substantial fragments of almost all novel fold proteins and the correct topology in several cases. In the final ranking they were awarded 31 points compared to a mere 10 of the second-best Friesner group. [310] Figure 5.6 shows the real structure for T0120, human DNA repair protein (XRCC4), and Baker group's prediction.

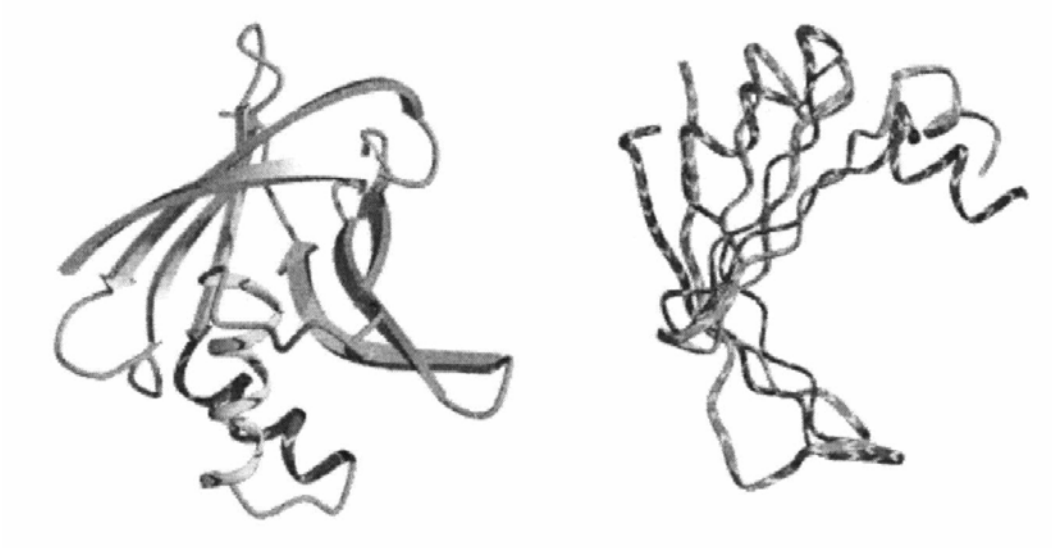

FIGURE 5.6. Comparison of the real structure (*left*) and Baker group's *ab initio* prediction (*right*) of T0120.

Friesner group [324] used a different strategy based on using predicted secondary structure for fold recognition and threading techniques to produce a limited list of possible remote homologues. Constraints, mainly describing typical distances in the templates, were extracted. These were then used to perform restrained energy minimization, attempting to mimic tertiary folding simulations. Variables in the simulation were (ϕ, ψ) angles of loop regions. Secondary structures were fixed with idealized torsion angles. For mainly β-sheet proteins additional constraints limiting $C_\alpha - C_\alpha$ distances were included.

A similar approach was followed by Skolnick's group [325]. They use a threading method and secondary structure predictions to find small fragments that can be assembled using a lattice model. These are optimized and refined into more detailed off-lattice models using a combination of statistical and physics-based energy potentials. A number of restraints is also used to improve the optimization.

Both the Friesner and Skolnick methods, while not as successful as Baker's, were able to correctly predict large backbone fragments of novel folds in CASP-4. They also somehow highlight the trend in CASP-4, where *ab initio* methods using much knowledge from existing structures were more successful than pure energy minimization methods. Whether these still deserve to be termed "*ab initio*" or not is still subject to debate.

A different *ab initio* approach, called MOLEGO, was implemented in our group by E. Bindewald [85]. MOLEGO uses a novel discrete search algorithm, called best-profile search, to sample the search space. This is a global opti-

mization scheme that tries to establish the "most promising" paths to attempt for energy minimization. It was shown to outperform classical branch & bound algorithms. Various geometry representations have been implemented, usually consisting of sampling highly populated regions of the Ramachandran map with four or more (ϕ, ψ) angle combinations. A novel orientation dependent knowledge-based potential is also used to evaluate the energy of a conformation. Constraints are also introduced, e.g. a radius of gyration term to produce compact structures and a β-strand pairing term. MOLEGO is able to predict proteins with less than 50 amino acids to less than 5 Å RMSD. [85] It was also used by our group during CASP-4 to predict small fragments not aligned with a template structure.

5.6 Common Optimization Methods

Some simple optimization methods that can be applied to all types of problems related to protein structures are frequently found in the literature. These are the Monte Carlo and simulated annealing algorithms. Even if not directly used in this thesis, they are nevertheless ubiquous and results will have to be compared with them on some occasions. A brief description will therefore cover their main strengths and weaknesses.

Monte Carlo algorithms essentially implement a random search. From a random starting point, random changes in the solution space are made. At each step, the energy is calculated. If the new energy value is lower, the random step is accepted. If the new energy is higher, it is accepted with a given probability. The generally used Metropolis criterium [229] sets this probability to $e^{-\frac{\Delta E}{kT}}$. The temperature T is a free parameter and is constant in simple Monte Carlo searches. The choice of T is important as it determines the rate of convergence to the global optimum. If T is too small, the system will be stuck in a local minimum, because passing the energy barrier is very improbable. If T is too large, the system will not converge to the global optimum, since it will keep exploring new conformations.

Kirkpatrick et al. [230] implemented the idea of simulated annealing as a way to avoid the choice of a fixed temperature. At the beginning of the conformational search the temperature T is set to a high initial value. The solution space can be freely explored and large energy barriers surpassed. During the search, T is gradually lowered until it reaches zero. This ensures that the system converges to a local, or hopefully the global, optimum.

From their description is apparent why Monte Carlo and simulated annealing enjoy such a popularity: They are very simple to implement and yield satisfactory results if enough computing power is used. This is paid for in slow convergence and the impossibility to establish whether the global minimum

has been found or not. This is the reason why Monte Carlo and simulated annealing protocols usually require a number of independent optimization runs to ensure that a good solution is found. For large systems the computation time can easily become prohibitive. Nevertheless, these algorithms form the base line for any optimization task.

Part II

Modeling of Protein Structures

6
State of the Art

Before the work done in thesis can be described, it is necessary to sketch the current state of the art concerning all major aspects related to the modeling of protein structures. (The major focus of this thesis, loop modeling, will be more thoroughly introduced in Part III.)

It is important to understand that the present description of the state of the art cannot be considered to give more than a brief introduction in each topic. The literature on protein structure prediction is becoming daunting, so an attempt to describe every method found in the literature in this thesis would be doomed to fail. Instead, the following description will be based mainly on the CASP experiment, which gives the most objective view of which methods seem to work best.

6.1 CASP

Attempts to predict the structure of proteins date back at least to the late 1960s [254], and most methodological advances necessary to make protein structure prediction viable were developed in the 1970s and 1980s. One of the first experiments to model unknown protein structures by homology, mammalian serine proteinases, was described by Greer in 1981 [255]. Since then, protein structure prediction has seen a growing interest, with many new approaches being developed.

At the start of the 1990s, a situation was reached where many publications already claimed to have "solved" the protein structure prediction problem. While this may be "true" for a very limited number of proteins, it was widely known to not be the case (yet).

This prompted J. Moult to create a true blind test to assess the real state of the art in protein structure prediction. The *Critical Assessment of techniques for protein Structure Prediction* (CASP) series of experiments was organized. The first was held in 1994 (CASP-1) and subsequent experiment follow every two years, with the most recent in 2000 (CASP-4).

CASP4 Target T0111

1. Protein Name
enolase
2. Organism Name
Escherichia coli
3. Number of amino acids (approx)
431
4. Accession number
P08324
5. Sequence Database
Swiss-prot
6. Amino acid sequence
SKIVKIIGREIIDSRGNPTVEAEVHLEGGFVGMAAAPSGASTGSREALEL
RDGDKSRFLGKGVTKAVAAVNGPIAQALIGKDARDQAGIDKIMIDLDGTE
NKSKFGANAILAVSLANAKAAAAAKGMPLYEHIAELNGTPGKYSMPVPMM
NIINGGEHADNNVDIQEFMIQPVGAKTVKEAIRMGSEVFHHLAKVLKAKG
MNTAVGDEGGYAPNLGSNAEALAVIAEAVKAAGYELGKDITLAMDCAASE
FYKDGKYVLAGEGNKAFTSEEFTHFLEELTKQYPIVSIEDGLDESDWDGF
AYQTKVLGDKIQLVGDDLFVTNTKILKEGIEKGIANSILIKFNQIGSLTE
TLAAIKMAKDAGYTAVISHRSGETEDATIADLAVGTAAGQIKTGSMSRSD
RVAKYNQLIRIEEALGEKAPYNGRKEIKGQA
7. Additional Information
oligomerization state: dimer in the presence of magnesium by dynamic light scattering and small angle x-ray solution scattering and
in the recently solved crystal structure.
8. Homologous Sequence of known structure
yes
9. Current state of the experimental work
Protein supply: overexpressed in E. coli
crystals: grown at 20 C from PEG 3550
diffraction quality: strong data to 2.5 A with good redundancy

Structure solved by molecular replacement. Currently, the refinement to 2.5 A resolution is near completion.
Current Rfree 27 % ; R 22 %

10. Interpretable map?
yes
11. Estimated date of chain tracing completion
complete
12. Estimated date of public release of structure
Dec 2000
13. Name
Unavailable until after public release of structure

FIGURE 6.1. Sample sequence from CASP-4. T0111, a 431-residue Enolase.

The CASP works as follows: Before the start of CASP, the organizers request structures from X-ray crystallographers and NMR spectroscopists which are about to be experimentally solved, and thus not yet publicly available. The corresponding **sequences** are collected and published on the CASP homepage [99], together with additional information such as name and reference to a sequence database. These sequences are usually called *targets*. A sample target from CASP-4 is shown in Figure 6.1. (The full list of CASP-4 targets can be found in Appendix A.1.)

The registered predictors are permitted enough time to predict the unknown **structures** (June to September). These predictions are collected by the organizers and forwarded in an anonymous form to the independent assessors

(September). Three categories are defined in CASP: homology modeling, fold recognition and *ab initio*[1]. The assessors compare the predictions with the experimental structures and decide a ranking of the submissions. This serves to select a limited number of prediction groups performing "*consistently well*".

In December a conference is held in Asilomar (CA), USA, where all predictors gather. The results are presented and the ranking is announced. The best groups are invited to describe their methods to the rest of the community. In addition, a special issue of the journal *Proteins* (e.g. for CASP-3 [2]) is published some months later, containing detailed descriptions of what has been achieved and articles describing the best methods. Additional information about the experiment, e.g. a detailed numerical evaluation, can be found at the CASP web site [99].

Since its creation in 1994 the CASP experiment has quickly become the most important event for research groups involved in protein structure prediction. All major groups usually participate in the CASP, and to be selected as one of the best performing groups is a major source of scientific reputation. In CASP-4 a total of 160 groups participated across all categories, predicting 43 targets. Fold recognition turned out to be the category drawing most interest with 127 participating groups.

Beginning with CASP-3 (1998) a parallel experiment was organized for automatic servers: CAFASP (*Critical Assessment of Fully Automated Structure Prediction*). This is reserved to publicly available web servers, with over 20 servers participating in CAFASP-2.

Dealing with protein structure prediction, it was felt that the methods developed during the dissertation had to be tested in the CASP-4 experiment in order to gain an objective view of their performance. The results of our group's participation in CASP-4 are summarized in Chapter 11. The full numerical evaluation can be found in Appendix A.2.

6.2 Homology Modeling

If the sequences of two proteins are significantly similar, their structures will roughly superimpose. As was established in Section 5.2, a sequence identity over 20-30% is generally sufficiently significant[2]. With the growing number of experimentally solved structures, this concept has become a powerful way to infer the structure of unknown proteins.

[1] The latter was called "*new folds*" in CASP-4 to avoid previous disputes.

[2] This statement is true for most naturally evolved proteins. Some studies suggest that it may not be the case for artificial ones.

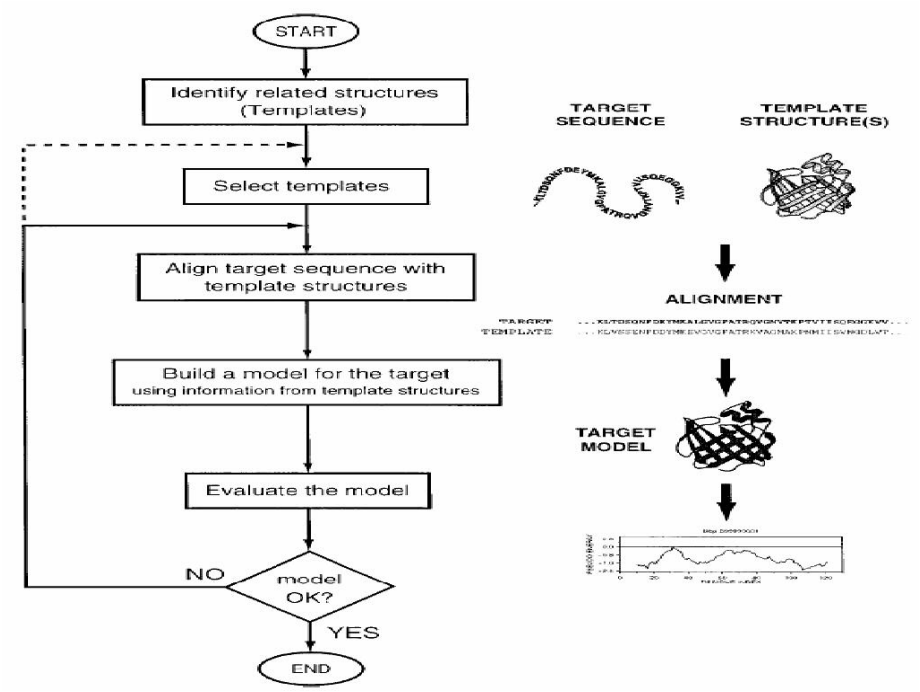

FIGURE 6.2. An overview of homology modeling.

Building a model by homology requires a general approach leading from the selection of a suitable template structure and its alignment with the target sequence to the refinement of the full model. Rules can be deduced from existing structures to guide and improve the modeling process. A flow diagram of the modeling process is shown in Figure 6.2.

The first step in homology modeling is to scan a structural database for suitable template structures. This can be done with sequence comparisons or more sophisticated fold recognition methods, the latter being the subject of the following section. Finding a significant sequence similarity implies inferring a plausible correspondence between residues of both sequences. Selection of the template structures is therefore generally combined with aligning the template sequence to the target (for a definition of alignment see Section 4.1). For higher sequence identities ($\geq$ 45%) this is straightforward. Most programs will roughly produce a similar alignment. The computation of an alignment becomes extremely difficult for very low identity (<= 25%). In fact, aligning the target sequence to the template structure is the most frequent and serious source of errors in homology modeling [239].

Building a model from a given alignment is fairly simple. The coordinates of the atoms in the template structure can either be copied directly or used to derive restraints that are used to optimize the model. The main problem here

is to define those fragments of the model, the structurally variable loops, which should not be copied from the model. In fact, loops are the most flexible parts of the structure and tend to vary even between closely homologous structures [239]. These difficulties have led to the development of specialized methods for modeling loop structures. Second in severity only to alignment errors, loop modeling is a subject of ongoing research. It will be treated in depth in Part III of this thesis.

A somewhat simpler problem is the placement of side chains, which will be thoroughly addressed in Section 6.5. While side chains tend to be reorganized between closely homologous structures, the errors arising from side chain placement are relatively minor. Moreover, it was observed that side chain quality depends on alignment accuracy [2]. Where the model is incorrectly aligned, side chains will be incorrectly placed as well. Particular care has to be taken when modeling side chains close to the active site of the protein. Errors in these side chains may compromise the utility of the model for studies of its biochemical function.

The final steps of the model building process consist in assessing the quality of the model and adjusting small errors. A limited energy minimization of the structure may also be employed to reduce local clashes between atoms. The stereochemical quality may be assessed with programs such as PROCHECK [76][77], which indicate deviations from ideal bond lengths and angles. Energy functions are useful to both estimate the overall quality and to perform energy minimization. These will be more thoroughly addressed in Section 6.4. A flow chart of this "classic" approach to homology modeling is shown in Figure 6.3.

As has already been established, the sequence to structure alignment is the prime source of errors in homology modeling. This has prompted the development of sophisticated alignment techniques that are frequently used for fold recognition. Their discussion is therefore presented in the next section. Here we will instead focus on the best database search tools currently available.

PSI-BLAST [128] has become the de facto standard for searching homologous templates and aligning these. It combines a fast heuristic with the sensitivity of more complex alignment programs by building a protein specific scoring matrix (*PSSM*). An iterative search with such a *PSSM* presently forms the best way to exploit all the available knowledge from protein sequences. The algorithm will be described in more detail in Section 7.2.

Even with PSI-BLAST the sequence-structure alignment is not always correct for building a model. Inclusion of structural information like secondary structure or conserved "key" residues (e.g. in the active site) is known to improve the alignment quality by limiting local shifts. Extracting knowledge on conserved positions from structural alignments between related proteins also helps to pinpoint those parts of the structure that can be confidently pre-

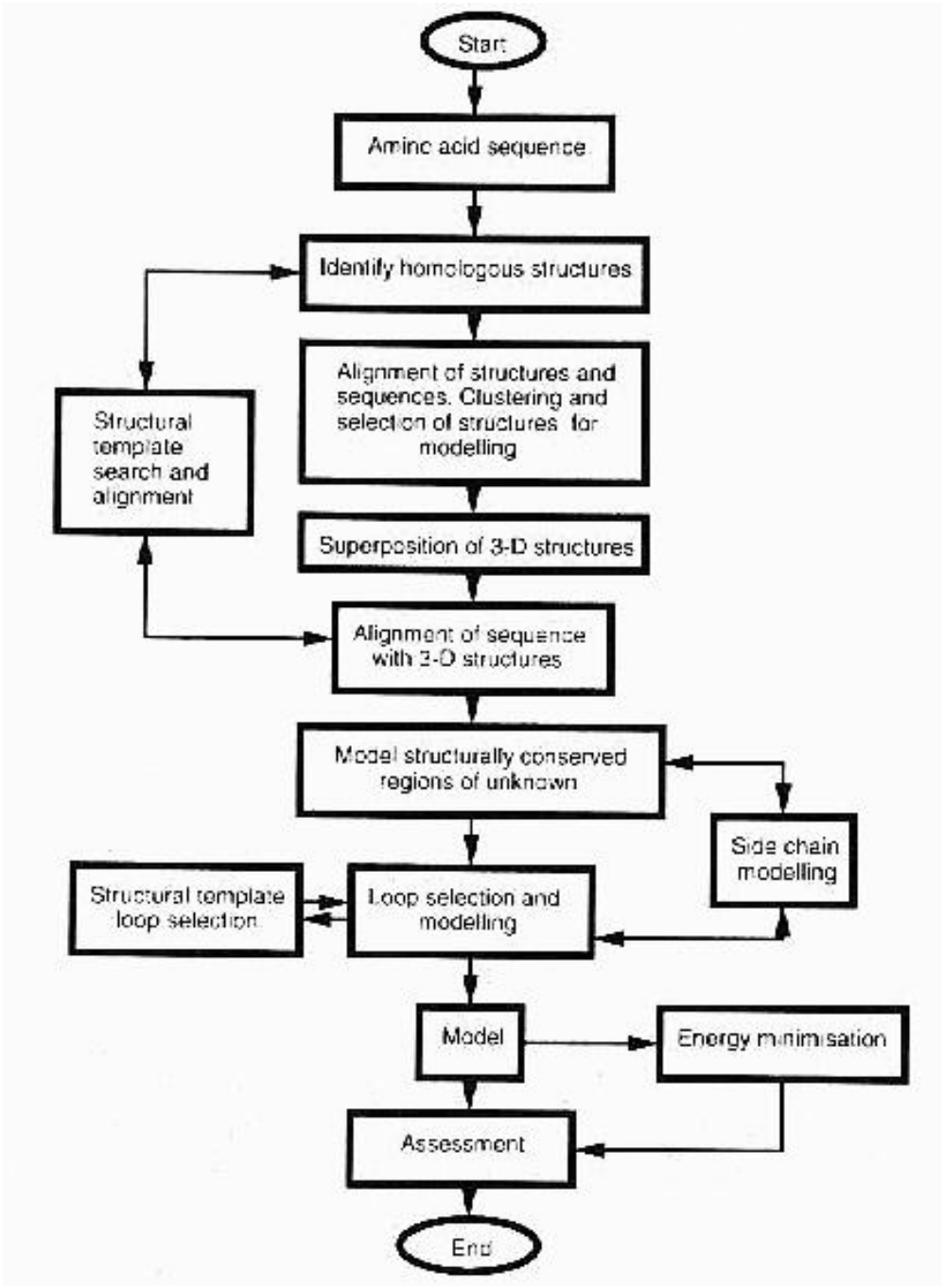

FIGURE 6.3. The"classic" approach to homology modeling, as implemented by *Composer*.

dicted. The other parts can be predicted with loop modeling or are sometimes left altogether unpredicted.

Another open problem are multi-domain proteins. For instance in In CASP-4, several proteins were composed of two or more domains which had to be modeled separately. Both parts were homologous to those in the database, but nothing was known about how to assemble them. To the best of this author's knowledge, no automated method exists for assembling multiple domains into complete models[3].

With homology modeling it is possible to build models with an RMSD of less than 2.0 Å for targets with high ($\geq 60\%$) sequence identity. Targets with lower identity ($\geq 30\%$) can generally be predicted with less than 5.0 Å overall RMSD [297]. Exceptions to both statements are possible, but tend to occur more frequently at lower identity levels. The overall RMSD is not equally spread over the structure. In general, the protein core containing secondary structures is almost perfectly modeled and may only be slightly shifted. Insertions, deletions and loops on the surface are where most of the errors concentrate. Errors at the start and/or end of helices and strands are also possible.

Almost all homology modeling methods tested at the CASP experiment retain some part of manual intervention, especially during the alignment stage. The most widely used program to build models is A. Šali's *Modeller* [185]. It takes an alignment and template structure(s) as input to extract a series of structural restraints which serve to build an optimized model. This procedure attempts to produce more "flexible" models, which try to accommodate local shifts. Loop modeling is still problematic for *Modeller*[245].

The best performing groups in CASP tend to use their own proprietary methods for homology modeling [2]. Venclovas et al. [323] produced some of the best sequence-structure alignments by using a combination of structural alignments, producing several alternative sequence alignments and manually selecting and editing the most probable one.

Blundell et al. [315] use their database of homologous structures, HOMSTRAD [316], to produce better alignments. An iterative approach is followed, where the alignment is expressed as a model and the structure re-aligned with the conserved features of the protein family. In addition, they explicitly use loop modeling [15] and side chain placement methods to complete the model.

Arguably the most automated homology modeling method is the one by Bates and Sternberg [320]. This was fully automated during CASP-4 and is available as the 3D-JIGSAW server [319]. For sequences with identity not larger than 40% to the template they use predicted secondary structure in addition to structural alignments of the protein family to improve the over-

[3]In CASP-4 only Baker's group and our own group submitted assembled models. Baker used an *ab initio* method to predict the whole model, circumventing the assembly problem. Our method was manual.

all alignment. The model is built either from a single or multiple templates, depending on the RMSD and sequence identity difference among members of the same family. Loop modeling is based on a database search of similar fragments. After the side chain placement step, the model is energetically minimized using the CHARMM force field[60].

All in all, it is fair to say that RMSD differences between models from several groups are generally below 1.0 Å. The main shortcomings, compared to the best possible solution, can be found in loop regions and selection of a non-optimal template. The latter is a bit of a gamble, as sometimes the template with the highest sequence identity may not correspond to the one with the lowest possible RMSD. The performance of a group on an individual protein is therefore not necessarily an indicator for the overall performance.

6.3 Fold Recognition

If no suitable homologous template structure can be found, fold recognition methods provide another option for constructing useful models. The goal of fold recognition is to use a known structure as a model of the fold for a new sequence rather than to predict the structure from physicochemical characteristics of the sequence information alone as in *ab initio* folding. Instead of searching the vast conformational space of the new protein, the search is confined to the conformations of known structures.

Homology modeling methods use the amino acid sequence for computing an alignment only and do not exploit 3D structural information. In 1991, Bowie et al. [258] developed an alternative method: instead of scoring the compatibility of a sequence by comparing it to a sequence of known structure, the sequence was compared with the structural information of known 3D structures. This method was termed structural 3D-1D profile. Since then, a variety of fold recognition methods have been published (e.g. [192][193][259][263][267]) and several reviews on these methods have appeared (e.g. [111][112][204][264][265][269]). The fold recognition methods can be roughly divided into four classes:

1. structural (3D-1D) profile methods [195][205][258][270].
2. threading methods [192][193][259][271].
3. sequence profile methods [128][194][206][207][266].
4. mapping methods [85][191][209][261][262].

Recent methods have begun to combine elements from two or more classes (e.g. [195][205]). The main components of protein fold recognition are:

1. a library of known template folds

2. a scoring function used to evaluate the compatibility between the probe sequence and the template fold

3. the algorithm used to search for the optimal alignment of the probe sequence to each template fold

4. the significance assessment obtained by ranking compatibility scores for sequence-fold pairs.

The approaches used differ in at least one component of fold recognition [260]. In the following, we describe the each component of these methods.

A library containing all known protein folds is desirable. As the time required to scan all known *PDB* structures would make fold recognition ineffective, representative subsets are employed. Typical fold libraries are extracted from a structural classification, such as SCOP, CATH or FSSP (see Section 4.3 for a description). Depending on the fold recognition method, these may undergo clustering to improve coverage and reduce redundancy. For example Kelley et al. [195] build a fold library from SCOP domains augmented by additional pseudofamilies defined by multiple alignments.

Compatibility functions are used by the alignment algorithm to find the most likely sequence-structure alignment. The functions used in fold recognition can be classified into two types: unipositional and multipositional. In general, the compatibility function associates a score to the match of one or more amino acids from the probe sequence to one or more structural positions in the target fold. Each structural position is characterized by some physicochemical features. The main difference between uni- and multipositional compatibility functions lies in the way the occurrences are counted. For unipositional functions, the occurrences are counted independently at each position. For multipositional functions, the occurrences are counted at more than one position simultaneously.

In 1991, Bowie et al. [258] matched sequences to a fold using an unipositional compatibility function that related a sequence to its residue's environments in the 3D structure. The environments are described in terms of:

1. total area of the side chain buried by other protein atoms;

2. fraction of the side-chain area covered by polar atoms or water; and

3. local secondary structure.

Based on these parameters, each residue position is classified into an environmental class. The authors describe this method as a 3D-1D profile, in which a 3D structure is encoded as a 1D string that represents the environmental class of each residue in the folded protein structure, as shown in Figure 6.4.

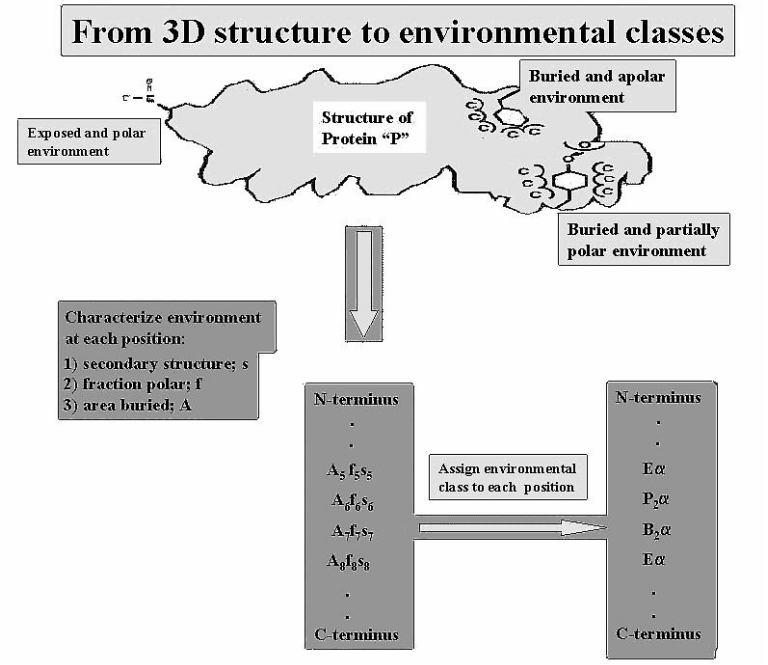

FIGURE 6.4. How a 3D structure is encoded as a 1D string that represents the environment class of each residue in the fold protein structure.

For each of the 20 amino acids a total of 18 discrete environmental classes is defined. The compatibility function in this case is an 18 by 20 table of the 3D-1D scores. Each score specifies the compatibility value of aligning an amino acid to an environment, as shown in Figure 6.5.

After determining the environmental class of a given position in a protein structure, it is possible to construct a 3D profile. A profile is a $n * m$ matrix containing: n is the position index in the structure, with each row corresponding to a residue in the structure; m is normally 20, each corresponding to an amino acid type. The entry (i, j) in the profile specifies the value of the matching of residue type j from sequence to environment at position i of the structure. The compatibility of the sequence for the 3D fold is the sum of the individual residue compatibilities in the alignment, corrected for gap penalties. This is exemplified in Figure 6.6.

Other fold recognition methods are the sequence profile methods [128] [266]. These methods allow the recognition of weak relationships between proteins that previously were considered "structure-only" similarities. Generally, these

Environment class	W	F	Y	L	I	V	M	A	G	P	C	T	S	Q	N	E	D	H	K	R
$B_1\alpha$	1.00	1.32	0.18	1.27	1.17	0.65	1.26	-0.66	-2.53	-1.16	-0.73	-1.28	-2.73	-1.06	-1.93	-1.74	-1.97	-0.84	-1.82	-1.67
$B_1\beta$	1.17	0.85	0.07	1.13	1.47	1.09	0.55	-0.79	-2.02	-0.84	-0.22	-1.12	-2.91	-1.07	-1.42	-1.93	-2.56	-1.91	-2.69	-1.16
B_1	1.05	1.45	0.17	1.10	1.11	1.02	0.96	-0.91	-1.92	0.26	-1.22	-1.53	-2.61	-1.17	-2.42	-2.52	-1.76	-1.12	-2.59	-2.16
$B_2\alpha$	0.50	0.90	0.85	1.01	0.63	0.68	1.12	-0.69	-1.49	-2.21	-0.10	-1.50	-1.47	-0.23	-0.61	-0.71	-1.62	0.23	-0.78	0.05
$B_2\beta$	0.01	1.18	1.05	0.76	1.31	1.06	0.64	-1.55	-2.26	-0.48	-0.87	-2.27	-1.77	-1.22	-2.07	-1.97	-1.41	-0.77	-1.14	-0.20
B_2	1.02	1.05	1.12	0.64	0.81	0.60	0.90	-0.66	-1.66	0.19	-0.05	-0.76	-1.17	-0.76	-0.55	-1.35	-1.28	0.48	-2.34	-0.80
$B_3\alpha$	0.92	-0.03	0.58	0.15	0.04	-0.02	0.69	-0.57	-1.88	-0.68	-1.58	-0.57	-0.96	0.22	-0.06	0.08	-0.50	0.73	0.43	0.96
$B_3\beta$	0.75	0.81	1.30	0.18	0.54	0.56	-0.57	-0.93	-1.93	-0.34	-0.54	-0.44	-0.74	0.21	-0.24	-0.14	-0.86	0.82	-0.53	0.13
B_3	1.07	0.70	1.13	0.35	-0.17	-0.03	0.23	-0.96	-0.98	-0.13	-1.20	-0.53	-0.54	0.05	0.04	-0.38	-1.05	1.01	0.10	0.66
$P_1\alpha$	-1.35	-0.82	-0.59	-0.62	-0.24	0.10	-0.03	0.73	-0.49	-0.25	0.95	0.31	0.34	-0.14	-0.54	-0.17	-0.25	-0.52	-0.21	-0.28
$P_1\beta$	0.36	-0.49	0.17	-1.03	0.20	0.46	-0.27	0.64	-0.82	-0.55	1.49	0.93	0.33	-2.27	-1.32	-0.73	-1.07	-0.42	-1.21	-0.77
P_1	-1.26	-1.20	-1.31	-0.62	-0.23	-0.01	-1.19	0.46	-0.24	0.65	1.35	0.56	0.49	-0.63	-0.13	-0.51	0.38	-1.12	-0.74	-1.29
$P_2\alpha$	-1.14	-1.43	-0.79	-0.35	-0.54	-0.48	-0.45	0.06	-0.50	-0.28	-0.93	-0.05	-0.18	0.55	-0.05	0.56	0.28	0.06	0.61	0.50
$P_2\beta$	-0.79	-0.54	-0.84	-1.30	-0.33	0.13	-0.72	-0.55	-0.98	-1.29	-0.57	0.84	0.59	-0.06	-0.16	0.32	0.19	-0.67	0.59	0.10
P_2	-0.82	-0.86	-0.51	-0.70	-1.09	-0.88	-0.89	-0.15	-0.40	0.44	-0.50	0.08	0.26	0.27	0.50	0.27	0.49	0.13	0.44	0.30
$E\alpha$	-1.35	-2.20	-2.10	-1.58	-2.76	-1.10	-0.72	0.46	0.56	0.04	-0.44	-0.17	0.15	0.36	0.28	0.59	0.44	-0.19	0.13	-0.34
$E\beta$	0.64	-0.90	0.30	-1.66	-1.47	-1.74	-0.68	0.06	1.46	-0.98	-0.24	0.14	0.65	-0.19	-0.06	-0.16	-0.76	-0.83	-0.52	-0.49
E	-2.14	-1.90	-0.94	-1.19	-1.61	-0.91	-1.67	0.12	1.13	0.20	-0.46	0.12	0.32	-0.03	0.41	0.03	0.22	-0.25	-0.14	-0.32

FIGURE 6.5. 3D-1D scoring table. The scores for pairing residue i with environment j are given. The environments are divided by secondary structure (α,β, coil) and six classes of solvent accessibility (3 * buried, 2 * partially exposted, exposed).

new approaches are based on the concept of a sequence profile [276]. Additionally, they incorporate two important ideas, namely construction of a position specific scoring matrix, *PSSM*, [128][275][277] and an iteration of the database search until weak relationships are detected. Conceptually similar ideas have been implemented in the family of sequence analysis methods based on hidden Markov models (*HMM*) [194].

Another generation of fold recognition methods such as mapping methods are based on 1D predictions [85][191][260][261][262]: first, 1D structure information (secondary structure and solvent accessibility) is predicted for a sequence of unknown structure, then the 1D information is extracted for a library of known structures, and finally the observed and predicted 1D structure strings are aligned by a standard dynamic programming algorithm.

Methods such as threading [192][193][259] use a class of multipositional compatibility functions, because in such functions the compatibility is computed by considering two or more positions in the alignment at the same time. In these methods, the functions are used to attempt to describe the numerous interactions that operate in a 3D protein fold in some simplified way. A common energy function consists of pairwise interatomic energy terms, with the structural role of any given residue described in terms of its interaction with the environment. Both the 3D distance and the sequence separation between the components of each pair may be included. One of these compatibility functions are the knowledge-based potentials developed by Sippl [184], which will be described in Section 6.4. To identify probable models for the unknown fold

		Amino acid type													Gap penalty	
Position in fold	Environment class	A	C	D	E	F	G	...	R	S	T	V	W	Y	Opn	Ext
1	E	12	-46	22	3	-190	113	...	-32	32	12	-91	-214	-94	2	0.02
2	B_2	-66	-5	-128	-135	105	-166	...	-80	-117	-76	60	102	112	2	0.02
3	E α	48	-44	44	59	-220	68	...	-34	15	-17	-110	-135	-210	200	200
4	P_2 α	6	-93	28	56	-143	-50	...	50	-18	-5	-48	-114	-79	200	200
5	E α	46	-44	44	59	-220	68	...	-34	15	-17	-110	-135	-210	200	200
6	P_2 α	6	-93	28	56	-143	-50	...	50	-18	-5	-48	-114	-79	200	200
7	B_2 α	-69	-10	-162	-71	90	-149	...	6	-147	-150	68	50	85	200	200
8	E α	46	-44	44	59	-220	68	...	-34	15	-17	-110	-135	-210	200	200
9	P_2 α	6	-93	28	56	-143	-50	...	50	-18	-5	-48	-114	-79	200	200
10	B_1 α	-66	-73	-197	-174	132	-253	...	-167	-273	-129	66	100	18	200	200
.	.	.	.	.	.	.	.		.	.	.		.	.	.	.
.	.	.	.	.	.	.	.		.	.	.		.	.	.	.
.	.	.	.	.	.	.	.		.	.	.		.	.	.	.

FIGURE 6.6. An example of a 3D profile of sperm-whale myoglobin. An environment group is listed for each position, followed by scores for placing each of the amino acids at that position. The example shows the first ten positions. The scores placed in each row are 3D-1D scores of Figure 6.5, multiplied by 100. Gap penalties are determined empirically.

of a given sequence S, the sequence is compared to all possible conformations in the database. A pool of conformations C_n is obtained by taking all possible fragments of length L from the database (L is the length of the amino acid sequence of interest). The sequence is mounted onto all fragments C_n so that the overall energy is minimized, then the conformations are sorted with respect to their energy. The fragments of low net energy obtained from the pool of conformations are candidates for the unknown conformation of S.

Jones et al. [193] used Sippl-like potentials but supplemented them with a solvent-accessibility term. Bryant and Lawrence [259] use a 2-positional compatibility function in which pairs are either in contact or not. Godzik et al. [267] use a unary, binary, and tertiary interaction compatibility function to build a "topological fingerprint" that considers the buried area of each residue plus the pairs and triplets of residues that are in contact. It can be used to calculate the pseudo-energy of any protein structure.

For every structure in the database these parameters are calculated and stored as the "topological fingerprint" library. This fingerprint does not use sequence information, but merely defines the characteristics of each position along the template chain. Thus, it is possible to calculate the energy of a corresponding system with the same interaction pattern but a different sequence.

The main difficulty of multipositional compatibility functions arises when used to compute an optimal alignment. Indeed, it has been demonstrated that the time required to find an optimal alignment with the multipositional compatibility functions present an NP-complete problem [272], which means

that the time required to find the solution grows exponentially as the size of the protein increases. Godzik et al. [267] simplify the problem by introducing a "frozen" approximation to calculate protein energy. This method approximates the multipositional compatibility functions as unipositional and allows the application of any dynamic programming algorithm to find the optimal alignment. In the "frozen" approximation of a 3-positional function, the compatibility value of aligning three amino acids from the sequence to three positions in the structure is computed using only one amino acid from the sequence and taking the second and the third from the structure. Here, the energies of the amino acids from sequence B are calculated as if they interact with their partners from protein A.

The choice of algorithm to obtain an optimal alignment depends on the type of compatibility function used by the method. Methods based on unipositional compatibility functions such as [258] can directly use a dynamic programming algorithm to find the optimal alignment. Methods based on multipositional compatibility functions have to transform the function into an unipositional one, or require a different alignment algorithm. Bryant and Lawrence [259] use a 2-positional compatibility function to find the optimal alignment. They do not convert the 2-positional function into a unipositional one and approximate the pairwise interactions by using a Monte Carlo sampling algorithm [259].

Once the scores for the compatibility of the probe sequence to each template fold are computed, they must be ranked. The simplest ranking is sorting in decreasing order of raw scores [193]. When a probe sequence is compared against a template fold library, there will always be a fold with the highest compatibility score. Using a measure of statistical significance, the Z-score, it is possible to quantify whether this compatibility score is high enough to be significant. The Z-score is the number of standard deviations that a score is above the mean score for all matches. Methods ranking with Z-scores automatically attach a reliability measure to the result [207]. A more sophisticated method for evaluating the confidence level is used in GenTHREADER [205]: A neural network is trained to rank the confidence based on features such as alignment score, alignment length, pairwise potential and solvation scores, with the output expressing a probability.

The confidence level returned by a fold recognition program is an important measure as it serves to discriminate between correct predictions and false positives. When this measure is above a program-specific threshold it is safe to assume that the prediction is correct.

The first truly successful example of protein structure prediction by fold recognition, replication terminator protein (**rtp**) from *B. subtilis* [268], has been conducted in CASP-1. The structure of **rtp** and template are shown in

Figure 6.7, with the corresponding alignment in Figure 6.8. This case showed that structures can be quite similar even if their sequences are unrelated and the successful prediction demonstrated that computational techniques are capable of recognizing this structural similarity.

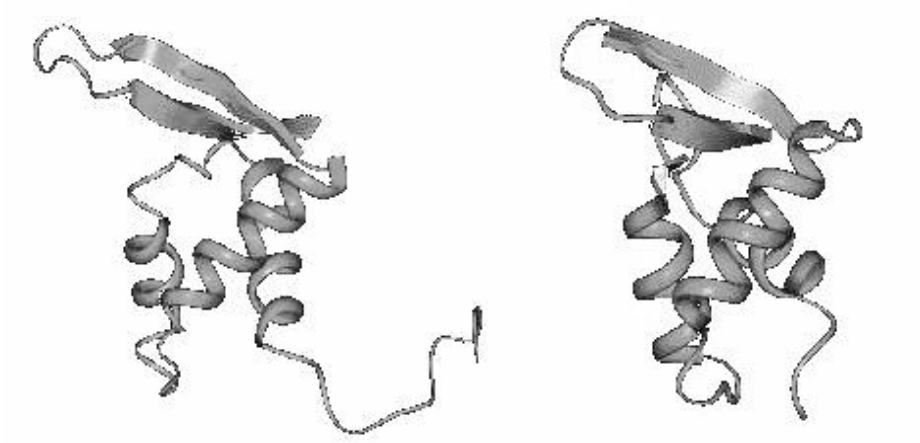

FIGURE 6.7. Structures of replication terminator protein from *B. subtilis* (`1bm9`) (*left*) and histone H5 from a chicken (`1hst`) (*right*). The C-terminal helix of rtp is not shown.

```
                      HHHHHHHHHHKHHHHH     HHHHHHHHHHH          HHHHHHHH
rtp     MKEEKRSSTGFLVKQRAFLKLYMITMTEQERLYGLKLLEVLRSEFKEIGFKPNHTEVYRSL

1hst    ....SHPT-----YSEMIAAAIRAE--KSRGGSSRQSIQKYIKSHYKVG-HNADLQIKLSI
                    HHHHHHHHHH          EEHHHHHHHHHHH        HHHHHHHHH

        HHHHH   EEEEEEE        EEEEEEE  HHHHHHHHHHHHHHHHHHHHHHHHHHHHH
rtp     HELLDDGILKQIKVKKEGAKLQEVVLYQFKDYEAAKLYKKQLKVELDRCKKLIEKALSDNF

1hst    RRLLAAG---VLKQTKGVGA-SGSFRLA-K.............................
        HHHHH      EEEE         EEEE
```

FIGURE 6.8. Sequence–structure alignment of rtp and 1hst. The secondary structure is indicated by H for helices and E for strands.

In CASP-3 the target proteins for fold recognition were divided into two groups: homologous folds, or members of the same superfamily, and common folds in the SCOP database, i.e. without detectable evolutionary relationship. Different results were obtained for the two groups. Very few common fold relationships were detected. Results for the homologous fold relationships were much better. For all but one of the targets, multiple predictors had identified correct folds. No predictor alone recognized more than half the relationship,

so that a combination of methods is needed to get reliable results [2]. In CASP-3 the "threading" methods received the most attention. Methods such as PSI-BLAST [128] and Hidden Markov Models [194], augmented by secondary structure prediction, were competing with threading to identify the correct fold. The quality of the model depends on the sequence structure-alignment. Alignment quality did improve significantly between CASPs 1 and 2, but not detectability between CASPs 2 and 3.

During CASP-4, the results of automatic predictions were already available at the CAFASP-2 homepage [97]. The following fold recognition servers turned out to be quite useful for compiling a consensus:

- **3D-PSSM** [195] (`http://www.bmm.icnet.uk/~3dpssm/`)
- **FFAS** [207] (`http://bioinformatics.ljcrf.edu/FFAS/`)
- **BioInBgu** [206] (`http://www.cs.bgu.ac.il/~bioinbgu/`)
- **GenTHREADER** [205] (`http://insulin.brunel.ac.uk/psipred/`)
- **SAM-T99** [194](`http://www.cse.ucsc.edu/research/compbio/`)

After CAFASP-2 an automated consensus method was introduced [278], which tries to combine the best server predictions. Indeed consensus methods from various servers appear to yield better automated results. Augmenting these with usage of expert knowledge to select correct templates and manual inspection to improve the alignment will provide the best results currently attainable.

The different methods quoted above give a good idea of what is nowadays possible to predict. As will be presented in Chapter 11, all of them are among the top scoring methods in CASP-4. A brief presentation follows.

3D-PSSM [195] and GenTHREADER [205] are both structural profile methods incorporating elements of other approaches. Both add predicted secondary structure to augment the sequence information. Dynamic programming is used to align the target sequence against a set of pre-calculated profiles representing protein families. Differences exist in the calculation of profiles and the scoring scheme. Where GenTHREADER uses only sequence alignments to generate profiles, 3D-PSSM also uses structural alignments to increment coverage. GenTHREADER uses knowledge-based potentials commonly employed in threading to assign scores and computes the ranking through a neural network, giving a probability as output. 3D-PSSM instead calculates an expectation value and uses functional information from SAWTED [188]. This is a method that scans a database of scientific abstracts for textual keywords representing hints about

protein function. It is an attempt to mimic the approach of A. Murzin (see below).

The remaining three methods (FFAS[207], BioInBgu[206], SAM-T99[194]) use sequence profiles. The first two are mainly extensions of the ideas behind PSI-BLAST, with BioInBgu also using secondary structure. SAM-T99 instead scores the target sequence against a set of iteratively pre-computed *HMMs*. It was recently extended to include secondary structure information [314].

The fold recognition method developed in our group by E. Bindewald is called MANIFOLD [85][209]. It is a mapping method similar to the ideas of Russel et al. [191]. Inputs are the sequence, predicted secondary structure (from either PSI-PRED [279] or SSPRO [215]), predicted solvent accessibility (e.g. [231]), length and function. The latter is unique to MANIFOLD and is represented by the enzyme classification (*EC*) number. The fold library was augmented by finding out the *EC* number of all domains, where possible. A set of rules is used to filter out unlikely structures before ranking. During CASP-4 this was still a simple "pareto score", meaning that folds appearing in one of five subrankings (one for each criterium) were awarded one "point" per occurrence [85]. Since then, a more complex evaluation using Z-scores was implemented [209]. The output is a list of probable folds. Figure 6.9 shows a schematic representation of MANIFOLD.

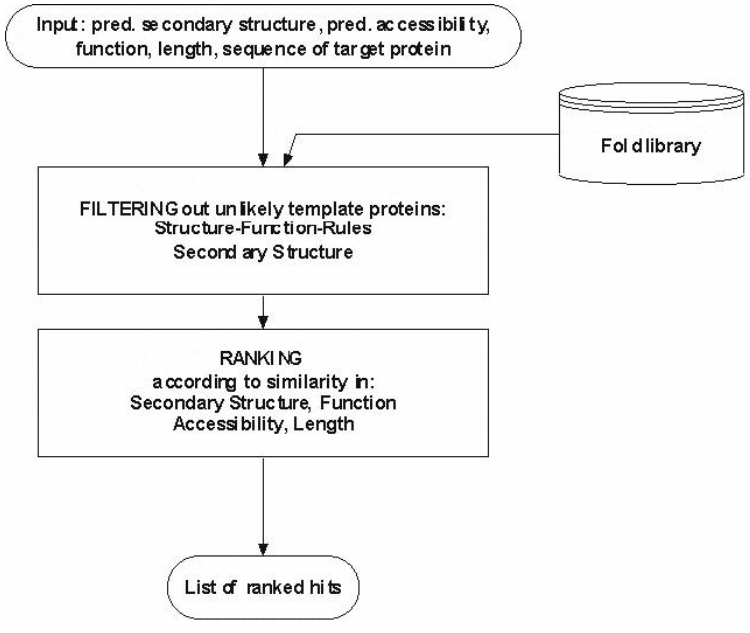

FIGURE 6.9. MANIFOLD flow diagram

No account of the state of the art in fold recognition would be complete without mentioning the phenomenon known as A. Murzin. The author of the

SCOP database [134] has "won" the fold recognition category in both the CASP-2 and CASP-4 experiments[4] with an approach that can well be defined as rather manual. The tools employed included only PSI-BLAST [128] and the SCOP [134] and PFAM [280] databases. Models were built by what can only be considered an exceptional display of expert knowledge. The commonly reproducible step in the knowledge employed by A. Murzin is to extract details about protein function and alignment of specific residues from the scientific literature. This has prompted the development of automated data mining methods such as SAWTED [188].

Perhaps more important is Murzin's knowledge of protein structures. This is something that cannot be readily described in algorithmic ways. It is what still makes the manual selection in fold recognition a determining factor, something reflected in the weaker performance of all automated servers in CAFASP compared to the corresponding CASP submissions.

6.4 Energy Functions

Energy functions are used in a variety of roles in protein modeling. An energy function precise enough to always discriminate the native protein structure from all possible decoys would not only simplify the protein structure prediction problem considerably. It would also increase our understanding of the protein folding process itself.

If feasible, one would like to use quantum mechanical models, being the most detailed representation, to calculate the energy of a protein. It can theoretically be done by solving the Schrödinger equation. This equation can be solved exactly for the hydrogen atom, but is no longer trivial for three or more particles. In recent years it has become possible to approximately solve the Schrödinger equation for systems up to hundred atoms with the Hartree-Fock or self-consistent field approximations. Their main idea is that the many-body interactions are reduced to several two-body interactions.

Two programs for performing quantum mechanical computations of molecules are e.g. TURBOMOLE [251] and GAUSSIAN 98 [252]. They can compute approximate solutions of the Schrödinger equation expressed as linear combinations of basis functions. However, the resulting energies in many cases show significant deviations from experimental data. For a more detailed treatment of quantum mechanical models see [94].

Currently the main problem with quantum mechanical models is the number of atoms for which an energy can be computed. This is limited to about 100 atoms, where most proteins consist of more than 1,000 atoms. In addition,

[4] He was the fold recognition assessor in CASP-3, and thus did not participate.

solvent interactions are not sufficiently approximated. Nor is it possible to incorporate such factors as temperature or ionic strength [94].

In absence of the computing power and other advances required to make quantum mechanical models appropriate for calculating the energy of proteins, two main alternative classes of energy functions have been investigated: force fields and knowledge-based potentials.

Force fields are empirical models that treat molecules as semi-classical systems relying on the Born-Oppenheimer approximation. This states that the energy of the molecule can be approximately calculated as a function of the coordinates of the nuclei. Force fields contain two types of interaction terms: bonded and non-bonded.

Bonded interactions typically include energy terms describing the deviations from bond length, bond angle and torsion angles. Others (e.g. improper dihedral angles) may be included. Non-bonded interactions are essentially represented van der Waals and electrostatic terms. Hydrogen bond or implicit solvation terms are also sometimes included. The most widely used force fields for proteins are AMBER [218], CHARMM [60] and GROMOS [219]. The functional forms representing the energy terms are quite similar [220], with differences mostly in parametrization [82]. For a more detailed representation refer to Section 8.1.

Force fields are still subject of research, as testified by a number of recent review articles (e.g. [225][226][227][253]). The focus of on-going research appears to be higher order parametrization techniques and implicit solvation models. Force fields have become more effective in recent years, partially due to the increase in computation power. Molecular dynamics studies, where the force field energy is used to infer trajectories of atom movements, are becoming feasible for longer simulations of protein folding.

The current state of the art in force fields may be best expressed by the following anecdote from the CASP-4 meeting. The final session was dedicated to energy functions with all major experts of the field present. The recent improvements in force field parametrization techniques were highlighted to postulate that free-energy calculations will soon be accurate enough to fully discriminate native structures. As one of the more pessimistic participants commented, "it has been stated for over ten years now that tomorrow we will have the ultimate force field, but this has yet to happen"[5].

Knowledge-based potential form an alternative approach to energy calculation. Instead of deriving a potential to capture the "local" structure of a molecule with a force field, which mainly represents interactions with its close

[5] public comment made on Dec 7^{th}, 2000 during the *Electrostatics and Hydrophobicity* discussion of the CASP-4 meeting in Asilomar (CA), USA.

neighbors, the "global" energy preference is inferred from solved protein structures. This "global" energy takes the form of a density function describing the probability of an amino acid being found at a given distance from another among the experimentally solved structures. The underlying theory, describing the relation between statistical preferences and energetics, is described in [177][184] [221]. A review comparing knowledge-based potentials and force fields can be found in [82].

This theory is general enough to allow application to all sorts of features that can be extracted from protein structures. The simplicity with which a potential can be derived has generated a large number of alternative implementations. These range from contact potentials [178][183] [224], where amino acids are simply described as a two-state system with a distance threshold, all the way to very detailed potentials trying to estimate the "global" energy of the system [78][85][176] [186].

Knowledge-based potentials have become increasingly important over the last couple of years, due to their capacity for abstraction in fold recognition problems. Relevant reviews are [180][201][222]. Where force fields present a rough energy surface, making calculations subject to local variations, knowledge–based potentials are known to present a more "stable" picture, indicating a general trend. More details can be found in Section 8.2.

6.5 Side Chain Placement

Side chain placement has developed into a sub-field of its own due to the relative independence from all other modeling steps. A good, albeit not overly recent, review of methods can be found in [157]. Side chains are almost always considered independently from the backbone, which is kept fixed. This is a computational simplification, since backbone flexibility is sometimes necessary to accommodate particular side chains in the native structure.

Another important approximation is given by the use of canonical structures, called rotamers. A strong preference for side chains to adopt specific torsion angles has been established [149][234][126][130] and used to derive rotamer libraries [119][141][158]. It has been argued that 5-30% of the amino acids, depending on type, do not conform to rotamers [235]. However, no evidence for an improved performance of continuum search methods was observed [150]. In practice, all newer side chain placement methods use rotamer libraries [157]. Recently, the concept of "flexible rotamers" has been proposed. In this approximation, each fixed rotamer is replaced by an ensemble (e.g. 1,000) of slightly different structures [120].

Different types of search methods for side chain optimization have been described. These range from local energy minimization [148][172], variations

of the Monte Carlo search [115][151][155][156] [162][169], genetic algorithms [119][160] and self-consistent ensemble optimization [114][124][168] all the way to systematic or combinatorial searches [118][129][147]. With the exception of systematic or combinatorial searches, which are computationally expensive, the other methods are not guaranteed to find the global energy minimum.

This situation was improved with the publication of the dead end elimination (*DEE*) theorem [141] and its subsequent extensions [140] [142][144]. A full overview of the *DEE* theory can be found in [108]. The theorem essentially reduces the combinatorial problem of finding the global energy minimum to a series of pairwise inequalities. Solutions that do not satisfy these inequalities can be removed from the solution space, typically reducing the search space by several orders of magnitude [142].

Publications concerning *DEE* in side chain placement [87][143][174] and its extensions [117][122][170] have proven to be very popular in recent years. Perhaps the only remaining problem is the search method to be used after the *DEE* has been performed. Larger proteins can still yield a considerable search space, making a combinatorial search impractical. Leach [117][122] has proposed to use the A^* search [92] to fill this gap. A more detailed description of *DEE* and A^* search can be found in Section 9.3.

As an alternative to energy based side chain placement, a simple statistical method, called *SCWRL*, was developed [125]. This heuristic method uses a backbone dependent statistical approach to place side chains in the most probable allowed conformation. It is both fast and produces quite good results, making it some kind of "base line" for more complex optimization algorithms such as *DEE*. It will be described in more detail in Section 9.2.

Side chain placement has also gained importance in protein engineering. Here, one is interested in designing artificial proteins that fulfill a particular role. This means asking the question "given a fold, which sequences will adopt it?". Energy–driven side chain optimization has proven to be effective at discriminating sequences compatible with a specific fold [161]. The general assumption is that sequences showing low energy will likely adopt the desired fold. This has led to the development of automated processes for sequence prediction [45][152][153] [154] [161][171].

Rotamers are used to keep the search space tractable, in addition to allowing every amino acid type at any position. It has been argued that the *DEE* theorem is the best choice for protein sequence design [121] as it is guaranteed to find the global energy minimum. A variation of the branch & bound algorithm, called branch & terminate, has been proposed to search the conformational space remaining after the *DEE* step [170]. This is quite similar to the A* search proposed for side chain placement in [117][122]. Achievements

of this kind of approach include the de novo design of a 30-residue zinc finger motif, composed of a $\beta\beta\alpha$-structure and two connecting turns [161].

6.6 Summary

The state of the art is described mostly in terms of what has been established in the CASP series of experiments. These blind tests for protein structure prediction take place every two years (CASP-4 in 2000) and have become the best way to assess methods that work consistently well. Being selected to speak at the CASP conference has also become a major source of scientific reputation.

Homology modeling is the type of prediction that yields the most accurate results. It can be applied to targets having at least 20-30% sequence identity to a known structure. The approach starts by scanning a sequence database of known structures for homologs and aligning these to the target. This step is fairly easy for sequences with high identity ($\geq$ 45%) but very difficult for low identity ($<=$ 25%). In fact, it is the primary source of errors.

Using PSI-BLAST to scan the database and align templates is the current de facto standard for this first step. Once a suitable target to template alignment has been defined, the 3D coordinates of structurally equivalent residues are copied. Structurally variable loops have to be predicted and are the second largest source of errors in homology modeling. Being a major focus of this thesis, they will be treated in Part III.

The structure is finalized by placing the side chain atoms and assessing model quality, with the possibility to perform small adjustments like limited energy minimization. The most successful methods (e.g. Sternberg [320] or Blundell [315]) are briefly discussed. However, differences between several methods tend to be less than 1.0 Å overall RMSD and may partially be due to the luck of selecting the most suitable template.

Fold recognition is an alternative approach for sequences without significant similarity to known structures. In addition to the sequence, structural information is used to augment the prediction. This includes predicted secondary structure and solvent accessibility. Some methods, mostly threading, also use 3D models to estimate the energy of a possible fold. Every fold recognition method requires four essential components: fold library, scoring function, alignment algorithm and ranking.

The fold library is typically a subset of the *PDB* extracted from a structural database (e.g. SCOP). The scoring function strongly depends on the method employed. Most combinations ranging from regular 20*20 amino acid substitution tables, extended matrices representing structural environment (e.g.

secondary structure class), position specific scoring matrices to uni- and multipositional knowledge-based potentials have been described.

Each fold in the library is aligned to the target sequence using the scoring function. For simpler scoring functions this can be done with dynamic programming. More complex scoring functions either require the "frozen" approximation (i.e. reduction to unipositional form) or different, and more time consuming, optimization methods. Ranking can be anything from sorting raw scores, to Z-scores measuring statistical significance and neural networks yielding a probability.

The most successful methods are either very sensitive sequence profile methods (e.g. FFAS [207], BioInBgu [206], SAM-T99 [194]), combine elements from structural profiles with threading (e.g. GenTHREADER [205]) or other sources of information (e.g. 3D-PSSM [195]). The fold recognition program of our group, MANIFOLD [209], uses the enzyme classification to improve prediction rate [211]. Use of information available in the literature is very important to explain the success of manual fold recognition methods, best exemplified by A. Murzin.

Energy functions are important to all aspects of protein structure prediction, as they give a measure of confidence for optimization. An ideal energy function would also explain the process of protein folding. The most detailed way to calculate energies are quantum mechanical methods. These are, to date, still overly time consuming and impractical. Two alternative classes of functions have been developed: force fields and knowledge-based potentials.

Force fields (e.g. AMBER [218]) are empirical models approximating the energy of a protein with bonded and non-bonded interactions, attempting to describe all contributions to the total energy. They tend to be very detailed and are prone to yield many erroneous local minima.

An alternative are knowledge-based potentials (e.g. [78]), where the "energy" is derived from the probability of a structure being similar to interaction patterns found in the database of known structures. This approach is very popular for fold recognition, as it produces a smoother "global" energy surface, allowing the detection of a general trend. Abstraction levels for knowledge-based potentials vary greatly, and several functional forms have been proposed.

Side chain placement has developed into an autonomous process, attempting to reproduce the position of side chain atoms based on statistical and/or energetic properties. A limited number of rotamer structures is found to approximate well the conformations of side chains in experimental structures.

Global energy minimization methods are greatly enhanced by the dead end elimination (*DEE*) theorem [141]. This reduces the search space several orders of magnitude by filtering out rotamers that cannot be part of the global

optimum. The remaining search space can be explored with A^* search [92] for example. Such methods are shown to perform well, albeit requiring some computation time. A heuristic method called *SCWRL* [125] forms a valid alternative based entirely on statistical preference of rotamers. It is very fast, but lacks the calculation of a real energy of the optimized system.

Energy optimization is required for another application of side chain placement: protein design. It has been demonstrated that a combination of *DEE*-based side chain optimization and experiments can produce novel sequences that fold into a particular 3D motif [161].

7
From Sequence to Model

The path leading from an amino acid sequence to a complete structural model can be traversed in several ways. This chapter introduces the general strategy followed throughout this thesis. It briefly addresses the steps that will be fully described later and highlights the main tasks at hand.

7.1 Approach: *Victor*

The project of this thesis was named *Victor* (*VIrtual Construction TOol for pRotein design*) to reflect the overall aim: to produce a state of the art method for protein structure prediction and modeling. This aim was of particular importance for the participation in the CASP-4 experiment, which allows a quantitative comparison with other methods.

In CASP-3 it was stated that the most serious problems in knowledge-based protein structure prediction (i.e. fold recognition and homology modeling) were template selection, alignment and loop modeling [2].

Template selection is of obvious importance, especially for "harder" fold recognition targets: Selecting a wrong template invariably leads to wrong results. This problem was addressed by E. Bindewald [85], who developed our group's automated fold recognition method MANIFOLD [211]. As we will see later in this section, it was supplemented by manual inspection using *ad hoc* knowledge of the protein.

The alignment problem is closely coupled with template selection. Therefore, the same programs usually perform both tasks. For "easy" and "medium" homology modeling targets this problem is adequately solved by using PSI-BLAST [128]. For "hard" homology modeling and fold recognition targets no

satisfactory solution is available yet, so again manual inspection is used to improve the alignment. One CASP participant said: "you make or break a good model with the alignment"[1].

Both problems described so far are too complex and not sufficiently understood to be fully automated if one seeks the best possible performance on single proteins. On the other hand, if the goal is to produce as many models as possible in short time (e.g. to predict all structures of an entire genome), there are satisfactory automated methods. Combining PSI-BLAST and a fold recognition method, e.g. MANIFOLD [85][211], enables the user to assign the many "easy" structures and highlight the difficult ones. These can then be modeled in more detail.

With this in mind, it was decided to focus the thesis mainly on the problems which arise **after** a template has been selected and an alignment generated. For the automated and fast prediction case, a state of the art protocol, PDB-BLAST, was implemented. This will be described in the following sections, together with the strategy followed in CASP-4 for the knowledge-based, manual template selection and alignment computation.

An outline of the general process implemented in this thesis is shown in the flow chart in Figure 7.1. This process resembles a "classic" homology modeling strategy: First information about the protein is collected and templates and alignments generated. This is then evaluated (automatically generated homology modeling targets are only modeled if above the "twilight zone") and a template selected. The alignment may be slightly adjusted before generating a model of the core residues, i.e. those residues that are assumed to be part of the conserved structure. The loop modeling procedure is used to fill in the remaining residues. Finally, the side chains are placed and the finished model evaluated. It is worth to note that the process was mostly performed manually during CASP-4 and automated afterwards, mainly based on the experiences acquired during the experiment.

Loop modeling was found to be one aspect of modeling protein structures, which is still problematic. This is in agreement with the previously cited CASP-3 evaluation [2]. Closing the gaps in alignments can prove to be quite difficult, especially for longer insertions.

In fact, it may be argued that the emphasis generally placed on the alignment could be a false problem: If one considers the case of strongly diverged structures it is frequently not possible to consider one part "correctly aligned" and the other "wrongly aligned". A "correct" alignment of structural fragments may simply be the lesser evil of two equally "wrong" structures. The

[1] personal communication

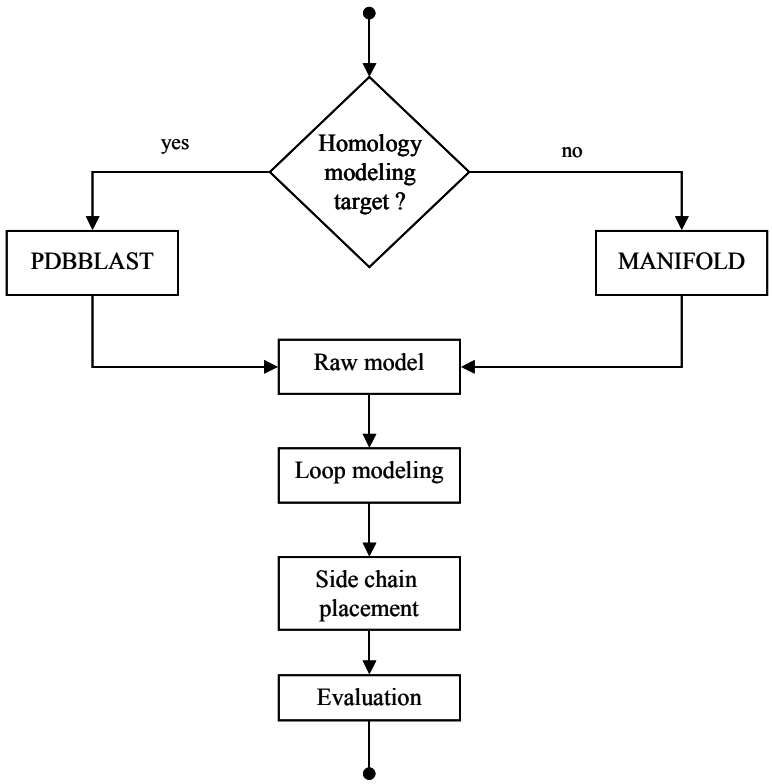

FIGURE 7.1. *Victor* flow diagram

question remains, however, how to position these fragments, since leaving them out will produce insufficient models for many fold recognition targets.

Adding flexibility to the modeling process can therefore help to solve this problem. A good loop modeling method is certainly a step in this direction. In particular, it is interesting to explore the possibility to perform loop modeling to place "undefined" parts of fold recognition targets. If it is possible to fill the gaps in these structures without increasing the overall RMSD, this would represent an important advance in modeling difficult structures. All of these thoughts were at the base of the decision to emphasize the loop modeling process throughout this thesis. A novel state of the art algorithm for fast *ab initio* loop modeling [210] was developed and will be described in detail in Part III.

7.2 Database Searches

Before the strategy used to select a template and corresponding alignment can be discussed, it is worth introducing what has *de facto* become the standard

method for searching sequence databases: The BLAST [131] algorithm with its extension PSI-BLAST [128].

Since its publication in 1990 by Altschul et al. [131] the BLAST algorithm has developed into the most widely used database search tool available. This success derives from the speed and relative accuracy with which BLAST is able to search even large sequence databases with hundreds of thousands of sequences in a matter of tens of seconds. In order to be used at maximum efficiency, the algorithm has to be understood. The inexperienced user may face a couple of pitfalls. The ideas behind the algorithm and the two main quality measures will now be described.

The BLAST algorithm consists of three main steps. These are schematically shown in Figure 7.2. The first step consists in preparing a look-up table of "words" (i.e. overlapping sequence fragments, typically 3 residues) that score above a threshold T according to the chosen scoring matrix. For proteins these "words" contain both the original sequence fragments themselves and all closely homologous amino acid combinations (according to the scoring matrix). This ensures maximum coverage of the look-up table for the subsequent step.

In the second step, each sequence from the database is scanned for **exact** matches with "words" from the look-up table. Since the look-up tables are complete, the task is simplified to exact matches, a computationally significantly less demanding task than searching similar "words". For sequences matching at least one such "word", the sequence position of the hit(s) is stored for the third step.

In this final step, each hit is extended in both directions of the sequence, until the score cannot be raised by adding more residues. These so-called maximum segment pairs (*MSP*) are compared to a threshold S. The *MSP* alignment is output for values greater than S. It is important to note that the MSP is not guaranteed to contain all residues that can be aligned. BLAST alignments tend to be truncated and are frequently slightly shorter than the optimal alignment. This is important in homology modeling where a few of $N-$ or $C-$terminal residues may be missed.

BLAST defines two important measures for evaluating an alignment. The first one is the normalized score S', which is said to be expressed in *bits*. The following formula is used to compute it:

$$S' = \frac{\lambda S - \ln K}{\ln 2} \tag{7.1}$$

S is the raw alignment score, calculated as the sum of alignment score at each position ($S = \sum_{ij} s_{ij}$). λ is a normalization constant and K is obtained from the Poisson distribution. This is obtained by considering the extreme value distribution. The normalized score allows to compare alignments between different proteins and estimate its significance.

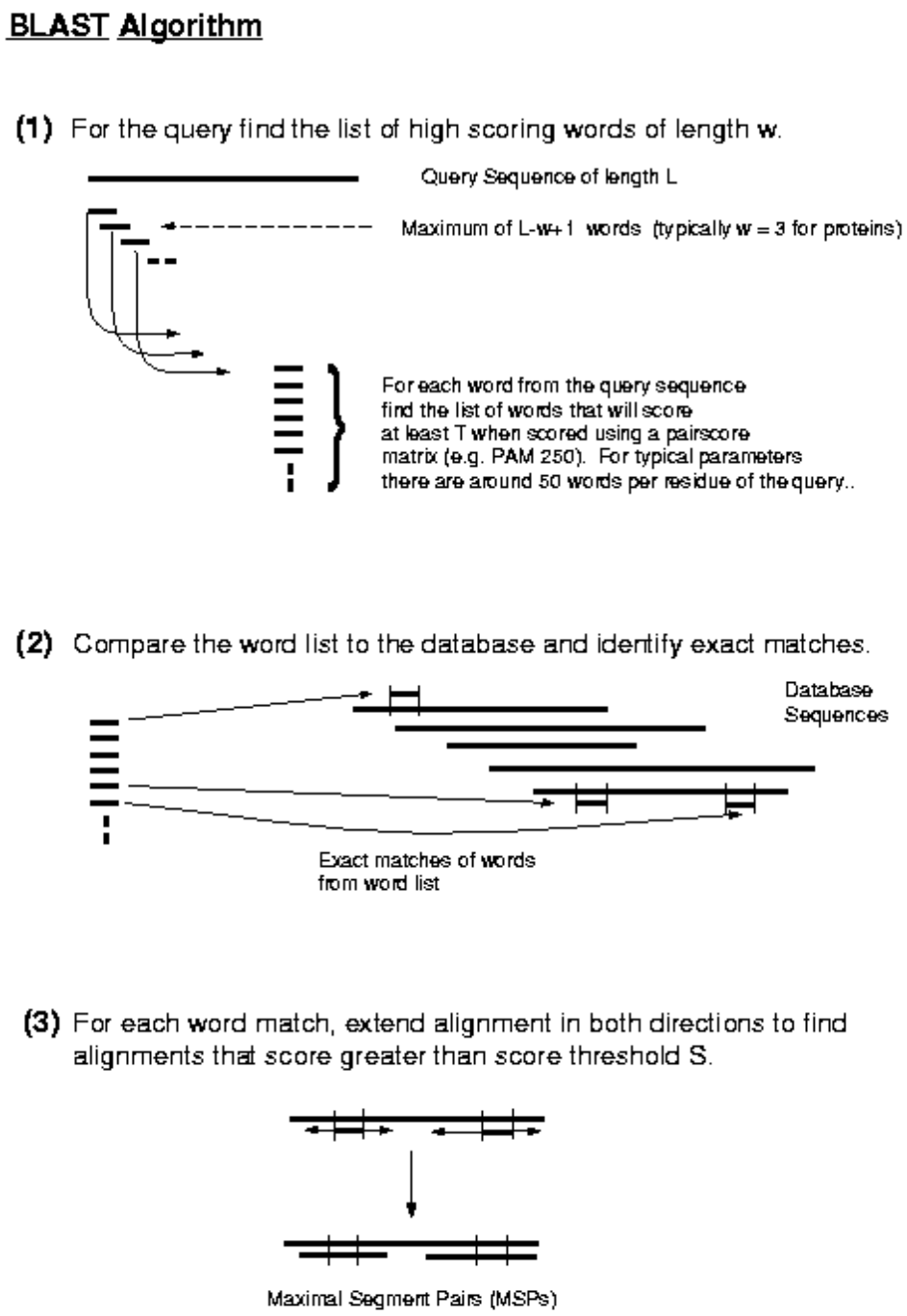

FIGURE 7.2. Schematic explanation of the BLAST algorithm.

Another, even more frequently used, measure of confidence is the expectation value (*E-value*). It estimates the probability of an alignment occurring by chance, i.e. between two non-related protein sequences. It is calculated as:

$$E = \frac{N}{2^{S'}} \tag{7.2}$$

where N is the search space (i.e. the size of the database) and S' is the normalized alignment score. The *E-value* offers a convenient way to assess the significance of an alignment. As a rule of thumb, and for searches against the PDB database, alignments with an *E-value* of less than 0.001 can be considered possible templates. For other sequence databases, containing many highly similar sequences, this value has to be lowered accordingly.

In a more recent paper Altschul et al. [128] describe an extension to the basic BLAST algorithm called PSI-BLAST (for *Position Specific Iterative* BLAST). In addition to a number of algorithmic improvements, which yield a program that runs three times faster and is more sensitive to detect alignments with weak similarity, an important new feature was introduced: *PSSM* (*Position Specific Scoring Matrix*). BLAST, which so far considered only information from pairwise alignments, was extended to sequence profiles, which *de facto* simulate multiple sequence alignments. Using information from multiple alignments regarding amino acid substitution probabilities at given positions in the sequence improves alignment quality [189].

PSI-BLAST combines the pairwise alignments obtained with the default scoring matrix to derive a *PSSM* that encapsulates the observed relative substitution frequencies. This *PSSM* can then be used to reiterate the database search and generally discovers additional weaker homologies. It also yields more robust alignments. No longer are these based on statistical assumptions about substitution probabilities, but rather on the observed values. The process can be repeated for a number of rounds until the search has converged, i.e. no new sequences are added.

The PSI-BLAST algorithm has become the de facto standard for searching homologies among protein sequences. It is both fast, with computation times in the order of tens of seconds, and very sensitive. Compared to standard BLAST its detection rate for distant homologues is significantly increased. It has recently even been found to outperform direct multiple sequence alignment methods such as CLUSTALW for low sequence identities [249].

7.3 Template Selection & Alignment

As has been already established (Section 7.1), two possible scenarios exist for modeling protein structures: Either the large scale modeling of hundreds or

thousands of structures or the single protein case, where one would like to include as detailed knowledge as possible about the structure to model. With this in mind, two alternative strategies have been developed throughout this thesis.

In the large scale modeling case, accuracy has to be traded for speed, and process automation is of paramount importance. Using a state of the art database search tool, which, if properly used, is known to produce very few false positives, seems an appropriate choice. This has been achieved by implementing the PDBBLAST protocol [207] for template and alignment selection.

PDBBLAST is frequently considered the "base line" for discriminating homology modeling from fold recognition targets [97]. It is a two-step protocol that tries to collect as much information as available on a protein family in order to improve the search of a matching homologous structure. In the first step, PSI-BLAST is used to search for homologous sequences in the non-redundant (*NR*) database. The *NR* database contains all publicly available protein sequences from Swiss-Prot, GenBank and TrEMBL, i.e. all known protein sequences. The PSI-BLAST search is iterated for a number of round, typically 4 or 5, to find a *PSSM* of the protein family. In the second step the generated *PSSM* is used to search the *PDB* database for matching sequences.

Templates found by PDBBLAST having a significant *E-value* (e.g. $E <= 0.001$) are known to represent the most probable fold. The corresponding alignment, although not always optimal, represents the best choice among a number of automated database search methods [136].

The opposite modeling case, where one wishes to include as much information as possible about a protein, is best represented by the CASP experiments. Automating this process is problematic, because the alignment frequently requires local adjustments to simplify modeling even for "easy" targets. For "hard" targets the selection of a suitable template can become the dominating factor [309]. In order to improve the perceived model quality a partially manual approach was chosen.

To this end, E. Bindewald developed the MANIFOLD [85] [211] fold recognition program, described in Section 6.3. This program searches for plausible folds mainly based on secondary structure and functional similarity in terms of enzyme code, where applicable. At the time of CASP-4, the program did not contain a reliable alignment module, so templates suggested by MANIFOLD were manually aligned using CLUSTALW [136]. The algorithm performs a global alignment of the input, always aligning the entire sequences regardless of domain boundaries. This has to be taken into account as we will see when discussing the results (Chapter 11).

For difficult CASP-4 targets, the templates suggested by MANIFOLD were compared with information from the publicly available CAFASP-2 server pre-

dictions. The SCOP classifications of MANIFOLD templates were manually checked against the most frequent CAFASP-2 server predictions. Templates appearing in both lists were marked as possible solutions and visually inspected. The final decision regarding template and alignment to use for modeling was based on a personal assessment by this author. It mainly included the perceived sequence, secondary structure and function similarities. The usefulness of this strategy will be discussed in Chapter 11.

7.4 Model Generation

The model generation step consists of transferring the information from the structural template(s) to the new structure. Automated and rule-based modeling procedures have been developed in order to reduce manual decisions, falling in two main classes [239]:

- use of restraints, such as interatomic distances, to construct models that best agree with the template(s)
- assembly of rigid fragments from the template(s)

Restraint-based methods use either distance geometry [242][243] or optimization techniques [185] to satisfy spatial restraints obtained from the alignment of the target sequence with the templates. The program *Modeller* [185] is the most widely used method of this category. It derives both interatomic distances (e.g. distances between structurally equivalent residues or hydrogen bond geometry) and torsion angle restraints (e.g. secondary structure, side chains) from the templates. Optimization is performed with the conjugate gradients method. The advantage of this type of approach is the flexibility of the constructed model. This is paid for by a rather time-consuming optimization procedure.

Fragment-based methods are still the most widely used ones [84], being faster and simpler to use than restraint-based ones. This is paid for by the rigidity of the produced models. They involve copying the atomic coordinates of known protein structures. These can be short fragments as demonstrated in [48][50] [53][167]. In practice the fragments are extracted from the template structure(s). It has been established that only the most accurate parts of the models are copied from the template structure(s) [240][241].

The main question in fragment-based methods is the number of template structures to include in the construction process. When using multiple templates, these have to be structurally aligned. Several methods are available for this task, e.g. CE [247], DALI [246] or SSAP [248]. An example of a structural alignment is shown in Figure 7.3.

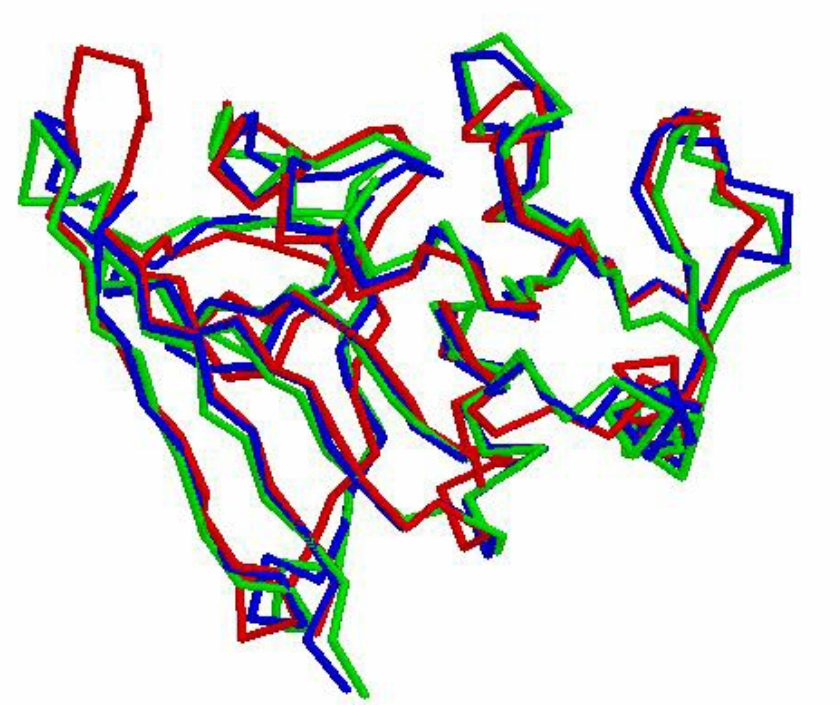

FIGURE 7.3. Sample structural alignment of dihydrofolate reductase (`3dfr`).

When using multiple templates it has to be decided, which residues are to be considered as equivalent. The coordinates of these can then be averaged and copied. Delimiting the structural core with equivalent residues can sometimes reduce the part of the structure that can be modeled directly. It has been found that using multiple templates slightly improves the overall model quality, but that many cases exist where fitting to a single basis structure improves results [244]. Alignment errors affect the overall model quality much more than the decision of the number of templates to use. Using a single template structure therefore remains the method of choice for many applications (e.g. CASP-3 [2]).

For this thesis, it was decided to use a fragment-based approach. Moreover, only single templates were used to build the model. This was done for two reasons. First of all, during CASP the templates and alignments undergo manual inspection to maximize the presumed model quality. Manually edited alignments generally refer to a single template structure.

Second, the benefit of multiple templates is limited to those cases where the templates show sufficient difference to warrant the risk of disturbing the geometry of a single template. Atomic coordinates in experimental structures of the same protein can vary up to about 1 Å [281]. This averaging between

templates can smear out the individual geometry, reducing overall accuracy. On the other hand, for distant homology or fold recognition targets, having strongly diverged, typically only a single structure acting as template can be defined.

It was therefore decided that using single template fragment-based modeling is sufficient for proof of principle in the present thesis. Extending the approach to multiple templates is not difficult, with the public availability of structural alignment methods such as CE [247]. After calculating a structural alignment for the template structures, the core residues can be identified using a RMSD cutoff (e.g. 3 Å).

7.5 Implementation: *Biopool 2000 & Homer*

The strategy outlined above has been implemented as part of this thesis. The classes and programs necessary to represent a protein structure and guide the process from automatic template selection to model building are divided in two packages: *Biopool2000* and *Homer*.

The base classes to represent a protein structure in the computer, with all necessary methods to manipulate it, are contained in *Biopool2000* (for *Biopolymer Object Oriented Library*, year *2000* version). This is a complete redesign of the classes used in [86], based on acquired experiences. It was necessary due to new requirements of the expanded scope of the project.

In particular, it should represent the polypeptide chain in a convenient way, including the possibility of reading a plain sequence or processing the structure in *PDB* file format. The latter was not possible with the version used in [86] due to the inherent representation of the atom coordinates.

Two alternative, somewhat contradictory, but nevertheless equally useful representations of atom positions exist in the literature. The Cartesian coordinates describe the atom position in terms of 3-D coordinates, i.e. relative to some arbitrary origin. Implementing this representation is trivial, in that it is what *PDB* files describe. Energy calculations benefit greatly from it, as the distances between atoms are apparent. However, it is quite difficult to modify the structure. E.g. changing a backbone torsion angle requires the immediate re-calculation of all subsequent atom positions, which can run into thousands, creating a substantial computational overhead.

The second alternative, internal coordinates, describes the atom position in terms of its relationship to previously positioned atoms. Rather than indicating the (global) 3-D coordinates, the structure is described in terms of bond length, bond angle and torsion angle. Three previous atoms are required to derive the 3-D coordinates as shown in Figure 7.4. This representation makes modifying the structure (e.g. changing a backbone torsion angle) very easy.

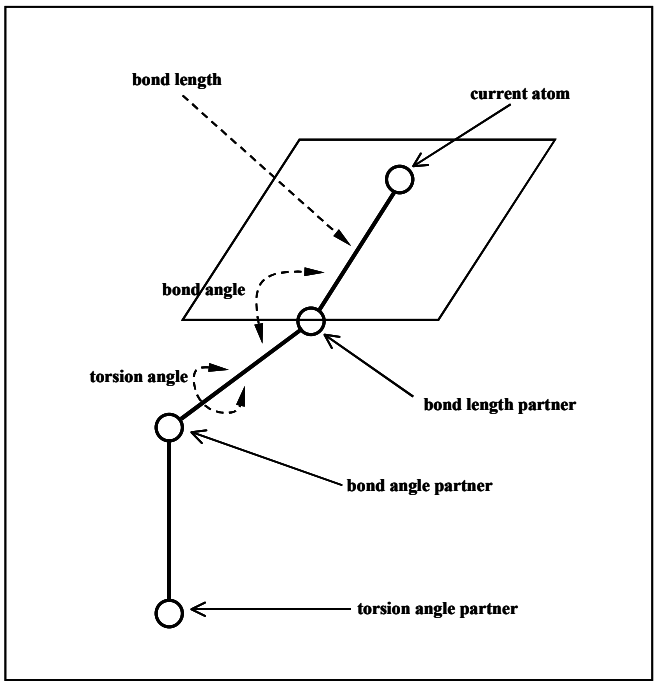

FIGURE 7.4. How internal coordinates are computed.

Calculating the 3-D position is more time consuming, as the internal coordinates of all atoms have to be converted first.

This dilemma was solved by choosing a structured double representation of the molecule in terms of global coordinates and local position using bonded atoms. Each atom has the global 3-D coordinates from the Cartesian representation. In addition, every atom position has the position (expressed as a translation and rotation) relative to a predecessor and an up-to-date flag. As long as the global 3-D coordinates are valid, the flag is `true`. When the structure is modified, this change is recursively communicated to the changed, and all following, atoms by setting the flag to `false`. The relative position is only changed for the first atom. Whenever the 3-D coordinates of an atom that is not up-to-date are requested, the molecule is tracked back to the first atom not being up-to-date and all atoms in-between updated. This treatment ensures that the minimum number of operations necessary are performed and only when needed. Since changes to the structure, e.g. torsion angles, are performed in groups, the updating process is delayed and performed only once.

The basis of the *Biopool2000* package are classes representing the physical entities involved in protein structure prediction, i.e. `Atom`, `SideChain`, `AminoAcid`. The latter two, being both groups of bound atoms, are derived from the class `Group` containing generic operations. A collection of bound amino acids, e.g. a protein, a domain or a fragment, is represented in the class

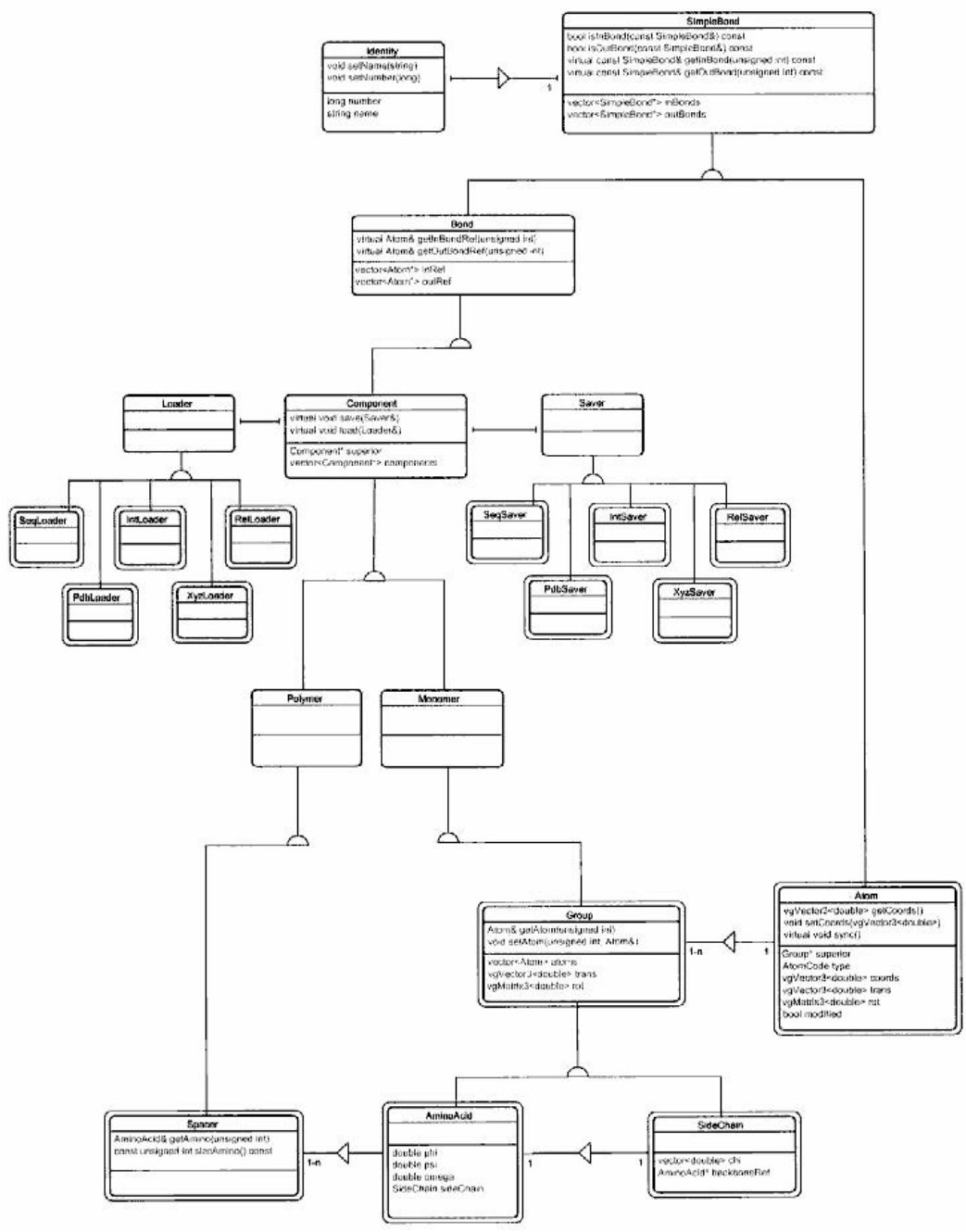

FIGURE 7.5. Class diagram for *Biopool2000*

`Spacer`. Usage of the "composite" pattern from Gamma et al. [95] ensures that a `Spacer` can be composed of `AminoAcid`, other `Spacer` or both types of objects. The only restriction is that the represented fragment be part of a single amino acid chain, i.e. only single N- and C-terminal residues are allowed.

A number of abstract base classes are required to fill roles in the design patterns. These are the `SimpleBond` and `Bond` classes to represent the covalent bonds between atoms and groups of atoms respectively. The `Monomer` and `Polymer` classes serve as base for the "composite" pattern, which enables the `Spacer` to contain other `Spacer` objects while ensuring that this cannot happen for single `AminoAcid` or `SideChain` objects.

The "visitor" pattern [95] is used to implement loaders and savers for various file formats. The most important ones are `PdbLoader` and `PdbSaver` to handle the standard *PDB* format. The remaining formats are rarely used, but can prove useful. In order to enable treatment of codes according to the *PDB* format, two more classes were implemented. `AtomCode` represents the atomic codes and can be used to intuitively query specific atoms in an amino acid (e.g. `CA` or `NZ`). The same functionality for amino acids is implemented in `AminoAcidCode` (e.g. `GLY` or `HIS`).

Last but not least, the functionality to query and modify bond lengths, bond angles and torsion angles can be found in `IntCoordConverter`. This class encapsulates the code required to perform the geometric conversions involved. It also contains methods for assembling protein fragments into larger structures and other operations requiring geometric transformations. A diagram describing the class hierarchy is shown in Figure 7.5.

Designing and coding the *Biopool2000* package occupied a fair part of the overall time required for this thesis. The functionality of this package, while at first glance appearing of limited use, has proven to considerably speed up the development of subsequent packages. Its flexibility has enabled the fast development and testing of energy functions and optimization methods.

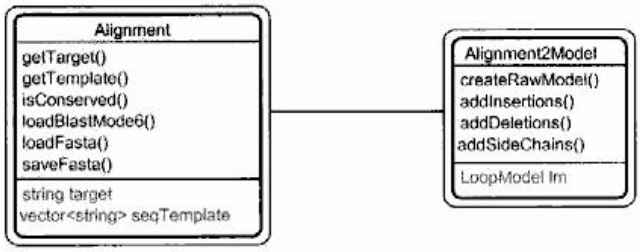

FIGURE 7.6. Class diagram for *Homer*

The functionality required to guide the model building process can be found in the *Homer* package. It embraces the single parts implemented in all the other packages described throughout this thesis. The program `homer` is a simple shell script that calls the `pdbblast` script, which implements the homonymous template and alignment strategy (see Section 7.3), retrieves the required *PDB* files and starts the model construction.

Retrieving the *PDB* files is a two step process. First, the program `ali2filter` is invoked to evaluate the proposed templates as produced by `pdbblast`. It uses a set of simple rule based filters to discard problematic templates. E.g. minimum sequence identity 35% and alignment length ≥ 20 residues. The allowed templates are written to `filelist` and parsed by the `getpdb` script, which retrieves the corresponding files from the *PDB* database.

The model building process is implemented in the `ali2model` program. This program accepts various parameters to influence model construction (e.g. disabling loop modeling or side chain placement). It loads the required information and sets up an `Alignment2Model` object that is used during the steps of model construction.

Two main classes are used internally to represent the data. The `Alignment` class, representing the homonymous entity, contains methods to load and save various specific alignment file formats (e.g. FASTA and BLAST) as well as to modify the actual sequence alignment. `Alignment2Model` encapsulates the calls to methods in other packages and offers a simplified interface for model construction, taking care of interna related to the other implementations. A class diagram is shown in Figure 7.6. The extension of the *Homer* package functioning as web server will be described in Chapter 10.

7.6 Summary

The most important decisions for building a rough model of a protein are discussed. As has been established in CASP-3 [2] for targets for which a similar structure can be found in the database, three main problems exist. Template selection and alignment generation are closely coupled and generally treated in the same step. Loop modeling, being more complicated than the former two, will be discussed in Part III.

For template selection and alignment computation, two alternative strategies were devised and implemented. The BLAST and PSI-BLAST algorithms were introduced in order to understand PDBBLAST, which is the presently best "base line" protocol for selecting sufficiently clear homologs. It allows to quickly model a large number of structures (e.g. genome-wide prediction). This protocol enhances the detection rate for PSI-BLAST [128] by adding as much sequence information as available.

In the case of detailed modeling of single proteins, best represented in the CASP-4 experiment, a combination of consensus predictions from MANIFOLD [85] and other CAFASP-2 servers and manual inspection was used. This serves to maximize the perceived quality of the template and alignment and is of paramount importance for difficult targets.

Once a suitable template and alignment have been found, a model of the conserved protein core is built. Two alternative techniques, fragment and restraint based, exist. Fragment-based modeling is considered better suited for this thesis and implemented in a single template fashion, by copying the atom coordinates of aligned residues.

The implementation of base classes to represent the physical entities (e.g. amino acid, protein, etc.) is described, with particular focus on the pitfalls created by alternative representations of atom coordinates. The implementation is found to allow the quick implementation of higher order functions, such as energy potentials and optimization algorithms. Finally, the program guiding and implementing the whole modeling strategy is also outlined.

8
Energy Functions

Energy functions are important to all aspects of protein structure modeling, generally serving as target functions for the optimization process. The various energy functions that were implemented and tested as part of the thesis will be addressed. These cover the whole range from force fields to knowledge-based potentials. Each of them has pros and cons, which make them particularly interesting for one application or another. A description of each of them, their specific use and implementation follows.

8.1 Force Fields

In absence of the computing power required to calculate the energy of proteins using quantum mechanical models, the so-called force fields have been developed. These are complex empirical models trying to approximate the actual energy of a molecule by representing all of its major contributions.

Force fields rely on the Born-Oppenheimer approximation: The energy of the molecule can be (approximately) written as a function of the coordinates of the atomic nuclei, since these are more than three orders of magnitude heavier than the electrons.

A simple force field contains the following energy terms [94]:

$$E(\mathbf{r}^N) = E_{bondlength} + E_{bondangle} + E_{torsionangle} + E_{non-bonded} \tag{8.1}$$

$$E_{bondlength} = \sum_{bonds} \frac{k_i^b}{2}(l_i - l_{i,0})^2 \tag{8.2}$$

$$E_{bondangle} = \sum_{angles} \frac{k_i^a}{2}(\theta_i - \theta_{i,0})^2 \tag{8.3}$$

$$E_{torsionangle} = \sum_{torsions} \frac{k_i^t}{2}(1 + \cos(n\omega - \gamma)) \tag{8.4}$$

$$E_{non-bonded} = \sum_{i=1}^{N}\sum_{i=1}^{N}\left(4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}\right) \tag{8.5}$$

$E(\mathbf{r}^N)$ denotes the potential energy which is a function of the positions ($\mathbf{r}$) of N atoms. The various contributions are schematically represented in Figure 8.1.

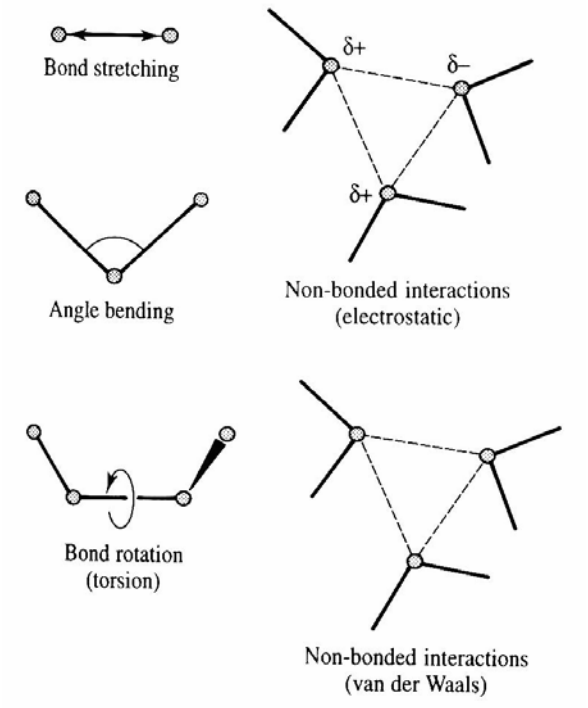

FIGURE 8.1. Schematic representation of the key contributions to a force field.

The first three terms ($E_{bondlength}$, $E_{bondangle}$, and $E_{torsionangle}$) describe the bonded character of atoms and the deviation from equilibrium values in bond lengths, bond angles, and torsion angles. The equilibrium values are measured experimentally. For bond lengths and bond angles, this is modeled as a harmonic potential with minimum at the equilibrium values $l_{i,0}$ and $\theta_{i,0}$ respectively. Torsion angles, being limited to values between $-180°$ and $+180°$, are modeled with the cosinus function. The non-bonded term, $E_{non-bonded}$, is calculated for all atoms that are separated by at least three bonds. In a simple force field, it usually contains a 12–6 Lennard-Jones potential for van der Waals interactions and a Coulomb potential for electrostatic interactions, schematically represented in Figures 8.2 and 8.3.

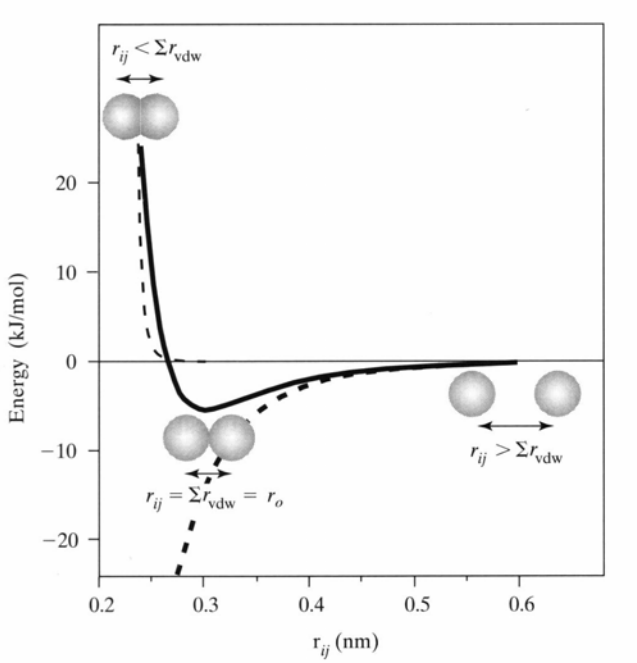

FIGURE 8.2. The Lennard/Jones potential. It is a sum of short-range attraction between atoms and extremely short-range steric repulsion. Together, they define an optimum distance r_0, which is the sum of the van-der-Waals radii.

The force field specific parameters are the energy contributions k_i^b, k_i^a, k_i^t, σ_{ij}, ε_{ij}, ε_0 and the partial charges q_i. Often different force fields are very similar in functional form, but differ in their parametrization. Two main approaches have been developed for obtaining the parameters [82]. The first involves fitting the parameters to those observed in small molecules. More recently partial charges for electrostatic interactions been obtained from quantum mechanical calculations on model compounds.

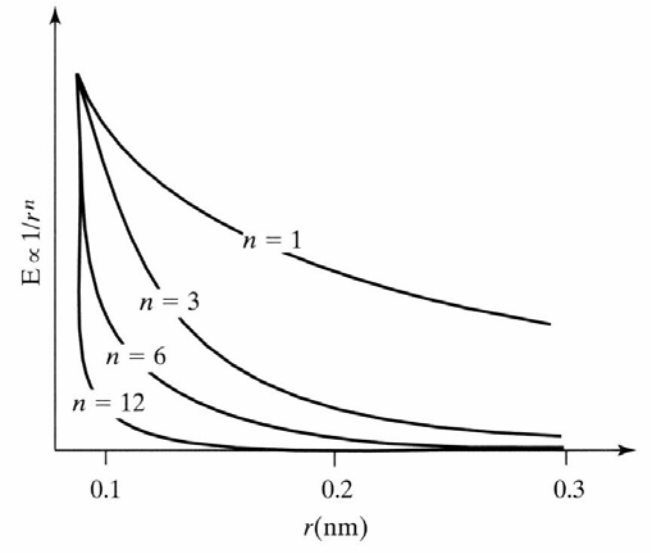

FIGURE 8.3. Energy functions inversely related to the distance (r). Electrostatic interactions typically have an $n = 1$ relationship. Higher powers are used to model shorter-range interactions. E.g. steric repulsion ($n = 12$).

The strategy of fitting force fields parameters by fitting as many properties of simple systems as possible has resulted in the emergence of many slightly different parametrizations and functional forms [82]. The most widely used force fields for proteins are AMBER [218], CHARMM [60] and GROMOS [219]. A review of functional forms can be found in Hünenberger & van Gunsteren [220]. Other reviews covering force fields are [225][226][227][253].

At the beginning of the present thesis, it was established that a C++ implementation of one of the major force fields was available for AMBER as part of the program QMOLVIEW of S. Siebert [329]. Therefore, considering the relatively minor difference in parametrization, it was decided to use this as basis for implementing a force field.

It was also decided early on to use only the non-bonded energy term and to use the *rigid rod* representation. This approximation keeps bond lengths and bond angles in idealized values. The energy of different allowed torsion angle combinations is assumed to be constant. The underlying assumption is that deviations from the equilibrium values are of minor importance and can be optimized in a later step. The following two terms of the AMBER force field are:

$$E_{AMBER,vdW} = E_m \left[-\left(\frac{r_m}{r_{ij}}\right)^{12} + 2\left(\frac{r_m}{r_{ij}}\right)^{6} \right] \tag{8.6}$$

$$E_{AMBER,estat} = \frac{C}{\varepsilon} * \frac{q_i q_j}{r_{ij}} \tag{8.7}$$

$E_{AMBER,vdW}$ is a classical Lennard-Jones potential representing van der Waals interactions between non-bonded atoms. E_m is the energy minimum reached at distance r_m between atoms i and j. $E_{AMBER,estat}$ is an electrostatics term derived from Coulomb's Law. $C = 332.05382$ is the Coulomb constant and ε the dielectric constant. $\varepsilon = 80$ in water, $1 <= \varepsilon <= 5$ in proteins. AMBER sets $\varepsilon = 1$.

The implementation of the AMBER non-bonded potential can be used to optimize the structure of proteins. Due to the relatively high level of detail, it was shown to perform particularly well for side chain placement. Here, the energy function has to be very sensitive to local changes in order to select the best local side chain conformation. For the optimization of larger structures, i.e. loops and whole proteins, the force field is less suited than the following functions. The energy surface of AMBER contains a number of local optima, making global optimization more difficult than using simpler energy functions.

8.2 Knowledge-based Potentials

As with many aspects of protein structure prediction where theoretical information is insufficient and/or impractical (e.g. computationally intractable), the possibility to derive an energy function from observed ("real") structures was investigated. Since the first work of Sippl in 1990 [184], this approach has developed into a field of its own, with many published articles describing variants of the basic theme.

The term "knowledge-based potential" is frequently used for expressions like [177]:

$$energy = -\ln\left(\frac{P_{observed}}{P_{expected}}\right) \tag{8.8}$$

or even

$$energy = -k_B T \ln\left(\frac{P_{observed}}{P_{expected}}\right) \tag{8.9}$$

The factor $k_B T$ implies a connection to statistical mechanics and is used to re-scale the energy in order to produce "realistic" values. $P_{observed}$ and $P_{expected}$ are the observed and expected probability (or frequency) of an event, e.g. two

atoms being at distance of k Å of each other. This formalism has been applied to a variety of "potentials" that attempt to measure different properties of protein structures to allow the prediction of plausible folds. (A special class of knowledge-based potentials will be treated within Section 8.3)

The idea behind knowledge-based potentials was introduced by Sippl in [184] and the theory further refined in [221]. Given two positions i and j in the structure at a given sequence separation $k = j - i$ and a given 3D distance s, the potentials specify the compatibility value of matching two amino acids from the sequence into positions i and j of the structure. A set of pairwise potentials was derived from statistical analysis of known protein structures. It relates the probability density of specific characteristics (i.e. atomic distances) to the energy of the system in equilibrium using statistical physics. The interested reader may find an excellent review of the principles of database potentials in [222].

The potentials $\Delta E_f^{ac,bd}(s)$ are calculated from a database of known protein structures. For specified atoms c and d in a pair of residues a and b the variable s represents the distance between atom c and b, respectively, and f is the separation of a and b along the amino acid sequence. The potential is given by following the expression:

$$\Delta E_f^{ac,bd}(s) = -kT * \ln \left[\frac{g_f^{ac,bd}(s)}{g_f^{c,d}(s)} \right] \tag{8.10}$$

The functions $g_f^{ac,bd}(s)$ represent the relative frequency of c and d in the distance interval corresponding to s. The reference state $g_f^{c,d}(s)$ represents the relative frequency of c and d as a function of s averaged over all amino acid pairs. k is Boltzmann's constant and T is the absolute temperature.

Once the potentials $\Delta E_f^{ac,bd}(s)$ are compiled from the database, the energy $\Delta E(S, C)$ of a given sequence S of length L with respect to a particular conformation C can be computed:

$$\Delta E(S, C) = \sum_{ij} E_f^{ac,bd}(s_{ij}) \tag{8.11}$$

where a, b, c, d, and k are functions of the atom indices i and j.

This class of energy function was successfully applied to a variety of optimization problems in protein structure prediction. E.g. discrimination between native and non-native folds [78][186], side chain placement [123], loop modeling [15][210], ab initio [85] [322], fold recognition [192][193][205], homology modeling [123], to name just a few. In fact, many of the mot successful approaches used in CASP-3 employed this type of energy function [2]. The relative ease

of implementation and efficacy of knowledge-based potentials has produced a wave of different publications on the subject, with many variations on the basic theme (e.g. [78][85][176] [178] [183][186] [224]). Relevant reviews can be found in [82][180][201] [222].

A commonly employed simplification of the knowledge-based potential are the amino acid based contact potentials [178][183][224]. The scoring function is computed using a $20 * 20$ contact matrix. The elements m_{ab} of the contact matrix represent the contribution when an amino acid of type a is in contact with that of type b. The total energy is a sum of the pairwise interactions:

$$E_{total} = \sum_{i=2}^{N} \sum_{j=1}^{j<i} E_{ij} \tag{8.12}$$

$$E_{ij} = \left\{ \begin{array}{ll} m_{type(i)type(j)} & \text{if } i \text{ and } j \text{ in contact} \\ 0 & \text{otherwise} \end{array} \right\} \tag{8.13}$$

Two amino acids are frequently considered in "contact" if the distance between at least one of their atoms (excluding hydrogens) is closer than 4.5 Å [178][224]. Another definition sets the contact threshold for C_α atoms to 8.5 Å [183]. In general, this type of contact potential is sufficiently detailed for *ab initio* or fold recognition prediction. However, it is too coarse for applications requiring discrimination between many very similar structures, such as homology and loop modeling.

Most simplifications of the knowledge-based potential lack the continuous nature required for assessing local variations of a structure and are likely to be too coarse for modeling applications. It was therefore decided to investigate more complex implementations. The potential had to be detailed enough to allow significant discrimination between similar structures and loops.

A potential satisfying these requirements is the residue-specific all-atom probability discriminatory function (RAPDF) of Samudrala and Moult [78]. In their paper, it was tested against five different decoy sets (i.e. sets of non-native structures) covering the whole range from very close to distant structures, in terms of C_α RMSD. The results show that the RAPDF potential is able to discriminate all correct conformations in three out of five test sets. In addition, it was also tested against a decoy set of loop structures, being able to discriminate a conformation within 1.0 Å of the lowest available in 10 out of 11 cases [78]. One key advantage of this potential is the possibility to download the parameter files from the PROSTAR website [217]. Together with the ease of implementation it constitutes a very good candidate for the optimization required as part of this thesis.

Samudrala and Moult [78] derive their knowledge-based potential from a statistical analysis of 265 protein structures with an X-ray resolution of at least 3.0 Å and less than 30% pairwise sequence identity. The latter is necessary to reduce the bias towards proteins with many representatives in the *PDB*.

Instead of using the mean force potential approach described above, the RAPDF potential is derived from the equivalent statistical formula. According to the authors this avoids the less clear assumptions made in the physics-based approach. Seeking the conformation with the lowest energy is formally equivalent to seeking the conformation with the one with the largest probability in terms of Bayesian statistics [78]. The following formula is thus introduced to calculate the conditional probability[1] (i.e. "energy"):

$$S(d_{ab}^{ij}) = -\sum_{ij} \ln \frac{P(d_{ab}^{ij}|C)}{P(d_{ab}^{ij})} \tag{8.14}$$

d_{ab}^{ij} is the distance between atoms i and j of type a and b, respectively. $P(d_{ab}^{ij}|C)$ is the probability of observing d_{ab}^{ij} in a correct structure. $P(d_{ab}^{ij})$ is the probability of observing such a distance in any structure, correct or incorrect. In order to calculate the conditional probability $S(d_{ab}^{ij})$, the distributions $P(d_{ab}^{ij}|C)$ and $P(d_{ab}^{ij})$ for all combinations of atom types at all observed distances are required. The former can be extracted from an analysis of the *PDB* structures:

$$P(d_{ab}|C) = f(d_{ab}) = \frac{N(d_{ab})}{\sum_d N(d_{ab})} \tag{8.15}$$

With $N(d_{ab})$ being the number of observations of atom types a and b in a particular distance interval d. The frequencies $f(d_{ab})$ obtained from the *PDB* are used as an approximation for the probabilities.

The distribution $P(d_{ab})$ is a prior distribution in terms of Bayesian statistics. It represents the probability of seeing a separation d between atom types a and b in any possible structure, correct or incorrect. Many choices for this are possible. The authors approximate the probability of finding atom types a and b in a distance bin d in *any* compact conformation as the probability of seeing *any* two atom types in a distance bin d:

$$P(d_{ab}) = \frac{\sum_{ab} N(d_{ab})}{\sum_d \sum_{ab} N(d_{ab})} \tag{8.16}$$

[1] Strictly speaking this is not a "probability" but a "likelihood". The term "probability" is used, because Samudrala and Moult [78] use it in their publication.

With the formulas to calculate the conditional probability (i.e. "energy") defined, the decision remains which atom types to include and how to partition the possible distances into intervals. Samudrala and Moult show that differentiating between all heavy atoms (i.e. non-hydrogens) of all amino acid types yields the best results. They divide the distances in 18 bins of 1 Å, with the first covering 0.0-2.0 Å. Considering distances up to 20.0 Å is shown to improve overall discrimination. For a more detailed description of the assumptions underlying the conditional probability formalism and other implementation details refer to [78].

The implementation of the RAPDF potential as part of this thesis was found to work well for optimizing whole proteins and especially in loop modeling. The correlation between RAPDF "energies" and constructed loop segments allows a good discrimination of the most plausible prediction. The potential was observed to differentiate well between alternatives, yet at the same time does not have such a rugged energy landscape as force fields, which would limit its ability to select the best solution among a set of non-native structures.

8.3 Solvation Potentials

Both classical force fields and knowledge-based potentials disregard an important effect contributing to the stability of protein structures: solvation. Proteins do not exist in vacuum but are immersed in aqueous solution. Interactions between amino acids on the protein surface and the solvent are known to affect the conformation of the residues. It has also been argued that solvent interactions strongly discriminate between native and misfolded structures [197]. Nevertheless, solvation effects are still considered a major challenge in molecular modeling [94].

Exact calculations of the solvation energy are not yet solved and few papers describing applications of such complex calculations for protein structure prediction have been published. One such application of a complex energy function for loop modeling was published by Rapp and Friesner [29]. However, the authors pointed out that results were so far only anecdotal and cannot yet be generalized [29].

To date, estimation of solvent interactions is to date mostly limited to simple models centered around the calculation of the solvent-accessible surface of the protein structure, which consists of the parts of the protein in contact with water molecules. The exact definition was formulated by Lee & Richards in 1971 [216]. Figure 8.4 shows an example for a solvent-accessible surface.

An empirical solvation model was proposed by Eisenberg & McLachlan in 1986 [203]. They assume that the solvation energy ΔG_s can be calculated as the sum of single atom contributions, with the latter being a function of solvent

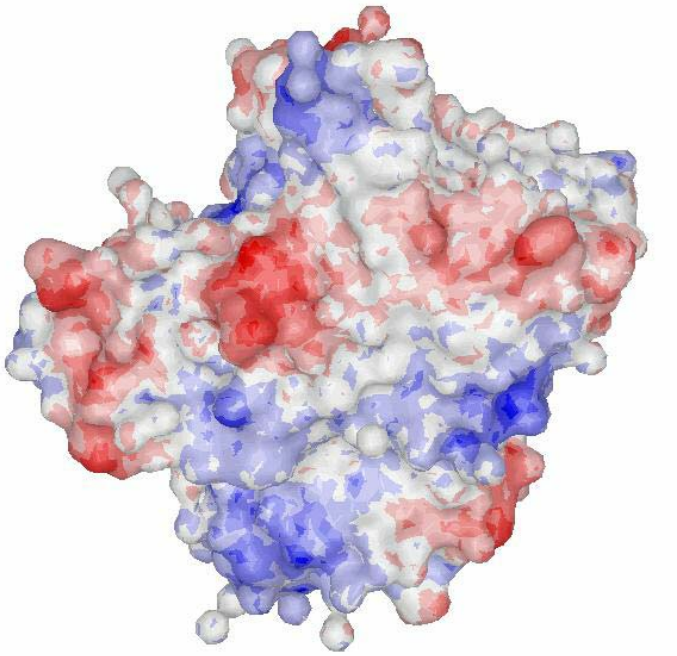

FIGURE 8.4. Sample solvent accessible surface of `3dfr`, colored by electrostatics.

accessible surface. Five classes of atoms (C, neutral N and O, O^-, N^+, and S) are defined and the solvation parameters $\Delta\sigma$ derived from experimental protein structures. The solvation energy can then be calculated as:

$$\Delta G_s = \sum_{i=\, C,N/O,O^-,N^+,S} \Delta\sigma\ (i) * \sum_{\nabla i} (A_i - A_i^r) \tag{8.17}$$

A_i is the solvent-accessible surface area of an atom in the current (i.e. folded) structure and A_i^r is that in the reference state. In the usual way in which the equation is used the reference state does not matter, as energies are only compared among each other.

The model is still too detailed and time consuming to calculate for many applications where a large number of energy calculations have to be performed in little time. This is definitely the case for fold recognition and threading, and so Jones et al. introduced a further simplification in 1995 [187]. Instead of calculating the contribution of single atoms, whole residues are treated as a single entity. The solvent-accessibility is calculated using DSSP [223] and a knowledge-based approach is used to derive a statistical solvation energy ΔE^a_{solv} from the observed frequencies:

$$\Delta E^a_{solv}(r) = -RT \ln \left(\frac{f^a(r)}{f(r)} \right) \tag{8.18}$$

r is the percent residue accessibility (relative to the fully extended $GGXGG$ pentapeptide), $f^a(r)$ is the frequency of occurrence of residue a with accessibility r, and $f(r)$ is the frequency of occurrence of all residues with accessibility r. After statistical analysis, five classes of relative accessibility were defined ($r < 12\%$, $12\% <= r < 36\%$, $36\% <= r < 44\%$, $44\% <= r < 87\%$, and $r \geq 87\%$). RT is taken to be 0.582 kcal/mol [187].

In 1999 Jones reports a further simplification of the solvation potential ΔE^a_{solv}[205]. Instead of calculating the more time-consuming solvent-accessibility, the number of C_β atoms falling within 10 Å of a residue's C_β atom is counted. This value replaces the previous one as frequency of occurrence in the above formula. Jones reports this new parameter to correlate well (correlation coefficient >0.85) with the relative solvent-accessibility of a residue [205].

Due to the simplicity of Jones' solvation potential ΔE^a_{solv}, it was decided to implement it as part of this thesis to overcome some of the drawbacks of previous energy functions. It was observed that force fields and especially knowledge-based potentials are less sensitive when considering errors and shifts in alignment. For example let us consider an alignment shifted by a single residue. All energy functions will detect an increased energy. The effect on force fields and knowledge-based potentials, both averaging over a large number of contributions, may be quite modest and unspecific. However, the solvation potential will invert the energy contribution, clearly indicating an error. This effect of the solvation potential is particularly interesting for improving the alignment in homology modeling.

8.4 Implementation: *Energy*

The energy functions presented in this chapter are fundamental for all optimization procedures performed as part of this thesis. The main requirement was flexibility and a general applicability in the context of the protein classes implemented in the *Biopool2000* package. It should be possible to invoke the energy calculation with any structure from all programs. At the same time the parameters of the energy models had to be stored externally to allow their rapid modification.

With this considerations in mind, the package *Energy* was designed to collect the classes and programs dealing with energy calculation. The main design decision was to use the "strategy" design pattern from Gamma et al. [95]. The abstract class `Potential` was defined to provide a common interface for energy calculation. It contains the necessary methods to load the energy parameters during initialization of an object. Computing the energy value for objects of the `Atom` and `Spacer` classes as well as a combination of both is allowed.

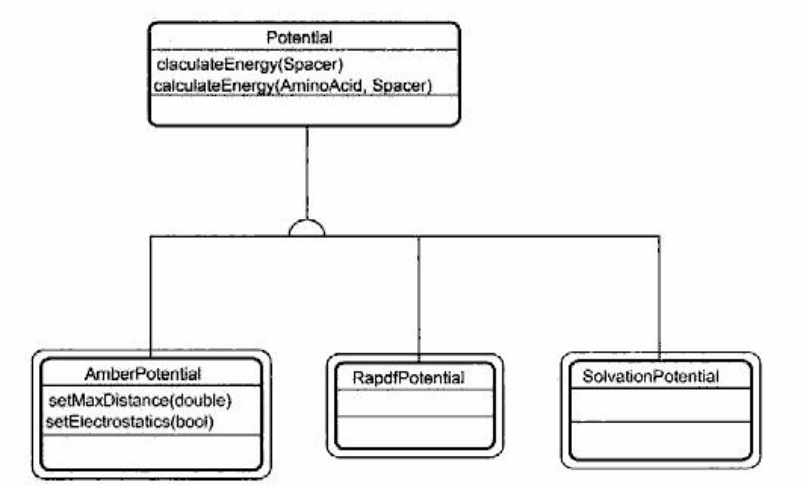

FIGURE 8.5. Class diageram for *Energy*

Three classes inherit the functionality of the base class: `AmberPotential`, `RapdfPotential` and `SolvationPotential`. The corresponding class diagram is shown in Figure 8.5.

The AMBER force field described in Section 8.1 is implemented in the class `AmberPotential`. This class was ported from the QMOLVIEW program of S. Siebert [329]. It uses the same parameter file as in the other program, called `AMBER.prm`. The actual energy calculation is limited to summing over the contribution of pairwise atom interactions. A maximum distance cutoff can be set with `setMaxDistance(double);` default is 10.0 Å. The electrostatics term can be switched on or off by invoking the method `withElectrostatics(bool)`.

The knowledge-based potential of Samudrala and Moult [78], as described in Section 8.2, is implemented as `RapdfPotential`. The calculation is limited to adding the pairwise interactions defined in the parameter file `ram.par` for the protein(s) of interest. It was downloaded from the Prostar web site [217] and contains the original values used by Samudrala and Moult [78].

The Jones solvation potential (Section 8.3) is contained in `SolvationPotential`. The same definition described in [205] is implemented. Since the frequency of occurrence for the twenty amino acids is not publicly available, this had to be re-calculated. The proteins from the PDBSELECT-95 list [75] were examined and the relative frequencies extracted. The corresponding parameter file is `solv.dat`. Since the solvation energy is only defined for structures containing multiple residues, it is only possible to calculate the energy for a `Spacer` or interaction between `AminoAcid` and `Spacer`.

In order to test the energy functions and to allow a quick examination of *PDB* structures, a program called `pdb2energy` was implemented. It calculates

the energy according to all three functions. The results returned can be readily parsed with Perl scripts to process large numbers of structures.

8.5 Summary

Energy functions serve as target functions for all kinds of optimizations related to protein structures. In order to produce good results, three different energy models were implemented during the thesis. All having their particular pros and cons. The two main alternative classes of energy functions for proteins, force fields and knowledge-based potentials, are introduced. This serves to motivate the choice of the implemented functions.

The non-bonded terms of AMBER [218], consisting of van der Waals and electrostatic interactions, were implemented in order to have an empirical force field. This is well-suited to local optimizations, such as side chain placement. It is, however, not equally well-suited to select conformations of larger fragments, such as loops or alternative templates. To overcome these limitations, especially for loop modeling, a knowledge-based potential was implemented. The knowledge-based RAPDF potential of Samudrala and Moult [78], whose parameters are publicly available, is able to predict good loop conformations and discriminate near-native structures. Selection of good alignments has benefitted from the implementation of a simple knowledge-based solvation potential, following the indications of Jones [205]. All three energy models were derived from a common interface, allowing the quick implementation and testing of more functions.

9
Side Chain Placement

Side chains are usually the last part of the protein structure to be modeled, because they are of minor importance for the backbone position. While the backbone may displace to accommodate different side chains, this process is not well understood and too computationally expensive to be included. Side chains are therefore considered detached from the backbone and need to be placed all at the same time. Even if the structure is well conserved, the side chain conformations tend to rearrange, forming a different pattern in the protein core. The arrangement of the side chains is, however, important to infer the function of a protein.

As has been established in Section 6.5, several different approaches for side chain placement have been described in the literature. These can be broadly classified in heuristic and deterministic optimization methods. One of the best performing approaches of both classes has been implemented as part of this thesis.

9.1 Rotamers

One of the first studies regarding the conformation of amino acid side chains was performed by Janin et al. in 1978 [234]. Plotting the distribution of observed side chain conformations from the *PDB* depending on χ torsion angles revealed the preference for certain χ angle combinations. This was confirmed, and the ranges tightened, by a similar study from Ponder and Richards in 1987 [149]. This limited number of canonical shapes is usually called rotamers.

Rotamers tend to correspond to low-energy structures in single-residue models [234]. Their conformation can be explained by the conformational analy-

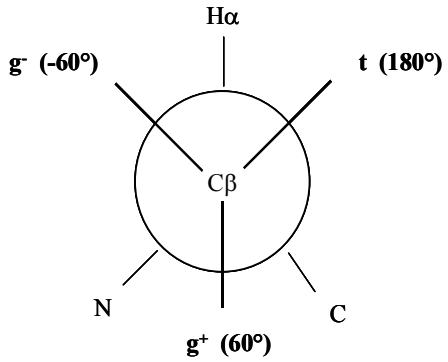

FIGURE 9.1. Position of the three canonical rotamers relative to the backbone, as seen from the C_α-C_β axis.

sis of hydrocarbons [130]. In tetrahedral carbon atoms, the atoms indirectly bonded to both sides of the central atom have a lower free energy if they are as distant as possible from each other. This is shown in Figure 9.1. This effect is also modeled in force fields by the torsion angle term and yields three preferred torsion angle states: t ($\chi = 180°$), g^+ ($\chi = +60°$) and g^- ($\chi = +60°$). Using related arguments, it is further possible to describe different preferences for subsequent χ torsion angles depending on each other [130].

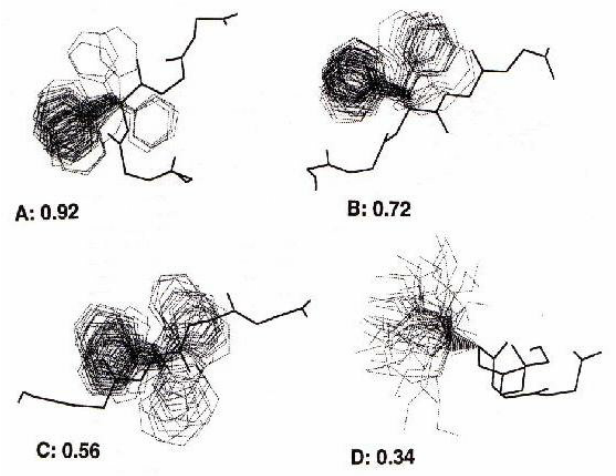

FIGURE 9.2. Rotamer probabilities. Four different distributions and the associated probability of the residue to be in the most probable rotamer are shown.

These idealized rotamer states are strained in actual protein structures. Real conformations are found to differ slightly depending on amino acid type and torsion angle combinations. An analysis of the real preferences was performed by Dunbrack [126][130][158]. The results are used to derive two different defi-

nitions of rotamer libraries (i.e. collections of densely populated states). The backbone independent rotamer library averages over all possible backbone φ, ψ torsion angle combinations and considers only χ_1, χ_2 interactions to derive probabilities for the single rotamer states. The backbone dependant library also considers φ, ψ torsion angles in steps of 20° to deliver a more detailed picture of rotamer probabilities. Both rotamer libraries developed by Dunbrack are freely available on the web and are used as part of this thesis. Other, more limited, rotamer libraries were developed for example in [119][141][149]. A sample probability distribution is shown in Figure 9.2.

The validity of the rotamer approximation has been examined in some detail. A considerable portion of side chain conformations, as observed in well resolved protein crystal structures, clearly does not conform to canonical rotamer structures [151][235]. This non-conforming fraction varies from 5-30%, depending on the amino acid type [235]. In practice, it has been difficult to observe a definite improvement in actual calculations of structure when comparing rotamer-based to continuum search methods [150]. When making correlations between packing energies and stability, however, these off-rotamer effects do become important [168][236]. More recently, it has been argued that a highly detailed rotamer library can improve speed [145] and accuracy [120][145] of side chain calculations.

9.2 Heuristic Optimization

A very fast and efficient heuristic optimization method is *SCWRL* (*Side Chains With Rotamer Library*) developed by R. Dunbrack [125]. Due to the simplicity behind its assumptions, it can be considered something of a base line for efficient side chain placement.

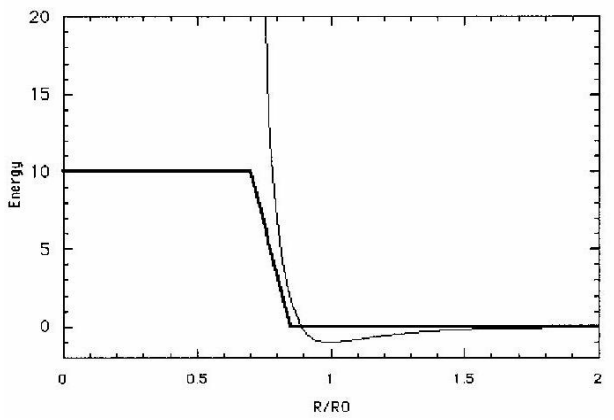

FIGURE 9.3. *SCWRL* van-der-Waals term.

SCWRL uses the backbone-dependant rotamer library described in [158]. Being a statistical method, it is shown that adapting the rotamer probabilities to the backbone torsion angles improves the results. The energy function is a simple repulsive van-der-Waals term for all heavy atoms described by the following formula (also schematically shown in Figure 9.3):

$$E = \left\{ \begin{array}{ll} 0.0 & R > R_0 \\ (-57.273) * (\frac{R}{R_0}) + 57.273 & R_0 \geq R \geq 0.83 R_0 \\ 10.0 & R < 0.83 R_0 \end{array} \right\} \tag{9.1}$$

R is the distance between two atoms and R_0 is the sum of their van der Waals radii. The radii are reduced by 15% to values which approximate the point where the Lennard-Jones potential becomes repulsive. The linear portion of the function approximates the repulsive curve of a Lennard-Jones potential. The energy is cut at 10.0 to allow optimization in presence of multiple clashes where a "true" Lennard-Jones potential produces exponential energy negating any optimization effort. In addition, the entropy of a rotamer is considered as the negative log likelihood, i.e. all things being equal making more probable rotamers have a lower energy.

The search strategy consists of three steps. In the first step all side chains are placed in the most probable rotamer position. Clashes (i.e. atomic collisions) with the backbone are relieved by iteratively replacing the offending rotamer with the next most probable one until no clashes are left. This produces a first solution where only side chain to side chain clashes may remain.

These are checked in the second step, where each pair of clashing residues is placed in a cluster. The cluster is enlarged by iteratively testing all rotamers for residues already in the cluster for clashes with new residues. Maximizing the size of the clusters is required to ensure that as few potentially optimal solutions as possible are discarded. In fact, side chains in the protein core can give rise to a form of "domino effect", where one misplaced side chain can affect all others interacting with it.

In the third step, the clusters are resolved by combinatorial optimization, i.e. testing all combinations and selecting the one with the lowest energy. Since the energy also considers the probability of each rotamer it is the statistical "optimum". If a cluster becomes too large to be efficiently searched (e.g. in less than 1 second), it is divided in two sub-clusters selecting the rotamer position with the fewest interactions. This criterium is schematically represented in Figure 9.4.

SCWRL adopts a heuristic that can be easily termed "common sense" strategy for placing side chains based on their relative probability. It is both efficient and fast, although it does not guarantee the global optimum. These characteristics have prompted its implementation as part of this thesis. Since the ap-

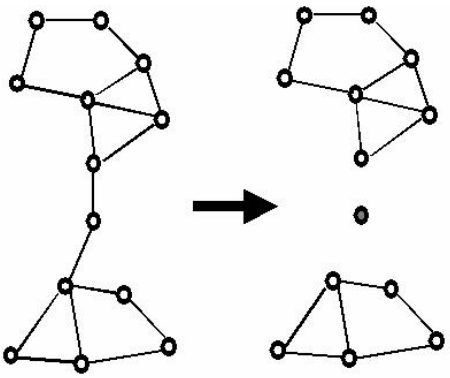

FIGURE 9.4. SCWRL cluster resolution strategy

proach is purely statistical it holds no statement about the true energy of the solution, making integration in energy-based optimization of whole proteins difficult. This lack of a "true" energy minimization procedure has prompted the investigation of another category of side chain placement algorithms, based on the dead end elimination theorem. Combining the first two steps of *SCWRL* with a subsequent energy minimization was also investigated.

9.3 Deterministic Optimization

The dead end elimination (*DEE*) theorem for side chain placement was first introduced by Desmet in 1992 [141]. Since then it has generated a number of publications ranging from theoretical improvements [108][140][142][144], implementations [87][116][143][174] and extensions [117][122][170].

The basic assumption behind the theorem is that the global energy E_{global} of a protein system can be written as:

$$E_{global} = E_{template} + \sum_{i} E(i_r) + \sum_{i} \sum_{j \neq i} E(i_r j_s) \tag{9.2}$$

$E_{template}$ is the template (i.e. the backbone atoms) self energy, $E(i_r)$ the potential energy of the side chain atoms in rotamer r at position i in the force field of the template and $E(i_r j_s)$ the non-bonded pairwise interaction energy between rotamers r at position i and s at position j.

The brute force method to determine the global minimum energy conformation (*GMEC*) for such an energy function would require to calculate all possible rotamer combinations. This is unfeasible for all practical purposes, due to the combinatorial explosion. The *DEE* theorem states that a rotamer r at position i that does not fulfill the following inequality can be excluded in favor of rotamer t if [141]:

$$\left(E_{template}(i_r) + \sum_{i \neq j} \min_s E(i_r j_s)\right) - \left(E_{template}(i_t) + \sum_{i \neq j} \max_s E(i_t j_s)\right) > 0 \tag{9.3}$$

$E_{template}(i_r)$ is the interaction energy with the template and $E(i_r j_s)$ is, again, the interaction between rotamers i_r and j_s.

In simple words: If at position i rotamer r always (i.e. regardless of all other rotamer combinations) has a higher energy than rotamer t it cannot be part of the *GMEC*. This is achieved if the minimum interaction energy (i.e. best case) of r has a higher energy than the maximum interaction energy (i.e. worst case) of t.

The above inequality turns out to be too loose and unduly limiting the number of eliminated rotamers. This prompted Goldstein [142] to formulate a sharper inequality in the following form:

$$E_{template}(i_r) - E_{template}(i_t) + \sum_{i \neq j} \min_s \left(E(i_r j_s) - E(i_t j_s)\right) > 0 \tag{9.4}$$

This states that i_r will not contribute to a local minimum if the energy of a conformation with i_r can always be lowered by solely exchanging i_r with i_t, keeping all other non-i residues frozen. The large difference in effectiveness between both inequalities is illustrated in Figure 9.5.

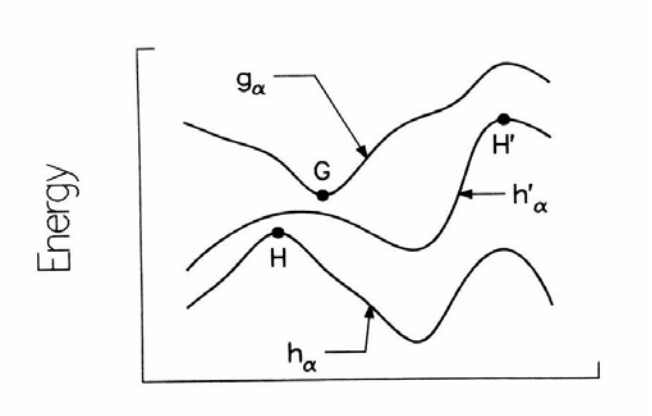

FIGURE 9.5. Advantages of a stricter inequality. The conformation of residue α is held fixed at rotamer g_α or some other point. g_α would be identified as dead ending relative to h_α by both Eq. 9.3 and Eq. 9.4. h'_α would be identified as dead ending by Eq. 9.4 but not by Eq. 9.3, because H' is higher than G.

The Goldstein *DEE* theorem is iteratively applied to all residues until no further rotamers can be eliminated. In theory, the DEE theorem has also been

postulated for pairs, triplets and higher orders of rotamer positions [141]. Its application is expensive and not always recommended [142]. Further improvements to the *DEE* theorem are limited to optimizing certain aspects, e.g. the speed at which the solution space is reduced, and go beyond the scope of the description in this thesis. The interested reader may find a very detailed and updated explanation of the theory behind the *DEE* theorem and its extensions in [108].

After the *DEE* has been performed, the solution space is reduced to a list of possible conformations, which is typically several orders of magnitude smaller than before. However, the problem remains how to search the *GMEC*. For larger proteins, this reduced space is still too large to allow an efficient solution by combinatorial optimization.

Leach [117][122] describes an elegant way to explore the reduced solution space using the A^* search algorithm [92]. The A^* algorithm is a variant of the best-first branch & bound search, belonging to the class of "informed" searches. Selection of the next state to be searched is guided by inclusion of an optimistic heuristic function h^* that underestimates the cost of reaching a solution. This underestimation of the true cost in h^* defines its "admissibility".

For "admissible" heuristics h^* the following has been demonstrated [237]: The first solution found by the A^* algorithm is the global optimum. The algorithm is also optimally efficient. Any algorithm using the same knowledge, which expands fewer nodes, sacrifices the optimality. The solution is no longer guaranteed to be the global optimum [237].

The choice of a good heuristic h^* is what makes the A^* algorithm efficient [238]. Using the trivial $h^* \equiv 0$ heuristic turns the search into a simple breadth-first search. Thus the heuristic should contain as much information about the solution as available at the time it is used to estimate the remaining cost.

For exploring the solution space in rotamer based side chain placement Leach [122] has proposed the following heuristic h^* for all remaining positions j to be optimized:

$$h^* = E_{template}(j_s) + \sum_{i=1}^{n} E(i_r j_s) + \sum_{k=n+1}^{N} \min_t E(k_t j_s) \tag{9.5}$$

The remaining cost is estimated as the sum of the template interactions, the interactions with the already optimized rotamer positions ($i = 1, ..., n$) and the minimum possible interaction with the remaining rotamer positions ($k = n+1, ..., N$). The A^* algorithm may be used for side chain placement with this obviously admissible heuristic. However, it was established that efficiency depends on the choice of the next rotamer position to be searched. Here, the

following term $v(j_s)$ is calculated for all positions j and rotamers s that still need to be optimized:

$$v(j_s) = E_{template}(j_s) + \sum_{k \neq j} \min E(k_t j_s) \tag{9.6}$$

The position j is expanded first, where the difference between the two lowest values $v(j_s)$ is greatest. This means that residues for which it is likely that there will be a single rotamer preferred to all others will be searched first, typically reducing the remaining search space considerably [87][122].

Despite this search method being the best-suited to explore large solution spaces [122], for large proteins it may reach the point where the available computer memory is no longer sufficient [87][122]. To still produce a solution in this case the diploma thesis of A. Kindler [87] studied a way to split the solution space in two or more parts, sacrificing the optimality of the global solution for applicability to larger protein structures.

Alternatively, the so-called "memory bounded" A^* search [92] was implemented. In this version of the A^* algorithm, memory is saved by modifying the memory structure such that only part of the nodes is expanded. This trades memory usage for speed, allowing larger problems to be solved.

The methods described in this section, *SCWRL* (steps 1 and 2), *DEE* and A^* search, were implemented as part of this thesis. The advantage of using *SCWRL* is the speed at which a part of the side chain positions can be placed. This is paid for in the lack of a "true" energy minimization. The combination of *DEE* and A^* search is slower, but allows a more detailed optimization of the rotamer positions. It is especially interesting for combining side chain placement with limited backbone flexibility, something usually disregarded which nevertheless is quite significant [157].

9.4 Implementation: *Peso*

A first prototype of the side chain placement methods was implemented as part of A. Kindler's diploma thesis [87]. It covered the non-iterative *DEE* theorem and A^* search algorithms. Later it was necessary to re-write the code in order to make the *DEE* method iterative and improve the overall efficiency. The first two steps of the *SCWRL* algorithm were also included at this stage. This implementation was mainly done by J. Maydt[1].

The program classes concerning side chain optimization are collected into a single package called *Peso* (for *Prot*ein *S*ide chain *O*ptimizer). It requires the

[1] Who worked as a "studentische Hilfskraft" at that time.

packages *Biopool2000* and *Energy*. The side chain placement was split into a number of classes in order to separate the functionality of the rotamer library, the problem space, the *DEE* and *SCWRL* algorithms, the *A** search and the energy calculation. A class diagram of *Peso* is shown in Figure 9.6.

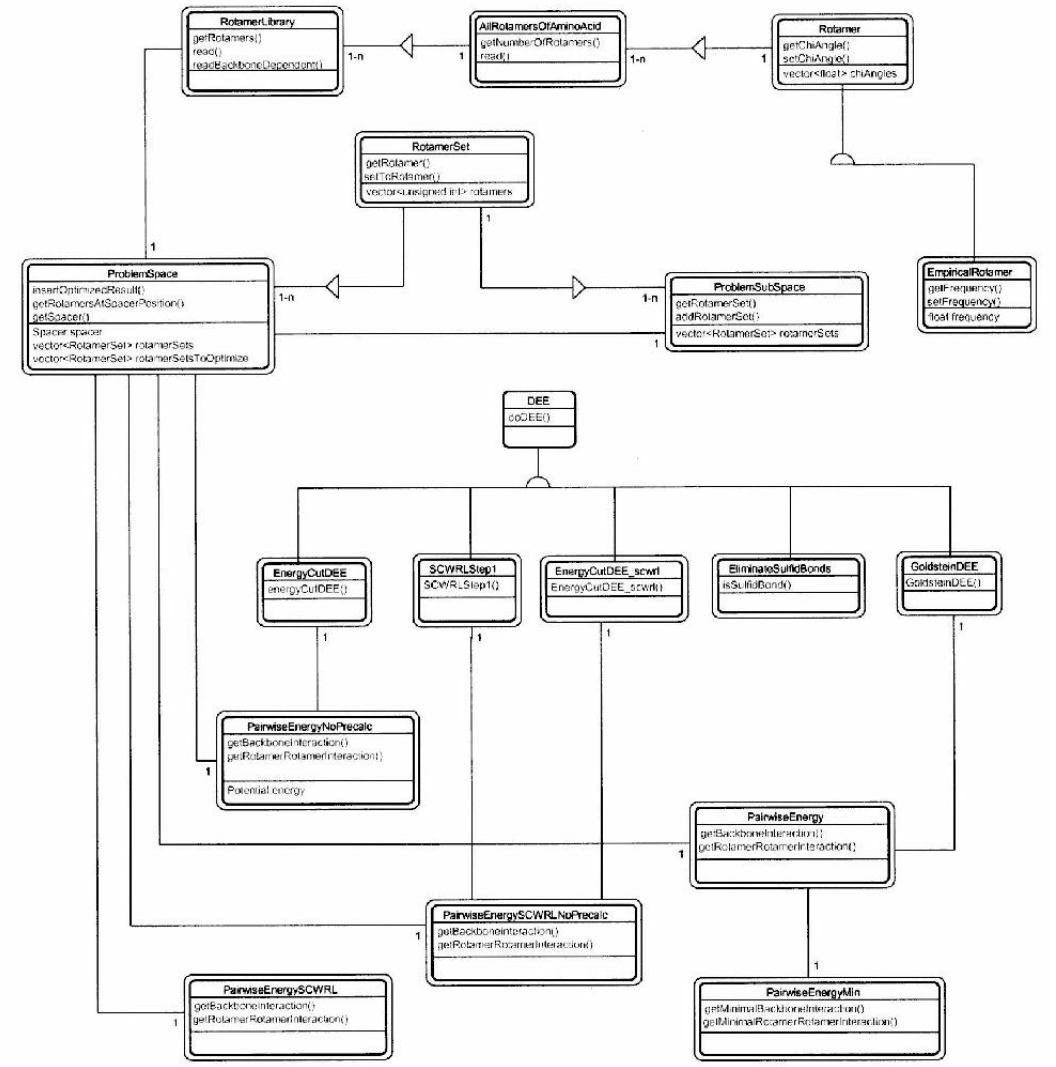

FIGURE 9.6. Class diagram for *Peso*

The rotamer library is represented in the class `RotamerLibrary`, which allows operations required to set the rotameric state of the amino acids. It also allows the input/output operations of the rotamer data like converting the rotamer library of Dunbrack [158] into the present format. The class uses `AllRotamersOfAminoAcid`, `EmpiricalRotamer` and `Rotamer` to represent the homonymous functionality.

The class `ProblemSpace` serves to unite the information from `RotamerLibrary`, the backbone template in `Spacer`, and the information about the remaining problem space. `RotamerSet` is used in `ProblemSpace` to control the rotamers available at a specific `Spacer` position.

Goldstein's *DEE* algorithm is implemented in `GoldsteinDEE`. For efficiency reasons, two initialization steps were implemented separately: `EliminateSulfidBonds` and `EnergyCutDEE`. The former tries to fix, where possible, cysteine residues in a disulfide bridge. `EnergyCutDEE` eliminates rotamers with unfavorable template interactions. An analogous class, `EnergyCutDEE_scwrl`, performs the same step for the *SCWRL* algorithm in `SCWRLStep1`.

A generic A^* search algorithm is implemented in `AStarSearch`, which can be re-used for other applications. To control the A^* search, the following two classes are required: `AStarNodeTemplate` and `EnergyHeuristicNode`. In order to split a problem space into smaller subspaces, whenever solving the original space would exceed a given time frame, uses `ProblemSubSpace`.

A set of specific energy calculations required to perform the *DEE* and *SCWRL* algorithms is implemented in `PairwiseEnergyNoPrecalc` (interactions rotamer to template), `PairwiseEnergySCWRLNoPrecalc` (same for *SCWRL*), `PairwiseEnergy` (interactions rotamer to template **and** rotamer), `PairwiseEnergySCWRL` (same for *SCWRL*), `PairwiseEnergyMin` (minimum interactions rotamer to rotamer), .

In addition to using the algorithms as part of *Homer*, a number of programs was written to test and benchmark the functionality. `dee` is the program to start a side chain placement procedure with *DEE* and A^* search and `scwrl` imitates the *SCWRL* algorithm. `bestfit` determines the best rotamer combination regardless of energy (i.e. lowest possible RMSD). `compare` serves to benchmark the RMSD difference between two *PDB* files differing only in side chain position. `convert_lib` is used to import a new version of Dunbrack's rotamer libraries.

9.5 Summary

Side chains are typically the last and an autonomous part of the protein construction process, after the backbone position has been fixed. They need to be placed simultaneously and rearrangements are more common than changes in backbone conformation.

The common main approximation in side chain placement is the usage of a rotamer library. This is a set of preferred torsion angle combinations. Two optimization methods of the state of the art for side chain placement are introduced and implemented.

SCWRL [125] is a heuristic based on the statistical occurrence of side chain rotamers. Placing each residue in its most favored rotamer, it checks for collisions and removes them by using less probable rotamers. This results in a fast method, which forms the base line for side chain placement. Due to its statistical character, it makes energetic optimization difficult and was complemented with a second method.

Combining the dead end elimination (*DEE*) algorithm [141] for reducing the solution space with the A^* search [122] yields a slower, yet efficient, energy optimization procedure to find the global optimum in large conformational spaces. The *DEE* theorem allows to reduce the conformational space quickly by several orders of magnitude when discarding unfavorable conformations. The A^* search is guaranteed to find the global optimum and to be optimally efficient for the utilized prior knowledge [237].

Much care was taken during implementation of both methods to separate the rotamer library and problem space description from the utilized energy functions and optimization strategies. This allows the fast implementation of new variants or additional algorithms.

10
Homology Modeling Server

The best methods for protein structure prediction are only as useful as they are employed by the user community[1]. With a community as wide as that studying proteins, availability is of key concern. Disseminating the advanced methods is best achieved by offering a web-based service, as will now be described.

10.1 Motivation

Due to the huge interest in biotechnology, it can be estimated that for every group developing structure prediction software there are at least two or three orders of mangnitude more potential users. Construction of models for proteins with unknown structure can yield good results for homologous sequences with medium or high sequence identity. Its effective usage to construct reliable models still requires knowledge about the process.

Interest in protein structures is growing as the number of sequenced genomes increases. [300] Tools capable of processing large amounts of data in little time are required to handle this vast amount of information. Automating the process of homology modeling is therefore desirable to improve the widespread use of structural models.

Over the last couple of years there has been a growing trend in bioinformatics to use the web as a way to spread knowledge among researchers. Web services, or servers as they are mostly called, are becoming the most practical

[1] In secondary structure prediction for example there have been significant advances over the last 10 years. With modern methods reaching a Q_3 value around 77%. Yet, there appear to be a consistent number of biologists who still employ the outdated GOR algorithm, averaging a Q_3 of 60%. (G. Pollastri, personal communication)

way to fulfill this task. An automated homology modeling server is a natural way to complete the goals of this thesis.

In addition to the CASP and CAFASP experiments, a continuous evaluation and benchmarking experiment for secondary structure prediction and homology modeling servers, called EVA, has been introduced by B. Rost [331]. Participating in this continuous evaluation is planned for the near future.

10.2 Available Servers

At present, and in contrast to an ever-growing number of fold recognition servers, only few web-based servers for homology modeling exist. The three most important ones are: Swiss-Model [317], CPH Models [179] and 3D-Jigsaw [319]. The common input consists of a sequence, title and e-mail address which are required for the server to run and return the results. Coordinates for the constructed model are returned by e-mail. The different implemented approaches used by the servers will be now explained.

Swiss-Model [317] is the oldest and best advertised existing web-based homology modeling server. It uses an internal database of structurally superimposed protein structures as possible templates. BLAST is used to generate a sequence alignment between query sequence and this internal database. This is used to select the templates to be used for model-building. The model framework is built from those residues in the alignment which occupy a similar portion of space in the structural alignment. The 3D coordinates of the framework are averaged between the templates and non-conserved side chain conformations removed. The framework is completed by building the lacking loop regions. The geometry of the anchor region is compared to a database of loop fragments extracted from the *PDB*. Missing side chains are built from a rotamer library with a statistical approach. The final model is evaluated for local errors considering the 3D environment of each residue and its packing density.

CPH Models [179] uses a radically different approach to homology modeling. It implements a restraint-based modeling method based on predicted inter-residue distances. An alignment is built from the query sequence. The information contained in the alignment is used as input for a neural network based predictor. The output consists in a number of inter-residue distances. Since the query sequence is a homology modeling target, the predicted information is accurate enough to use a restraint-based method to derive the C_α atom coordinates of the model. Compared to fragment based methods, CPH Models produces models that capture the overall topology at least as well, but which may contain deformed backbone segments requiring further refinement.

The only web-based server to perform consistently well at the CAFASP-2 and CASP-4 meetings was 3D-Jigsaw by P.A. Bates [319]. PSI-BLAST is used to search both the non-redundant (NR) sequence database and the *PDB* for possible templates. Up to five templates are selected based on a mixture of sequence similarity and data quality (e.g. resolution and number of missing atoms) and structurally superimposed. The resulting structural alignment is aligned to the previously built sequence profile. For targets with less than 40% sequence identity, a secondary structure prediction is further considered in order to improve the alignment. Coordinates from residues in the framework are copied from the templates. Non-conserved loops are built from a database of *PDB* fragments and optimized using a modified mean field approach to gap closure [124]. Side chains are constructed as close as possible to the template and a second mean field calculation is used where necessary. The final model was refined using 100 steps of steepest descent optimization in the CHARMM force field [60].

As has been stated before, this server performed consistently well in CAFASP-2 and CASP-4, outperforming several expert groups. It would therefore appear to be the best currently available automated method. Unfortunately, it is not (yet) attached to the continuous server evaluation EVA. Detailed results for EVA still have to be released.

10.3 Implementation

The *HOMER* server implements the classic fragment based homology modeling approach described in Chapter 7 as a web-based service. It shares with existing servers the concept of submitting a sequence, title and e-mail address via web form. The results containing the finished model are returned by e-mail. The web interface is shown in Figure 10.1.

Two different modes of operation are implemented. In the 'automatic' alignment mode it searches for suitable template structures and generates an alignment. This is performed with the PDBBLAST protocol [207] already described in Chapter 7. PSI-BLAST [128] is first used to search for homologous sequences in the non-redundant (NR) database of protein sequences. Typically, a total of four iterations is performed in order to generate a profile comprising as much information as possible on the protein family. This profile is then used to search against the *PDB* database of protein structures. Targets that have less than 30% sequence identity or less than 20 residues aligned to a template *PDB* structure are not modeled as it would be uncertain whether the predicted structure would be correct [202]. Thresholds for template selection may be submitted by the user.

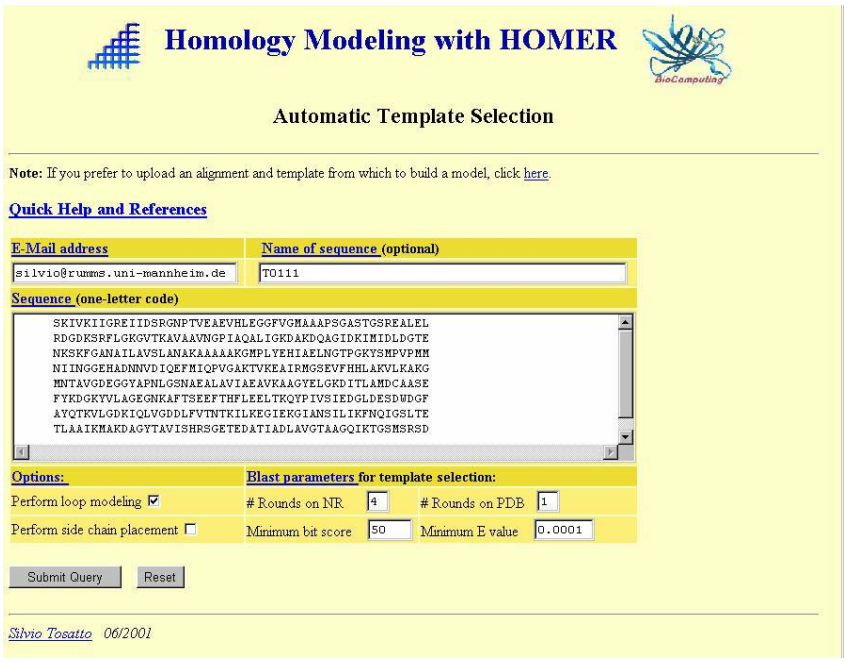

FIGURE 10.1. Web interface of the *HOMER* server.

In the alternative 'manual' alignment mode the user can submit an alignment in FASTA format between the query sequence and a template structure. The *PDB* file corresponding to the template sequence also has to be submitted. This mode can be useful for the experienced user who wishes to manually edit an alignment to improve the constructed model. It is not present in other homology modeling servers.

Once a suitable template is found, the corresponding *PDB* entry is retrieved. The raw model is computed from a single template structure, by substituting the amino acids and copying the 3D coordinates of the protein core. Insertions and deletions are optionally modeled using the fast *ab initio* loop modeling algorithm based on the divide-and-conquer approach described in Chapter 12.5. The 3D coordinates of side chains conserved between target and template are copied from the template structure. The model can optionally be finalized by optimizing the position of non-conserved side chains with a rotamer based dead-end elimination theorem method as described in Chapter 9.

The results are sent back to the user in an e-mail which contains information about the model building process. The final model is sent in standard *PDB* format as an attachment and can be read with Rasmol or the Chime plug-in for web browsers. Figure 10.2 contains a sample model constructed with the *HOMER* server and returned by e-mail.

The web server uses the standard Apache web server software [318], requiring little specific adjustments. The HTML page accessible to the user con-

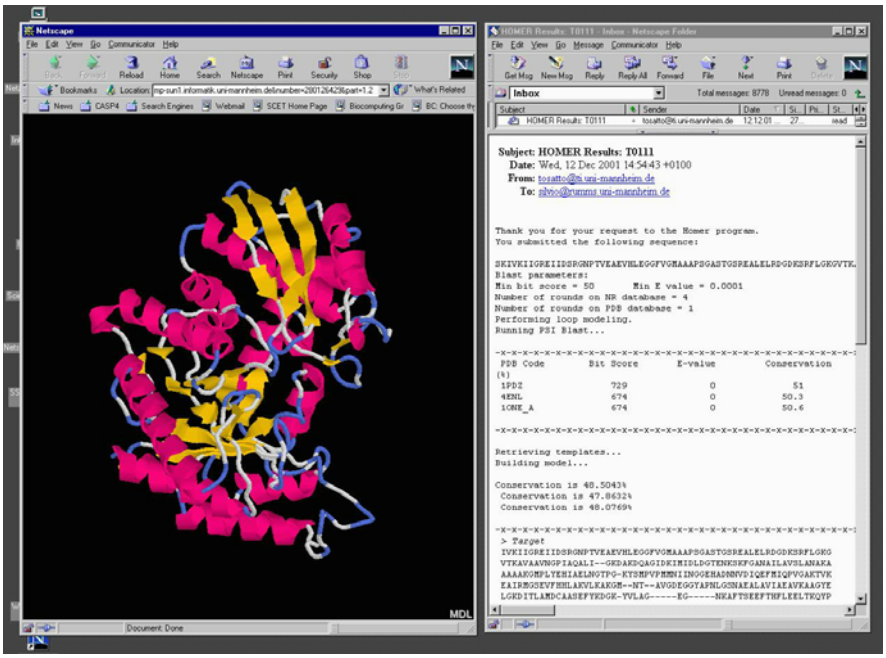

FIGURE 10.2. Sample results for the *HOMER* server. The e-mail contains information about the model construction process (*right*) as well as the constructed model (*left*).

tains a standard web form transmitting the submitted data to the CGI script `homer.cgi`. This script parses the input and checks for obvious mistakes which would invalidate the request (e.g. forbidden characters in the e-mail address field). If the input is valid a standard HTML page is generated thanking the user and explaining that results will be returned by e-mail. The data is passed on to the Perl script `homerrunemail.pl` which writes a log entry, coordinates the data processing and activates the e-mail response. The actual calculations are initiated in `homerrun.pl` and take the form of two program calls. The first, only required in 'automatic' mode, calls the PDBBLAST script and collects the admissible template structures via the `ali2filter` program. The second program call, which is always executed, goes to the `ali2model` program. This implements all the functionality of the *Homer* package and is described in Chapter 7. The created model is temporarily stored before handing back the control to the `homerrun.pl` script. The `mymailto.pl` script is used to dispatch the e-mail describing the optimization process, including the model as a MIME attachment. Finally, another log entry is written and all temporary files deleted. A flow chart of the server process is shown in Figure 10.3.

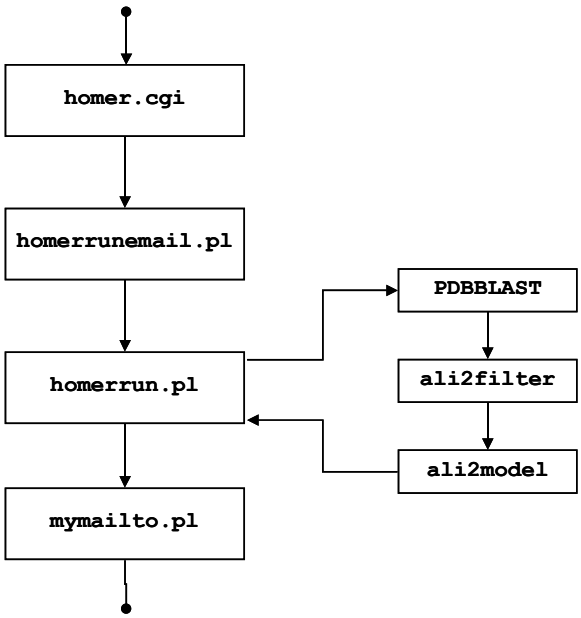

FIGURE 10.3. *HOMER* server flow diagram.

10.4 Summary

Homology modeling has a large number of potential users and a growing interest, but still requires expert knowledge. An automated homology modeling server facilitates the usage of structural models by non-experts. Currently only three homology modeling servers, implementing different approaches, are available. Implementing such a server was considered as a natural way to complete the goals of this thesis.

The *HOMER* server offers the possibility to construct models of protein structures with automatic or manual alignment generation. It bundles the previously described technology, ranging from template selection to side chain placement, in a web interface and returns the constructed models as e-mail attachments. Internally, the server uses a number of simple Perl scripts to coordinate data processing and uses the previously described *Homer* package.

11
Results

With all methods and protocols for knowledge-based protein structure predicion having been described, it is now possible to focus on the results of the work. To this end, the performance of the methods will be judged by the results of our participation in the CASP-4 competition in 2000.

11.1 CASP-4

This section describes the results presented by the three independent assessors at the CASP-4 conference held in Asilomar, California, in December 2000. At the time of writing the special issue of *Proteins* containing the CASP-4 assessment papers had not been published yet. Only the papers of the fold recognition (M. Sippl) [309] and *ab initio* (A. Lesk) [310] assessors are available as preprints. A preprint from the homology modeling assessor (A. Tramontano) is not available.

In CASP-4 the overall results were similar to those in CASP-3, but with more difficult targets. Over a dozen sequences were multi-domain proteins. The average length increased, with several ranging between 300 and 400 residues and the largest one being 811 residues long. It is fair to say that predictions are becoming useful for such larger proteins. The first true blind test of structure prediction servers (CAFASP-2) was also held in parallel to CASP-4 [99][332]. The servers performed as well as most manual submissions for easy targets. For hard targets the results still indicate manual intervention to be of prime importance to filter out wrong predictions and improve the results. The servers as a group identified roughly double the number of correct folds than the best of the servers.

Results for the homology modeling category were not conclusive. A. Tramontano used the algorithms provided by the prediction center to analyze the submitted models. Her main comment was that most models were equally valuable, with only minor differences partially caused by chance. She therefore chose to bypass the traditional presentation of six groups performing consistently well and only presented two methods. Sternberg's 3D-JIGSAW server was selected because it performed at least as well as most manual submissions, strengthening the impression that homology modeling has become tractable by automated methods. Venclovas was selected because, in addition to perform consistently well, he had been able to pinpoint segments of the protein in two models which, according to the assessor, were best left unpredicted. For the remaining groups no direct ranking was established. Tramontano stated that the differences were of minor importance. Our group was nevertheless ranked in this group of more or less equally well-performing methods[1].

The fold recognition assessor, M. Sippl, produced a detailed ranking of the predictions. For each target domain a number of points was awarded for correct fold (up to 2.0 points) and correct alignment (up to 4.0 points). Two overall scores were derived. T_i is the total score of group i over all N_{sub} submitted models. Q_i is the average score per submission (i.e. $Q_i = \frac{T_i}{n}$). The three top-ranking groups of each measure were invited to present their results. Among these, Sternberg and Karplus have "classic" fold recognition approaches based on sequence profiles. Baker used a mixture of homology modeling and *ab initio* predictions with manual intervention. Perhaps the most remarkable performance is that of A. Murzin who reached the highest Q_i score by using his vast knowledge of protein structures[2] and essentially assembled the models by hand.

During the subsequent discussion it became clear that the T_i ranking is generally considered more significant. Table 11.1 shows the top 20 ranking according to T_i. As can be seen from the table, our group ranks 15th out of 125 participants in the fold recognition category [309]. Almost half of the groups, 56 out of 125, had a T_i score of 5.0 or less. An interesting case is given by the second domain of T0115, where our group was the only one to have a score > 0.0.

Assessment of the *ab initio* predictions was carried out by A. Lesk in a similar fashion to that of fold recognition, in that scores were also used to quantify the success of a prediction. The Baker group had an outstanding result compared to all other *ab initio* groups, having by far the best total score. 31 points compared to 10 for the second best group (Friesner). Despite not

[1] A. Tramontano, personal communication.

[2] A. Murzin, author of the SCOP structural classification of proteins, is said to be able to classify any structure by simply looking at its model.

Rank	Group Name	code	N_{sub}	T_i	Q_i
1	Baker	354	34	41	1.24
2	Murzin	384	15	37	2.47
3	Karplus	94	28	34	1.21
4	Sternberg	126	23	33.5	1.46
5	Rychlewski	31	29	33	1.14
6	ORNL-Prospect	88	32	30.5	0.95
7	CAFASP Consensus	359	29	27	0.93
8	Friesner	414	30	26.5	0.88
9	Rost	77	23	25.5	1.11
10	Godzik	197	26	25.5	0.98
11	Walts-Wondrous-Wizards	44	30	25.5	0.85
12	Sternberg-3DPSSM	132	29	24.5	0.84
13	Honig-Barry	42	22	23	1.05
14	SBfold	381	16	23	1.44
15	BinToHes	255	33	22	0.67
16	Blundell-tl	95	9	22	2.44
17	Lomize-Andrei	2	20	21.5	1.07
18	Fischer-Daniel	357	30	21	0.7
19	Jones	23	29	21	0.72
20	bioinbgu-seqprf	106	32	17.5	0.55

TABLE 11.1. CASP-4 fold recognition ranking. This is the official top 20 ranking (from [309]), sorted by total score (T_i). For a definition of the scores see text.

having submitted "true" *ab initio* predictions, our group nevertheless appears in the ranking because we submitted predictions for the most difficult targets of the fold recognition and novel fold categories. In the overall *ab initio* ranking our group scores 3 points and is ranked 21st. Counting only novel folds, we are ranked 9th with 1 point. In this category the best two groups had 9 points (Baker) and 3 points (Friesner) [310].

Having described the general trend in CASP-4, results for some of the more representative targets will now be discussed in more detail (in ascending order of difficulty).

T0111: Enolase, E. coli

One of the easier homology modeling targets was the 431 residue-long Enolase from E. coli. This protein, annotated in Swiss-Prot, is responsible for the processing and degradation of RNA. Forming a homodimer, it was solved with X-ray crystallography at a resolution of 2.5 Å using molecular replacement.

Running PSI-BLAST it was easily established that the target sequence is over 50% identical to four template structures: `1pdz`, `1pdy`, `5enl`, `7enl`. Comparing the secondary structure predicted from SSpro to the templates revealed

the structural core to be well conserved. Using FSSP it was possible to identify very limited structural differences, mainly concentrated in the loop regions. The choice of a template structure was therefore based on considerations of X-ray resolution of the PDB structures and possible error sources, such as ligand-induced conformational changes. Structure `1pdy` was selected as the most probable template. A multiple alignment between the four template sequences and the target was performed with CLUSTALW, resulting in the following alignment:

```
1pdy -SITKVFARTIFDSRGNPTVEVDLYTSKGLF-RAAVPSGASTGVHEALEMRDGDKSKYHG
1pdz -SITKVFARTIFDSRGNPTVEVDLYTSKGLF-RAAVPSGASTGVHEALEMRDGDKSKYHG
5enl -AVSKVYARSVYDSRGNPTVEVELTTEKGVF-RSIVPSGASTGVHEALEMRDGDKSKWMG
7enl -AVSKVYARSVYDSRGNPTVEVELTTEKGVF-RSIVPSGASTGVHEALEMRDGDKSKWMG
t111 SKIVKIIGREIIDSRGNPTVEAEVHLEGGFVGMAAAPSGASTGSREALELRDGDKSRFLG
       : *: .* : *********.::  . *..  : .******* :****:******:: *

1pdy KSVFNAVKNVNDVIVPEIIKSGLKVTQQKECDEFMCKLDGTENKSSLGANAILGVSLAIC
1pdz KSVFNAVKNVNDVIVPEIIKSGLKVTQQKECDEFMCKLDGTENKSSLGANAILGVSLAIC
5enl KGVLHAVKNVNDVIAPAFVKANIDVSDQKAVDDFLISLDGTANKSKLGANAILGVSLAAS
7enl KGVLHAVKNVNDVIAPAFVKANIDVSDQKAVDDFLISLDGTANKSKLGANAILGVSLAAS
t111 KGVTKAVAAVNGPIAQALIGK--DAKDQAGIDKIMIDLDGTENKSKFGANAILAVSLANA
     *.* :**  **. *.  ::    ...:*   *.:: .**** ***.:******.**** .

1pdy KAGAAELGIPLYRHIANLAN--YDEVILPVPAFNVINGGSHAGNKLAMQEFMILPTGATS
1pdz KAGAAELGIPLYRHIANLAN--YDEVILPVPAFNVINGGSHAGNKLAMQEFMILPTGATS
5enl RAAAAEKNVPLYKHLADLSKSKTSPYVLPVPFLNVLNGGSHAGGALALQEFMIAPTGAKT
7enl RAAAAEKNVPLYKHLADLSKSKTSPYVLPVPFLNVLNGGSHAGGALALQEFMIAPTGAKT
t111 KAAAAAKGMPLYEHIAELNG-TPGKYSMPVPMMNIINGGEHADNNVDIQEFMIQPVGAKT
     :*.**  .:***.*:*:*     .   :*** :*::***.**.. : :***** *.**.:

1pdy FTEAMRMGTEVYHHLKAVIKARFGLDATAVGDEGGFAPNILNNKDALDLIQEAIKKAGYT
1pdz FTEAMRMGTEVYHHLKAVIKARFGLDATAVGDEGGFAPNILNNKDALDLIQEAIKKAGYT
5enl FAEALRIGSEVYHNLKSLTKKRYGASAGNVGDEGGVAPNIQTAEEALDLIVDAIKAAGHD
7enl FAEALRIGSEVYHNLKSLTKKRYGASAGNVGDEGGVAPNIQTAEEALDLIVDAIKAAGHD
t111 VKEAIRMGSEVFHHLAKVLKAKG--MNTAVGDEGGYAPNLGSNAEALAVIAEAVKAAGYE
     . **:*:*:**:*:*  : * :       ****** ***: .  :** :* :*:* **:

1pdy G--KIEIGMDVAASEFYKQNNIYDLDFKTANNDGSQKISGDQLRDMYMEFCKDFPIVSIE
1pdz G--KIEIGMDVAASEFYKQNNIYDLDFKTANNDGSQKISGDQLRDMYMEFCKDFPIVSIE
5enl G--KVKIGLDCASSEFFK-DGKYDLDFKNPNSDKSKWLTGPQLADLYHSLMKRYPIVSIE
7enl G--KVKIGLDCASSEFFK-DGKYDLDFKNPNSDKSKWLTGPQLADLYHSLMKRYPIVSIE
t111 LGKDITLAMDCAASEFYK-DGKYVLAG-----EGNKAFTSEEFTHFLEELTKQYPIVSIE
```

```
        .: :.:* *:***:* :. * *       : .: ::. :: .:  .: * :******

1pdy DPFDQDDWETWSKMTSGTT--IQIVGDDLTVTNPKRITTAVEKKACKCLLLKVNQIGSVT
1pdz DPFDQDDWETWSKMTSGTT--IQIVGDDLTVTNPKRITTAVEKKACKCLLLKVNQIGSVT
5enl DPFAEDDWEAWSHFFKTAG--IQIVADDLTVTNPKRIATAIEKKAADALLLKVNQIGTLS
7enl DPFAEDDWEAWSHFFKTAG--IQIVADDLTVTNPKRIATAIEKKAADALLLKVNQIGTLS
t111 DGLDESDWDGFAYQTKVLGDKIQLVGDDLFVTNTKILKEGIEKGIANSILIKFNQIGSLT
     * : :.**: ::   .     **:*.*** ***.* :  .:**  ...:*:*.****:::

1pdy ESIDAHLLAKKNGWGTMVSHRSGETEDCFIADLVVGLCTGQIKTGAPCRSERLAKYNQIL
1pdz ESIDAHLLAKKNGWGTMVSHRSGETEDCFIADLVVGLCTGQIKTGAPCRSERLAKYNQIL
5enl ESIKAAQDSFAAGWGVMVSHRSGETEDTFIADLVVGLRTGQIKTGAPARSERLAKLNQLL
7enl ESIKAAQDSFAAGWGVMVSHRSGETEDTFIADLVVGLRTGQIKTGAPARSERLAKLNQLL
t111 ETLAAIKMAKDAGYTAVISHRSGETEDATIADLAVGTAAGQIKTGSMSRSDRVAKYNQLI
     *:: *   :   *: .::*********  ****.**  :******: .**:*:** **::

1pdy RIEEELGSGAKFAGKNFRAPS-- 433
1pdz RIEEELGSGAKFAGKNFRAPS-- 433
5enl RIEEELGDNAVFAGENFHHGDKL 436
7enl RIEEELGDNAVFAGENFHHGDKL 436
t111 RIEEALGEKAPYNGRKEIKGQA- 431
     **** **. * : *.:    .
```

Adjustments in the alignment were done to optimize the distance between anchor residues for insertions and deletions. Model construction was straightforward with no significant manual intervention.

Analysis by the assessors established that roughly 95% of the X-ray target structure was superimposable to the closest PDB structure with 0.87 Å C_α RMSD. The model submitted by our group was among the best performing, with a global C_α RMSD of just 1.89 Å over the entire structure. The core, about 94% of the submitted model, superimposes with 1.2 Å, while the loops (6%) have a global C_α RMSD of 6.07 Å. For comparison, Venclovas had an overall C_α RMSD of 1.85 Å, with the core at 0.97 Å and loops at 6.46 Å. For Sternberg's group the values are 3.04 Å, 2.37 Å and 8.13 Å respectively, but with only 96.3% of the total structure modeled. The superimposition between our model and the target is shown in Figure 11.1.

T0122: Tryptophan Synthase α-subunit, Pyrococcus Furiosus

A more demanding homology modeling target was the 248 residue-long Tryptophan Synthase α-subunit, from Pyrococcus Furiosus. The structure of a homologous protein domain had been previously solved in complex with the β-subunit for a different organism. The protein, which is not directly anno-

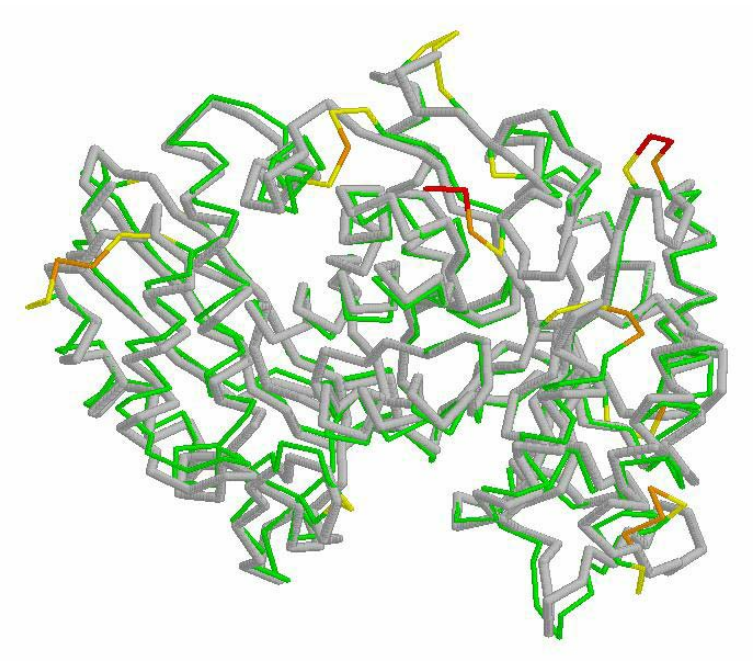

FIGURE 11.1. Structural superposition between prediction (*green to red*) and real structure (*grey*) for T0111.

tated in a database, contributes to synthesizing Tryptophan. It was solved with X-ray crystallography at a resolution of 2.0 Å. Molecular replacement using PDB structure `1bks` had failed and heavy metal atoms were instead being used to solve the phase problem.

In addition to the inherent difficulty of the modeling process, the publication of this target sequence only weeks before the CASP deadline imposed severe time limitations on the modeling process. A PSI-BLAST search identified two possible template structures in PDB with about 32% sequence identity: `1bks` and `2tys`. With the knowledge about the failed molecular replacement using `1bks` (X-ray resolution 2.2 Å) it was quickly decided to use `2tys` (resolution 1.9 Å) as template structure. The CLUSTALW alignment between target and template was not further edited manually due to the limited time available before the deadline. Anchor regions for loop modeling of insertions and deletions were not optimized, instead relying on the ranking procedure in loop modeling to select good candidates.

The CASP analysis established that despite the lower sequence identity, as much as 87% of the X-ray target structure were superimposable to the closest PDB structure with 1.23 Å C_α RMSD.

The model submitted by our group performed well, with a global C_α RMSD of 3.04 Å over the entire structure. The core, comprising 87% of the residues, superimposes with 2.15 Å, while the loops (13%) have a global C_α RMSD

of 6.28 Å. In comparison, Venclovas had an overall RMSD of 2.42 Å, with the core at 1.85 Å and loops at 4.77 Å C_α RMSD. However, only 98.8% of the overall structure and 93.8% of the loop residues were predicted. For the Sternberg group these values are 3.00 Å, 2.23 Å and 5.93 Å respectively. The superimposition between our model and the target is shown in Figure 11.2.

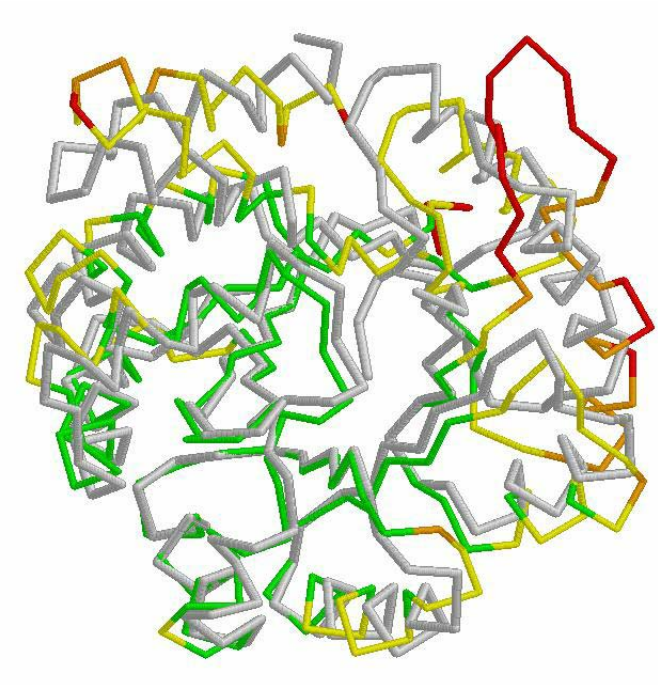

FIGURE 11.2. Structural superposition between prediction (*green to red*) and real structure (*grey*) for T0122.

T0107: Family 9 Carbohydrate Binding Module, T. maritima

A difficult fold recognition target without detectable homology to any known structure was the Family 9 Carbohydrate Binding Module from T. maritima. With no database entry provided by the crystallographer, it was unknown during the CASP-4 prediction whether this 188 residue protein belonged to a known fold class at all.

The prediction process therefore started with a secondary structure prediction using SSpro. Using MANIFOLD gave a ranking of about ten different template structures. The top ranking solution `1axi_B` had a significantly higher score than the following solutions (pareto score 8.0 vs. 6.0). Comparison with the solutions generated by the CAFASP servers did not yield further hints, as the results from single servers were disagreeing strongly. Manual analysis of the secondary structure pattern and cross-referencing available information about target and template supported the choice of `1axi_B` as template. The

alignment was derived from a manual correction of the CLUSTALW pairwise global alignment. Despite the possibility that only part of the structure would be predictable, we decided to model the entire structure, including loops.

The CASP analysis established T0107 to have a strong structural similarity to an existing fold but only weak sequence similarity in the structural alignment. It was therefore classified as analogous to a PDB structure. The maximum score awarded by Sippl to any group was 2.0 points.

The model submitted by our group received 2.0 points for having 78 out of 188 equivalent residues in a sequence independent superposition. The best submission had 84 equivalent residues, and only a total of four groups had more than our number of equivalent residues. Sequence dependent superposition is not as good, with an average shift of 31.4 residues for our model compared to about 10-12 residues for other models of the same quality. This is a direct consequence of using CLUSTALW to align the two structures. Of the overall best-performing fold recognition groups only Rychlewski, SBFOLD and Friesner had a similar score, the others performing significantly worse. The superimposition between our model and the target is shown in Figure 11.3.

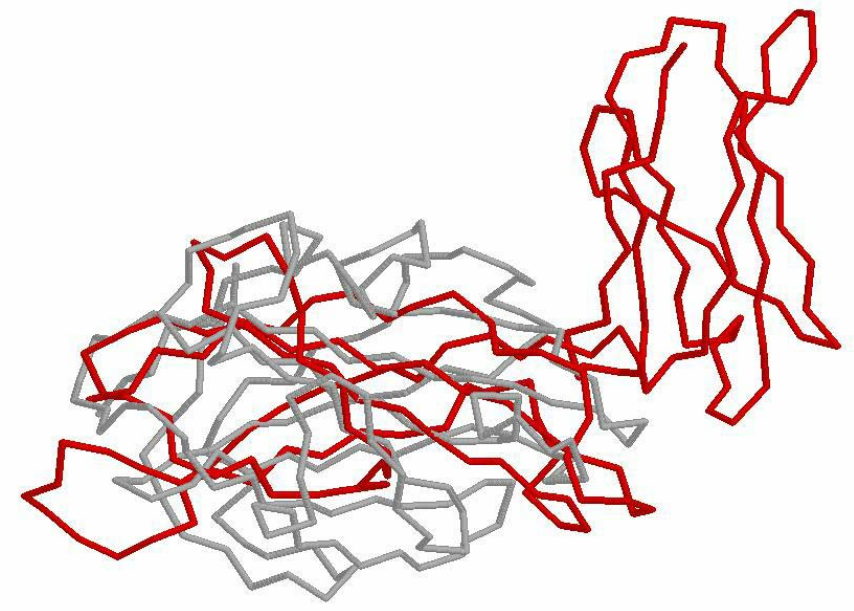

FIGURE 11.3. Structural superposition between prediction (*red*) and real structure (*grey*) for T0107.

T0116: MutS, Thermus aquaticus

Perhaps the most challenging protein to model during CASP-4 was the DNA mismatch repair protein MutS from Thermus aquaticus. Information from the

crystallographer suggested this 811-residue protein to contain a domain of the ABC ATPase superfamily near the C-terminus. Database information from Swiss-Prot supported this, indicating a potential ATP binding site at residues 583 to 590. Not much was known about the rest of the sequence.

The prediction process was started by predicting the secondary structure with SSpro and using PSI-BLAST to find homologous sequence fragments. The secondary structure prediction contained an extremely long α–helix in the central part of the protein. Due to its unique nature, we submitted the sequence to Psi-Pred to confirm this. Both servers agreed. This unusual feature can only be explained with the size of the protein and was assumed to form an autonomous domain stabilizing the remaining structure. It was later modeled as a single idealized α–helix. Of the remaining two sequence fragments no clear homology was established by PSI-BLAST to the presumed ATPase domain in the C-terminus. Instead, PSI-BLAST produced a confident prediction for `1d9x_A`, matching the predicted secondary structure, for part of the sequence prior to the N-terminus of the central α–helix. This domain was modeled directly from a CLUSTALW alignment with `1d9x_A`.

Two sequence fragments of over 200 residues each remained to model at the N- and C-terminus of the protein. MANIFOLD was used to produce possible templates. Due to a low confidence level, these predictions were compared with those of the CAFASP servers to find recurrent predictions. For the N-terminal domain this was the rank 4 structure `1a5t` and for the C-terminal domain `5tmp_A`, ranked 9th. Again, CLUSTALW was used to produce the alignments.

The last, and most problematic, step consisted in assembling the four independent domains into a single model. To the best of this author's knowledge, no automated tools exist. We therefore assembled multi-domain proteins manually, by changing the torsion angles at the junction between two domains until the composed structure appeared more or less "compact". Due to the extreme work-load during the final CASP-4 days, the assembly had to be performed less than 45 *minutes* before the submission deadline on August 31st. In this case the resulting assembled structure is therefore not compact as shown in Figure 11.4.

The CASP assessors decided to base the evaluation of this target on single domains. Based on the experimental structure four separate domains were defined. These match the division from our own prediction quite well. Only 7 out of 77 groups submitting structures for T0116 attempted to predict the entire structure, with the lowest C_α RMSD being 32.7 Å and the highest 79.4 Å. Our submitted model had a C_α RMSD of 45.4 Å.

The third domain, containing the long α–helix, was classified as a novel fold. Our prediction was correct insofar as it contained the single α–helix but did not cope with the curved nature in the experimental structure. Our model for domain three is nevertheless ranked first based on the GDT measure and was

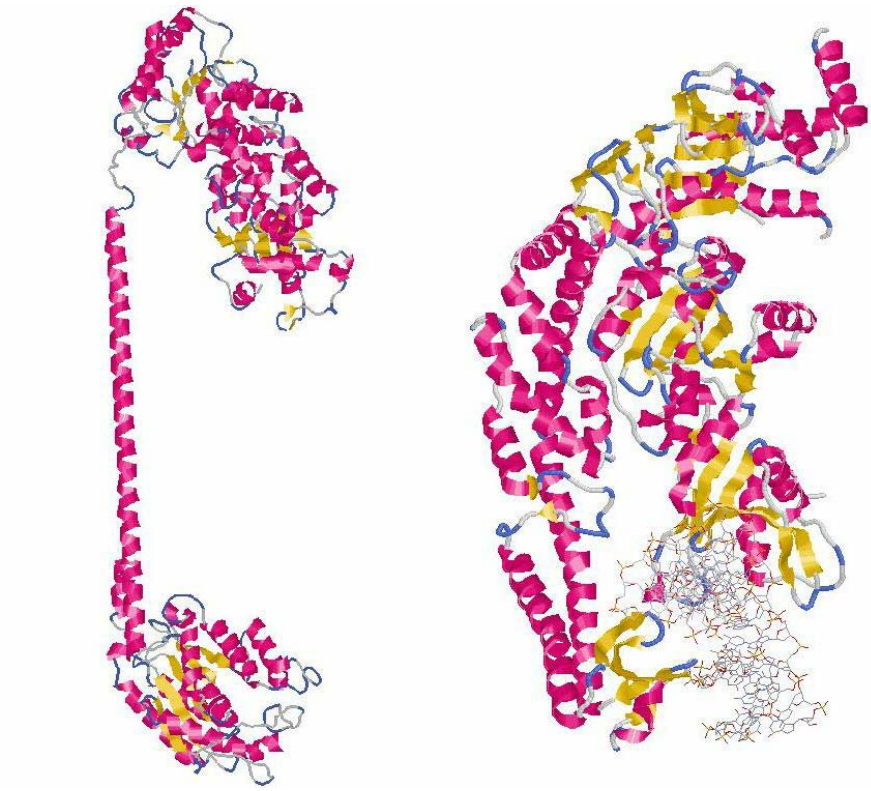

FIGURE 11.4. Comparison between predicted (*left*) and real (*right*) structure of T0116. The bound DNA fragment (*right*) in the experimental structure is displayed in "ball-and-stick" mode.

one of only four models receiving a positive score from the ab initio assessor. The first domain was classified as a non-homologous fold recognition target. Only two groups received a positive score from the fold recognition assessor: Baker and our group, with a marginally better RMSD for the equivalent residues in our model.

The second domain was also classified as analogous, while the fourth domain was a distantly homologous fold recognition target. The performance of our group was not as extraordinary for these two domains, ranking in the top 10 for domain two and obtaining 1.0 points compared to a maximum of 2.5 awarded for domain four. Overall, it is fair to say that our model of this difficult four-domain protein was among the best produced by any CASP-4 participant and is possibly only second to Baker's group. None of the other top-ranking fold recognition groups predicted all four domains except Friesner, who performed less well than our group.

11.2 Overall Performance

In addition to the thorough evaluation provided by the CASP-4 results, two more aspects of the thesis are worth describing in this section: side chain placement and performance of the automated modeling process. Both are not part of the main CASP-4 evaluation process and will now be addressed.

Side chain placement is not properly evaluated in CASP because it depends strongly on the alignment. Side chains placed from different alignments cannot be compared directly. To benchmark the placement algorithm, a representative test set of 22 proteins was used. This set covers a wide range of structures, ranging from small to large and mainly α to mainly β. Table 11.2 shows the test set composition and computation time. In addition to all-atom RMSD two side chain specific measures are used to evaluate the results: $\chi_1$and χ_{12}. The former measures the percentage of χ_1 torsion angles placed within $\pm 40^\circ$ of the native structure. Such torsion angles are considered to be "correct" within the rotamer approximation. χ_{12} measures the percentage of side chains having **both** χ_1 and χ_2 within $\pm 40^\circ$ of the native structure. This value is obviously lower than χ_1, since placement errors are cumulative.

Table 11.3 shows the results for the test set. Two values are given for each measure: *start* and *opt*. The first refers to the first, unoptimized, placement derived by adding side chains in their most probable conformation to the backbone, disregarding atomic collisions. It takes less than 2 seconds to compute and forms the baseline for further optimization. *opt* refers to the optimized conformation after using the algorithm from Chapter 9. It requires variable computation times, and only rarely containing atomic collisions.

From looking at the results it can be seen that the optimization significantly improves the RMSD in all but two cases. In most cases the optimized $\chi_1$and χ_{12} measures are significantly better than the starting conformation. With χ_1 on average around 75% and χ_{12} on average around 55% the results are in line with current methods (e.g. [308]).

The other aspect worth elucidating is the automated modeling process. The most significant CASP-4 homology modeling targets were submitted to the HOMER server (see Chapter 10) for benchmarking. Table 11.4 uses the program CE [247] to compare the results of the automated process with the models submitted to CASP-4.

The automatic results are quite similar to the manually edited ones, supporting the quality of the automated model building process. For T0122 the program CE divides the structural superposition in two fragments. When averaging the RMSD of both fragments the difference in RMSD becomes less significant (i.e. 1.59 vs. 1.74 Å). An interesting case is T0128 where the model submitted to CASP-4 obviously contained some easily corrected alignment error, supporting the validity of automating the model generation process.

PDB Code	Time	$N_{residues}$	$N_{rotamers}$	Class
2end	29:28	118	2722	α
1amm	33:51	158	3145	$\left\{\begin{matrix}\beta,\\ \beta\end{matrix}\right\}$
2ihl	12:28	106	2074	α
2erl	0:21	34	418	α
1ptx	2:32	54	981	α/β
1plc	4:15	82	1156	β
5rxn	1:14	48	724	β
1igd	1:59	50	926	α/β
1whi	18:09	101	2587	β
1xnb	13:28	151	1938	β
2hbg	10:02	97	2034	α
1arb	23:50	202	2744	$\left\{\begin{matrix}\beta,\\ \beta\end{matrix}\right\}$
1ctj	3:08	61	1021	α
1cex	22:07	146	2556	α/β
1crn	0:19	37	355	α/β
2cro	4:54	56	1435	α
1ctf	3:36	47	1301	α/β
4fxn	13:25	118	2193	α/β
1lz1	18:04	105	2302	α
3app	31:26	259	2412	$\left\{\begin{matrix}\beta,\\ \beta\end{matrix}\right\}$
3rn3	12:37	109	1958	α/β
3tln	62:23	252	3562	$\left\{\begin{matrix}\alpha/\beta,\\ \alpha\end{matrix}\right\}$

TABLE 11.2. Side chain placement benchmark, part I. The PDB code and time (min:sec) required to optimize the side chains is reported. $N_{residues}$ is the number of residues and $N_{rotamers}$ is the number of rotamers to optimize in the structure. Note that Gly and Ala residues are not counted. Class is the CATH classification (α, β or α/β). Two-domain structures have their two classes indicated in brackets.

PDB Code	$RMSD_{start}$	$RMSD_{opt}$	$\chi 1_{start}$	$\chi 1_{opt}$	$\chi 12_{start}$	$\chi 12_{opt}$
2end	3.22	2.44	58.5	73.7	41.5	49.2
1amm	2.91	2.74	58.9	79.1	44.9	53.2
2ihl	2.70	2.25	61.3	84.0	45.3	66.0
2erl	3.51	2.92	58.8	70.6	44.1	52.9
1ptx	3.68	2.64	53.7	72.2	42.6	50.0
1plc	2.98	1.93	50.0	80.5	36.6	53.7
5rxn	3.51	2.00	52.1	75.0	33.3	50.0
1igd	2.81	1.60	50.0	78.0	36.0	64.0
1whi	2.51	3.22	61.4	70.3	49.5	51.5
1xnb	3.78	2.27	51.0	76.8	39.1	54.3
2hbg	2.43	2.45	63.9	78.4	50.5	53.6
1arb	3.05	2.07	58.9	78.2	47.5	58.4
1ctj	3.12	2.64	65.6	75.4	47.5	47.5
1cex	3.05	2.26	58.2	81.5	45.9	61.6
1crn	3.30	0.88	67.6	97.3	56.8	86.5
2cro	3.19	2.52	55.4	75.0	37.5	55.4
1ctf	1.89	1.95	68.1	74.5	55.3	55.3
4fxn	3.44	2.18	48.3	69.5	28.8	44.1
1lz1	2.79	2.06	61.0	81.0	46.7	58.1
3app	3.22	1.46	58.7	79.5	47.1	57.9
3rn3	3.03	2.00	53.2	76.1	41.3	58.7
3tln	3.20	1.97	53.6	75.0	37.3	49.6

TABLE 11.3. Side chain placement benchmark, part II. The starting and optimized structures are benchmarked in terms of RMSD, $\chi 1$ and $\chi 12$ correct. See text for a description of the measures.

Target			Manual			Homer		
Id	S_{Id}	F_{Sup}	RMSD	N_{ali}	Z-score	RMSD	N_{ali}	Z-score
T0099	34	84	4.22	53	2.6	4.61	55	2.8
T0111	52	96	1.58	430	8.0	1.46	418	8.0
T0113	28	92	2.18	241	7.2	2.08	234	7.0
T0122*	32	95	1.53	160	6.8	1.46	158	6.7
			1.74	63	4.6	2.47	62	3.9
T0123	68	85	2.97	151	6.2	2.78	143	6.3
T0125	18	81	2.87	124	6.1	2.86	122	5.9
T0128	59	97	2.20	188	7.1	1.46	190	7.1

TABLE 11.4. Benchmark for the automatic model generation. The models submitted to CASP-4 are compared to the automatic prediction made by the *HOMER* server. Both are superposed to the experimental structure of the target with CE [247]. The RMSD and Z-score (i.e. "similarity") are reported. S_{id} is the percentage sequence identity between the best template and the target. F_{sup} is the percentage of the target that is structurally equivalent to the best template. N_{ali} is the number of residues aligned by CE. * Due to a missing loop in the experimental structure CE treats T0122 as two separate fragments.

Typical times for server generated models returned by E-mail are about 1-5 minutes for alignment generation and raw model generation. These rise to about 5-15 minutes including loop modeling and 10-45 minutes in total for models including side chain optimization.

11.3 Discussion

Structure prediction methods in CASP-4 have reached a point at which reasonable models for proteins can be found on a routine basis, particularly if a homologous fold can be detected. For all but the easier targets the best predictions are achieved when there is the opportunity for manual intervention to incorporate expert knowledge concerning structure and function. This is difficult to encapsulate in algorithms at the moment. Building a consensus from the predictions of several methods therefore yields a significantly higher success rate for difficult targets.

The performance of the methods developed in this thesis has been presented based on the CASP-4 results. This comparison is the most thorough and realistic determination of the state of the art possible, since all leading protein structure prediction groups participate. Some groups, such as Baker or Sternberg, have well over ten years of experience in structure prediction. They have been among the top-ranking groups for several CASP experiments in a row. Measuring up with them for a new group participating in CASP for the first time can be considered an extraordinary success. The methods developed in

this thesis have proven to be state of the art in a very competitive field, performing better than some of the most successful groups from CASP-3 (e.g. Jones, Fischer or Levitt) [2].

Despite these good results, it is important to ask which were the most relevant problems in the CASP-4 approach and how to avoid them in the future. Comparing the results of our group to those of others it become apparent that the quality of our alignments was inferior. This class of errors can be divided in at least two categories.

The first category of errors are mistakes introduced by manual model construction. E.g. in our T0121 model, an otherwise accurate prediction was spoiled by shifting the alignment by two residues. This happened because the model generation process involved calling a program with parameters for alignment start and end. These were counted manually from the alignment file. The increasing pressure at the end of CASP, where more than one model had to be finished in the middle of the night to meet an imminent submission deadline, led to some miscounts. Such an erroneous model for a relatively "easy" target automatically causes a major loss in terms of ranking. This category of errors is easy to avoid by automating those parts of the modeling process where human intervention can only cause additional errors. This has been achieved as part of this thesis.

The second, and more insidious, problem concerns the program used to align two sequences. During CASP-4 we used CLUSTALW [136] to compute pairwise alignments. This approach was warrantable as our fold recognition software lacked a dedicated alignment module. According to Sauder et al. [249] this alignment strategy is inherently inferior to others for low sequence similarity cases. CLUSTALW calculates a global alignment, i.e. it will always align the entire sequences no matter if they are related or not. For multi-domain proteins we avoided aligning unrelated fragments by cutting the sequence along its probable domain boundaries. This strategy solved some part of the problem, but did not improve alignment on the remaining sequence fragment.

The situation is even more complex for fold recognition targets where no homology is detectable. CLUSTALW will align parts of the structure that cannot be aligned because they are unrelated, e.g. helices or strands outside the structural core. All the information derived from sequence profiles and/or secondary structure prediction is ignored and reduced to a mere sequence similarity measure. This explains why in fold recognition, our group in comparison to others scored better on the harder targets, i.e. in a hit or miss situation, than on easier targets, i.e. where most points are awarded for alignment accuracy.

The problem could be alleviated by using PSI-BLAST profiles or HMMs as seed alignments for manual improvement. Simultaneous alignment of sequence and secondary structure would have been an option for targets without de-

tectable homology to the template. In the long run it is desirable to develop a dedicated alignment module that takes into account the alternatives sketched above and automatically decides how to select the most probable alignment.

The initial errors presented above derived from a lack of experience with certain aspects of structure prediction. Having learned to avoid such errors was the one of the "achievements" of the best-performing groups between CASP-1 and CASP-2. With the experience gained during the CASP-4 experiment, it should be possible to perform better in the next CASP-5 experiment.

11.4 Summary

The results are presented largely in terms of what our group has achieved during the CASP-4 experiment, since the protocols used by our group were developed by this author. With predictions for all 43 targets submitted to the assessors, the results have been evaluated in all three categories: homology modeling, fold recognition and *ab initio*. Only the fold recognition [309] and *ab initio* [310] assessment papers were available as preprints at the time of writing.

Results for homology modeling were described as inconclusive by the assessor, with our group ranked in an unspecified place among the better predictions. In fold recognition our group ranked 15th with 22.0 points out of 125 participants. Almost half of these (56 groups) had a score $\leq$ 5.0. Despite not having submitted *ab initio* predictions, our group ranked 21st in the overall *ab initio* ranking and 9th when considering only novel folds. Both the fold recognition and *ab initio* categories were dominated by the Baker group's outstanding predictions.

Results for four selected targets are described in more detail. Two of these (T0111, T0122) were homology modeling targets. Of the remaining two targets, one was a difficult fold recognition case (T0107) where our group had the highest score. The other (T0116) is perhaps the most difficult CASP-4 target. This 811-residue protein contained four different domains, ranging from homology modeling to difficult fold recognition and *ab initio*. Our group was one of the very few submitting an assembled prediction of all four domains. The quality of this model is perhaps only second to that of the Baker group.

In order to evaluate the side chain placement algorithm and the performance of the automated modeling protocol, which was not available during CASP-4, further tests are described. These support the idea that both parts of the approach can be considered state of the art.

The two problems encountered in CASP-4, process automation to avoid human errors and alignment accuracy are discussed. Putting the results achieved by our group into the context of starting from scratch and participating for

the first time in the CASP experiment and nevertheless ranking higher than very experienced groups, it is fair to say that this thesis has achieved all that could be realistically expected.

Part III

Loop Modeling

12
State of the Art

Loops, the structurally most variable regions parts of the protein backbone cannot be modeled from a template structure. Their prediction remains one of the main problems in comparative protein modeling [2]. As was described in Section 7.1, loop modeling is a major focus of the present thesis. Before the novel algorithm developed in this thesis can be introduced, the problem has to be defined accurately and the state of the art described.

12.1 Loops

Loops can be defined as the parts of the protein outside regular secondary structure. A typical globular protein contains approximately two-thirds of its residues in helices and strands and one-third in loops [110]. Loops serve to connect secondary structures, but may additionally have functional and structural roles. In some proteins, loop residues are part of the active site and interact with ligands and cofactors.

Loops often show the greatest flexibility in amino acid sequence and are usually less restrained in conformation than the core regions. Most insertions and deletions between homologous proteins are located in loops. For protein structure prediction this means that they cannot easily be taken from the parent structure during model building. Unless the loops are structurally conserved across related proteins, their conformation has to be predicted.

This problem occurs for both homology modeling and fold recognition targets. In homology modeling one is especially interested in modeling loops to complete the information about the active site. Generation of a set of solutions within minutes is acceptable and allows the user to include information that

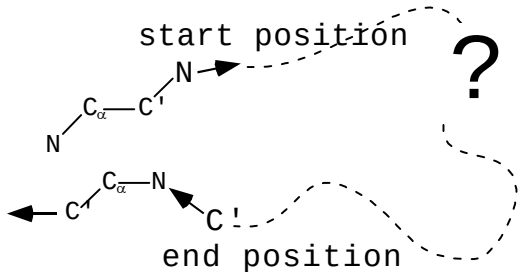

FIGURE 12.1. The problem setting of loop modeling.

cannot be expressed algorithmically. Modeling experts prefer to have a set of rapidly generated alternatives rather than relying on a slow method producing a single answer.

Due to the complexity of predicting a loop, and lack of a fast and efficient method, loops are to date not modeled in fold recognition. With many more fragments of the protein that cannot be modeled from the template, prediction would greatly benefit from a fast and robust loop modeling method. With the more coarse modeling in fold recognition, the loop prediction would have to be fast rather than accurate. It should also be able to produce solutions for a broad range of more or less deformed local geometries.

The following definition can be given for loop modeling: Given the position of two *anchor regions*, consisting of at least one residue flanking the loop on the N- and C-terminus, and a fixed number of residues of specific type (i.e. the loop). How is it possible to construct conformations that fit the geometry of the *anchors* and ideally have the native conformation? This is shown in Figure 12.1.

The main problem is the generation of a good set of alternative structures which have to be evaluated with a scoring or energy function. A number of different approaches have been investigated in the literature to tackle this problem, which can be divided in at least two categories: *ab initio* and database methods. Evaluation of the candidates can be considered a problem of its own.

Before the methods described in the literature can be introduced, it is important to define the criteria for evaluation. The accuracy of a loop prediction is evaluated by comparing it with the native conformation, taken from an experimental structure. Due to the inherent flexibility of loops, the conformations of loops determined with X-ray crystallography show a resolution-dependent behavior. It is well known that the errors in loops from *PDB* files rise quickly above 2.5 Å X-ray resolution [77]. This has to be taken into account when comparing loop predictions with *PDB* structures.

A variety of reasonable criteria for comparing loop conformations exist, with a variation of the RMSD being the most common. It is further possible

to distinguish between "**local**" and "**global**" RMSD. The former considers a superposition of the two loops to calculate the relative internal deviation, whereas the latter superimposes the whole structure to calculate the overall displacement of the two loops. It is apparent that "**local**" RMSD will be lower than "**global**" RMSD, as it excludes the possibility that the loop conformation may be correctly predicted, but poorly oriented to the rest of the protein. As has been argued by Fiser et.al. [46] the two measures are correlated, with "**global**" RMSD on average being equivalent to at least 1.5 times "**local**" RMSD. In the present thesis we have based our observations on "**global**" RMS, as it is the stricter measure and also solves the optimization problem of defining the correct orientation of the loop towards the protein framework. Results measured with "**global**" RMSD are immediately applicable for comparative modeling.

The actual RMSD is calculated on the backbone atoms. Unfortunately different definitions exist in the literature. The N, C_α and C' atoms are always included. C_β is generally not included, but inclusion of the O atom depends on the respective publication. Recognizing this frequently unspecified detail is important, since it significantly alters a comparison: The same results vary around 0.2-0.4 Å depending on the inclusion of the O atom. Whenever RMSD values are described in the following, inclusion or exclusion of the O atom will be stated in parentheses.

12.2 *Ab Initio* Methods

Ab initio methods aim to predict the conformation of loops using only knowledge about the geometry and energy of the loop, without reference to experimentally solved structures. They can be further divided into at least three subcategories: analytical, combinatorial, and energy minimizing.

Analytical methods try to solve the loop modeling problem by geometrical transformations, solving a set of equations. They date back to the pioneering work of Go and Scheraga [3], who found that it is possible to determine the conformation of fragments with up to six rotable torsion angles using rigid geometry (i.e. idealized bond lengths and bond angles).

They define a local coordinate system i with respect to some arbitrary origin. Bond lengths d_i, bond angles θ_i and torsion angles ω_i are shown in Figure 12.2 (*left*). In this coordinate system, the positions of atoms $i-1$, i and $i+1$ are $(0,0,0)$, $(d_i,0,0)$ and $([d_i + d_{i+1}cos\theta], d_{i+1}sin\theta_i, 0)$ respectively. A given point in space can be expressed by position vectors r_i and r_{i-1} with respect to the ith and $(i-1)$th coordinate system by the following relation [3]:

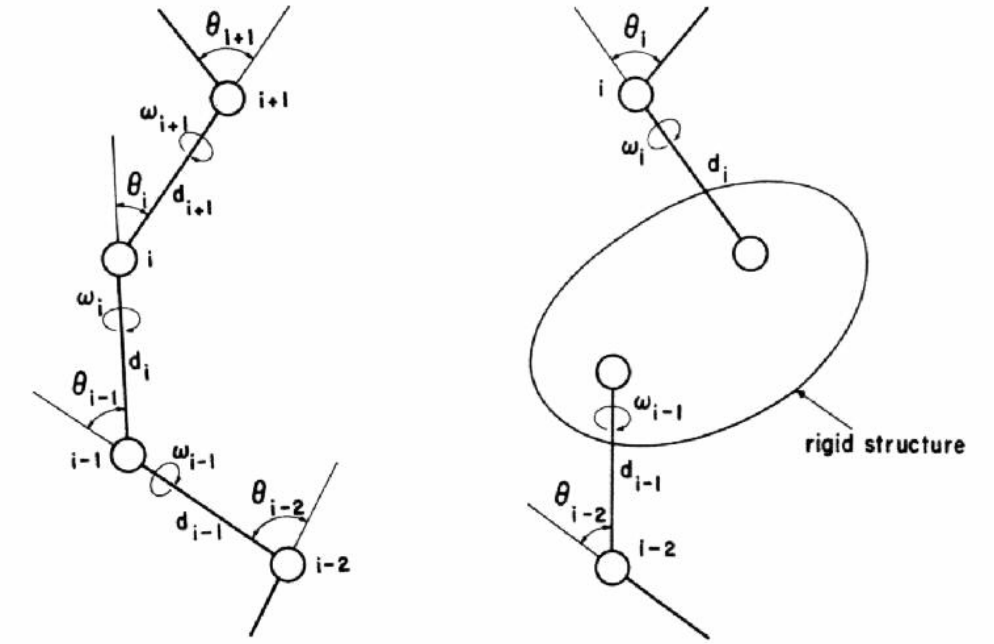

FIGURE 12.2. Analytical loop closure. Definitions of bond length d, bond angle θ and torsion angle ω for loop closure.

$$r_{i-1} = \mathbf{T}_{i-1}\mathbf{R}_i r_i + p_{i-1} \tag{12.1}$$

where

$$\mathbf{T}_{i-1} = \begin{pmatrix} \cos\theta_{i-1} & -\sin\theta_{i-1} & 0 \\ \sin\theta_{i-1} & \cos\theta_{i-1} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{12.2}$$

$$\mathbf{R}_{i-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega_i & -\sin\omega_i \\ 0 & \sin\omega_i & \cos\omega_i \end{pmatrix} \tag{12.3}$$

The vector $p_{i-1} = (d_{i-1}, 0, 0)$ describes the translation between the two coordinate systems. In case of a protein, the peptide torsion angle (i.e. along the $C - N$ bond) is assumed to be planar and does not require a coordinate system. This rigid structure is shown in Figure 12.2 (*right*).

Knowing the position of two amino acids to connect with a chain of fixed length is equivalent to finding a series of coordinate transformations that transform one endpoint coordinate system into the other. This requires six degrees

of freedom, three for translation of the origin and three for the rotations necessary to superimpose the axes [3]. Since there is one free torsion for each transformation, six free torsions are necessary to solve the problem. With each residue having two free torsions, three amino acids can be predicted.

A number of publications have further addressed this problem [4][5] [6][7], but the results show that no generalized analytical solution beyond six torsion angles is possible [3][7]. Bruccoleri and Karplus [8] have extended the approach, solving small fragments analytically and enumerating the solutions of larger ones.

Combinatorial approaches for loop modeling have been studied by several groups [9][10][11][12][13][14][15]. The allowed positions of loop residues are enumerated. Discretization of solution space is required to limit the combinatorial explosion. A restricted set of (φ, ψ) torsion angles is used to approximate all possible conformations. This ranges from uniform conformational sampling, where each torsion angle is sampled in fixed intervals (e.g. in steps of 60°), to distributions biased towards more populated regions of the (φ, ψ) map. In addition, techniques to limit the combinatorial explosion have been described, e.g. pruning parts of the search tree which are too far apart in 3D space to be spanned. The search algorithm can either generate the conformations on the fly or separately from modeling. For example, Sudarsanam et al. [58] use a previously compiled database of all possible dimers (i.e. two-residue fragments) to construct loops in a similar way to enumerative methods.

Number	φ angle	ψ angle
1	−63	−40
2	−125	135
3	−85	75
4	−78	149
5	−95	−5
6	55	40
7	85	5
8	−85	175

TABLE 12.1. Eight torsion angle pairs selected by Deane and Blundell [15] for loop modeling.

Recently Deane and Blundell [15] have presented an interesting combinatorial method to predict loop conformations up to eight residues in length. Analyzing loop segments found in the PDB, they derive a set of eight carefully chosen (φ, ψ) torsion angles, shown in Table 12.1. It represents over 96% of all possible five residue fragments with less than 1 Å RMSD. A database

enumerating all combinations of these eight (φ, ψ) torsion angles up to twelve residues in length is generated and stored, requiring over 3 GB of disk space[1].

The search algorithm uses a two residue overlap on each side of the loop to scans the database and select fitting fragments, according to the distances between C_α atoms of the overlapping residues. Loops up to eight residues long can be predicted. The average **global** backbone RMSD (*including O* atom) ranges between 1.4 Å for three residue loops and 3.9 Å for eight residue loops. The computation for one loop requires up to 20 minutes [15].

Many energy minimization methods have been proposed, ranging from incorporating geometrical considerations to more or less pure minimization of an energy function. Methods relying on local optimization of the geometrical structure are the minimum perturbation "random tweak" [16] [17][18] or "local moves" [19]. Both try to minimize the difference in torsion angle positions at the anchor regions of the loop with an iteration of small adjustments.

A different approach is taken by the "scaling relaxation and multiple copy sampling" series of papers [38][39][40][41][42] [43][44]. Model construction is initiated by placing the atoms very close together (*scaling*). An iterative energy minimization procedure guides the gradual increase of bond lengths and bond angles (*relaxation*) to standard values. Less geometrical information is used in the "importance sampling by local minimization of randomly generated conformations" [20][21][22] and "global energy minimization by mapping a trajectory of local minima" [23][24] approaches.

Methods relating to the optimization of an energy function include molecular dynamics simulations [25][26][27][28], Monte Carlo and molecular dynamics [29], biased probability Monte Carlo search [169][31][32], Monte Carlo with simulated annealing [33][34][35][36][37] and self-consistent mean field optimization [124]. The resulting loop conformations constructed with any of the energy minimization methods may cover only a subset of the solution space and are not necessarily close to the native structure.

Perhaps one of the most typical energy minimization approaches was recently published by Fiser et al. [46]. They initialize the modeling process by placing all atoms on a straight line between the anchor residues and displacing them randomly with some predefined maximum distance. Conjugate gradient optimization and Monte Carlo simulated annealing or molecular dynamics are iteratively applied to "guide" the random start conformation into an energetically favorable one. They modify CHARMM [60] as energy function to allow atoms to pass "through" each other during the early simulation stages. The method is typically applied to 500 independent simulations to generate a ranking. Each modeled loop requires up to 30 hours CPU time and the av-

[1] C. Deane, personal communication.

erage **local** backbone RMSD (*including O* atom) varies between 0.59 Å for 4 residue loops, 1.16 Å for 8 residue loops and 2.61 Å for 12 residue loops [46]. While this method appears to be very accurate it is generally not suitable for to comparative modeling or fold recognition: Considering a typical modeling target with up to a dozen or more loops would require weeks of computation for a single model. This is beyond the scope for typical modeling applications. Rather it should be considered a good benchmark for quantifying what is possible to predict.

12.3 Database Methods

Database methods aim to predict the conformation of loops using knowledge from experimentally solved structures. The underlying assumption is that the possible conformations can be reduced into a representative subset, which can be extracted from the *PDB*. These methods can be further divided into at least two subcategories: fragment-based and taxonomy-based.

The idea of fragment-based methods was developed by Jones and Thirup [48], who selected fragments from the *PDB* for electron density fitting in X-ray crystallography based on geometric criteria. Similar approaches have been used in loop modeling [49][50][51][52][167][54][55][56][57]. Fragments are selected from a database of many known structures based on overlap with the framework on both ends and sorted according to geometric criteria or sequence similarity. Different definitions of anchor regions, i.e. overlap between loop and framework, ranging from one to three residues were used in the literature. Using more than one residue for overlap was shown to improve discrimination of good candidates, but reduces the number of available loops. Coverage of conformational space for fragment databases is known to deteriorate quickly for lengths above 5-6 residues, making prediction of longer loops difficult [14]. The overlap between fragment and framework alone is unlikely to yield satisfactory results [52].

An algorithm combining database and combinatorial search has been proposed by Martin et al. [30]. Short loops are predicted by the mixture of combinatorial search and analytical method presented by Bruccoleri and Karplus [4]. Long loops, where a combinatorial search would be ineffective, are first approximated by selection of good candidates from a database of backbone conformations. The central part of the loop is again predicted with the method of Bruccoleri and Karplus [4]. Since the central part of a loop shows the greatest flexibility, this approach tends to reduce the limitations of an incomplete fragment database.

Van Vlijmen and Karplus [59] have tested ways to improve the performance of database methods by means of energy optimization. They use the

CHARMM energy function [60] to minimize a set of candidate loops extracted from the PDB. They report a **global** RMSD (*excluding* O atom) better than 1.07 Å for 8 out of 18 target loops. These figures refer to optimization of the 50 candidate loops closest to the target loop (i.e. lowest RMSD) and may therefore be overly optimistic [59]. Since the method also requires up to about 30 hours computation time, the same restrictions pointed out for Fiser et al. [46] apply (see Section 12.2).

As with tertiary structure, loops have been subject of study for the development of taxonomies. Certain conformational classes have been identified. An example of clearly classifiable loops are the β-hairpins [68], short 3- or 4-residues loops connecting two β-strands.

Since database methods are able to approximate most of the antibody hypervariable loops quite closely, this suggested that these proteins form a specific sub-space of folding based on certain key residues allowing easy classification [61][49]. The concept of key residues states that the entire loop conformation is dominated by a particular residue, e.g. a proline, which drastically limits the available conformational space. Antibody loops form similar structures, allowing a strict classification based on key residues [49][56][66][67]. This concept has been generalized to determine the conformation of other loops, but only with limited success [62][63][64][65].

Many groups have developed classification methods for loops [57][63][64][68] [69][70][71][72][282]. The most common criteria for classification include sequence, loop length, torsion angle conformation and type of adjacent secondary structure. The SLoop database [282] for example divides loops into 560 well-populated classes. In addition to the usual classification criteria, hydrogen bonding patterns and solvent accessibility are analyzed. Also, rather than using sequence information alone, environmentally constrained substitution tables are employed: Probabilities, depending on residue type and (φ, ψ) angle, are derived from a database of homologous proteins to quantify the similarity of loops in terms of homology. The correct structural class is predicted in 35% of cases, rising to 65% considering the top three answers. These results correspond to an average **global** RMSD (*including* O atom) of 2.6 Å for five residue loops.

Perhaps one of the most representative loop classifications applicable to comparative modeling is the one described by Wojcik et al. [65]. Using a database of 13,563 loops extracted from *PDB* structures with less than 95% sequence identity, they derive a clustering of loop families connecting secondary structures. This is analyzed for sequence conservation, conformations and endpoint C_α distances. They find significant preferences for sequence patterns to adopt certain loop conformations. Depending on similarity at the

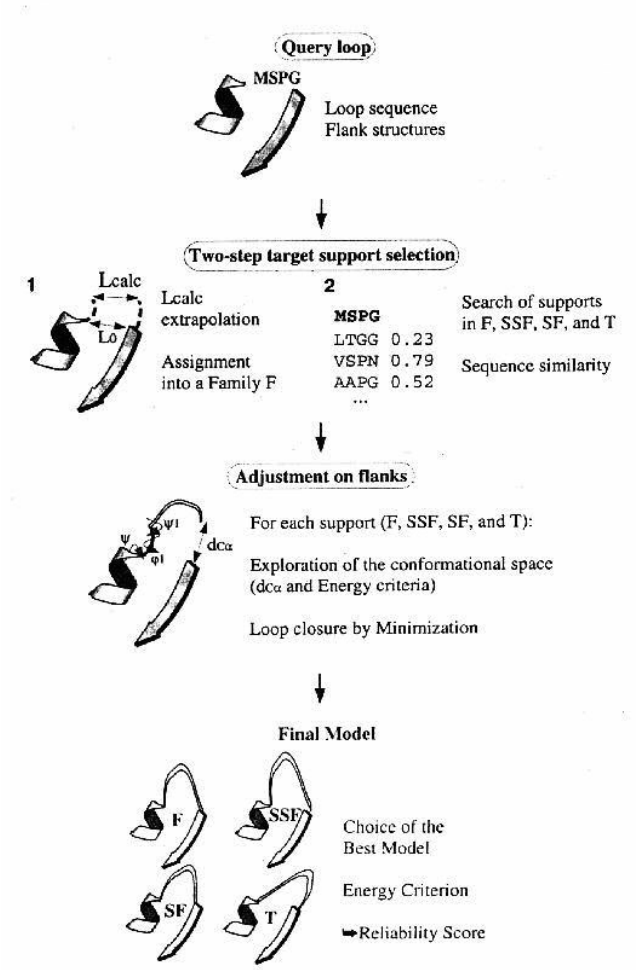

FIGURE 12.3. Loop modeling using a database classification.

anchor regions (i.e. RMSD), they assign each database loop to a tree of hierarchical similarity levels: families (F), sub-superfamilies (SSF), superfamilies (SF) and all (T).

A query loop is assigned one representative structure from each of the four levels (F,SSF,SF,T) depending on sequence and anchor region compatibility. These four candidates are fitted to the framework and ranked according to an energy criterion, as shown in Figure 12.3. The method has been benchmarked against the loop database using a Jackknife test (i.e. prohibiting the method to select the query loop). The average **global** backbone RMSD (*excluding O* atom) ranges between 1.1 Å for three residue loops and 3.8 Å for eight residue loops. Computation for a single loop requires around 1 minute [65] and is therefore well-suited for comparative modeling.

12.4 Ranking

A problem common to all loop modeling methods is how to rank the candidate loops and select the presumably best one. Whether the candidate loops are taken from existing structures or artificially generated, two easily measurable characteristics exist: geometry and energy.

Geometry refers to the deviation of the modeled loop compared to the anchor regions. Except for analytical methods, which guarantee a perfect match, all other methods will produce candidate loops with different deviations from the framework. The conformation of short loops is strongly determined by the necessity to match the framework [9], so it is possible to infer a correct conformation from a good geometric fit. This signal becomes weaker for longer loops and is not sufficient to discriminate loops longer than 5-6 residues [9]. A sample geometric fit is shown in Figure 12.4.

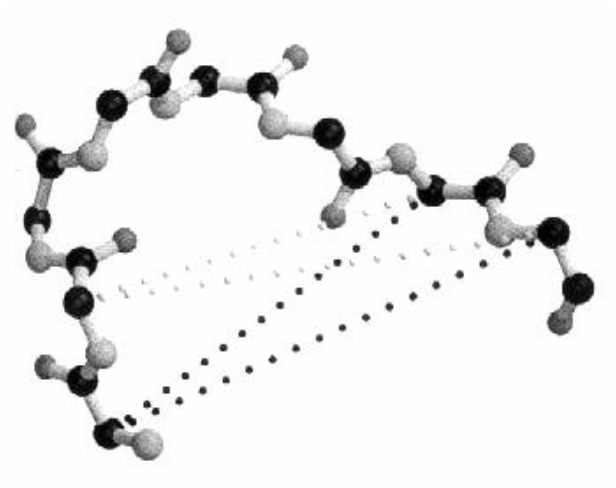

FIGURE 12.4. Sample anchor fragment distance definition. Two C_α atoms on each side of the loop are used to derive a total of four distances.

Calculating the energy of a candidate loop is also a common way to assess the quality of solutions. Since the native loop structure should give a low energy under many energy functions, it is reasonable to select the candidate with the lowest energy score as the most probable solution. Different energy functions have been employed for loop modeling. These range from force fields such as CHARMM (e.g. [46][59]) to knowledge-based potentials. Deane and Blundell have found the RAPDF potential [78] to discriminate well among candidate loops. This is in agreement with the original publication of Samudrala and Moult [78], which showed the potential to correctly discriminate 10 out of 11 loop structures within 1 Å of the best solution in a decoy set.

Amino acids are known to have different preferences for areas of the Ramachandran map, sometimes known as propensities. Measuring the (φ, ψ) torsion angles of candidate loops is a way to estimate the quality of a solution. Some methods use this information to guide the minimization process (e.g. [59]) or to select good database fragments (e.g. [65]), but it can also be used for ranking [15]. Wojcik et al. [65] even define special propensities for the first and last residue in a loop, since these have a different distribution than central residues.

Other less frequent characteristics are hydrogen bonds and solvent accessibility (e.g. [282]). Topham et al. [54] use a combination of hydrogen bonding possibilities, solvent accessibility, and (φ, ψ) torsion angles to derive a set of environmentally constrained substitution tables. These tables calculated from a database of homologous proteins express the probability of a particular residue being mutated from type A to type B in two structures. It can also be used as a measure for database methods.

Whenever using several measures, the question of how to combine them into a single ranking arises. For loop modeling, this is usually limited to a combination of two approaches: filters and linear scoring functions.

Prior to ranking, some solutions may be eliminated, because they do not fulfill a strict criterion. For example, Deane and Blundell [15] use propensities to sieve out very improbable conformations for a sequence (e.g. a proline residue in a disallowed region of the Ramachandran map).

The raw scores are computed by multiplying the single measures with a scaling factor and adding them. The ranking is made by sorting according to raw scores. More complex classification schemes, such as the Z-scores used in fold recognition are generally not employed for loop modeling.

12.5 Summary

Loops are the structurally variable regions outside regular secondary structure. They usually cannot be copied from the template structure during modeling and have to be predicted. Loop modeling is important for both homology modeling and fold recognition. It is not yet used for the latter due to insufficient robustness and speed.

The problem can be stated as finding a way how to connect two anchor regions using the chain corresponding to the loop sequence. Two main classes of approaches for loop modeling exist: *ab initio* and database methods. When comparing the predicted loop with its experimental structure, it is important to distinguish between different definitions of RMSD in the literature: global or local superimposition, with or without the main chain O atom.

Many alternative *ab initio* methods have been described. For up to three residues, it is possible to deduce the loop conformation by solving a series of geometric transformations. Conformational space for longer loops can be enumerated with some simplifications or a global optimization used. A good enumerative method using a set of eight torsion angle pairs was recently published by Deane and Blundell [15]. It computes accurate solutions in minutes. For global optimization, Fiser et al. [46] have developed a method finding very accurate solutions in 30 hours, which is currently out of scope for typical modeling applications.

It is possible to distinguish among database methods between fragment-based methods and taxonomies. The former concentrate on finding a set of representative loop fragments in the *PDB* to use for loop modeling. A method combining such loop fragments with global optimization was described by van Vlijmen and Karplus [59]. It produces accurate solutions in a similar time frame to Fiser et al. [46], but the same time limitations apply. Taxonomies instead concentrate on classifying structural families of loops in a similar way to tertiary structure classifications. A recent catalogue of loop structures is the SLoop database of Burke and Deane [282]. Direct application to loop modeling of an exhaustive loop classification has been described by Wojcik et al. [65]. Their method is able to produce accurate solutions in a matter of minutes.

Ranking of the candidate loops is usually restricted to a combination of geometric fit of the anchor regions and an energy function. Another less frequent criterion is the sequence-dependent propensity for areas of the Ramachandran map. The ranking is generally computed as a linear combination of single criteria, with the possibility to filter out impossible solutions beforehand.

13
Approach

The novel loop prediction method developed in this thesis is based on the so-called divide & conquer approach. The basic algorithm will be introduced and the underlying assumptions discussed in the following. A special representation of the loop conformation based on vectors is given together with the necessary operations for the divide & conquer algorithm. The details of its implementation will be presented in the next chapter.

13.1 Divide & Conquer

As has been established in the previous chapter, current database methods using solely experimentally determined loop fragments do not cover all possible loop conformations, especially for longer fragments. On the other hand it is not feasible to use a combinatorial search of all possible torsion angle combinations. For an algorithm to be efficient, a compromise has to be found.

One improvement in *ab initio* loop modeling is the use of look-up tables (*LUT*) to avoid the repetitive calculation of loop fragments. LUTs can be generated once and stored, only requiring loading during loop modeling. Using a set of LUTs reduces the computational time significantly.

The next problem is how to best explore the conformational space. Especially for longer loops, it is useful to generate a set of different candidate loops to exclude improbable ones by ranking. The method should therefore be able to select different loops by global exploration of the conformational space independently of starting conditions. Methods building the loop stepwise from one anchor residue to the other bias the solutions depending on choices made

in conformation of the first few residues. Rather a global approach to the optimization is required.

This criterion is fulfilled by the divide & conquer algorithm, which is recursively described by the following steps [289]:

1. if $start = end$, compute result;
2. else use algorithm for:
   (a) *start* to $end/2$
   (b) $end/2$ to *end*
3. combine the partial solutions into the full result.

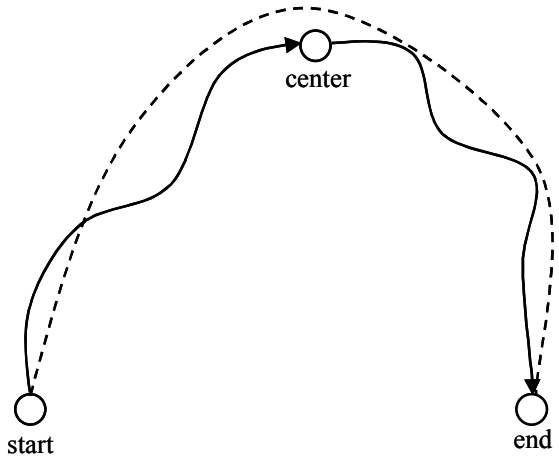

FIGURE 13.1. Loop selection through divide & conquer.

Applied to loop modeling, the basic idea of a divide & conquer approach is to divide the loop into two segments of half the original length choosing a good central position, as shown in Figure 13.1. The segments can be recursively *divided* and transformed, until the problem is small enough to be solved analytically (*conquered*). The positions of main-chain atoms for segments of a single amino acid can be calculated analytically, using the vector representation described below. Longer loop segments can be stored in LUTs and their coordinates extracted by geometrically transforming the coordinates for single amino acids back into the context of the initial problem. To this end we need to define an unambiguous way to represent the conformation of any given residue along the chain and a set of operations to concatenate and decompose loop segments.

13.2 Vector Representation

Using rigid geometry, i.e. idealized values for both bond length and bond angles, the conformation of an amino acid backbone is fully described by the positions of its three backbone atoms, N, C_α and C'. This corresponds to three vectors, one for each atom. The absolute position of any atom in Cartesian space can also be expressed in relation to a neighboring atom. It was decided to represent the conformation relative to the C' atom. Its absolute position forms the end point (EP). The vector from C' to the N atom of the following residue ($N+$) is called end direction (ED), whereas the end normal (EN) is the normal vector of the plane defined by C_α, C' and N, as shown in Figure 13.2. Additional vectors are required to include the conformation of amino acids in the context of the backbone.

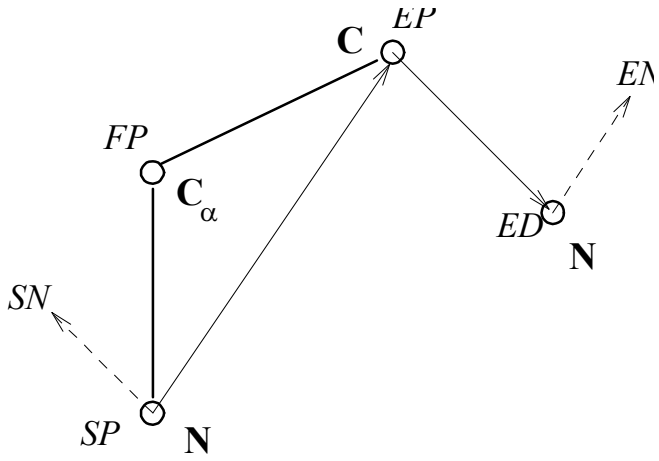

FIGURE 13.2. Generic loop representation relative to some arbitrary origin.

Let us first consider a single residue. The position can be expressed in a local coordinate system that the N atom, the start point (SP), is located in the origin and the C_α atom, the first point (FP), is along the y–axis. The start normal (SN) to the reference plane formed between N, C_α and C' would point in the z–axis for $\varphi = 0°$. The (φ, ψ) torsion angles can be derived directly. Let EP_0, ED_0 and EN_0 be the vector representation for $\varphi = 0°, \psi = 0°$. According to the given definition, EP_0 and ED_0 are in the (x, y) plane. The φ angle is shown in Figure 13.3 and can be expressed as:

$$\varphi = \arccos\left(\frac{EP * EP_0}{\parallel EP \parallel \cdot \parallel EP_0 \parallel}\right) \tag{13.1}$$

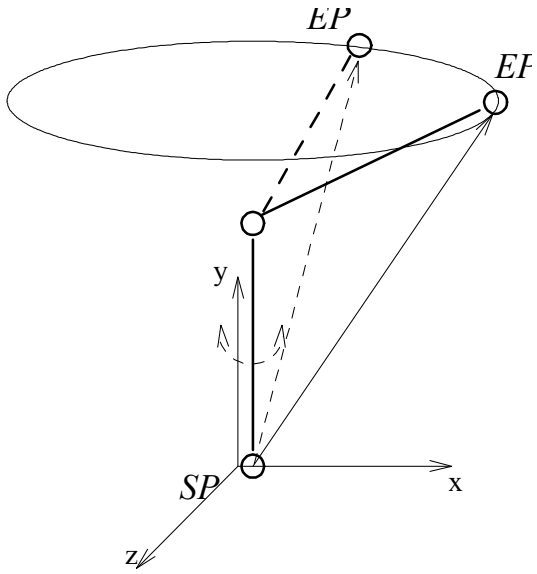

FIGURE 13.3. The φ torsion angle in vector representation.

Let EP', ED' and EN' be the vectors corresponding to rotating the original vectors back into the (x, y) plane, i.e. applying $-\varphi$. The ψ angle is shown in Figure 13.4 and can be expressed as:

$$\psi = \arccos\left(\frac{ED' * ED_0}{\parallel ED' \parallel \cdot \parallel ED_0 \parallel}\right) \tag{13.2}$$

Longer fragments of a polypeptide backbone can be represented as a fixed structure with the same set of six vectors. Three for the first residue (SP, FP, SN) and three for the last residue (EP, ED, EN) in the segment.

This representation allows the definition of the operations to concatenate and decompose loop fragments, by transforming their relative orientation, which is necessary for the divide and conquer method.

13.3 Vector Operations

To introduce the vector operations, it is first necessary to describe the relationship between two connected structures V and W. This is given by the following relationship:

1. $SP_W = EP_V + ED_V$
2. $FP_W = SP_W + B_{N \to C_\alpha}$, where $B_{N \to C_\alpha}$ is the fixed bond length $N \to C_\alpha$ transformed into the context of EN_V

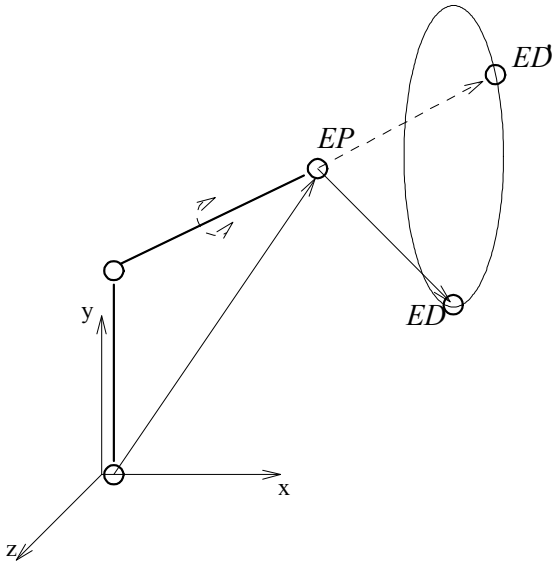

FIGURE 13.4. The ψ torsion angle in vector representation.

3. $SN_W = -EN_V$, because of $\omega = 180°$

This relationship is also shown in Figure 13.5. Given this relationship, the geometric transformations for concatenating two loop segments and decomposing them are defined as follows.

The algorithm for concatenating two segments S ("source") and D ("destination") consists of the following three major steps:

1. Rotate D to be parallel with S:
   (a) Rotate SN_D to superimpose with EN_S.
   (b) Establish the virtual position of the C_α atom of ED_S. This corresponds to the "should be" position of FP_D.
   (c) Calculate the angle between FP_D and the virtual position of the C_α atom. Rotate D around this angle.
2. If the position of D in the chain is even, rotate D by 180° around ED_S to compensate the ω–angle.
3. Translate EP_D by $EP_S + ED_S$.

To concatenate the two segments, the first step consists in orienting the plane at the start of D, given by SN_D, parallel to the end of S, given by EN_S. This can be done by rotating D by angle δ_1 along the axis A_1 as shown in Figure 13.6. The rotation angle and axis are defined as:

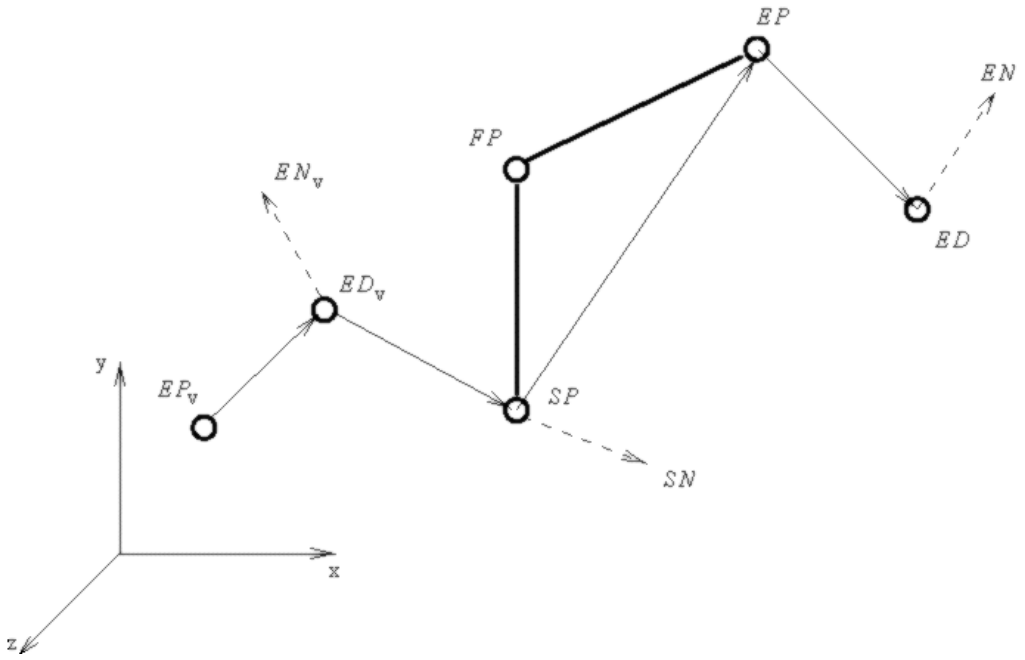

FIGURE 13.5. Generic representation of two concatenated segments.

$$\begin{aligned} \delta_1 &= SN_D * EN_S \\ A_1 &= SN_D \times EN_S \end{aligned}$$

The position of the virtual C_α atom following S, termed RP_S, has to be established. This is done by rotating $-ED_S$ by the bond angle $\beta_{N\rightarrow C_\alpha}$ along the axis EN_S, as shown in Figure 13.7. D has to be made parallel to RP_S by rotating it by the angle δ_2 along the axis A_2 as shown in Figure 13.8. The rotation angle and axis are defined as:

$$\begin{aligned} \delta_2 &= FP_D * RP_S \\ A_2 &= FP_D \times RP_S \end{aligned}$$

Once D is correctly superimposed it remains to consider the ω–angle before the translation of the length of S, that is $EP_S + ED_S$, can be applied. Let us consider the case of $\varphi = 0°, \psi = 0°, \forall\varphi, \psi$. Since $\omega \equiv 180°$, the orientation of the first residue is identical to the orientation of the third and every odd residue as shown in Figure 13.9. The concatenated segment is shown in Figure 13.10.

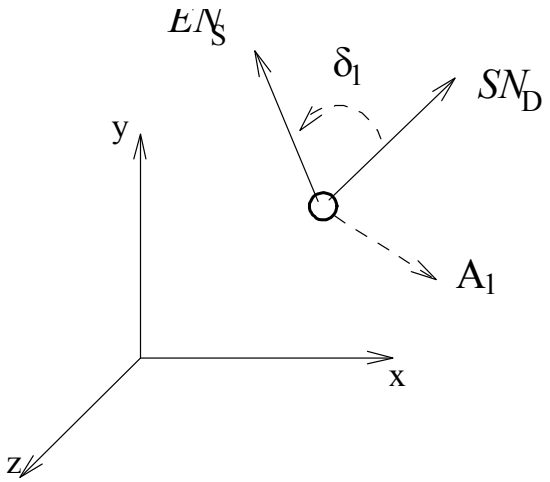

FIGURE 13.6. Step 1a of the concatenation

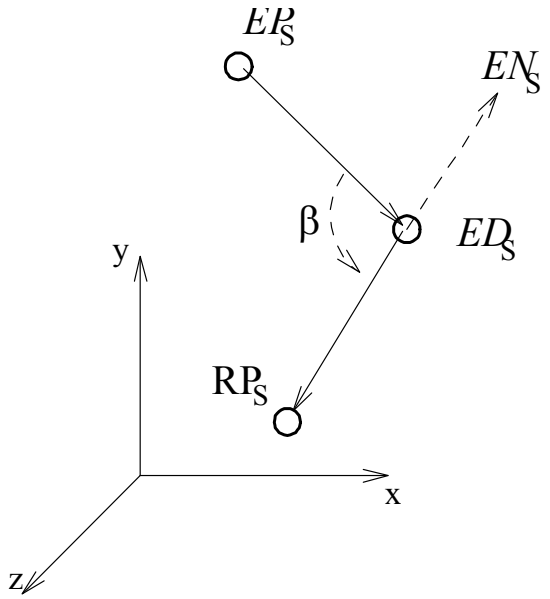

FIGURE 13.7. Step 1b of the concatenation

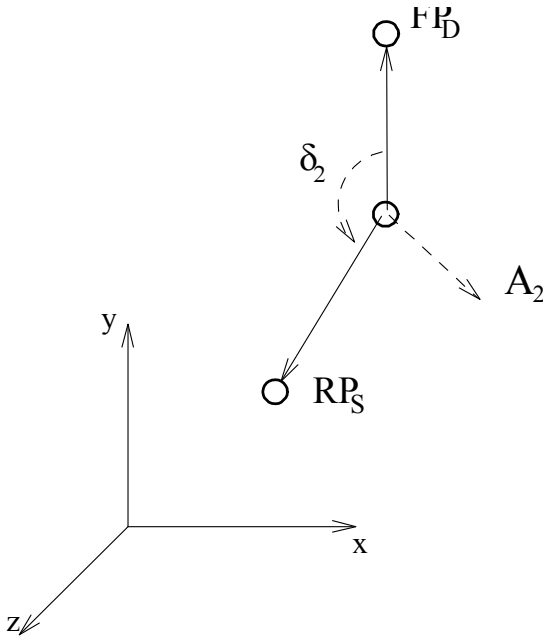

FIGURE 13.8. Step 1c of the concatenation

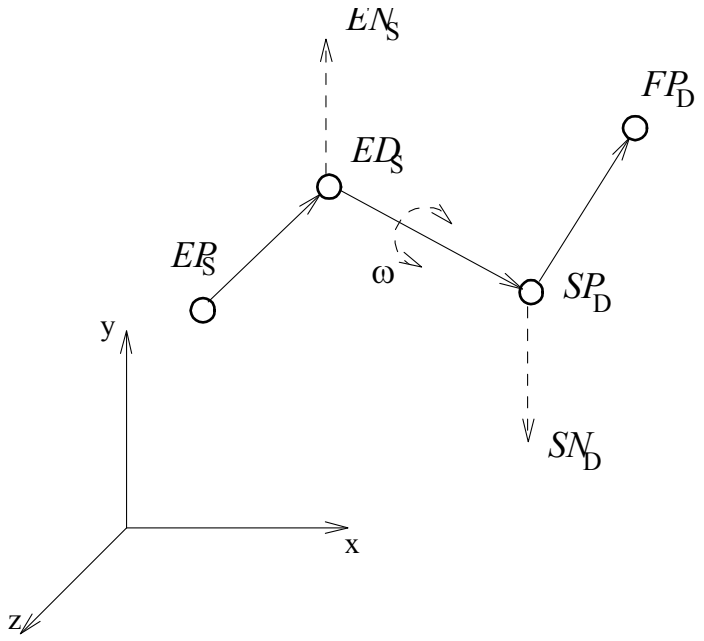

FIGURE 13.9. Step 2 of the concatenation

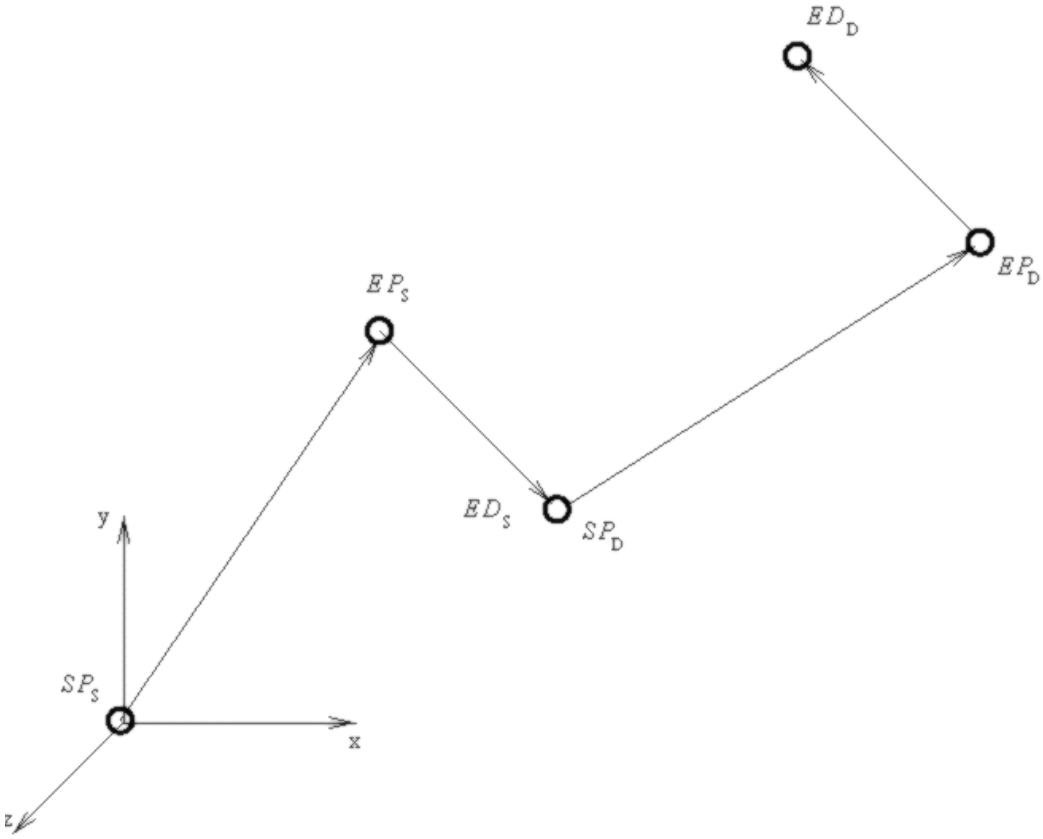

FIGURE 13.10. Step 3 of the concatenation

The decomposition of two connected segments S and D works by reversing the operations done during concatenation. This is described by the following three major steps:

1. Translate EP_D by $-(EP_S + ED_S)$.

2. If the position of D in the chain is even, rotate D by 180° around ED_S to compensate the ω angle.

3. Calculate the virtual position of S with $\varphi = 0°, \psi = 0°$ and rotate D to match this virtual position:

   (a) Rotate EN_S to superimpose with $(0, 0, 1)$.

   (b) Establish the virtual position of the C_α atom of ED_S. This corresponds to the "should be" position of FP_D.

   (c) Calculate the angle between FP_D and the virtual position of the C_α atom. Rotate D around the negative angle.

These operations are sufficient to apply the divide & conquer approach to loop modeling. Its application is the subject of the next chapter.

13.4 Summary

The concept of a novel divide & conquer algorithm for loop modeling is presented. It uses pre-calculated look-up tables (LUTs) that represent loop fragments of various sizes to speed up the calculation. Conformations are produced by recursively dividing the segment until the backbone coordinates can be derived analytically.

A particular vector representation is required for the algorithm to work. The loop fragments are defined using two sets of three vectors representing the start and end of the segment. The end of the segment is encoded from the position of the backbone atoms by end position (EP), the end direction (ED) and end normal (EN). The start conformation is encoded in an analogous way.

The two operations required for the algorithm to work are described in detail. During generation of the LUTs, it is necessary to concatenate two segments. This is done by three major geometrical transformations. The dual operation of decomposing a segment into two halves is done by reversing the geometric transformations in the decomposition.

14
Realization

With the underlying concepts of the divide & conquer algorithm for loop modeling described in the previous chapter, it is now possible to deal with the issues related to its implementation. The specific details of look-up table generation and search algorithm will be covered on the data generation side. The screening of generated solution for the best-fitting one will also be described in depth.

14.1 Look-up Tables

The construction of the look-up tables (LUTs) is separated from modeling and has to be executed only once. A number of LUTs, covering the conformational space for loop segments of lengths $2, ..., n$, are generated and used to improve the performance in terms of both computing time and accuracy of the loop construction.

The actual database generation requires a list of (φ, ψ) angle pairs from a Ramachandran plot [73] distribution to be compiled. The Feb 2001 version of PDBSELECT 90 [74][75] list, containing PDB identifiers with less than 90% sequence identity, was processed to extract the (φ, ψ) angles of loop regions. The rationale behind this high sequence cutoff was to retain as much variation in the loops with near identical sequence as possible, in order to better sample the weaker represented areas of the Ramachandran plot.

In addition, only high-resolution X-ray structures solved at 2.5 Å or better were used, as lower resolution structures tend to contain more errors in the loops. This statement was verified by comparing the prediction quality on three different Ramachandran plots. An artificial plot modeled with three Gaussian

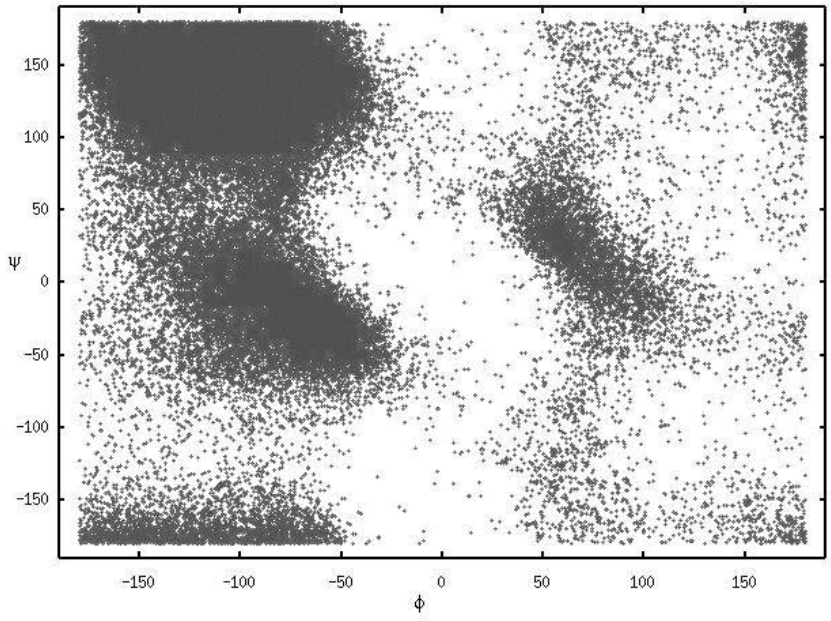

FIGURE 14.1. Ramachandran plot of the over 600,000 (φ, ψ) angles used to construct the look-up tables.

distributions for the major areas taken from [294] was compared with two "natural" Ramachandran maps, one with 2.5 Å resolution cutoff and the other without resolution cutoff. The resolution cutoff improved prediction accuracy by allowing tighter screening thresholds to be used for ranking.

The (φ, ψ) angles were computed using the DSSP [223] software and segments of regular secondary structure discarded. This reduces redundancy caused by widely populated regions of the Ramachandran plot associated with α-helices and β-sheets. The (φ, ψ) angles of over 600,000 residues were extracted and stored in a single table, in random order. The resulting distribution is shown in Figure 14.1. Whenever a new residue is considered during the subsequent database generation, different conformations are generated from these (φ, ψ) angles.

The database generation is initiated by concatenating different conformations of two single-residue fragments in rigid geometry, i.e. with fixed bond lengths and bond angles. Alternatively, it is possible to simulate "flexible" structures by allowing the bond lengths and bond angles to variate slightly around their mean values. The variation is expressed in terms of number of standard deviations and modeled as a Gaussian distribution centered on the mean value. The performance of both approaches will be discussed in Section 15.1. The mean values and standard deviations were taken from PROCHECK [76][77] and are summarized in Table 14.1.

		μ	σ
bond length	$C_\alpha \rightarrow C'$	1.52	0.02
	$C' \rightarrow N$	1.33	0.015
	$N \rightarrow C_\alpha$	1.458	0.016
bond angle	$N \rightarrow C_\alpha \rightarrow C'$	111.6°	2.5°
	$C_\alpha \rightarrow C' \rightarrow N$	116.4°	2.0°
	$C' \rightarrow N \rightarrow C_\alpha$	121.7°	1.7°

TABLE 14.1. Mean value (μ) and standard deviation (σ) for bond lengths and angles in proteins, as esteblished by PROCHECK [76] [77].

Between 10,000 and 1,000,000 different conformations are generated using Monte Carlo sampling. This sampling scheme means that the (φ, ψ) angles are randomly selected from the Ramachandran distribution. Due to the random character of the process, the resulting conformations approximate the true distribution of conformations observed in protein structures. The sampling error decreases with increasing number of samples [288] as will be seen in Section 15.1.

Both the end location of each segment and its central point are stored in the LUT using the vector representation. The central point is the overlapping residue (i.e. E_i and S_j) between the two segments from which the table entry was concatenated. It contains information for dividing the segment during database searches. The location of the starting residue (S_i) needs not to be stored, as it is assumed to lie in the origin of Cartesian space. During database searches the query will thus have to be re-oriented to match this implicit starting conformation.

Tables with higher order than two residue segments are then created, starting with three residues, then four, etc. This process relies on the ability to concatenate the conformations stored in lower order tables to extrapolate longer loop segments. It is made possible by using the previously defined vector operations. Monte Carlo sampling is again used to cover conformation space in randomly selecting segments for concatenation. The process is repeated until all tables up to a chosen length have been completed. The database generation is not limited to any specific loop length, although it can be expected that the coverage of solution space decreases for longer loops.

14.2 Search Algorithm

The search algorithm requires the position of the two anchor regions and the number of residues spanning them. The anchor regions are defined as the single amino acids preceding and following the loop structure (transformed in vector representation).

Using the divide & conquer approach, a loop of length n with an orientation O will be first matched against the LUT for that length. After loading, the LUT entries, each containing information about a central and end position, are randomly oriented and have to be rotated into the x, y–plane to allow comparison. The angle δ_0 needed to rotate an entry i into the x, y–plane is given by the following equation, also shown in Figure 14.2:

$$\delta_0 = \arccos\left(\frac{x_{EP_i}}{\parallel EP_i \parallel}\right) \tag{14.1}$$

EP_i is the end position of i. x_{EP_i} is the x–component of EP_i and $\parallel EP_i \parallel$ the vector length. The rotation axis is obviously $e_y = (0, 1, 0)$. This step cannot be performed during LUT construction, as it was observed to cause the accumulation of errors if performed during concatenation.

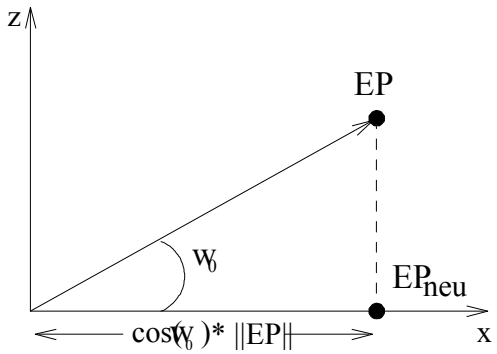

FIGURE 14.2. Rotating a LUT entry into the x, y–plane.

The loop orientation O will be re-oriented to allow comparison with the database entries. The LUT is searched for a list of matching candidates, each with its central residue conformation. The candidate loop is divided into two loop segments of length $n/2$ (or $n/2 + 1$ and $n/2$ if n is an odd number). Using its central point information, the loop is re-oriented and compared with a table of length $n/2$ in the following step. The process is repeated until the query conformation has reached a single residue. At this point the coordinates of the three backbone atoms can be calculated, by transforming them back into the original orientation O.

The search algorithm was designed to produce a list of possible solutions within seconds. The look-up table content is stored in a hash container [289] sorted by the Euclidean distance D between the two anchor regions:

$$D = \sqrt{\sum_{i=1,2,3} (EP_i)^2} \tag{14.2}$$

The hash container can be divided into any number of bins. More bins mean a faster search of matching loop segments, but require more time to fill when the LUT is loaded. E.g. when using over about 500 bins, the total computation time for a typical single loop modeling increases. Some tests were performed to find a good trade-off. The container currently divides the look-up table in 64 bins. Instead of searching all entries, it is possible to search only a fraction of each table, typically between 5% and 20%, to retrieve all entries with distances below a given cutoff.

The search criterium SC for selecting a bin in the tables is given by the distance between the target anchor region, transformed in vector representation, and each table entry:

$$\begin{aligned} SC &= \lambda_{EP} * \sum_{i=1,2,3} (EP1_i - EP2_i)^2 \\ &\quad + \lambda_{ED} * \sum_{i=1,2,3} (ED1_i - ED2_i)^2 \\ &\quad + \lambda_{EN} * \sum_{i=1,2,3} (EN1_i - EN2_i)^2 \end{aligned} \tag{14.3}$$

To save computing time the square-root was omitted from the formula. λ_{EP}, λ_{ED} and λ_{EN} are scaling factors used to adjust the relative weight of the three vectors. λ_{ED} and λ_{EN} are generally set to 1. Increasing λ_{EP} above 1 will reduce the impact of chain orientation towards the anchor fragment, whereas reducing it will increase the propensity to select conformations with better orientation to the anchor fragment. Optimization of λ_{EP} will be described in Section 15.1.

14.3 Filters & Ranking

The search algorithm on average produces less than five hundred conformations or any number the user chooses. These are subjected to a number of filters and a ranking is computed. Different criteria will now be introduced, before the ranking strategy can be explained.

Two simple filters are immediately applicable to significantly reduce the number of possible solutions. A van-der-Waals filter (VDW) checks for interatomic collisions between non-bonded atoms, eliminating those conformations

showing distances between two loop backbone atoms or loop and framework atoms of less than 2.0 Å. The geometry of the residue preceding the C–terminal anchor region can also be measured.

Due to the loop being constructed from the N– to C–terminus, any deviation from the idealized rigid geometry will deform the residue preceding the C–terminal anchor region. Let n be the C–terminal residue and $n-1$ the one preceding it. The chain continuity filter CC checks the following conditions:

1. bond length C'_{n-1} to N_n is 1.4 ± 0.5 Å
2. bond angle $C_{\alpha_{n-1}}$, C'_{n-1} to N_n is $121° \pm 15°$
3. w_{n-1} torsion angle is $180° \pm 20°$

No allowance is made for *cis* prolines in the present implementation, as these constitute less than 5% of all proline residues and less than 0.04% of all peptide bonds [293]. The high tolerance for variations in bond length is due to technical reasons: Since the algorithm tends to accumulate deviations from standard geometry on the last C–N bond length, using this high tolerance was empirically shown to preserve potentially favorable solutions. Conformations passing this filter are assumed to be close enough in rigid geometry that a constrained local optimization will be sufficient to close the gap in backbone continuity.

The following criteria can be used both for screening out improbable solutions and for ranking. The first one is based on the preference of amino acids for areas of the Ramachandran plot. The propensity P is calculated for all conformations based on the method described by Deane and Blundell [15]. The Ramachandran plot is divided into regions of $10° \times 10°$. $P_{i,A}$, the propensity of amino acid A for area i, is calculated as follows:

$$P_{i,A} = \frac{\frac{n_{i,A}}{n_{tot,A}}}{\frac{n_i}{n_{tot}}} \tag{14.4}$$

$n_{i,A}$ is the number of amino acids of type A in region i and $n_{tot,A}$ the total number of amino acids of that type. n_i is the number of all amino acids found in region i and n_{tot} the total number of all amino acid types in the Ramachandran plot. The overall propensity for a fragment of length n is then given as:

$$V = \frac{\prod_{i=1,\ldots,n} P_{i,a}}{n} \tag{14.5}$$

		α	β	*coil*
α	L_1	0.38	0.294	0.327
	L_{n-1}	0.388	0.533	0.089
	L_n	0.268	0.633	0.098
β	L_1	0.484	0.362	0.155
	L_{n-1}	0.340	0.541	0.119
	L_n	0.559	0.203	0.238
	L_{avg}	0.414	0.420	0.156

TABLE 14.2. Loop propensities depending on flanking regions. The propensity of a specific residue (L_1, L_{n-1}, L_n) with a given flanking region (*first column*) in the loop to adopt a specific conformation (*top row*) is given. For a description of the residue types see text.

All conformations below a given cutoff for V are removed. The propensity serves to exclude conformations for amino acids that are in a particularly unfavorable region of the Ramachandran plot.

Another kind of propensity was observed by Wojcik et al. [65] when analyzing *PDB* structures. The loop residues close to the anchor regions show distinct statistical (φ, ψ) preferences depending on the flanking secondary structure. Such preferences were observed for the first (L_1) and the last two residues of the loop (L_{n-1},L_n) compared to the average distribution (L_{avg}) as shown in Table 14.2. The flanking propensity FL is calculated as follows:

$$FL = \frac{L_1}{L_{avg}} * \frac{L_{n-1}}{L_{avg}} * \frac{L_n}{L_{avg}} \tag{14.6}$$

Since the filters may be unable to discriminate surface loops pointing away from the protein core, a further filter was implemented as an attempt to select conformations showing a compact structure. The compactness criterion CD is calculated as the sum of the minimal distances between the C_α atoms of every residue in the loop and the protein framework C_α atoms. In order to limit bias around the anchor regions, three residues before and after the loop are not considered.

$$CD = \sum_{i=1,\ldots,k} \min_{j=1,\ldots,k} \sqrt{(C_{\alpha i} - C_{\alpha j})^2} \tag{14.7}$$

A similar criterion is the Ooi number or packing index PI [333]. This measure counts the number of contacts between C_α atoms of the loop with those of the framework. Two C_α atoms are considered in contact if the distance is below 8 Å. PI gives an estimate of how the loop packs against the framework.

Another index HB counts the backbone hydrogen bonds between the loop and framework and in the loop itself. A hydrogen bond is counted whenever donor and acceptor atoms are inside an allowed range, regardless of their orientation. This crude approximation is faster to calculate and less sensitive to distortions in the backbone geometry created by the loop modeling process.

The energy E_{pot} of each fragment was calculated using the residue-specific all-atom distance-dependent probability function (RAPDF) of Samudrala and Moult [78], described in Section 8.2. E_{pot} is calculated only on the main chain and C_β atoms. Inclusion of Jones' [205] simplified solvation energy E_{solv}, described in Section 8.3, was also investigated.

For all fragments, the geometric fit to the anchor regions E_{rms}, is calculated. This takes the form of the RMSD to the C–terminal anchor region, since the N–terminal is fixed. It can expressed as:

$$E_{rms} = \sum_{atom=N,C\alpha,C'} \sqrt{(atom_{loop} - atom_{anchor})^2} \tag{14.8}$$

With all the different criteria described, how can these be used to produce a good estimate of the quality of the candidate loops? Any single criterion is unlikely to yield satisfactory results. Energy functions do not always discriminate the correct solution and all other criteria only consider a specific aspect of the structure. It is therefore necessary to combine the information. The process used to select good fragments is based on two steps.

First, screening is performed to remove the most improbable solutions. This is based on the observation that a single criterion, while unable to discriminate all solutions, is still able to indicate some particularly improbable ones. The best example are the VDW and CC filters, which are always applied before the remaining criteria are used on the screened solutions. Candidates having severe atomic clashes or lacking basic chain continuity can never be the best solution because the stereochemistry of proteins does not allow it. In the second step, the remaining solutions are ranked according to a combination of several criteria. The ranking term is a linear combination, with scaling factors adjusting the relative weight of each criterion. Linear ranking was chosen as it is simple to implement and robust [287]. The latter is very important as biological data tends to contain many local errors and apparent contradictions. Using a complex ranking scheme makes it subject to the selection and coverage of parametrization data, something very difficult to estimate.

The main contributions to the "correct" ranking are easily identifiable as coming from E_{pot} and E_{rms}. This is in line with published results (e.g. [15][46] [59][65]). In order to establish good thresholds for screening and the correct relationship in ranking, the following parametrization scheme was used. All

loops in a parametrization set were modeled and values for the single criteria recorded. Z-scores and correlation coefficients were calculated for this data depending on loop length. Z-score thresholds for eliminating improbable conformations were calculated by enumerating possible values and selecting one that guarantees to maintain the best solutions. This was repeated for all criteria until a limited subset remains. The ranking was optimized on this subset by selecting only criteria that contribute to the discrimination, as established by the correlation with loop RMSD. The scaling factors were again chosen by enumerating possible values and selecting the combination yielding the lowest average RMSD. This was achieved by maximizing the correlation between RMSD and score, a method recommended by Baldi et al. [295] in their overview of evaluation methods. The results of this optimization are described in Section 15.1.

After the ranking has been computed, a number of optional adjustments can be performed. The simplest one is to cluster the most similar fragments in the ranking to get a less redundant set of loops. The clustering process removes lower ranked fragments which fall below a fixed similarity threshold to a higher ranked one. It ensures to keep the presumably best solutions, while weeding out less favored ones.

Since any deviation from the idealized rigid loop geometry will deform the residue preceding the *C*–terminal anchor region, expressed by E_{rms}, it is possible to refine the loop fragments by reducing this deformation. Displacing the whole loop slightly serves to distribute the error. The difference between the backbone atom coordinates of the *C*–terminal anchor region and the last residue in the loop gives three distance vectors. This distance can be reduced (e.g. halved) by displacing the loop atoms accordingly.

The final optional step consists in the local optimization of the loop fragments. A simple Monte Carlo method was implemented to test the effects of simple local optimization. It is used to randomly displace the backbone atoms slightly. The displacement is calculated from a random sample in a Gaussian distribution with maximum displacement typically below 0.2 Å. The target function for the optimization is the knowledge-based E_{pot} energy. This is the most time consuming and not necessarily the most successful optional step.

14.4 Implementation: *Nazgûl*

The algorithms and data structures required for the loop modeling process are implemented in the *Nazgûl*[1] package. The main requirements for the classes

[1] The name *Nazgûl* comes from J.R.R. Tolkien's *Lord of the Rings* [286]. It indicates the evil ring wraiths which are controlled by the One Ring. An alternative name for the loop modeling problem, sometimes

were extensibility (e.g. to implement new ranking schemes) and simplicity of interfaces. Since the algorithm is centered on the LUTs, its design largely revolves around this particular data structure. The class diagram is shown in Figure 14.3.

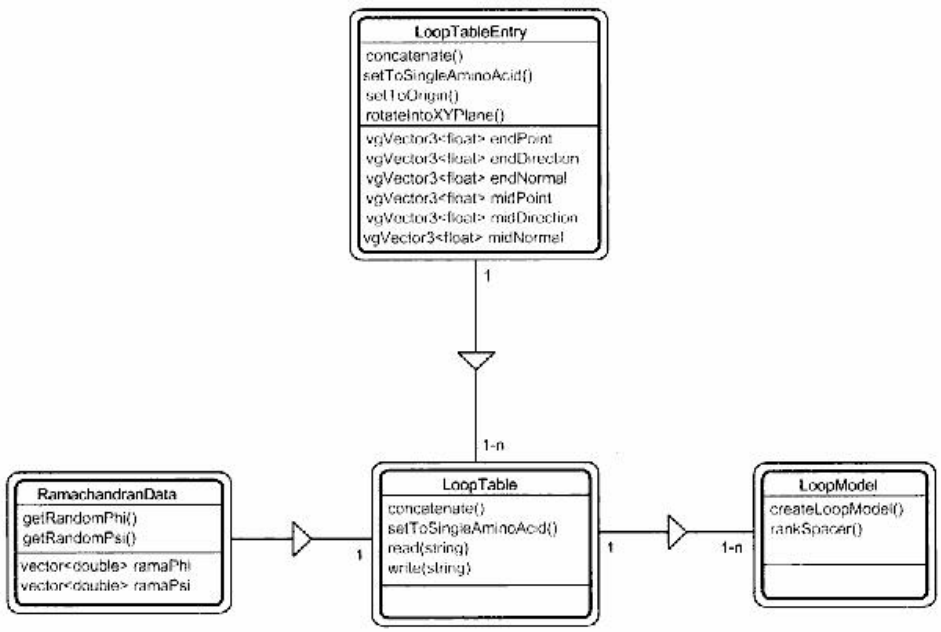

FIGURE 14.3. Class diagram for *Nazgûl*.

A `LoopTableEntry` object stores a single loop fragment in vector representation for a LUT. It contains all methods to perform the vector transformations that are necessary during LUT construction and evaluation. The data from a Ramachandran distribution as well as methods to load and query it are encapsulated in `RamachandranData`, which is a static member shared by all `LoopTableEntry` objects.

`LoopTable` is the central class representing a single autonomous LUT for a loop fragment of fixed size. It contains methods to construct LUTs by concatenation of smaller fragments, or to start the construction from a single residue using `RamachandranData`. In addition, methods to load and save the LUT data are implemented. These compress the single vector components of the LUT from `double` to `short int`, thereby reducing disk space usage. The hash container and a method to search for the best matching entry completes the repertoire of this class.

---

found in the literature (e.g. [3]), is ring closure. The difficulties in implementing the algorithm make for an interesting parallel between both.

The functionality to search for good LUT solutions to a given loop modeling problem is encoded in the `LoopModel` class. This includes the actual divide & conquer algorithm as well as methods to build the loop candidates on the original framework and calculate the RMSD. More importantly, it also includes the methods that filter the solutions and calculate a ranking. The criteria for ranking are implemented as individual functions. In order to make the algorithm more flexible, a number of status variables can be switched through the interface. Otherwise the interface is fairly simple, allowing the easy inclusion in homology modeling for instance.

Several main programs were written to use the classes described above. These range from `LoopTableTest` to construct the LUTs and `LoopTablePlot` to plot their content to more complex programs to perform loop modeling. Depending on the loop modeling task at hand, one of several programs can be used.

`LoopModelTest` is developed for the simplest case where one wishes to model a single loop from a given protein. It gives the user full flexibility concerning the setting of parameters for ranking and modeling. It is the standard program to use for applying loop modeling, unless using the automated approach coded in *Homer*.

`FullModelTest` and `LoopIteration` are tools to test the performance of the method on sets of proteins. `FullModelTest` divides a protein in overlapping fragments of fixed length covering all types of secondary structure and calculates the average RMSD and standard deviation for reconstructing them with loop modeling. `LoopIteration` instead reads a batch of protein files and automatically selects all existing loops for modeling. The results of each run are logged and a global statistic computed.

All loop modeling programs have been extended to optionally produce scatter plots of the ranking. These show the correspondence between each criterion and the RMSD of the solution. `scatEdit` can be used to generate files for plotting programs (e.g. `gnuplot`) and to analyze them. This analysis is at the base of the optimized parameter setting used for ranking. It can be used to calculate correlation factors and estimate filter cutoffs.

14.5 Summary

The main issues concerning implementation of the previously introduced divide & conquer method for loop modeling are addressed in this chapter. The look-up tables (LUTs) are constructed once prior to the actual loop modeling process from a Ramachandran distribution of (φ, ψ) torsion angles extracted from PDB structures. The LUTs may either be constructed with rigid geome-

try or allowing bond length and bond angle variations according to a Gaussian distribution around their mean values.

The search algorithm uses the LUTs to find matching loop candidates. It employs a hash container which finds solutions searching only between 5% and 20% of the LUT. The similarity measure is based on the weighted RMSD of the three vectors (EP, ED, EN). The weight λ_{EP} is found to affect prediction quality.

The candidate loops are subjected to a number of criteria ranging from van der Waals and chain continuity filters, sequence or structural features to knowledge-based potentials and geometric fit on the framework. The resulting data is used to first filter out improbable solutions and then rank the remaining ones according to a mixture of criteria. An optimization was performed by maximizing the correlation between RMSD and score, as recommended by Baldi et al. [295].

The algorithms are implemented in the package *Nazgûl*. The LUT is represented by `LoopTable`, which contains single `LoopTableEntry` conformations. The divide & conquer algorithm and ranking criteria are implemented in `LoopModel`, with a minimal interface for simple inclusion in other packages. A range of main programs, ranging from single loop prediction to benchmarking of whole sets of proteins complements the package.

15
Results

The results for the divide & conquer loop modeling algorithm will be presented in this chapter. In order to demonstrate its state of the art performance, it will be compared to some of the newest and best performing methods from the literature. Before this can be done, the main parameter choices and overall performance of the algorithm will be introduced.

15.1 Overall Performance

To evaluate a method systematically, it is first necessary to define a test set. Because the accuracy of the predictions for different loops may vary considerably, it is desirable to parametrize and test the method on many different loops. A list including all loops from 400 non-homologous proteins (less than 25% sequence identity with each other) was extracted from the *PDB*, using random selection from the Feb 2001 version of the PDBSELECT25 list [74][75]. As in loop construction, only structures solved at a resolution of 2.5 Å or better were used. The regions outside regular secondary structures, as identified from evaluating the selected proteins with DSSP [223], were defined as loops for this test. Loop segments between 3 and 12 residues in length were selected according to the following criteria:

- no overlap between any two loops,
- B factors for all backbone atoms are $<= 25$ Å^2 and
- the N– and C–termini are not used as test loops.

This list was divided into independent parametrization and test sets. The parametrization set is composed of 200 protein structures with 777 loops in total. The test set consists of 637 loops from the remaining 200 proteins. Table 15.1 shows the distribution per loop length.

Loop length	Number of Loops Parametrisation Set	Test Set
3	175	156
4	149	144
5	114	102
6	88	80
7	81	50
8	64	35
9	48	26
10	23	20
11	25	12
12	10	12

TABLE 15.1. Distribution of loops in the parametrization and test sets.

The quality of the divide & conquer algorithm was evaluated using the test set, which is composed of 637 loops of length between 3 and 12 residues. It was first investigated how well the algorithm was able to cover the solution space, i.e. how accurate in terms of global RMSD the best solution is. Since the look-up tables are built from a large number of (φ, ψ) angles, coverage is supposed to be high. However, the sampling is performed with a fixed number of entries per table, so it was interesting to determine how the accuracy scales with the number of entries per LUT. The results for various table sizes are shown in Table 15.2. The method performs better with larger LUTs, due to the greater number of alternative loop conformations. LUTs with 100,000 entries even perform better than LUTs with 1,000,000 for a number of cases. Fewer solutions can be generated from 100,000 entries, making the ranking more difficult. The performance after ranking was found to be slightly inferior. The following tests are therefore performed on the largest look-up table size, i.e. 1,000,000 entries.

The theoretical complexity for the algorithm is given by the following considerations. Generating s solutions for loop length l and LUTs with n entries implies the following considerations:

- For each of the s solutions, the LUTs have to be scanned, i.e. $O(s)$.

	Large		Medium		Small	
Length	μ_d	σ_d	μ_d	σ_d	μ_d	σ_d
3	0.60	0.40	0.62	0.26	0.75	0.42
4	1.00	0.97	1.09	0.94	1.19	0.91
5	1.30	0.90	1.39	0.93	1.45	0.83
6	1.67	1.43	1.73	1.39	1.78	1.39
7	2.13	1.24	1.90	0.64	2.16	0.73
8	2.22	0.83	2.05	0.64	2.11	0.67
9	2.92	0.87	2.54	0.63	2.48	0.53
10	3.87	3.17	3.90	3.25	3.95	3.13
11	3.86	1.47	3.40	1.02	3.61	0.40
12	3.50	0.54	3.48	0.34	3.91	1.59

TABLE 15.2. Lowest RMSD of the loop modeling method based on size. The lowest average (μ_d) and standard deviation (σ_d) is given for 10,000 entries (*small*), 100,000 entries (*medium*) and 1,000,000 entries (*large*) LUTs.

- Generating one solution of size l requires scanning the LUTs of size $l, \frac{l}{2}, ..., 2$ a total of $1, 2, ..., \frac{l}{2}$ times, i.e. $O(l)$.
- Scanning a LUT depends on the number of entries n, but is optimized to search only a fraction of the data, i.e. $O(\log n)$.

The complexity is therefore $O(s * t * \log n)$, i.e. it scales linearly with loop length, number of evaluated solutions and logarithmic with number of entries per LUT. Execution times range between roughly 20 seconds (2 residue loop) and 120 seconds (12 residue loop) for sixty solutions generated from LUTs with one million entries on a 500 MHz PC. For tables with 100,000 and 10,000 entries these values are respectively one and two orders of magnitude smaller. Storage of the look-up tables on hard-disk requires 36 bytes per table entry: 6 vectors with 3 components each of the type `short int` (2 bytes) have to be stored. Adding the overhead results in less than 37 MB per table for one million entries. A database to predict loops up to twelve residues in length therefore requires less than 450 MB disk space. Main memory scales linearly with the number of entries per look-up table. Keeping all necessary tables in memory for a given loop requires roughly 300 MB for loops of length twelve residues and look-up tables with one million entries. Smaller tables require about 30 MB (100,000 entries) and less than 5 MB (10,000 entries). Memory requirements can be traded for computation speed by reading the tables from hard-disk during the database search.

Different Ramachandran plot distributions were investigated to create the look-up tables. Alternatives included different sets of loops extracted from the PDB and an artificial distribution with Gaussian distributions approximizing main areas of (φ, ψ) angle space. No significant difference was encountered.

Given the possibility to find reasonable solutions in the conformations produced by the algorithm, the candidate selection and ranking process had to be parametrized. Parameters were fitted using the previously described parametrization set of 777 loops.

The first step consisted in the optimization of the selection criterium SC. This has three interdependent parameters, λ_{EP}, λ_{ED} and λ_{EN}, one for each of the three vectors. λ_{EP} differs from the other two insofar as the endpoint EP can vary the most. Variations of λ_{EP}, ranging from fixed values to linear and quadratic length-dependent functions, to search the parametrization set were tested. Using a fixed $\lambda_{EP} = 0.5$ yielded marginally better overall results as shown in Figure 15.1. Changing λ_{ED} and λ_{EN} produced very similar results with no clear trend (data not shown).

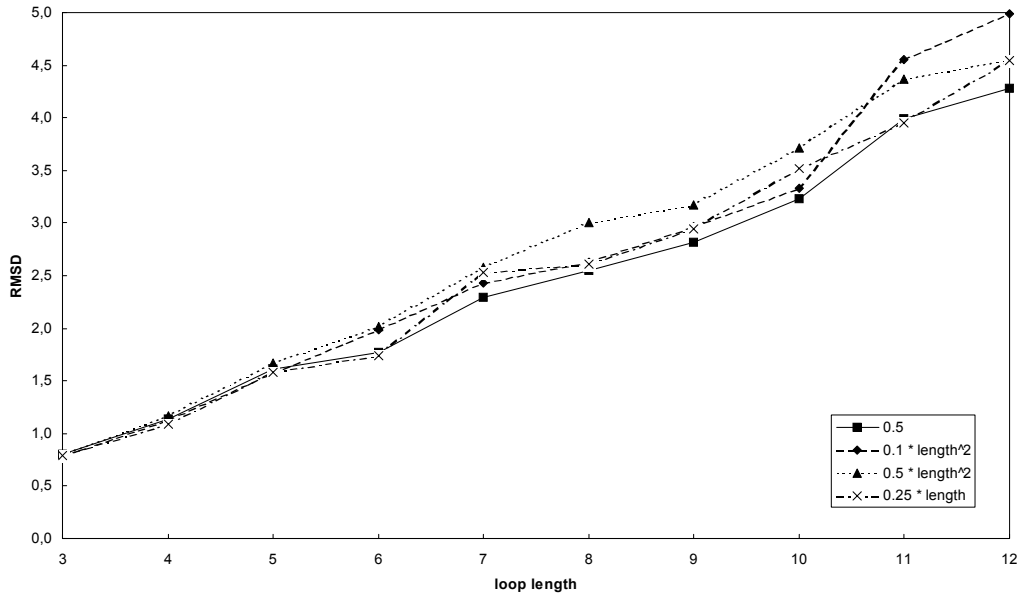

FIGURE 15.1. Influence of λ_{EP} on the solutions. The lowest RMSD of a solution is plotted against loop length for four different functions of λ_{EP}.

In order to reduce the number of candidate loops to evaluate with a more time-consuming scoring function, a set of computationally inexpensive filters was fitted on the parametrization set. Due to the way the algorithm builds the loop backbone from the fixed N–terminal anchor residue to the C–terminal anchor residue, the CC filter was first introduced, which eliminates conformations lacking elementary chain continuity. The VDW filter was also chosen to discard conformations invalidated by strong steric clashes. In general both filters eliminate roughly 40% of the candidates.

The surviving solutions are viable backbone conformations, but they still do not reflect the specific nature of the amino acid sequence. E.g. a single Proline residue may be in a prohibited area of the Ramachandran plot. To eliminate such invalid solutions, the propensity filter V is used with a threshold of $<= 0.001$. Such conformations are virtually impossible in the native structure. In order not to eliminate the best solution, the threshold cannot be raised. The propensity V was not found to be a positive ranking indicator, i.e. a higher propensity does not imply a better solution. V was therefore excluded from the subsequent analysis and optimization of the ranking. This is in agreement with the results found by Deane and Blundell [15]. Usage of the V filter eliminates another 10-20% of the solutions.

The remaining criteria described in Section 14.3 do not exclude "wrong" conformations. They merely indicate a tendency for some conformations to be "better" than others and cannot be used as filters. To decide which criteria to use for the ranking, the raw data from the test sets was collected and analyzed in terms of correlation between RMSD to each single criterion. This process follows the recommendations of Baldi et al. [295] for effective design of classification schemes. The resulting correlation coefficients, for each loop length, are shown in Table 15.3.

	$Score$	E_{rms}	E_{pot}	E_{solv}	PI	HB	CD	FL
overall	0.708	0.609	0.554	0.286	0.041	−0.611	−0.112	0.072
3	0.690	0.742	0.269	0.255	−0.045	−0.042	−0.096	0.064
4	0.775	0.761	0.380	0.071	0.141	0.087	−0.033	0.085
5	0.723	0.683	0.458	0.305	−0.079	−0.020	0.071	−0.070
6	0.675	0.597	0.455	0.279	−0.157	−0.058	0.075	0.172
7	0.541	0.516	0.261	0.090	−0.057	−0.001	−0.019	0.043
8	0.359	0.354	0.173	0.046	−0.042	−0.078	−0.082	0.133
9	0.442	0.425	0.169	0.097	0.030	−0.158	−0.056	−0.119
10	0.376	0.416	0.126	−0.266	0.142	0.091	0.091	−0.019
11	0.520	0.445	0.257	0.063	−0.084	−0.029	−0.146	0.075
12	0.313	0.282	0.220	0.158	−0.264	−0.142	−0.108	−0.005

TABLE 15.3. Correlation coefficients RMSD to single criteria. The correlation coefficients between the single criteria (for a description see Section 14.3) and the RMSD based on loop length and overall is given.

The geometric fit on the anchor fragment E_{rms} correlates best with the loop RMSD. This is not unexpected, as "good" solutions have to fit well on the loop anchors. The knowledge-based potential E_{pot} also correlates well with the RMSD. Indeed from another experiment on the test set, it is known that E_{pot} would be able to rank the native loop first in over 99.5% of all cases, should

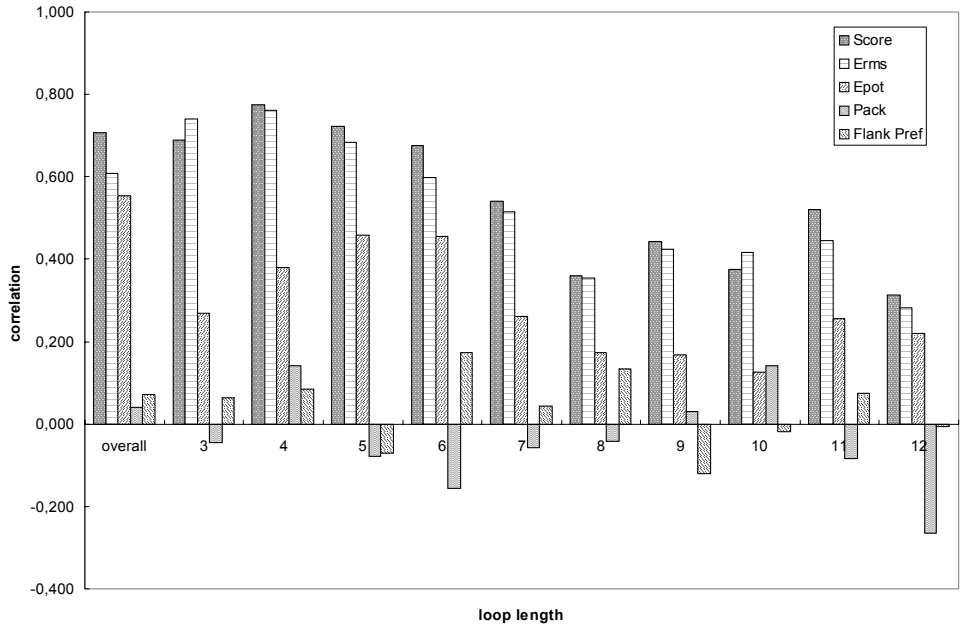

FIGURE 15.2. Correlation of the criteria with RMSD. Some correlation coefficients are shown per loop length. The optimized score almost always outperforms the best criterion, i.e. E_{rms}.

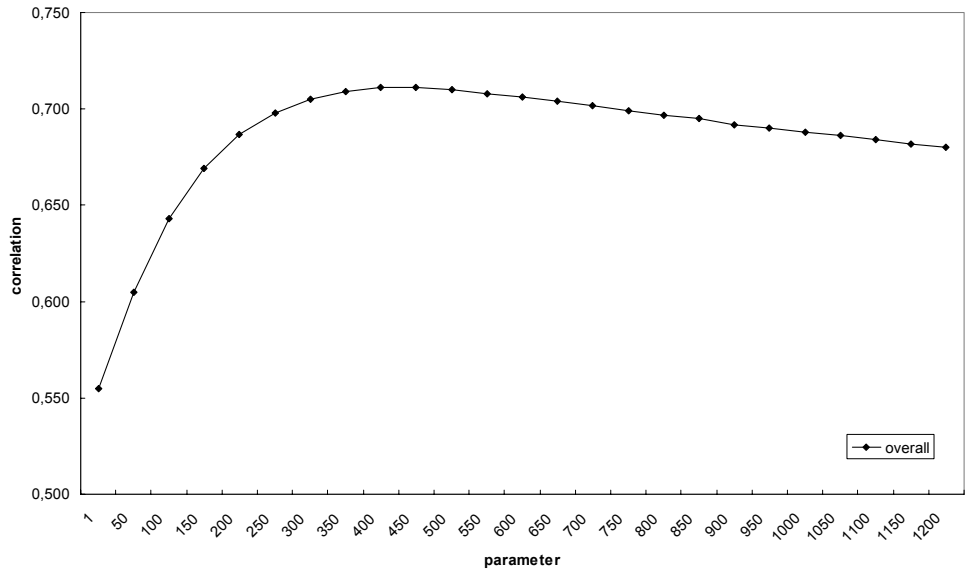

FIGURE 15.3. Variation of the correlation coefficient depending on the choice of the scaling factor for E_{rms}. A plateau is reached for values between 400 and 600.

this be among the candidates. Since the remaining criteria have a correlation coefficient close to zero they were not considered for ranking. Preliminary experiments confirmed that including them would reduce the robustness of the ranking.

For the final ranking, different combinations of E_{rms} and E_{pot} were studied. A complete search was performed for linear combinations. The linear classification is found to be sufficient with an optimal scaling factor for E_{rms} of 554.0. Its correlation factor is also shown in Table 15.3. Using a length-dependent scaling factor did only add sparsity to the data without significant improvements. Since, as shown in Figure 15.3, variations to the scaling factor do not significantly alter the overall accuracy, the ranking is indeed very robust.

The knowledge-based potential E_{pot} seems to be largely redundant in the final ranking, only marginally improving the combined correlation coefficient beyond the value for E_{rms} alone. This can only be explained with the VDW filter and propensity V eliminating most unfavorable solutions.

The optimized ranking is found to be on average a good indicator for a probable loop conformation, albeit with some variations. In some cases it is possible to have several conformations with high RMSDs obscuring the best solution. Table 15.4 shows the top X results ($X = 1, 3, 5, 10, 20$) for the test set with the final set of filters and cutoffs, as published in [210]. Figure 15.4 shows the superposition between predicted and real loops from the test set.

	1		3		5		10		20	
Loop	μ_d	σ_d	μ_d	σ_d	μ_d	σ_d	μ_d	σ_d	μ_d	σ_d
3	1.06	0.60	0.89	0.57	0.80	0.49	0.74	0.46	0.70	0.44
4	1.62	1.15	1.37	1.07	1.27	1.03	1.21	1.01	1.15	0.99
5	2.22	1.47	1.74	1.18	1.62	1.12	1.50	1.02	1.44	1.00
6	2.89	1.86	2.38	1.67	2.21	1.63	1.99	1.52	1.86	1.45
7	3.62	1.91	2.84	1.45	2.72	1.41	2.58	1.35	2.44	1.29
8	3.72	1.58	3.05	1.20	2.79	1.05	2.60	1.05	2.52	0.92
9	4.95	1.84	4.17	1.65	3.82	1.72	3.32	0.97	3.19	0.94
10	6.92	3.69	5.31	3.31	4.69	3.11	4.60	3.12	4.41	3.13
11	5.88	2.45	5.40	2.21	5.30	2.16	4.49	1.47	4.44	1.44
12	6.73	2.75	5.18	1.79	4.93	1.52	4.39	1.22	4.19	0.90

TABLE 15.4. Performance of the loop modeling method using fixed LUTs. The average (μ_d) and standard deviation (σ_d) of the global RMSD (including O atoms) among the X top ranking predictions ($X = 1, 3, 5, 10, 20$) is given.

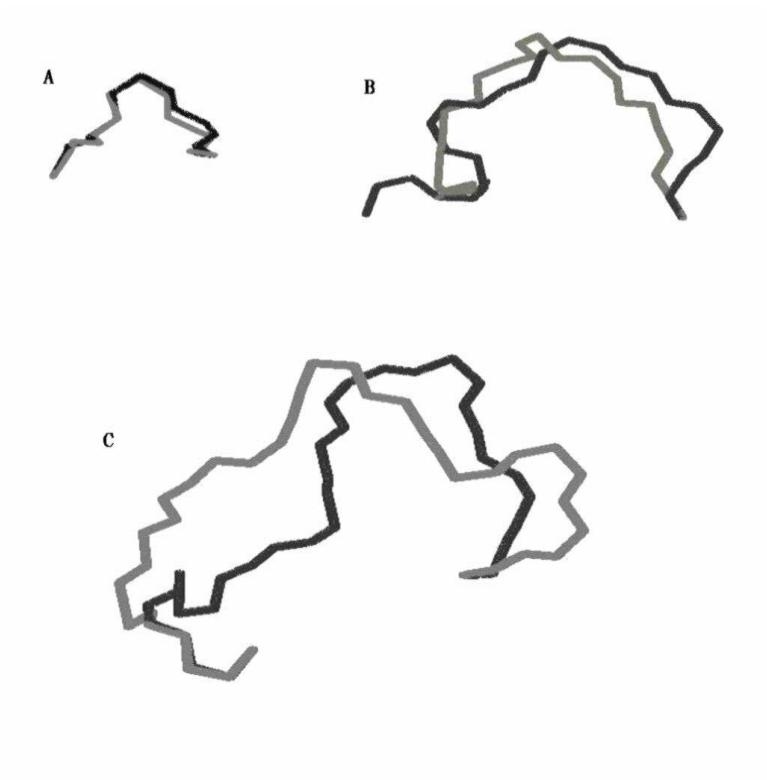

FIGURE 15.4. Sample loops from `1ohk`. The global superposition between real structure (*dark grey*) and prediction (*light grey*) is shown for three cases. (*A*) residues 172 to 175 (RMSD 0.35 Å). (*B*) residues 124 to 130 (RMSD 1.83 Å). (*C*) residues 77 to 87 (RMSD 3.37 Å).

15.2 Extension to Flexible Geometry

An extension to the basic algorithm outlined above, and published in [210], is the inclusion of limited backbone "flexibility" during LUT construction. Instead of using fixed bond lengths and angles, these are allowed to vary around their mean values, as described in Section 14.1.

The free parameter for this "flexible" approach is the number of standard deviations n_σ each value is allowed to vary around its average in a Gaussian function. The parameter was optimized by constructing different LUT sets and testing their performance, with previously optimal search parameters, on the parametrization and test sets measured. Representative results are shown in Table 15.5.

The best n_σ was found to be 2. The effects of this limited flexibility on the LUTs are shown in Figures 15.5 to 15.8. The largest difference can be encountered for small LUTs, where even limited shifts in atom position make the previously sharply defined area become quite fuzzy. For longer loop fragments

$n_\sigma =$	0		1		2		4	
Length	μ_d	σ_d	μ_d	σ_d	μ_d	σ_d	μ_d	σ_d
3	0.60	0.40	0.56	0.42	0.52	0.28	0.58	0.35
4	1.00	0.97	0.96	0.92	0.93	0.90	0.98	0.87
5	1.30	0.90	1.15	0.81	1.23	0.81	1.28	0.76
6	1.67	1.43	1.57	1.42	1.59	1.37	1.74	1.39
7	2.13	1.24	1.94	0.68	1.97	0.66	2.06	1.15
8	2.22	0.83	2.20	0.72	2.10	0.65	2.19	0.61
9	2.92	0.87	2.82	1.57	2.60	0.75	2.73	0.53
10	3.87	3.17	4.15	3.16	3.71	3.26	3.99	3.20
11	3.86	1.47	3.32	0.84	3.44	1.08	3.31	0.71
12	3.50	0.54	3.51	0.54	3.66	0.58	3.85	1.02

TABLE 15.5. Lowest global RMSD (including O atoms) of the flexible loop modeling method based on the flexibility parameter n_σ (see text). The lowest average (μ_d) and standard deviation (σ_d) is given.

the variations appear to cancel out and merely distribute the end positions more equally.

Optimization of the scoring function along the lines described in the previous section produced only marginally different parameters. From a comparison with results for the previous parameters these offer no improvement and are instead less robust. It was therefore decided to use the previous set of parameters. The final results are shown in Table 15.6.

	1		3		5		10		20	
Loop	μ_d	σ_d	μ_d	σ_d	μ_d	σ_d	μ_d	σ_d	μ_d	σ_d
3	0.97	0.58	0.78	0.47	0.67	0.38	0.60	0.34	0.57	0.32
4	1.56	1.14	1.28	1.05	1.19	1.01	1.09	0.95	1.02	0.93
5	2.10	1.50	1.67	1.21	1.52	1.01	1.37	0.91	1.32	0.89
6	2.94	1.74	2.42	1.63	2.11	1.54	1.90	1.43	1.79	1.39
7	3.26	1.49	2.56	0.99	2.43	0.96	2.25	0.82	2.10	0.74
8	3.96	1.89	3.21	1.29	2.92	0.93	2.62	0.88	2.36	0.85
9	4.47	1.92	3.72	1.41	3.55	1.31	3.22	0.93	2.94	0.76
10	5.86	3.42	4.83	3.46	4.70	3.48	4.49	3.50	4.31	3.45
11	6.38	2.26	5.37	1.94	4.74	1.51	4.09	1.27	3.87	1.08
12	6.27	1.90	4.43	0.92	4.38	0.96	4.32	0.95	4.03	0.57

TABLE 15.6. Performance of the loop modeling method using flexible LUTs. The average (μ_d) and standard deviation (σ_d) of the global RMSD (including O atoms) among the X top ranking predictions ($X = 1, 3, 5, 10, 20$) is given.

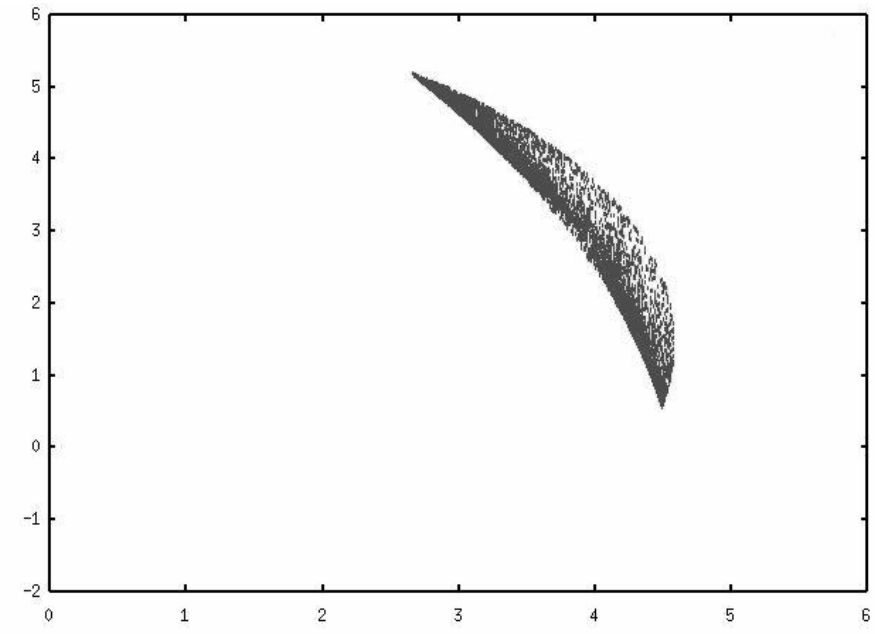

FIGURE 15.5. End points for a LUT spanning two residues constructed with fixed geometry.

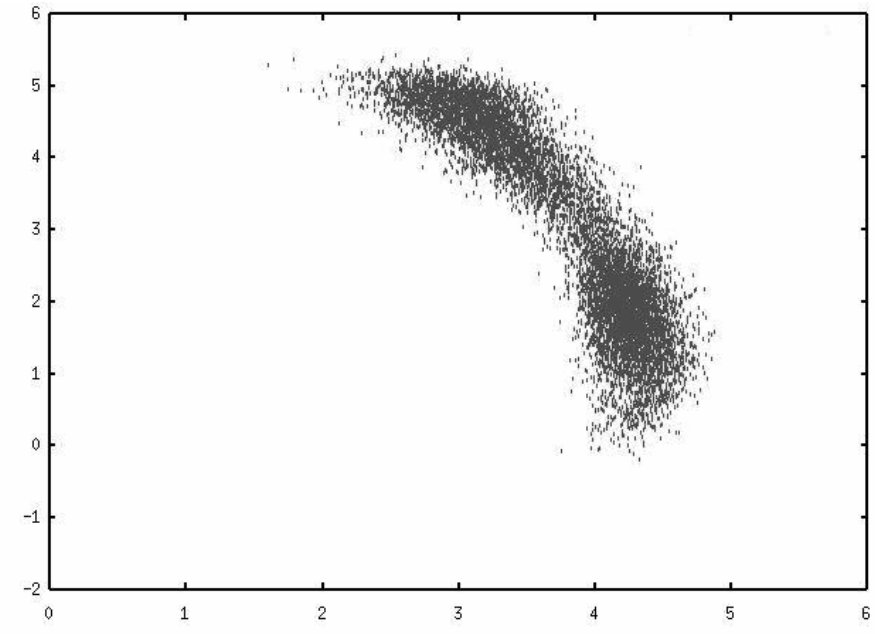

FIGURE 15.6. End points for a LUT spanning two residues constructed with flexible geometry.

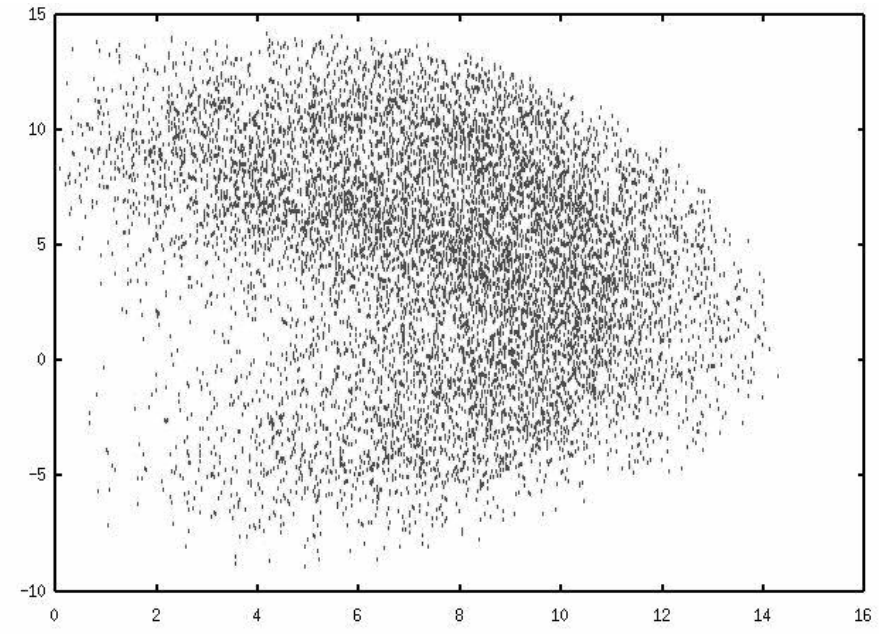

FIGURE 15.7. End points for a LUT spanning five residues constructed with fixed geometry.

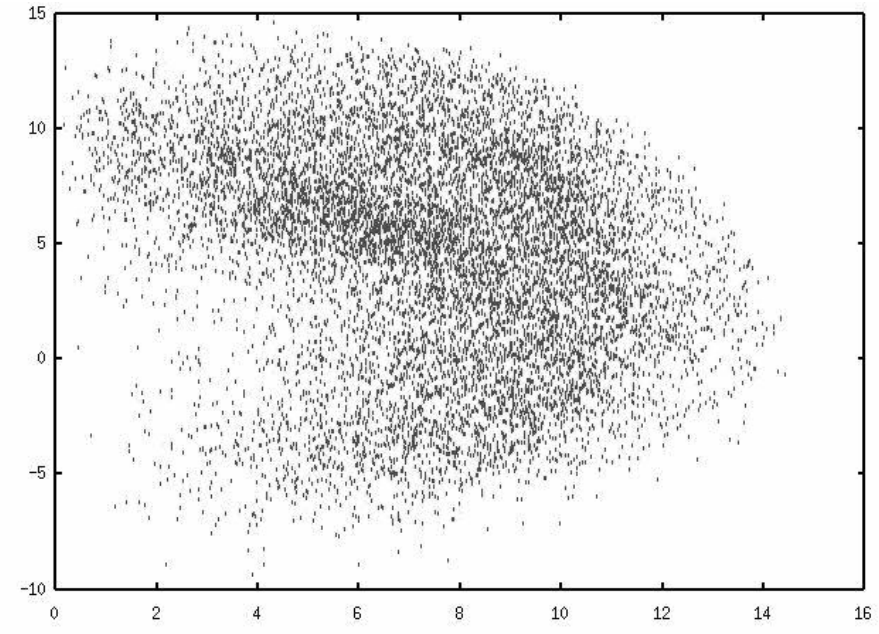

FIGURE 15.8. End points for a LUT spanning five residues constructed with flexible geometry.

15.3 Comparison with Other Methods

While there are a number of existing loop modeling methods, comparison is made difficult because of several reasons. Different methods used for calculating the RMSD give rise to divergent results. It was therefore decided to compare the results with two state of the art methods operating in a similar time frame: Deane and Blundell [15] and Wojcik et al. [65].

	Top		Best	
Loop length	μ_d	σ_d	μ_d	σ_d
3	1.3	1.1	1.0	0.3
4	1.9	0.9	1.2	0.5
5	2.5	1.1	1.4	0.5
6	2.9	1.2	1.6	0.6
7	3.6	1.6	1.8	0.7
8	3.9	1.5	2.3	0.8

TABLE 15.7. Performance of the Deane and Blundell method [15]. The average (μ_d) and standard deviation (σ_d) of the global RMSD (including O atom) are shown.

Loop length	μ_d	σ_d
3	1.1	0.5
4	1.7	0.9
5	2.2	1.2
6	2.7	1.4
7	3.4	1.8
8	3.8	2.1

TABLE 15.8. Performance of the Wojcik et al. method [65]. The average (μ_d) and standard deviation (σ_d) of the global RMSD (excluding O atom) are shown.

The *ab initio* method published by Deane and Blundell [15] shares several similar ideas with the present work. It is also based on an algorithm for searching and ranking a database of pre-calculated loop conformations. Their strategy is to compute a complete enumeration of a simplified set of eight torsion angle combinations. They report an upper limit of loop length eight due to the combinatorial explosion. The present approach works with arbitrary loop lengths. Both methods make use of the same knowledge-based contact potential [78] to improve the ranking. Their method computes a set of loop conformations in the order of up to twenty minutes [15], whereas the present work takes about two minutes on a 500 MHz PC. They use a test set of 400 high-resolution loops to validate their method. The main results taken from [15] are summarized in Table 15.7.

The database method of Wojcik et al. [65] is a loop classification based on the analysis of sequence patterns. It defines a collection of loop families with sequence patterns and anchor fragment distances. This can be queried for loop modeling. The ranking is based on the minimization of a candidate selected among four different classes. Computation time is vaguely described to be in the order of minutes. Their method is tested on a large set of loops taken from the PDB. The main results are summarized in Table 15.8. Before comparing the results with other methods, it is important to note that Wojcik et al. did not include the backbone O atom in the RMSD calculation. This omission lowers the RMSD on average by 0.2-0.3 Å compared to the other methods.

The test set used in this paper differs from that used by the other two methods, because the other test sets are not publicly available. We will nevertheless attempt to draw some conclusions. As can be seen in Table 15.6, the present method performs significantly better for loops up to five residues in length. For loops six and eight residues long it performs more or less as well as the method of Deane and Blundell, and again better for length seven. Taking into account increase in RMSD of about 0.2-0.3 Å necessary for a fair comparison with Wojcik et al., the present method performs better on all loop lengths except length six, where it performs equally well. Longer loops cannot be compared due length restrictions on the other methods. The diminishing improvement for longer loops can be explained considering the fixed size of the look-up tables. For long loops it becomes increasingly probable that some conformations are missed out altogether. This is supported by the performance on loops of more than ten residues, where the average RMSD can become prohibitive.

To evaluate the prediction accuracy of the method on any fragment in a protein, the prediction of all overlapping five residue segments in **1igd** (*immunoglobin binding protein*), a small 61 residue protein containing both α–helices and β–strands has been repeated. It has been already argued that this test is of particular interest to comparative modeling, where secondary structure elements are not well defined [15]. The results are shown in Table 15.9.

As can be seen, the present method has a significantly lower RMSD than the Deane and Blundell method in all types of segments for the test protein. The most significant improvement being for α–helical and mixed segments, i.e. loops. This supports the results from the previous test for loops of length five.

Having presented a comparison with other methods operating in a similar time frame, it is worth asking what the lowest achievable RMSD for a set of test loops disregarding time limitations could be. The method of Fiser et al. [46] and that of van Vlijmen and Karplus [59] both require in the order of thirty hours CPU time to compute a single loop. Both use a common test set of eleven loops. These methods combine a very slow optimization protocol

	*Top*1		*DB*	
Type of Structue	μ_d	σ_d	μ_d	σ_d
$\alpha - helix$	0.57	0.23	1.9	1.0
$\beta - strand$	0.93	0.28	1.2	0.6
$Mixture$	1.51	0.93	2.6	1.5
$Overall$	1.22	0.83	2.2	1.4

TABLE 15.9. Performance on entire structures. The average (μ_d) and standard deviation (σ_d) of the global RMSD (including O atom) for all overlapping length 5 segments of `1igd` is given. Top 1 denotes the top ranking result for the present method. DB denotes the results from Deane and Blundell [15].

with a complex energy function. The focus of the present method on the other hand was the fast generation of good solutions. Using a more complex energy function, like the other methods do, will improve the ranking but was out of scope for the work presented here. It is nevertheless straightforward to implement. The two methods are therefore best compared with the top 10 result of the present algorithm as shown in Table 15.10. Compared to van Vlijmen and Karplus, out of eleven cases a better solution is found in six and a comparable solution in another two cases. Compared with Fiser et al. a better solution is found in three and a comparable solution in another two cases. It should be emphasized that a difference in computation time of three orders of magnitude cannot yield a fair comparison. Moreover, due to the limited size of the test set the evidence should not be considered representative.

				RMSD	
PDB	Loop	Length	Nazgûl	VK	FS
2apr	76-83	8	2.52	5.16	1.31
8abp	203-208	6	3.58	0.28	0.38
2act	198-205	8	0.91	1.58	2.04
3grs	83-89	7	1.52	4.55	0.42
5cpa	231-237	7	1.91	2.14	0.95
2fb4	H26-H32	7	1.85	1.62	4.20
2fbj	H100-H106	7	2.66	0.49	0.84
3dfr	20-23	4	0.67	2.64	1.15
3dfr	89-93	5	1.11	1.62	1.02
3dfr	120-124	5	0.83	0.47	0.26
3blm	164-168	5	1.55	0.82	0.16

TABLE 15.10. Performance on the van Vlijmen & Karplus [59] and Fiser et al. [46] test set. The global RMSD (excluding O atoms) of the best prediction for the present method (*Nazgûl*) is compared to the top ranking solution of van Vlijmen & Karplus [59] (*VK*) and Fiser et al. [46] (*FS*).

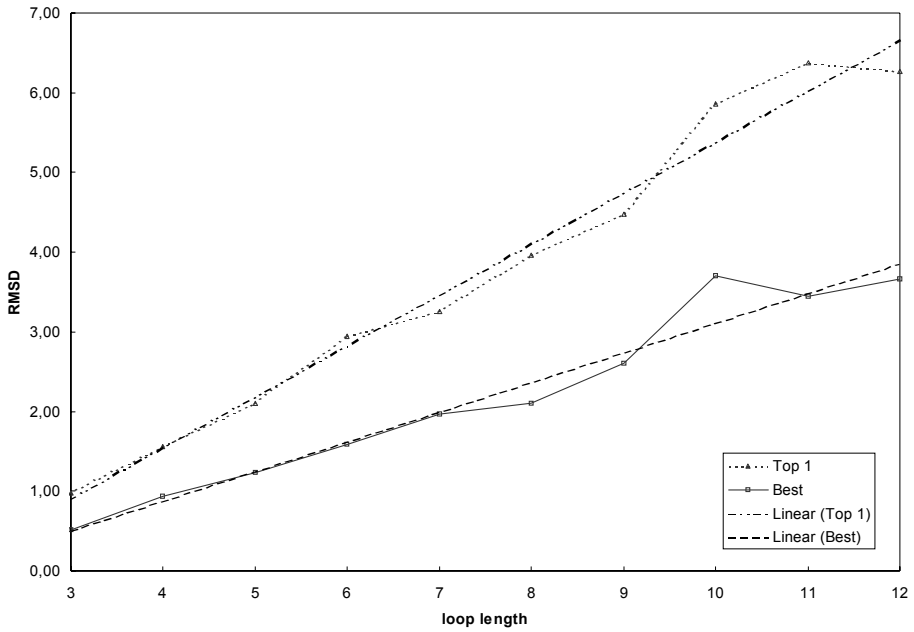

FIGURE 15.9. Graphical representation of the prediction accuracy. The top 1 solution and best prediction are plotted per loop length. Two linear functions are also fitted to the data, showing very good agreement (correlation better than 0.96).

15.4 Discussion

The novel divide & conquer algorithm for fast loop modeling developed in the present thesis has been shown to produce state of the art results. As shown in Figure 15.9, the results indicate that the RMSD increases as a linear function of loop length. This is not unexpected considering the fixed size of the LUTs.

Compared to other state of the art methods requiring up to one order of magnitude more CPU time, it was shown to significantly improve prediction accuracy for loops up to five residues. Longer loops are predicted equally well as in said methods. The resources required to run this algorithm, both in terms of hard disk space and main memory, are lower than those of the other algorithms. E.g. the Deane and Blundell method [15] requires 4 GB disk space, compared to about 0.3 GB for the present method under identical conditions. Computing time is also lower by a factor of ten. More importantly, the requirements for the present algorithm are linear in computation time, main memory and hard disk space.

Comparison with time-intensive methods requiring three orders of magnitude more CPU time, reveals the potential for future improvement. Such methods rely on extensive energy minimization using the CHARMM potential. They use crude methods for generating starting conformations which require

much CPU time to produce valid solutions. E.g. Fiser et al. [46] randomly displace the loop atoms along a straight line between the anchor fragments to start the energy minimization. In contrast, the present method requires only seconds to generate many valid alternative conformations. However the subsequent CHARMM energy minimization brings the solution of Fiser et al. closer to the native structure than the fast ranking used in the divide & conquer algorithm. Extending the algorithm to perform a similar energy minimization of the solutions generated from the LUTs will close the gap in prediction accuracy. The divide & conquer algorithm could then serve as a starting point for complex simulations, such as molecular dynamics. Implementing it was however out of scope for the present thesis.

The present scoring function used to rank the candidate loops appears to make good use of the information available in the various ranking criteria. Since the accuracy of the best structure among the top ranking 20 is nevertheless significantly better than that of the highest ranking one, there seems to be potential for further improvement. Using a more complex non-linear scheme seems unlikely to improve the top 1 accuracy. Further changes to the data on which the LUTs are constructed are also unlikely to yield improvements.

An interesting new approach to rank the candidate loops would be to create a consensus from two or more predictions. Since the divide & conquer algorithm uses artificially constructed loop segments, it may be possible to extract more information from combining its ranking with a method using database loops from the PDB. The idea is that loop conformations appearing in both rankings are more probable to correspond to the native structure. A simple way to achieve such an alternative ranking would be to modify the LUT construction to generate a set of tables containing only loops entirely present in the database. Combining this modified prediction with the regular one is likely to yield an improvement in accuracy.

In its present form, the divide & conquer algorithm is a valid contribution to the knowledge-based protein modeling protocol developed in Part II of this thesis. Available data, e.g. the `1igd` test from the previous section (see Table 15.9), support the idea that the divide & conquer method is robust in producing predictions for any type of fragment to be modeled, whether it is a loop or not. This is perhaps its most interesting feature, since it opens up new possibilities. Due to its speed, it is possible to model those missing fragments of fold recognition targets that cannot be copied from the template structure. To date this is not done for fold recognition targets. Indeed, during CASP-4 our group was, to the best of this author's knowledge, the only one to model variable regions of fold recognition targets[1]. One direction of further research

[1] Other groups predicting entire structures, e.g. Baker or Friesner, used *ab initio* methods for the entire structure.

will be to benchmark the contribution of modeling missing fragments for fold recognition targets.

In conclusion, it is fair to say that the previously described divide & conquer method for fast loop modeling is a valid contribution to the current state of the art in loop modeling. It improves results both in terms of accuracy and efficiency and opens up new possibilities for making more extensive use of loop modeling in new situations.

15.5 Summary

In order to evaluate the results for the divide & conquer algorithm, two independent data sets containing loops from 200 proteins each were derived. These were used to establish the performance of the method. The complexity is found to be $O(s * t * n)$, for generating s solutions of loop length l using LUTs with n entries. Typical execution times range from 20 to 120 seconds on a 500 MHz PC. Hard disk and main memory requirements are also linear.

The parametrization set was used to derive simple filter cutoffs that eliminate at least 50% of the candidate loops. Parameters for the linear ranking scheme were derived by optimizing the correlation coefficient between RMSD and the scoring function. The final ranking is found to be very robust and a good indicator of the most probable solution, albeit with some variations.

An extension of the basic algorithm is the inclusion of limited backbone "flexibility" during LUT construction. This is achieved by modeling bond lengths and angle with a Gaussian function around their mean values. This flexibility was optimized to improve the overall results of the basic algorithm.

Results for the algorithm were compared with two state of the art algorithm operating in a similar time frame. These represent the main alternatives of *ab initio* and database methods. The divide & conquer algorithm is found to significantly improve the accuracy for loops up to five residues in length. Results for loops of length six to eight are roughly equivalent. Longer loops cannot be modeled by the other methods. In all cases the present algorithm makes more efficient use of available computer resources. Testing the algorithm on all overlapping fragments of a protein demonstrates that its ability to predict fragments is not limited to loops.

Comparison with methods requiring three orders of magnitude more computation time, which are too slow to be used for homology modeling, demonstrate the potential for future improvement. Two alternatives are shown for future research. The first one involves using the divide & conquer method to generate starting conformations for slower energy minimization methods. The second is to form a consensus of results from the present method with other predictions, such as those using loops extracted from the database. Both approaches

are likely to improve the overall results. In conclusion, the divide & conquer method is found to be a state of the art extension of the knowledge-based protein modeling protocol described in Part II. Due to its speed it opens up new possibilities for application of loop modeling in different situations.

16
Outlook

With the work done in this thesis already described, it is now worth asking which parts could be improved and the direction further research should take. Once again, this is divided into the two main objects of the thesis, knowledge-based modeling of entire structures and ab initio loop modeling.

The knowledge-based protein modeling protocol developed in this thesis has performed very well in the recent CASP-4 experiment. In particular, it was able to select the most suitable template for very weakly homologous targets, as shown in the fold recognition results. Its complete automation will avoid further errors caused by manual intervention in building the model core. On the downside, like all modeling approaches participating in the CASP experiments, the sequence to structure alignment still needs to be improved.

The best way to improve alignment generation is to augment programs such as PSI-BLAST or CLUSTALW with additional restraints. These can be deduced by scanning the literature on the protein in question for biochemical or structural information. E.g. for metal-binding proteins the residues forming the active site may be known. This can be stated as a distance restraint between some residues, thereby greatly reducing the number of plausible alignments.

A similar, but more uncertain, extension would be to use contact predictions to generate additional spacial restraints. The problem here is the relative inaccuracy, with false positives being up to 80% of all predicted pairwise long-range contacts. Using only two-state contact number or accessibility predictions produces fewer false positives but yields less overall information. Elucidating the contribution of contact predictions to alignment accuracy is nevertheless an interesting research field.

A straight-forward extension of the approach to construct the conserved core is the inclusion of multiple templates. Structural superimposition of alternative template structures reveals the conserved structural motifs. Averaging these is the standard technique to construct the core, which is only slightly more accurate than using a single template. The problem in using the multiple structural alignment consists in assessing the effect of shifts and rotations to the relative orientation of template fragments. This requires both a new target function and a different kind of search algorithm to those present in the literature. As target function it would be intuitive to use a combination of energy, secondary structure, sequence similarity and torsion angle propensity. To combine single fragments it would be possible to use a modified form of the divide & conquer algorithm. Single fragments could be recursively selected and assembled into "optimal" structures, shifting the base alignment where necessary. The advantage of such an approach consists in its scalability and the possibility to use the additional spacial restraints mentioned above. Implementing such a new alignment and model construction method would be a promising new research project.

Coupled with "flexible" loop modeling and side chain placement algorithms, an improved model building procedure may serve as a basis for designing artificially stable protein structures. To this end it would be necessary to improve the results of the side chain placement method. For energy-based placement, it is probably useful to elucidate alternative and more complex energy functions, taking into account more diverse effects. Development of an alternative energy-based heuristic for quickly estimating the validity of a set of conformations may also be warranted. Such an estimate could serve to improve the loop modeling and model construction steps, by eliminating impossible structures early on.

The divide & conquer algorithm developed in this thesis has been shown to work very well in practice for fast loop modeling. It outperforms state of the art methods operating in a similar time frame. Methods which are slower by several orders of magnitude may be interesting from a theoretical point of view to establish how much better an algorithm may become, but are of little value in practical modeling exercises. It is nevertheless interesting to understand which parts of the method can be improved.

As was seen from the difference between the best solution produced by the method and the top ranking ones, there is still room for improvement. Ranking is currently dominated by the geometric loop closure constraint. This is necessary due to the local instability of the energy function. Developing a specific knowledge-based potential for loop modeling seems an interesting option for reducing this problem.

An alternative approach to circumvent the ranking problem in homology modeling would be to pass information of observed loop structures in homologous proteins to the algorithm. This could take the form of restrictions, with the atomic coordinates of the central residue in the simplest form. Such restrictions would allow negative screening, allowing to sieve out totally wrong predictions. This would work mainly for loops with "key residues", where the entire conformation is strongly influenced by the position of few atoms.

Taking this approach one step further would prompt the combination of *ab initio* loop modeling with database methods. Using a set of loop conformations derived from the PDB as basis for the divide & conquer algorithm may improve prediction in the case of more conserved structures. Clustering two distinct sets of predictions, artificial and database-derived, may give additional criteria for ranking and selection.

Another interesting extension of the loop modeling process would be to integrate a simple side chain placement method. It is not uncommon to select loop structures which cause irreparable clashes among side chain atoms. This is usually found out only in a later step, making the whole modeling process inefficient. Using a simplified side chain placement method to predict whether a loop conformation will allow all missing side chain atoms to be positioned would greatly improve the overall efficiency.

Again, taking this improvement one step further means to be able to modify the local backbone structure to accommodate amino acid mutations. This can be performed, if the positional restraints mentioned above are implemented. In case of an irreparable side chain clash, the backbone atoms would be displaced far enough to relieve the collision. These new positions would restrain the loop modeling process, ensuring that only allowed conformations are selected.

This "flexibility" of the backbone is one of the major problems in estimating the effects of point mutations and protein design. A generalized "flexible" loop modeling algorithm would greatly improve the efficiency of protein design. Estimation of the structural effects of amino acid substitution would be possible to quantify. Reliably predicting cooperative modifications of the structure would ultimately open up the possibility to design artificially optimized proteins for a variety of biotechnological purposes.

17
Summary

Proteins are the "molecular machines" of living organisms. Their function is determined by the 3D structure. Predicting protein structure, considered the "Holy Grail" of structural biology, would largely improve understanding of protein function, and ultimately life itself. The present thesis deals with the improvement and automation of such knowledge-based methods for protein structure prediction and modeling. It is divided in three parts.

Part I starts with a description of the basic characteristics of proteins, such as chemical properties and torsion angles as main free parameters. How they assemble in 3D, from secondary to quaternary structure and the concept of protein domains is explained. Recent theories for protein folding, including the "new view", are discussed (Chapter 2).

The two main experimental methods for protein structure determination, X-ray crystallography and NMR spectroscopy are introduced in Chapter 3. These are time consuming, but produce high (X-ray) or medium (NMR) resolution structures. The depository of experimental structures, the *PDB* databank, is also introduced.

Measures and tools for sequence and structural similarity, such as alignment and sequence identity, are defined in Chapter 4. RMSD is introduced for measuring structural similarity and its main pitfalls highlighted. The concept of homology is discussed, with the implications for convergent and divergent evolution. The three major methods for structure classification (SCOP, CATH, FSSP) are explained.

Chapter 5 gives an overview of computational methods for structure prediction. The three main categories (homology modeling, fold recognition and *ab initio*) are introduced and the limits for homology modeling, around 20%

to 30% sequence identity, discussed. Brief descriptions of the state of the art in secondary structure and contact & accessibility prediction are followed by an overview of the basic concepts of *ab initio* methods. Some of the most successful *ab initio* methods, the Baker group's in particular, are described. Finally, common base line optimization methods (Monte Carlo and simulated annealing) are defined.

Part II covers the knowledge-based protein structure prediction approach followed in this thesis. It starts with an extensive description of the state of the art (Chapter 6). It is described mostly in terms of what has been established in the CASP experiments. These blind tests take place every two years (CASP-4 in 2000) and have become the best way to assess methods that work consistently well. Homology modeling and fold recognition are thoroughly described. Two specific sub-problems, energy functions and side chain placement are also addressed.

Homology modeling is the type of prediction that yields the most accurate results. The approach starts by scanning a sequence database of known structures for homologs and aligning these to the target. Alignment is the primary source of errors. The 3D coordinates of aligned residues are copied. Structurally variable loops are the second largest source of errors. The structure is finalized by placing the side chain atoms and assessing model quality. The most successful methods are briefly discussed, although differences between several methods tend to be limited.

Fold recognition is used for sequences without significant similarity to known structures. Structural information is used to augment the prediction. Four essential components are required: fold library, scoring function, alignment algorithm and ranking. The fold library is a subset of a structural database. Each fold in the library is aligned to the target sequence using the scoring function. Ranking can vary from sorting raw scores to statistical measures. The most successful methods use profiles and combine various sources of information. Manual intervention is still an important factor.

Energy functions give a measure of confidence for optimization. Two alternative classes of functions have been developed: force fields and knowledge-based potentials. Force fields are empirical models approximating the energy of a protein. In knowledge-based potentials the "energy" is derived from the probability of interaction patterns found in the *PDB*, with varying levels of abstraction.

Side chain placement attempts to reproduce the position of side chain atoms based on statistical and/or energetic properties using a number of rotamer structures. A combination of side chain optimization and experiments can be used for protein design.

The most important decisions for building a first model of a protein are discussed in Chapter 7. Three main problems exist. Template selection, alignment generation and loop modeling. For template selection and alignment two alternative strategies were implemented. For large scale modeling the presently best "base line" protocol for selecting clear homologs, PDBBLAST, is used. The underlying BLAST algorithms were also introduced. For the single protein case a combination of consensus predictions and manual inspection is used, maximizing the quality of the alignment. A model of the conserved protein core can be built either with fragment or restraint based techniques. Fragment-based modeling is implemented for single templates. The implementation of base classes is described and the program guiding the modeling strategy presented.

Chapter 8 describes the three different energy models implemented during the thesis, each having pros and cons. The non-bonded term of a force field is well suited for side chain placement. A knowledge-based potential is well suited to predict good loop conformations and discriminate near-native structures. Selection of good alignments has benefitted from the implementation of a simple knowledge-based solvation potential. All three energy models were derived from a common interface, allowing the easy implementation of additional functions.

Side chains are an autonomous part of the protein construction process, described in Chapter 9. The main approximation is usage of a rotamer library, a set of torsion angle combinations. Two state of the art optimization methods for side chain placement are introduced and implemented. A heuristic using statistical occurrence of rotamers makes for a fast base line method. The necessity to perform energetic minimization prompted combination with the dead end elimination (*DEE*) algorithm and A^* search. The *DEE* theorem reduces the conformational space by several orders of magnitude, while A^* search is guaranteed to be optimally efficient. The rotamer library and problem space implementations are separated from energy functions and optimization methods, allowing easy extension.

Chapter 10 describes the development of the homology modeling server. It facilitates the usage of structural models by non-experts. The *HOMER* server offers the possibility to construct models of protein structures with automatic or manual alignment generation. It bundles the previously described technology in a web interface and returns the constructed models as e-mail attachments.

Chapter 11 presents the results largely in terms of what our group has achieved during the CASP-4 experiment. All 43 targets were predicted. Results for homology modeling were inconclusive, with our group ranked among the better predictions. In fold recognition our group ranked 15th out of 125 participants. Despite not having submitted *ab initio* predictions, our group

ranked 21st overall *ab initio* and 9th for novel folds only. Results for four selected targets are described in more detail. The main problems encountered in CASP-4 are also discussed.

Part III describes the loop modeling problem and the innovative divide & conquer algorithm developed to solve it. It begins with an introduction to the problem and extensive description of the state of the art (Chapter 12).

Loops are the structurally variable regions outside regular secondary structure which have to be predicted. The problem can be stated as finding a way to connect two anchor regions using the chain corresponding to the loop sequence. Two main classes of approaches for loop modeling exist: *ab initio* and database methods.

Many alternative *ab initio* methods have been described. Up to three residues can be predicted by analytical means. Longer loops can be enumerated with some simplifications or a global optimization used. A fast enumerative method using eight torsion angle pairs was published by Deane and Blundell. Slow global optimization methods may produce accurate solutions, but are out of scope for typical modeling applications. Database methods concentrate on finding a set of representative loop fragments in the *PDB* to classify typical conformations and use for loop modeling. Direct application of a database method has been described by Wojcik et al. Ranking of the candidates in both classes of methods is usually restricted to linear combinations of geometric fit of the anchor regions and an energy function.

The concept of a novel divide & conquer algorithm for loop modeling is presented in Chapter 13. It uses pre-calculated look-up tables (LUTs) representing loop fragments for the calculation. Conformations are produced by recursively dividing the segment, until the backbone coordinates can be derived analytically. A particular vector representation required for the algorithm to work is presented. The necessary geometrical transformations are also described in detail.

Chapter 14 deals with the main issues concerning implementation of the divide & conquer method. The LUTs are constructed prior to the actual loop modeling process from a Ramachandran distribution of (φ, ψ) torsion angles, either with rigid geometry or allowing flexible bond lengths and bond angles. The search algorithm uses these LUTs to find matching loop candidates. Employing a hash container requires only between 5% and 20% of the LUT to be searched. The candidate loops are subjected to a number of criteria ranging from van-der-Waals and chain continuity filters, sequence or structural features to knowledge-based potentials and geometric fit on the framework. Improbable solutions are filtered out and the remaining solutions ranked according to a mixture of criteria. This protocol was optimized by maximizing the corre-

lation between RMSD and score. Implementation details and programs used for benchmarking are finally described.

The main results for the divide & conquer method are described in Chapter 15. The complexity is found to be linear, with execution times in the order of seconds or a few minutes. Candidate selection is based on filtering out improbable solutions before using a linear ranking scheme. Usage of limited backbone flexibility is found to improve the overall results. A comparison with two state of the art methods operating in a similar time frame shows a significant improvement for short and medium-sized loops and an equivalent performance of longer loops, with computer resources used more efficiently by the present method. Comparison with methods requiring three orders of magnitude more computing time demonstrate the potential for further improvement by using slower energy minimization methods on the ranked structures. The divide & conquer method is found to be a state of the art extension of the knowledge-based protein modeling protocol described in Part II which opens up new possibilities for loop modeling.

The outlook in Chapter 16 highlights two main areas for extending the present work and future research. The knowledge-based modeling of entire protein structures will benefit from the implementation of a new alignment module. The divide & conquer algorithm seems well suited to combine the present approach with additional restraints from multiple templates and biochemical information. The loop modeling method may be improved by taking into account information from loops of homologous structures and combination with a fast side chain placement heuristic.

18
Glossary

This section describes some terms which are often used throughout the present thesis.

1D : one dimensional.

3D : three dimensional.

ab initio : Methods which try to compute the structure of a protein from "first principles", i.e. based only on physicochemical properties in absence of knowledge from existing protein structures. See Section 5.5. This term has been subject of debate due to the different definitions of absence of knowledge from existing proteins.

alignment : The result of matching two (or more) amino acid sequences. The residues of a protein chain are superimposed in such a way that the matched residues are identical or "similar" as often as possible. At the same time it is desirable to have as few gaps as possible. Often there are several reasonable alignments between two (or more) sequences, depending on the definition of "similarity". Methods for explaining sequence alignments are described in Section 4.1.

analogy : Proteins are said to be *analogous* if they share a similar structure but are not assumed to derive from a common ancestor. See also homology.

BLAST : Basic Local Alignment Search Tool. Widely used alignment method with good speed and accuracy. See Section 7.2 for a description.

CASP : *Critical Assessment of Techniques for Protein Structure Prediction.* Important scientific experiment held every two years. See Section 6.1 for a description.

comparative modeling : Construction of a 3D protein model from homologous structures. Requires a relatively high sequence identity (e.g. $\geq$30%) with a known structure from the database. Method of choice, where applicable, for producing the most detailed 3D models. See Section 6.2 for an overview.

core : The conserved part of a protein structure. This forms the framework common to different homologous proteins, which is the easiest part to build in homology modeling.

domain : An autonomously folding protein structure, this may occur both as an entire protein or part of a more complex, so-called *multi-domain*, protein.

family : Group of closely related proteins sharing both a unique structure and the same function.

fold : The native structure adopted by a protein. Two proteins sharing the same fold need not be homologous, but can instead be analogous.

fold recognition : Broad class of different protein structure prediction methods that use knowledge about existing folds to predict unknown structures. It differs from *ab initio* methods by inclusion of fold libraries or other statistic features of existing proteins without strict theoretical foundation. In contrast to comparative modeling is usually exploits more than sequence information and may not produce full 3D models. See Section 6.3 for an overview.

gap : Position in an alignment where a residue from one sequence is not matched against any residue from the other sequence. In 3D this corresponds to "non existent".

HMM : Hidden Markov Model. Can be used for fold recognition, arguably achieving higher detection rates than PSI-BLAST for remote homologues.

homology : Proteins are said to be *homologous* if they are assumed to have evolved from a common ancestor. This usually implies a similar structure. On the sequence level this category can be further derived in *close homologues* (sequences which are easily identified as related by any simple alignment methods) and *remote homologues* (sequences which can

be only identified as related by very sensitive alignment methods and/or fold recognition methods).

homology modeling : Used as synonym for comparative modeling.

loop : Part of the protein structure, outside *helical* and *extended* conformational classes, which is less conserved than the core. Frequently it cannot be superimposed on homologous structures, even at high sequence similarity.

LUT : look-up table. Used for storing data in the loop modeling algorithm.

PDB : Protein Data Bank. Database containing all publicly available protein structures. See [127] for reference.

***PDB* code** : A four-digit code, composed of numbers and letters, assigned to every protein in the Protein Data Bank. It is used to retrieve structures from the *PDB*. A five-digit code denotes a particular chain from the protein. E.g. code "`1ewiA`" denotes chain "`A`" of the protein structure "`1ewi`".

***PDB* file format** : Standard format to communicate protein structures with atom coordinates. The current (draft) version of the *PDB* format is 2.1. [127].

PSSM : position specific scoring matrix. An amino acid substitution matrix adapted to capture the details of a protein family. It is used by PSI-BLAST and several fold recognition methods.

primary structure : The amino acid sequence of a protein.

profile : A set of aligned sequences, generally representing a single protein family. Profiles are used to encapsulate the knowledge from multiple sequence alignments, by allowing the direct view of conserved positions.

PSI-BLAST : Position Specific Iterative Blast. Arguably one of the best and most sensitive alignment methods.

native structure : The structure a protein assumes in its natural environment.

quaternary structure : The relative arrangement of several proteins or domains as an interacting 3D unit.

$\mathbf{Q_k}$: Measure for percentage correctness in a system with k-states. Its most frequent usage is as Q_3 for secondary structure prediction.

random coil : Part of the protein structure outside *helical* and *extended* conformational classes, usually a loop.

residue : An amino acid.

RMSD : Root mean square deviation. Used as a measure for similarity it is calculated by:

$$RMSD = \sqrt{\frac{\sum (r_{ai} - r_{bi})^2}{n}}$$

r_{ai} and r_{bi} are the positions of atom i of structure a and structure b.

secondary structure : The conformational class (*helical*, *extended* or *random coil*) of each residue in a protein. See Section 5.3 for a description of prediction methods.

sequence identity : A measure for assessing the sequence similarity of two (or more) proteins, under a given alignment. It is defined as the number of identical residues of the aligned sequences divided by the total length of the alignment. Different alignments may yield slightly different sequence identities between the same sequences.

sequence similarity : This is a way to express how related two (or more) proteins are on the sequence level. As a measure it is dependent on the matrix used for generating an alignment and thus somewhat arbitrary.

superfamily : Group of related proteins sharing a similar structure. Function is usually related.

superimposition : The procedure of bringing the coordinates of two (or more) protein structures to overlap in 3D, minimizing the RMSD between aligned residues.

target : Protein whose structure is to be predicted.

template : Protein with known structure used to align against a target.

tertiary structure : The 3D structure of a protein.

threading : A specific set of methods for fold recognition. It tries to predict a structure by adapting ("threading") the target sequence through a library of possible templates. The alignments are evaluated using an energy function and the best one chosen as the most probable structure.

References

[1] Schulz GE, Schirmer RH. Principles of Protein structure. Springer, 7th printing, 1985.

[2] Moult J, Hubbard T, Fidelis K, Pedersen JT. Critical assessment of methods of protein structure prediction (CASP): Round III. Proteins Suppl 1999; 3:2-6.

[3] Go N, Scheraga HA. Ring closure and local conformational deformations of chain molecules. Macromolecules 1970; 3:178-187.

[4] Bruccoleri RE, Karplus M. Chain closure with bond angle variations. Macromolecules 1985; 18:1767-1773.

[5] Manocha D, Zhu Y, Wright W. Conformational analysis of molecular chains using nano-kinematics. CABIOS 1995; 11:71-86.

[6] Wedemeyer WJ, Scheraga HA. Exact analytical loop closure in proteins using polynomial equations. J Comp Chem 1999; 20: 819-844.

[7] Palmer KA, Scheraga HA. Standard-Geometry Chains Fitted to X-Ray Derived Structures: Validation of the rigid-geometry approximation. I. Chain closure through a limited search of "Loop" Conformations. J Comp Chem 1991; 12: 505-526.

[8] Bruccoleri RE, Karplus M. Prediction of the folding of short polypeptide segments by uniform conformational sampling. Biopiolymers 1987; 26:137-168.

[9] Moult J, James MNG. An algorithm for determining the conformation of polypeptide segments in proteins by systematic search. Proteins 1986; 1:146-163.

[10] Pedersen J, Moult J. Ab initio structure prediction for small polypeptides and protein fragments using genetic algorithms. Proteins 1995; 23:454-460.

[11] Bruccoleri RE, Haber E, Novotny J. Srtructure of antibody hypervariable loops reproduced by a conformational search algorithm. Nature 1988; 335: 564-568.

[12] Brower RC, Vasmatzis G, Silverman M, DeLisi C. Exhaustive conformational search and simulated annealing for models of lattice peptides. Biopolymers 1993; 33: 320-334.

[13] Bruccoleri RE. Application of systematic conformational search to protein modeling. Mol Simul 1993; 10: 151-174.

[14] Fidelis K, Stern PS, Bacon D, Moult J. Comparison of systematic search and database methods for constructing segments of protein structure. Protein Eng 1994; 1:377-384.

[15] Deane CM, Blundell TL. A novel exhaustive search algorithm for predicting the conformation of polypeptide segments in proteins. Proteins 2000; 40:135-144.

[16] Fine RM, Wang H, Shenkin PS, Yarmush DL, Levinthal C. Predicting antibody hypervariable loop conformations. II. Minimization and molecular dynamics studies of MCP603 from many randomly generated loop conformations. Proteins 1986; 1: 342-362.

[17] Shenkin PS, Yarmush DL, Fine RM, Wang H, Levinthal C. Predicting antibody hypervariable loop conformations. I. Ensembles of random conformations for ring-like structures. Biopolymers 1987; 26: 2053-2085.

[18] Smith KC, Honig B. Evaluation of the conformational free energies of loops in proteins. Proteins 1994; 18: 119-132.

[19] Eloffsson A, Le Grand S, Eisenberg D. Local moves: An efficient algorithm for simulation of protein folding. Proteins 1995; 23:73-82.

[20] Lambert MH, Scheraga HA. Pattern recognition in the prediction of protein structure. I. Tripeptide conformational probabilities calculated from the amino acid sequence. J Comp Chem 1989; 10: 770-797.

[21] Lambert MH, Scheraga HA. Pattern recognition in the prediction of protein structure. II. Chain conformation from a probability-directed search procedure. J Comp Chem 1989; 10: 798-816.

[22] Lambert MH, Scheraga HA. Pattern recognition in the prediction of protein structure. III. An importance-sampling minimization procedure. J Comp Chem 1989; 10: 798-816.

[23] Dudek MJ, Scheraga HA. Protein structure prediction using a combination of sequence homology and global energy minimization. I. Global energy minimization of surface loops. J Comp Chem 1990; 11: 121-151.

[24] Dudek MJ, Ramnarayan K, Ponder JW. Protein structure prediction using a combination of sequence homology and global energy minimization. II. Energy functions. J Comp Chem 1998; 19: 548-573.

[25] Bruccoleri RE, Karplus M. Conformational sampling using high temperature molecular dynamics. Biopolymers 1990; 29: 1847-1862.

[26] Tanner JJ, Nell LJ, McCammon JA. Anti-insulin antibody structure and conformation. II. Molecular Dynamics with explicit solvent. Biopolymers 1992; 32: 23-32.

[27] Rao U, Teeter MM. Improvement of turn structure prediction by molecular dynamics: A case study of a-purithionin. Protein Eng 1993; 6: 837-847.

[28] Nakajima N, Higo J, Kidera A. Free energy landscapes of peptides by enhanced conformational sampling. J Mol Biol 2000; 296: 197-216.

[29] Rapp CS, Friesner RA. Prediction of loop geometries using a generalized Born model of solvation effects. Proteins 1999; 35:173-183.

[30] Martin AC, Cheetham JC, Rees AR. Modeling antibody hypervariable loops: a combined algorithm. Proc Natl Acad Sci USA 1989; 86:9268-9272.

[31] Evans JS, Mathiowetz AM, Chan SI, Goddard WA III. De novo prediction of polypeptide conformations using dihedral probability grid Monte Carlo methodology. Protein Sci 1995; 4: 1203-1216.

[32] Thanki N, Zeelen JP, Mathieu M, Laenicke R, Abagyan RA, Wierenga RK, Schliebs W. Protein engineering with monomeric triosephosphate isomerase (monoTIM): The modeling and structure verification of a seven-residue loop. Protein Eng 1997; 10: 159-167.

[33] Higo J, Collura V, Garnier J. Development of an extended simulated annealing method: Application to the modeling of complementary determining regions of immunoglobins. Biopolymers 1992; 32: 33-43.

[34] Carlacci L, Englander SW. The loop problem in Proteins: A Monte Carlo simulated annealing approach. Biopolymers 1993; 33:1271-1286.

[35] Carlacci L, Englander SW. Loop problem in proteins: Developments on the Monte Carlo simulated annealing approach. Comp Chem 1996; 17: 1002-1012.

[36] Collura V, Higo J, Gernier J. Modeling of protein loops by simulated annealing. Protein Sci 1993; 2: 1502-1510.

[37] Vasmatzis G, Brower RC, DeLisi C. Predicting immunoglobin-like hypervariable loops. Biopolymers 1994; 1669-1680.

[38] Rosenfeld R, Zheng Q, Vajda S, DeLisi C. Computing the structure of bound peptides: Application to antigen recognition by class I MHCs. J Mol Biol 1993; 234: 515-521.

[39] Zheng Q, Rosenfeld R, Vajda S, DeLisi C. Determining protein loop conformation using scaling-relaxation techniques. Protein Sci 1993; 2: 1242-1248.

[40] Zheng Q, Rosenfeld R, Vajda S, DeLisi C. Loop closure via bond scaling and relaxation. J Comp Chem 1993; 14: 556-565.

[41] Zheng Q, Rosenfeld R, DeLisi C, Kyle DJ. Multiple copy sampling in protein loop modeling: computational efficiency and sensitivity to dihedral angle perturbations. Protein Sci 1994; 3: 493-506.

[42] Zheng Q, Kyle DJ. Multiple copy sampling: rigid versus flexible protein. Proteins 1994; 19: 324-329.

[43] Rosenbach D, Rosenfeld R. Simultaneous modeling of multiple loops in proteins. Protein Sci 1995; 4: 496-505.

[44] Zheng Q, Kyle DJ. Accuracy and reliability of the scaling-relaxation method for loop closure: An evaluation based on extensive and multiple copy conformational samplings. Proteins 1996; 209-217.

[45] Su A, Mayo SL. Coupling backbone flexibility and amino acid sequence selection in protein design. Pro Sci 1997; 6:1701-1707.

[46] Fiser A, Kinh Giang Do R, Šali A. Modeling of loops in protein structures. Protein Sci 2000; 9:1753-1773.

[47] Abola EE, Sussman JL, Prilusky J, Manning NO. Protein data bank archives of three-dimensional macromolecular structures. Methods Enzymol 1997; 277: 556-571.

[48] Jones TA, Thirup S. Using known substructures in protein model building and crystallography. EMBO J 1986; 5:819-822.

[49] Chothia C, Lesk AM. Canonical structures for the hypervariable regions of immunoglobulins. J Mol Biol 1987; 196:901-917.

[50] Claessens M, Cutsem EV, Lasters I, Wodak S. Modeling the polypeptide backbone with ´spare parts´from known protein structures. Protein Eng 1989; 335-345.

[51] Summers NL, Karplus M. Modeling of globular proteins: A distance-based search procedure for the construction of insertion/deletion regions and pro -> non-pro mutations. J Mol Biol 1990; 216: 991-1016.

[52] Tramontano A, Lesk AM. Common features of the conformations of antigen-binding loops in immunoglobins and application to modeling loop conformations. Proteins 1992; 13:231-245.

[53] Unger R, Harel D, Wherland W, Sussman JL. A 3-D building blocks approach to analyzing and predicting structure of proteins. Proteins 1989; 5:355-373.

[54] Topham CM, McLeod A, Eisenmenger F, Overington JP, Johnson MS, Blundell TL. Fragment ranking in modeling of protein structure. Conformationally constrained amino acid substitution tables. J Mol Biol 1993; 229: 194-220.

[55] Lessel U, Schomburg D. Similarities between protein 3D structures. Protein Eng 1994; 7: 1175-1187.

[56] Martin ACR, Thornton JM. Structual families in loops of monologous proteins: automatic classification, modeling and application to antibodies. J Mol Biol 1996; 263:800-815.

[57] Li W, Liu Z, Lai L. Protein loops on structurally similiar scaffolds: database and conformational analysis. Biopolymers 1999; 489:481-495.

[58] Sudarsanam S, DuBose RF, March CJ, Srinivasan S. Modeling protein loops using a fi-1,yi dimer database. Protein Sci 1995; 4: 1412-1420.

[59] van Vlijmen HWT, Karplus M. PDB-based protein loop prediction: parameters for selection and methods for optimization. J Mol Biol 1997; 267:975-1001.

[60] MacKerell JAD, Bashford D, Bellott M, Dunbrack RL, Evanseck J, Field MJ, Fischer S, Gao J, Guo H, Ha S, et al. All-hydrogen empirical potential for molecular modeling and dynamics studies of proteins using the CHARMM22 force field. J Phys Chem B 1998; 102: 3586-3616.

[61] Chothia C, Lesk AM. The relation between the divergence of sequence and structure in proteins. EMBO J 1986; 5: 823-826.

[62] Ring CS, Kneller DG, Langridge R, Cohen FE. Taxonomy and conformational analysis of loops in proteins. J Mol Biol 1992; 224: 685-699.

[63] Oliva B, Bates PA, Querol E, Aviles FX, Sternberg MJE. An automated classification of the structure of protein loops. J Mol Biol 1997; 266:814-830.

[64] Rufino SD, Donate LE, Canard LHJ, Blundell TL. Predicting the conformational class of short and medium size loops connecting regular secondary structures: application to comparative modelling. J Mol Biol 1997; 267:352-367.

[65] Wojcik J, Mornon JP, Chomilier J. New efficient statistical sequence-dependent structure prediction of short to medium-sized protein loops based on an exhaustive loop classification. J Mol Biol 1999; 1469-1490.

[66] Morea V, Tramontano A, Rustici M, Chothia C, Lesk AM. Conformations of the third hypervariable region in the VH domain of immunoglobulins. J Mol Biol 1998; 275:269-294.

[67] Chothia C, Lesk AM, Tramontano A, Levitt M, Smith Gill SJ, Air G, Sheriff S, Padlan EA, Davies D, Tulip WR, et. al. Conformation of immunoglobulin hypervariable regions. Nature 1989; 342: 877-883.

[68] Sibanda BL, Thornton JM. Beta-hairpin families in globular proteins. Nature 1985; 316: 170-174.

[69] Kwasigroch J-M, Chomilier J, Mornon J-P. A global taxonomy of loops in globular proteins. J Mol Biol 1996; 259:855-872. (Published erratum appears in J Mol Biol 1996; 261:673)

[70] Sun Z, Jiang B. Patterns and conformations of commonly occurring supersecondary structures (basic motifs) in protein data bank. J Protein Chem 1996; 15:675-690.

[71] Geetha V, Munson PJ. Linkers of secondary structures in proteins. Protein Sci 1997; 6:2538-2547.

[72] Wintjens RT, Rooman MJ, Wodak S. Automatic classification and analysis of aa-turn motifs in proteins. J Mol Biol 1997; 255:235-253.

[73] Ramachandran GN, Sasisekharan V. Conformation of polypeptides and proteins. Adv Protein Chem 1968; 28:283-437.

[74] Hobohm U, Scharf M, Schneider R, Sander C. Selection of a representative set of structures from the Brookhaven Protein Data Bank. Protein Sci 1992; 1:409-417.

[75] Hobohm U, Sander C. Enlarged representative set of protein structures. Protein Sci 1994; 3:522.

[76] Laskowski RA, MacArthur MW, Moss DS, Thornton JM PROCHECK: a program to check the stereochemical quality of protein structures. J Appl Cryst 1993; 26:283-291.

[77] Morris AL, MacArthur MW, Hutchinson EG, Thornton JM. Stereochemical quality of protein structure coordinates. Proteins 1992; 12:345-364.

[78] Samudrala R, Moult J. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. J Mol Biol 1998; 275: 895-916.

[79] Murzin AG. Progress in protein structure prediction. Nature Struct Biol 2001; 8:110-112.

[80] Kleywegt GJ, Jones TA. Good model-building and refinement practice. Methods in Enzymology 1997; 277:208-230.

[81] Danchin A. From protein sequence to function. Curr Opin Struct Biol 1999; 9:363-367.

[82] Moult J. Comparison of database potentials and molecular mechanics force fields. Curr Opin Struct Biol 1997; 7:194-199.

[83] Murzin AG, Patthy L. Sequences and topology: From sequence to structure to function. Curr Opin Struct Biol 1999; 9:359-362.

[84] Sanchez R, Šali A. Advances in comparative protein-structure modelling. Curr Opin Struct Biol 1997; 7:206-214.

[85] Bindewald E. Protein structure prediction with combinatorial optimization and fold recognition. Doctoral Thesis, Universität Mannheim, 2000.

[86] Tosatto SCE. Das Ringschlußproblem bei der Proteinfaltung. Diploma Thesis, Universität Mannheim, 1998.

[87] Kindler A. Entwicklung eines Verfahrens zur Optimierung der Struktur von Proteinsequenzen. Diploma Thesis, Universität Mannheim, 2000.

[88] Trabold A. Verfahren zur Modellierung von Loops in Proteinen. Diploma Thesis, Universität Mannheim, 2001.

[89] Adolph S. Implementierung einer grafischen Oberfläche zur Proteinstrukturvorhersage. Diploma Thesis, Universität Mannheim, 2000.

[90] Stryer L. Biochemie. Spektrum Akad. Verlag, 1991.

[91] Creighton T. Proteins: Structures and molecular properties. Freeman and Company, 2nd edition, 1993.

[92] Russel S, Norvig P. Artificial Intelligence - A modern approach. Prentice Hall, 1995.

[93] Böhm HJ, Klebe G, Kubinyi H. Wirkstoffdesign. Spektrum Akad. Verlag, 1996.

[94] Leach AR. Molecular Modelling - principles and applications. Addison Wesley, 1996.

[95] Gamma E, Helm R, Johnson R, Vlissides J. Design patterns. Addison Wesley, 1995.

[96] Stroustrup B. The C++ programming language. Addison Wesley, 1997.

[97] World wide web address of CAFASP-2. URL: `http://www.cs.bgu.ac.il/~dfischer/CAFASP2/`

[98] Meyers S. Effektiv C++ programmieren. Addison Wesley, 1995.

[99] Homepage of the CASP-4 experiment. URL: `http://PredictionCenter.llnl.gov/casp4/Casp4.html`

[100] Booch G. Objektorientierte Analyse und Design. Addison Wesley, 1995.

[101] Coad P, Yourdon E. Object-oriented analysis. Prentice-Hall, 1991.

[102] Alberts B, Bray D, Lewis J, Raff M, Roberts K, Watson JD. Molekularbiologie der Zelle. VCH Wiley, 1997.

[103] Bindewald E. Proteinstrukturvorhersage mit genetischen Algorithmen. Diploma Thesis, Universität Mannheim, 1997.

[104] Coad P, North D, Mayfield M. Object models - Strategies, patterns and applications. 2nd edition. Yourdon Press, 1997.

[105] Schader M, Rundshagen M. Objektorientierte Systemanalyse. 2. Auflage, Springer, 1996.

[106] Sedgewick R. Algorithmen in C++. Addison Wesley, 1992.

[107] Bruccoleri RE. Ab initio loop modeling and its application to homology modeling. Methods in Mol Biol, 2001; 143:247-263.

[108] De Maeyer M, Desmet J, Lasters I. The dead-end elimination theorem: Mathematical aspects, implementation, optimizations, evaluation and performance. Methods in Mol Biol, 2001; 143:265-304.

[109] Sternberg M (ed.). Protein Structure Prediction: A practical approach. The practical approach series, 1997.

[110] Lesk A. Introduction to protein architecture. Oxford University Press, 2001.

[111] Jones DT. Progress in protein structure prediction. Curr Opin Struct Biol, 1997; 7:377-387.

[112] Sternberg MJE, Bates PA, Kelley LA, MacCallum RM. Progress in protein structure prediction: assessment of CASP3. Curr Opin Struct Biol 1999; 9:368-373.

[113] Orengo CA, Todd AE, Thornton JM. From protein structure to function. Curr Opin Struct Biol 1999; 9:374-382.

[114] Koehl P, Delarue M. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. J Mol Biol 1994; 239:249-275.

[115] Hwang JK, Liao WF. Side-chain prediction by neural networks and simulated annealing optimization. Pro Eng 1995; 8:363-370.

[116] Tanimura R, Kidera A, Nakamura H. Determinants of protein side-chain packing. Protein Sci 1994; 3:2358-2365.

[117] Leach AR. Ligand docking to proteins with discrete side-chain flexibility. J Mol Biol 1994; 235:345-356.

[118] Petrella RJ, Lazaridis T, Karplus M. Protein sidechain conformer prediction: a test of the energy function. Fold Des 1998; 3:353-377.

[119] Tuffery P, Etchebest C, Hazout S, Lavery R. A new approach to the rapid determination of protein side chain conformations. J Biomol Struct & Dynamics, 1991; 8:1267-1289.

[120] Mendes J, Baptista AM, Carrondo MA, Soares CM. Improved modeling of side-chains in proteins with rotamer-based methods: A flexible rotamer model. Proteins 1999; 37:530-543.

[121] Voigt CA, Gordon DB, Mayo SL. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. J Mol Biol 2000; 299:789-803.

[122] Leach AR, Lemon AP. Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm. Proteins 1998; 33:227-239.

[123] Samudrala R, Moult J. A graph-theoretic algorithm for comparative modeling of protein structure. J Mol Biol 1998; 279:287-302.

[124] Koehl P, Delarue M. A self consistent mean field approach to simultaneous gap closure and side-chain positioning in homology modeling. Nature Struct Biol 1995; 2:163-170.

[125] Bower MJ, Cohen FE, Dunbrack RL. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: A new homology modeling tool. J Mol Biol 1997; 267: 1268-1282.

[126] Dunbrack RL, Karplus M. Backbone-dependent rotamer library for proteins. J Mol Biol 1993; 230:543-574.

[127] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissing H, Shindyalov IN, Bourne PE. The protein data bank. Nucl Acid Res 2000; 28:235-242. URL: `http://www.rcsb.org/PDB/`

[128] Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucl Acid Res 1997; 25:3389-3402.

[129] Chinea G, Padron G, Hooft RWW, Sander C, Vriend G. The use of position-specific rotamers in model building by homology. Proteins 1995; 23:415-421.

[130] Dunbrack RL, Karplus M. Conformational analysis of the backbone-dependent rotamer preferences of protein sidechains. Nature Struct Biol 1994; 1:334-340.

[131] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic Local Alignment Search Tool. J Mol Biol 1990; 215:403-410.

[132] Bairoch A, Apweiler R. The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1998. Nucl Acid Res 1998; 26:38-42.

[133] Holm L, Sander C. The FSSP database of structurally aligned protein fold families. Nucl Acid Res 1994; 22:3600-3609.

[134] Murzin AG, Brenner SE, Hubbard T, Chothia C. SCOP: A structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 1995; 247:536-540.

[135] Orengo CA, Michie AD, Jones S, Jones DT, Swindells MB, Thornton JM. CATH - a hierarchic classification of protein domain structures. Structure 1997; 5:1093-1108.

[136] Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucl Acid Res 1994; 22:4673-4680.

[137] Altschul SF, Boguski MS, Gish W, Wootton JC. Issues in searching molecular sequence databases. Nature Genetics 1994; 6:119-129.

[138] Vingron M. Near-optimal sequence alignment. Curr Opin Struct Biol 1996; 6:346-352.

[139] Rodriguez R, Chinea G, Lopez N, Pons T, Vriend G. Homology modeling, model and software evaluation: three relaed resources. Bioinformatics 1998; 14:523-528.

[140] Lasters I, De Maeyer M, Desmet J. Enhanced dead-end elimination in the search for the global minimum energy conformation of a collection of protein side chains. Protein Eng 1995; 8:815-822.

[141] Desmet J, De Maeyer M, Hazes B, Lasters I. The dead-end elimination theorem and its use in protein side-chain positioning. Nature 1992; 356:539-542.

[142] Goldstein RF. Efficient rotamer elimination applied to protein side-chains and related spin glasses. Biophys J 1994; 66:1335-1340.

[143] Lasters I, Desmet J. The fuzzy-end elimination theorem: correctly implementing the side chain placement algorithm based on the dead-end elimination theorem. Protein Eng 1993; 6:717-722.

[144] Gordon DB, Mayo SL. Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. J Comp Chem 1998; 19:1505-1514.

[145] De Maeyer M, Desmet J, Lasters I. All in one: a highly detailed rotamer library improves both accuracy and speed in the modelling of sidechains by dead-end elimination. Fold Des 1997; 2:53-66.

[146] Miller S, Janin J, Lesk AM, Chothia C. Interior and surface of monomeric proteins. J Mol Biol 1987; 196:641-656.

[147] Wilson C, Gregoret LM, Agard DA. Modeling side-chain conformation for homologous proteins using an energy-based rotamer search. J Mol Biol 1993; 996-1006.

[148] Kono H, Doi J. Energy minimization method using automata network for sequence and side-chain conformation prediction from given backbone geometry. Proteins 1994; 19:244-255.

[149] Ponder JW, Richards FM. Tertiary templates for proteins. Use of packing criteria in the enumeration of allowed sequences for different structural classes. J Mol Biol 1987; 193:775-791.

[150] Vásquez M. An evaluation of discrete and continuum search techniques for conformational analysis of side chains in proteins. Biopolymers 1995; 36:53-70.

[151] Lee C, Subbiah S. Prediction of protein side-chain conformation by packing optimization. J Mol Biol 1991; 217:373-388.

[152] Dahiyat BI, Mayo SL. Protein design automation. Protein Sci 1996; 5:895-903.

[153] Dahiyat BI, Gordon DB, Mayo SL. Automated design of the surface positions of protein helices. Protein Sci 1997; 6:1333-1337.

[154] Harbury PB, Tidor B, Kim PS. Repacking protein cores with backbone freedom: structure prediction for coiled coils. Proc Natl Acad Sci USA 1995; 92:8408-8412.

[155] Holm L, Sander C. Fast and simple Monte Carlo algorithm for side chain optimization in proteins: Application to model building by homology. Proteins 1992; 14:213-223.

[156] Laughton CA. Prediction of protein side-chain conformations from local three-dimensional homology relationships. J Mol Biol 1994; 235:1088-1097.

[157] Vásquez M. Modeling side-chain conformation. Curr Opin Struct Biol 1996; 6:217-221.

[158] Dunbrack RL, Cohen FE. Bayesian statistical analysis of protein side-chain rotamer preferences. Protein Sci 1997; 6:1661-1681.

[159] Doig AJ, Sternberg MJE. Side-chain conformational entropy in protein folding. Protein Sci 1995; 4:2247-2251.

[160] Tufféry P, Etchebest C, Hazout S. Prediction of protein side chain conformations: a study on the influence of backbone accuracy on conformation stability in the rotamer space. Protein Eng 1997; 10:361-372.

[161] Dahiyat BI, Mayo SL. De novo protein design: Fully automated sequence selection. Science 1997; 278:82-87.

[162] Shenkin PS, Farid H, Fetrow JS. Prediction and evaluation of side-chain conformations for protein backbone structures. Proteins 1996; 26:323-352.

[163] Tufféry P, Lavery R. Packing and recognition of protein structural elements: a new approach applied to the 4-helix bundle of myohemerythrin. Proteins 1993; 15:413-425.

[164] Pickett SD, Sternberg MJE. Empirical scale of side-chain conformational entropy in protein folding. J Mol Biol 1993; 231:825-839.

[165] Creamer TP, Rose GD. α-helix-forming propensities in peptides and proteins. Proteins 1994; 19:85-97.

[166] McGregor MJ, Islam SA, Sternberg MJE. Analysis of the relationship between side-chain conformation and secondary structure in globular proteins. J Mol Biol 1987; 198:295-310.

[167] Levitt M. Accurate modeling of protein conformation by automatic segment matching. J Mol Biol 1992; 226:507-533.

[168] Lee C. Predicting protein mutant energetics by self-consistent ensemble optimization. J Mol Biol 1994; 236:918-939.

[169] Abagyan R, Totrov M. Biased probability Monte Carlo conformational searches and electrostatic calculations for peptides and proteins. J Mol Biol 1994; 235:983-1002.

[170] Gordon DB, Mayo SL. Branch-and-Terminate: a combinatorial optimization algorithm for protein design. Structure 1999; 7:1089-1098.

[171] Lazar GA, Desjarlais JR, Handel TM. De novo design of the hydrophobic core of ubiquitin. Protein Sci 1997; 6:1167-1178.

[172] Eisenmenger F, Argos P, Abagyan R. A method to configure protein side-chains from the main-chain trace in homology modeling. J Mol Biol 1993; 231:849-860.

[173] Cregut D, Liautard JP, Chiche L. Homology modeling of annexin I: implicit solvation improves side-chain prediction and combination of evaluation criteria allows recognition of different types of conformational error. Protein Eng 1994; 7:1333-1344.

[174] Keller DA, Shibata M, Marcus E, Ornstein RL, Rein R. Finding the global minimum: a fuzzy end elimination implementation. Protein Eng 1995; 8:893-904.

[175] Schiffer CA, Caldwell JW, Kollman PA, Stroud RM. Prediction of homologous protein structures based on conformational searches and energetics. Proteins 1990; 8:30-43.

[176] Miyazawa S, Jernigan RL. Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. J Mol Biol 1996; 256:623-644.

[177] Koppensteiner WA, Sippl MJ. Knowledge-based potentials - back to the roots. Biochemistry (Moscow) 1998; 63:247-252.

[178] Miyazawa S, Jernigan RL. Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. Macromolecules 1985; 18:534-552.

[179] Lund O, Frimand K, Gorodkin J, Bohr H, Bohr J, Hansen J, Brunak S. Protein distance constraints predicted by neural networks and probability density functions. Protein Eng 1997; 10:1241-1248.

[180] Sippl MJ. Knowledge-based potentials for proteins. Curr Opin Struct Biol 1995; 5:229-235.

[181] Finkelstein AV. Protein structure: what is it possible to predict now? Curr Opin Struct Biol 1997; 7:60-71.

[182] Sander C, Schneider R. Database of homology-derived protein structures and the structural meaning of sequence alignment. Proteins 1991; 9:56-68.

[183] Tobi D, Shafran G, Linial N, Elber R. On the design and analysis of protein folding potentials. Proteins 2000; 40:71-85.

[184] Sippl MJ. Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins. J Mol Biol 1990; 213:859-883.

[185] Šali A, Blundell TL. Comparative protein modelling by satisfaction of spatial restraints. J Mol Biol 1993; 234:779-815.

[186] Casari G, Sippl MJ. Structure-derived hydrophobic potential. Hydrophobic potential derived from X-ray structures of globular proteins is able to identify native folds. J Mol Biol 1992; 224:725-732.

[187] Jones DT, Miller RT, Thornton JM. Successful protein fold recognition by optimal sequence threading validated by rigorous blind testing. Proteins 1995; 23:387-397.

[188] Turcotte M, Muggleton SH, Sternberg MJE. Automated discovery of structural signatures of protein fold and function. J Mol Biol 2001; 306:591-605.

[189] Park J, Karplus K, Barrett C, Hughey R, Haussler D, Hubbard T, Chothia C. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. J Mol Biol 1998; 284:1201-1210.

[190] Osguthorpe DJ. Ab initio protein folding. Curr Opin Struct Biol 2000; 10:146-152.

[191] Russell RB, Copley RR, Barton GJ. Protein fold recognition by mapping predicted secondary structures. J Mol Biol 1996; 259:349-365.

[192] Sippl MJ, Weitckus S. Detection of native-like models for amino acid sequences of unknown three-dimensional structure in a data base of known protein conformations. Proteins 1992; 13:258-271.

[193] Jones DT, Taylor WR, Thornton JM. A new approach to protein fold recognition. Nature 1992; 358:86-89.

[194] Karplus K, Barrett C, Hughey R. Hidden markov models for detecting remote protein homologies. Bioinformatics 1998; 14:846-856.

[195] Kelley LA, MacCallum RM, Sternberg MJE. Enhanced genome annotation using structural profiles in the program 3D-PSSM. J Mol Biol 2000; 299:499-520.

[196] Huang ES, Subbiah S, Levitt M. Recognizing native folds by the arrangement of hydrophobic and polar residues. J Mol Biol 1995; 252:709-720.

[197] Novotny J, Rashin AA, Bruccoleri RE. Criteria that discriminate between native proteins and incorrectly folded models. Proteins 1988; 4:19-30.

[198] Bujnicki JM, Elofsson A, Fischer D, Rychlewski L. LiveBench-1: Continuous benchmarking of protein structure prediction servers. Protein Sci 2001; 10:352-361.

[199] Martin ACR, Orengo CA, Hutchinson EG, Jones S, Karmirantzou M, Laskowski RA, Mitchell JBO, Taroni C, Thornton JM. Protein folds and functions. Structure 1998; 8:875-884.

[200] Orengo CA, Pearl FMG, Bray JE, Todd AE, Martin AC, Lo Conte L, Thornton JM. The CATH database provides insights into protein structure/function relationships. Nucl Acid Res 1999; 27:275-279.

[201] Godzik A. Knowledge-based potentials for protein folding: what can we learn from known protein structures? Structure 1996; 4:363-366.

[202] Rost B. Twilight zone of protein sequence alignments. Protein Eng 1999; 12:85-94.

[203] Eisenberg D, McLachlan AD. Solvation energy in protein folding and binding. Nature 1986; 319:199-203.

[204] Jones DT. Protein structure prediction in the postgenomic era. Curr Opin Struct Biol 2000; 10:371-379.

[205] Jones DT. GenTHREADER: An efficient and reliable protein fold recognition method for genomic sequences. J Mol Biol 1999; 287:797-815.

[206] Fischer D. Hybrid fold recognition: Combining sequence derived properties with evolutionary information. Pac Symp Biocomput 2000; 119-130.

[207] Rychlewski L, Jaroszewski L, Li W, Godzik A. Comparison of sequence profiles. Strategies for structural predictions using sequence information. Protein Sci 2000; 9:232-241.

[208] Tosatto S, Bindewald E, Maydt J, Trabold A, Hesser J, Männer R. Ab initio loop modeling with precalculated synthetic loops and sidechain placement. CASP4 Manual 2000; A-119.

[209] Bindewald E, Tosatto S, Maydt J, Trabold A, Hesser J, Männer R. Secondary structure and function based protein fold recognition. CASP4 Manual 2000; A-9.

[210] Tosatto SCE, Bindewald E, Hesser J, Männer R. A divide and conquer approach to fast loop modeling. Protein Eng 2002; in press.

[211] Bindewald E, Tosatto SCE, Hesser J, Männer R. Protein fold recognition based on function, secondary structure and sequence similarity. Submitted to Protein Eng 2002.

[212] Winter R, Noll F. Methoden der biophysikalischen Chemie. Teubner Studienbücher Chemie, 1998.

[213] van Holde KE, Johnson WC, Ho PS. Principles of physical biochemistry. Prentice Hall, 1998.

[214] Chou PY, Fasman GD. Prediction of protein conformation. Biochemistry 1974; 13: 222-245.

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

[216] Lee B, Richards FM. The interpretation of protein structures: Estimation of static accessibility. J Mol Biol, 1971; 55:379-400.

[217] PROSTAR: The protein potential test site. URL: `http://prostar.carb.nist.gov/`

[218] Cornell WD, Ciepak P, Bayly CI, Gould IR, Merz KM Jr, Ferguson DM, Spelleyer DC, Fox T, Caldwell JW, Kollman PA. A second generation force field for the simulation of proteins, nucleic acids and organic molecules. Biochemistry 1995; 117:5179-5197.

[219] van Gunsteren WF, Billeter SR, Eising AA, Hünenberger PH, Krüger P, Mark AE, Scott WRP, Tironi IG. Biomolecular simulation: the GROMOS96 manual and user guide. Hochschulverlag an der ETH Zürich 1996.

[220] Hühnenberger PH, van Gunsteren WF. Empirical classical interaction functions for molecular simulation. In computer simulation of biomolecular systems, theoretical and experimental applications, vol III. Edited by van Gunsteren WF, Weiner PK, Wilkinson AJ. ESCOM 1997.

[221] Sippl MJ, Ortner M, Jaritz M, Lackner P, Flöckner H. Helmholtz free energies of atom pair interactions in proteins. Fold Des 1996; 1:289-298.

[222] Jernigan RL, Bahar I. Structure-derived potentials and protein simulations. Curr Opin Struct Biol 1996; 6:195-209.

[223] Kabsch W, Sander C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 1983; 22:2577-2637.

[224] Skolnick J, Jaroszewski L, Kolinski A, Godzik A. Derivation and testing of pair potentials for protein folding. When is the quasichemical approximation correct? Pro Sci 1997; 6:676-688.

[225] Lazaridis T, Karplus M. Effective energy functions for protein structure prediction. Curr Opin Struct Biol 2000; 10:139-145.

[226] Vajda S, Sippl M, Novotny J. Empirical potentials and functions for protein folding and binding. Curr Opin Struct Biol 1997; 7:222-228.

[227] Hao MH, Scheraga HA. Designing potential energy functions for protein folding. Curr Opin Struct Biol 1999; 9:184-188.

[228] Anfinsen CB. Principles that govern the folding of protein chains. Science 1973; 181:223-230.

[229] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculations by fast computing machines. J Chem Phys 1953; 21:1087-1092.

[230] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science 1983; 220:671-680.

[231] Pollastri G, Baldi P, Fariselli P, Casadio R. Improved prediction of the number of residue contacts in proteins by recurrent neural networks. Proceedings of the ISMB01 Conference 2001; AAAI Press.

[232] Richardson CJ, Barlow DJ. The bottom line for prediction of residue solvent accessibility. Protein Eng 1999; 12:1051-1054.

[233] Fariselli P, Casadio R. Prediction of the number of residue contacts in proteins. Proceedings of the ISMB00 Conference 2000; AAAI Press, 146-151.

[234] Janin J, Wodak S, Levitt M, Maigret B. Conformation of amino acid side chains in proteins. J Mol Biol 1978; 125:357-386.

[235] Schrauber H, Eisenhaber F, Argos P. Rotamers: to be or not to be? An analysis of amino acid side-chain conformations in globular proteins. J Mol Biol 1993; 230:592-612.

[236] Lee C, Levitt M. Accurate prediction of the stability and activity effects of site-directed mutagenesis on a protein core. Nature 1991; 352:448-451.

[237] Dechter R, Pearl J. Generalized best-first search strategies and the optimality of A*. Journal of the Association for Computing Machinery 1985; 32:505-536.

[238] Pearl J, Kim J. Studies in semi-admissible heuristics. IEEE Transactions on pattern analysis and machine intelligence 1982; PAMI-4:4.

[239] Srinivasan N, Guruprasad K, Blundell TL. Comparative modelling of proteins. In M. Sternberg (ed.): Protein Structure Prediction: A practical approach, 1995. The practical approach series, 111-140.

[240] Jones TA, Kleywegt GJ. CASP3 comparative modeling evaluation. Proteins 1999; S3:30-46.

[241] Martin ACR, MacArthur MW, Thornton JM. Proteins 1997; Assessment of comparative modeling in CASP2. Proteins 1997; S1:14-28.

[242] Havel TF, Snow ME. A new method for building protein conformations from sequence alignments with homologues of known structure. J Mol Biol 1991; 217:1-7.

[243] Srinivasan S, March CJ, Sudarsanam S. An automated method for modeling proteins on known templates using distance geometry. Pro Sci 1993; 2:227-289.

[244] Deane CM, Kaas Q, Blundell TL. SCORE: Predicting the core of protein models. Bioinformatics 2001; 17:541-550.

[245] Šali A, Potterton L, Yuan F, van Vlijmen H, Karplus M. Evaluation of comparative modeling by MODELLER. Proteins 1995; 28:818-326.

[246] Holm L, Sander C. Protein structure comparison by alignment of distance matrices. J Mol Biol 1993; 233:123-138.

[247] Shindyalov IN, Bourne PE. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Prot Eng 1998; 11:739-747.

[248] Orengo CA, Taylor WR. SSAP: sequential structure alignment program for protein structure comparison. Meth Enz 1996; 266:617-635.

[249] Sauder JM, Arthur JW, Dunbrack RL Jr. Large-scale comparison of protein sequence alignment algorithms with structural alignments. Proteins 2000; 40:6-22.

[250] Blundell TL, Sibanda BL, Sternberg MJE, Thornton JM. Knowledge-based prediction of protein structures and the design of novel molecules. Nature 1987; 326:347-352.

[251] Ahlrichs R, Bär M, Häser M, Horn H, Kölmel C. Electronic structure calculations of workstation computers: The program system turbomole. Chem Phys Letters 1989; 162:165-169.

[252] Frisch MJ, Trucks GW, Schlegel HB, Scuseria GE, Robb MA, Cheeseman JR, Zakrewski VG, et al. Gaussian 98 (Revision A.7). Gaussian, Inc. 1998.

[253] Shakhnovich EI. Theoretical studies of protein-folding thermodynamics and kinetics. Curr Opin Struct Biol 1997; 7:29-40.

[254] Browne WJ, North ACT, Philips DC, Drew K, Vanaman TC, Hill RL. J Mol Biol 1969; 42:65.

[255] Greer J. J Mol Biol 1981; 153: 1027.

[256] Smith TF, Waterman MS. Identification of common molecular subsequences. J Mol Biol 1981; 147:195-197.

[257] Vingron M, Waterman MS. Sequence alignment and penalty choice: review of concepts, case studies and implications. J Mol Biol 1994; 235:1-12.

[258] Bowie JU, Lüthy R, Eisenberg D. A method to identify protein sequences that fold into a known three-dimensional structure. Science 1991; 253:164-170.

[259] Bryant SH, Lawrence CE. An empirical energy function for threading protein-sequence through the folding motif. Proteins 1993; 16:92-112.

[260] Fischer D, Eisenberg D. Protein fold recognition using sequence-derived predictions. Protein Sci 1996; 5:947-955.

[261] Rost B, Sander C. Progress of 1D protein structure prediction at last. Proteins 1995; 23:295-300.

[262] Rost B. Fitting 1D predictions into 3D structures. In Protein Folds. A Distance Based Approach. Edited by Bohr H, Brunak S. CRC Press 1995; 132-151.

[263] Krogh A, Brown M, Mian IS, Sjölander K, Haussler D. Hidden Markov models in computational biology: Applications to protein modeling. J Mol Biol 1994; 235:1501-1531.

[264] Wodak SJ, Rooman MJ. Generating and testing protein folds. Curr Opin Struct Biol 1993; 3:247-259.

[265] Bowie JU, Eisenberg D. Inverted protein structure prediction. Curr Opin Struct Biol 1993; 3:437-444.

[266] Eddy SR. Hidden Markov Models. Curr Opin Struct Biol 1996; 6:361-365.

[267] Godzik A, Skolnick J. Sequence-structure matching in globular proteins: application to supersecondary and tertiary structure determination. Proc Natl Acad Sci USA 1992; 89:12098-12102.

[268] Flöckner H, Braxenthaler M, Lackner P, Jaritz M, Ortner M, Sippl MJ. Progress in fold recognition. Proteins 1995; 23:376-386.

[269] Kocher JPA, Rooman MJ, Wodak SJ. Factors influencing the ability of knowledge-based potentials to identify native sequence-structure matches. J Mol Biol 1994; 235:1598-1613.

[270] Alexandrov NN, Nussinov R, Zimmer RM. Fast protein recognition via sequence to structure alignment and contact capacity potentials. Pac Symp Biocomput 1996; 53-72.

[271] Thiele R, Zimmer RM, Lengauer T. Protein threading by recursive dynamic programming. J Mol Biol 1999; 290:757-779.

[272] Lathrop RH, Smith TF. Global optimum protein threading with gapped alignment and empirical pair score functions. J Mol Biol 1996; 255:641-665.

[273] Baker D, Šali A. Protein structure prediction and structural genomics. Science 2001; 294:93-96.

[274] Standley DM, Eyrich VA, Felts AK, Friesner RA, McDermott AE. A branch and bound algorithm for protein structure refinement from sparse NMR data sets. J Mol Biol 1999; 1691-1710.

[275] Sjölander K, Karplus K, Brown MP, Hughey R, Krogh A, Mian IS, Haussler D. Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology. Comp Applic Biosci 1996; 12:327-345.

[276] Gribskov M, McLachlan AD, Eisenberg D. Profile analysis: detection of distantly related proteins. Proc Natl Acad Sci USA 1987; 84:4355-4358.

[277] Tatusov RL, Altschul SF, Koonin EV. Detection of conserved segments in proteins: iterative scanning of sequence databases with alignment blocks. Proc Natl Acad Sci 1994; 91:12091-12095.

[278] Bujnicki JM, Elofsson A, Fischer D, Rychlewski L. Structure prediction meta server. Bioinformatics 2001; 17:750-751.

[279] Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 1999; 292:195-202.

[280] Bateman A, Birney E, Durbin R, Eddy SR, Howe KL, Sonnhammer ELL. The Pfam protein families database. Nucleic Acids Res 2000; 263-266.

[281] Clore GM, Robien MA, Gronenborn AM. Exploring the limits of precision and accuracy of protein structures determined by nuclear magnetic resonance spectroscopy. J Mol Biol 1993; 231:82-102.

[282] Burke DF, Deane CM. Improved protein loop prediction from sequence alone. Protein Eng 2001; 14:474-478.

[283] Webster DM (ed.). Protein Structure Prediction: Methods and Protocols. Humana Press 2000.

[284] Branden C, Tooze J. Introduction to protein structure. 2nd edition. Garland Publishing 1999.

[285] Banaszak LJ. Foundations of structural biology. Academic Press 2000.

[286] Tolkien, JRR. The Lord of the Rings. George Allen & Unwin Ltd., 1966.

[287] Fukunaga K. Introduction to Statistical Pattern Recognition. 2nd edition. Academic Press 1990.

[288] MacKeown PK, Newman DJ. Computational Techniques in Physics. Adam Hilger 1987.

[289] Ottmann T, Widmayer P. Algorithmen und Datenstrukturen. 3rd edition. Spektrum Akad. Verlag 1996.

[290] Bonneau R, Baker D. Ab initio protein structure prediction: Progress and prospects. Ann Rev Biophys Biomol Struct 2001; 30:173-89.

[291] Mirny L, Shakhnovich E. Protein folding theory: From lattice to all-atom models. Ann Rev Biophys Biomol Struct 2001; 30:361-396.

[292] Wang W, Donini O, Reyes CM, Kollman PA. Biomolecular simulations: Recent developments in force fields, simulations of enzyme catalysis, protein-ligand, protein-protein and protein-nucleic acid noncovalent interactions. Ann Rev Biophys Biomol Struct 2001; 30:211-243.

[293] Reimer U, Scherer G, Drewello M, Kruber S, Schutkowski M, Fischer G. Side-chain effects on peptidyl-prolyl cis/trans isomerisation. J Mol Biol 1998; 279:449–460.

[294] Bindewald E. Proteinstrukturvorhersage mit genetischen Algorithmen. Diploma Thesis, Universität Mannheim, 1997.

[295] Baldi P, Brunak S, Chauvin Y, Andersen CAF, Nielsen H. Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 2000; 16:412-424.

[296] Elofsson A, Sonnhammer ELL. A comparison of sequence and structure protein domain families as a basis for structural genomics. Bioinformatics 1999; 15:480-500.

[297] D'Alfonso G, Tramontrano A, Lahm A. Structural conservation in single-domain proteins: Implications for homology modeling. J Struct Biol 2001; 134:246-256.

[298] Gerstein M, Jansen R. The current excitement in bioinformatics - analysis of whole-genome expression data: how does it relate to protein structure and function? Curr Opin Struct Biol 2000; 10:574-584.

[299] Shapiro L, Harris T. Finding function through structural genomics. Curr Opin Biotech 2000; 11:31-35.

[300] Baker D, Šali A. Protein structure prediction and structural genomics. Science 2001; 294:93-96.

[301] Pokala N, Handel TM. Protein design — where we were, where we are, where we're going. J Struct Biol 2001; 134:269-281.

[302] Dietmann S, Holm L. Identification of homology in protein structure classification. Nature Struct Biol 2001; 8:953-957.

[303] Lindahl E, Elofsson A. Identification of related proteins on family, superfamily and fold level. J Mol Biol 2000; 295:613-625.

[304] Panchenko AR, Marchler-Bauer A, Bryant SH. Combination of threading potentials and sequence profiles improves fold recognition. J Mol Biol 2000; 296:1319-1331.

[305] Domingues FS, Lackner P, Andreeva A, Sippl MJ. Structure-based evaluation of sequence comparison and fold recognition alignment accuracy. J Mol Biol 2000; 297:1003-1013.

[306] Matsuo Y, Bryant SH. Identification of homologous core structures. Proteins 1999; 35:70-79.

[307] Gough J, Karplus K, Hughey R, Chothia C. Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. J Mol Biol 2001; 903-919.

[308] Mendes J, Soares CM, Carrondo MA. Improvement of side-chain modeling in proteins with the self-consistent mean field theory method based on an analysis of the factors influencing prediction. Biopolymers 1999; 50:111-131.

[309] Sippl MJ, Lackner P, Domingues FS, Pric A, Malik R, Andreeva A, Wiedensteiner M. Assessment of the CASP4 fold recognition category. Proteins 2001; in press.

[310] Lesk AM, Lo Conte L, Hubbard TJP. Assessment of novel fold targets in CASP4: Predictions of three-dimensional structures, secondary structures and interresidue contacts. Proteins 2001; in press.

[311] Rost B, Sander C. Prediction of protein secondary structure at better than 70% accuracy. J Mol Biol 1993; 232:584-599.

[312] Cuff JA, Clamp ME, Siddiqui AS, Finlay M, Barton GJ. Jpred: a consensus secondary structure prediction server. Bioinformatics 1998; 15:937-946.

[313] Fariselli P, Casadio R. A neural network based predictor of residue contacts in proteins. Protein Eng 1999; 12:15-21.

[314] Karplus K, Barrett C, Karchin R, Hughey R, Diekhans M, Grate L, Cline M. SAM-T2K protein structure predictions. CASP-4 Manual 2000; A-81.

[315] Burke DF, Campillo N, Deane C, de Bakker P, Chen L, Innis A, Lovell S, Mueller J, Mizuguchi K, Nagendra HG, Nunez R, Shi J, Shirai H,

Williams MG, Blundell TL. Comparative modelling incorporating structural features and environmental properties. CASP-4 Manual 2000; A-11.

[316] Mizuguchi K, Deane CM, Blundell TL, Overington JP. HOMSTRAD: A database of protein structure alignments for homologous families. Protein Sci 1998; 7:2469-2471.

[317] Guex N, Diemand A, Peitsch MC. Protein modelling for all. Trends Bio Sci 1999; 24:364-367. URL: `http://www.expasy.ch/swissmod/SWISS-MODEL.html`

[318] URL: `http://www.apache.org/`

[319] URL: `http://www.bmm.icnet.uk/3djigsaw/`

[320] Bates PA, Sternberg MJE. Model building by comparison: selecting and improving algorithms via expert knowledge. CASP-4 Manual 2000; A-132.

[321] Bonneau R, Tsai J, Ruczinski I, Baker D. Ab initio structure prediction using Rosetta. CASP-4 Manual 2000; A-118.

[322] Simons KT, Bonneau R, Ruczinski I, Baker D. Ab initio protein structure prediction of CASP-3 targets using Rosetta. Proteins 1999; Sup 3:171-176.

[323] Venclovas C. Sequence-structure alignment selection by 3D structure evaluation. CASP-4 Manual 2000; A-106.

[324] An Y, Eyrich VA, Gunn J, Pincus DL, Standley DM, Friesner RA. Protein structure prediction using a combination of threading and restrained energy minimization. CASP-4 Manual 2000; A-31.

[325] Kolinski A, Kihara D, Betancourt M, Rotkiewicz P, Boniecki M, Skolnick J. Ab initio protein folding using threading based, tertiary restraints. CASP-4 Manual 2000; A-95.

[326] Dill KA, Sun Chan H. From Levinthal to pathways to funnels. Nature Struct Biol 1997; 4:10-19.

[327] Dobson CM, Šali A, Karpls M. Proteinfaltung aus theoretischer und experimenteller Sicht. Angewandte Chemie 1998; 110:908-935.

[328] Crippen GM, Ohkubo Y. Statistical mechanics of protein folding by exhaustive enumeration. Proteins 1998; 32:425-437.

[329] Siebert S. Konformationsanalyse und Fluoreszenz-Mustererkennung von farbstoffmarkierten Konjugaten mit neuen Algorithmen. Doctoral Thesis, Universität Heidelberg, 1998.

[330] Levinthal C. How to fold graciously. Mossbauer spectroscopy in biological systems. (ed.) Debrunner P, Tsibris JCM. University Illinois, Urbana Champaign, 1969; 22-24.

[331] URL: `http://cubic.bioc.columbia.edu/~eva/`

[332] Fischer D, Elofsson A, Rychlewski L. The 2000 Olympic games of protein structure prediction; fully automated programs are being evaluated *vis-à-vis* human teams in the protein structure prediction experiment CAFASP2. Protein Eng 2000; 13:667-669.

[333] Shindyalov IN, Bourne P. An alternative view of protein fold space. Proteins 2000; 38:247-260.

[334] Reeck GR, de Haen C, Teller DC, Doolittle RF, Fitch WM, Dickerson RE, Chambon P, McLachlan AD, Margolish E, Jukes TH, Zuckerland E. Cell 1987; 50:667.

[335] Gouet P, Courelle E, Stuart DI, Metoz F. ESPript: multiple sequence alignments in PostScript. Bioinformatics 1999; 15:305-308.

Appendix A
CASP4 Material

This Appendix presents additional data related to the CASP-4 experiment in 2000 [99]. The prediction targets are listed, before the numerical data relative to our group's prediction is presented. Finally, graphical summaries for each target to show the relative performance compared to all other participants are given.

Please note that the information below is not exhaustive, as the CASP organizers have produced a larger number of different indicators, more than could be possibly printed in this Appendix. For the full information see the CASP-4 homepage [99].

A.1 Prediction Targets

The following listing contains information on all CASP-4 targets. From left to right this is: the target identifier and name; its number of residues; the experimental method used to determine the structure; the entry date when it was originally posted on the CASP website; the expiration date or submission deadline; and a short description of the protein in question.

Target	Name	Nres	Method	Entry	Expiry	Description
T0086	UBIC	164	X-ray	11 May	20 Jul	Chorismate lyase, E. coli
T0087	PPX1	310	X-ray	12 May	7 Sep	PPase, S. mutans
T0088	GAFD	156	X-ray	12 May	7 Sep	GafD, E. coli
T0089	FTSA	419	X-ray	18 May	1 Sep	FtsA, T. maritima

T0090	YQIE	209	X-ray	18	May	6 Jul	ADP-ribose pyrophosphatase, E. coli
T0091	YBAB	109	X-ray	20	May	8 Sep	Hypothetical protein HI0442, H. influenzae
T0092	YECO	241	X-ray	20	May	22 Jul	Hypothetical protein HI0319, H. influenzae
T0093	YIBK	160	X-ray	20	May	22 Jul	Hypothetical protein HI0766, H. influenzae
T0094	CPDase	181	X-ray	25	May	1 Aug	Cyclic phosphodiesterase, A.thaliana
T0095	CTN1	244	X-ray	5	Jun	8 Sep	Alpha(E)-catenin fragment, mouse
T0096	FADR	239	X-ray	8	Jun	8 Sep	FadR, E. coli
T0097	ER29	105	NMR	8	Jun	31 Aug	C-terminal domain of ERp29, rat
T0098	SP0A	121	X-ray	12	Jun	15 Aug	C-terminal domain of Spo0A, B. stearothermophilus
T0099	-	56	NMR	12	Jun	25 Jul	-
T0100	PMEA	342	X-ray	13	Jun	3 Jul	Pectin Methylesterase, E. chrysanthemi
T0101	PELL	400	X-ray	13	Jun	11 Sep	Pectate lyase PelL, E. chrysanthemi
T0102	AS48	70	NMR	16	Jun	30 Aug	Bacteriocin AS-48, E. faecalis
T0103	PICP	372	X-ray	27	Jun	11 Sep	Pepstatin insensitive carboxyl proteinase, Pseudomonas sp.
T0104	YJEE	158	X-ray	27	Jun	11 Sep	Hypothetical protein HI0065, H. influenzae
T0105	SP100	94	NMR	5	Jul	31 Aug	Protein Sp100b, human
T0106	SFRP3	128	X-ray	5	Jul	25 Jul	Secreted frizzled protein 3, mouse
T0107	CBD9	188	X-ray	10	Jul	29 Aug	Family 9 carbohydrate

						binding module, T. maritima
T0108	CBD17	206	X-ray	10 Jul	1 Sep	Family 17 carbohydrate binding module, C. cellulovorans
T0109	ORN	182	X-ray	10 Jul	1 Sep	Oligoribonuclease, H. influenzae
T0110	RBFA	128	X-ray	10 Jul	12 Sep	Ribosome-binding factor A, H. influenzae
T0111	ENO	431	X-ray	12 Jul	12 Sep	Enolase, E. coli
T0112	DHSO	352	X-ray	13 Jul	31 Aug	Ketose Reductase / Sorbitol Dehydrogenase, B. argentifolii
T0113	HCD2	261	X-ray	13 Jul	4 Aug	Short chain 3-hydroxyacyl-coa dehydrogenase, rat
T0114	AFP1	87	NMR	13 Jul	4 Aug	Antifungal protein AFP-1, S. tendae
T0115	KHSE	300	X-ray	18 Jul	12 Sep	Homoserine kinase, M. jannaschii
T0116	MUTS	811	X-ray	24 Jul	31 Aug	MutS, T. Aquaticus
T0117	DNK	250	X-ray	25 Jul	13 Sep	Deoxyribonucleoside kinase, D. melanogaster
T0118	ENRN	149	X-ray	26 Jul	13 Sep	Endodeoxyribonuclease I, Bacteriophage T7
T0119	BENC	338	X-ray	26 Jul	13 Sep	Benzoate dioxygenase reductase, Acinetobacter sp.
T0120	XRCC4	336	X-ray	26 Jul	14 Sep	DNA repair protein XRCC4, human
T0121	MALK	372	X-ray	18 Aug	14 Sep	MalK, T. litoralis
T0122	TRPA	248	X-ray	18 Aug	14 Sep	Tryptophan Synthase alpha subunit, P. furiosus
T0123	LACB	160	X-ray	18 Aug	15 Sep	Beta-lactoglobulin, pig
T0124	PLCB	242	X-ray	22 Aug	15 Sep	Phospholipase C beta C-terminus, turkey
T0125	SP18	141	X-ray	23 Aug	15 Sep	Sp18 protein, H. fulgens
T0126	OMP	163	X-ray	23 Aug	15 Sep	Olfactory marker

						protein, mouse
T0127	BCHI	350	X-ray	29 Aug	15 Sep	Magnesium chelatase, R. capsulatus
T0128	SODM	222	X-ray	29 Aug	15 Sep	Manganese superoxide dismutase homolog, P. aerophilum

A.2 Numerical Evaluation

The numerical evaluation performed by the CASP organizers is divided in two parts. All predictions have been evaluated with a set of global quality measures. Comparative modeling targets were analyzed in more detail. Additional measures, such as RMSD of "loops" and "core", can only be calculated for this class of models. Both evaluations will be presented, starting with the comparative modeling targets. The following measures are listed next to each target identifier:

CRMSCA ALL_ATOMS : RMSD for all C_α atoms in the structure.

ATOMCA_PP ALL_ATOMS : percent C_α atoms submitted.

CRMSCA LSHIFTS/INS : RMSD for C_α atoms belonging to large shifts and insertions, i.e. "loops".

CRMSCA CORE : RMSD for C_α atoms belonging to the "core" of the structure.

PRINCIPAL PARENT : Template structure with lowest RMSD.

CRMSCA T-P : RMSD for C_α atoms between target and principal parent.

CRN : RMSD for C_α atoms divided by number of predicted C_α atoms, i.e. per residue RMSD.

GDT_TS : Global distance test total score. Computed as the average of the percentage residues superposed with less than 1, 2, 4 and 8 Å RMSD.

SOV_O : Segment overlap measure for secondary structure elements (percentage).

Comparative Modeling Targets

Prediction	CRMSCA ALL_ATOMS	CRMSCA LSHIFTS/INS	CRMSCA CORE	PRINCIPAL PARENT	CRMSCA T-P	CRN
T0089	19.88	19.53	20.51	1dkg_D	1.77	0.0526
T0089_1	24.88	23.84	25.54	1ats	1.43	0.1716
T0089_2	16.06	21.69	6.75	1dga_A	1.67	0.3089
T0089_4	14.39	9.60	16.80	1dej_A	1.37	0.1424
T0090	17.03	18.29	10.62	1mut.m_15	1.92	0.0856
T0090_2	11.16	11.42	10.36	1mut.m_14	1.79	0.0781
T0092	15.70	14.00	17.87	1d2c_A	1.57	0.0692
T0099	10.57	11.02	9.72	1lck_A	1.59	0.1888
T0103	16.19	18.39	13.41	1ak9	1.23	0.0440
T0111	1.89	6.07	1.20	5enl	0.87	0.0044
T0111_1	1.41	1.52	1.41	1ebh_B	0.91	0.0110
T0111_2	2.03	6.68	1.05	6enl	0.83	0.0067
T0112	4.73	5.80	4.14	1bxz_D	1.52	0.0136
T0112_1	3.18	4.11	2.84	1bxz_C	1.39	0.0145
T0112_2	4.43	5.87	3.86	1agn_D	1.22	0.0346
T0113	9.03	9.32	8.97	1ahi_B	1.13	0.0354
T0117	19.62	18.46	20.31	1vtk	1.46	0.0996
T0121	30.55	35.42	10.70	1b0u_A	1.63	0.0821
T0121_1	9.10	10.34	8.52	1b0u_A	1.31	0.0536
T0121_1a	8.13	7.84	8.46	1b0u_A	1.56	0.0339
T0121_2	19.70	23.83	17.09	1b9n_A	1.16	0.1493
T0122	3.04	6.28	2.15	1c29_A	1.23	0.0126
T0123	4.86	7.49	1.39	1beb_B	1.24	0.0304
T0125	6.40	8.37	3.93	2lyn_B	1.54	0.0467
T0128	5.15	15.22	4.03	1b06_D	0.84	0.0244
T0128_1	4.31	5.60	4.28	1b06_F	0.56	0.0490
T0128_2	5.72	16.82	3.91	1b06_E	0.74	0.0465

All Targets

Prediction	CRMSCA ALL_ATOMS	ATOMCA_PP ALL_ATOMS	CRN	GDT_TS	SOV_O
T0086TS255_1	19.07	100.0	0.1163	11.59	43.00
T0087TS255_1	19.82	100.0	0.0641	9.87	47.10
T0088TS255_1	-	-	-	-	-
T0089TS255_1	19.56	100.0	0.0517	9.72	39.90
T0090TS255_1	12.24	100.0	0.0615	20.73	30.50
T0091TS255_1	15.66	100.0	0.1740	26.94	26.50

T0092TS255_1	14.98	100.0	0.0660	21.81	57.20
T0093TS255_1	-	-	-	-	-
T0094TS255_1	18.67	100.0	0.1055	13.56	51.50
T0095TS255_1	17.47	100.0	0.0725	18.26	78.60
T0096TS255_1	17.16	100.0	0.0773	17.34	61.30
T0097TS255_1	18.12	100.0	0.1726	23.09	37.40
T0098TS255_1	12.69	100.0	0.1067	25.63	58.80
T0099TS255_1	6.33	100.0	0.1130	39.73	38.70
T0100TS255_1	19.29	100.0	0.0564	10.23	32.40
T0101TS255_1	18.22	100.0	0.0455	19.94	44.80
T0102TS255_1	11.46	100.0	0.1638	30.00	22.70
T0103TS255_1	15.23	100.0	0.0414	30.23	52.70
T0104TS255_1	15.36	100.0	0.0978	17.35	44.20
T0105TS255_1	12.99	100.0	0.1382	20.48	41.70
T0106TS255_1	18.65	100.0	0.1492	16.00	42.50
T0107TS255_1	22.71	100.0	0.1208	8.38	52.00
T0108TS255_1	22.09	100.0	0.1234	11.73	37.90
T0109TS255_1	14.08	100.0	0.0774	16.76	53.40
T0110TS255_1	13.16	100.0	0.1386	28.95	64.30
T0111TS255_1	1.87	100.0	0.0044	88.31	90.30
T0112TS255_1	4.22	100.0	0.0121	57.90	74.20
T0113TS255_1	6.57	100.0	0.0258	35.69	74.50
T0114TS255_1	14.98	100.0	0.1721	16.95	44.20
T0115TS255_1	19.19	100.0	0.0648	8.28	31.80
T0116TS255_1	45.41	100.0	0.0598	5.33	43.60
T0117TS255_1	17.71	100.0	0.0899	15.35	54.60
T0118TS255_1	19.71	100.0	0.1528	14.92	36.00
T0119TS255_1	-	-	-	-	-
T0120TS255_1	28.65	100.0	0.1411	10.71	22.30
T0121TS255_1	18.56	100.0	0.0499	21.78	63.20
T0122TS255_1	2.93	100.0	0.0122	74.69	92.50
T0123TS255_1	4.50	100.0	0.0281	66.72	91.60
T0124TS255_1	37.46	100.0	0.1548	11.26	21.70
T0125TS255_1	5.23	100.0	0.0382	49.45	97.30
T0126TS255_1	18.79	100.0	0.1160	11.11	30.10
T0127TS255_1	24.45	100.0	0.0736	6.93	34.30
T0128TS255_1	4.89	100.0	0.0232	79.15	80.60

A.3 Graphical Summaries

The GDT (*global distance threshold*) graphs for each CASP-4 target is shown below. Our group's prediction is blue, whereas all other models are orange. The graphs measure the percentage of the structure (*x-axis*) that has less than k Å RMSD (*y-axis*). The best prediction is the one that has the lowest slope compared to all other predictions.

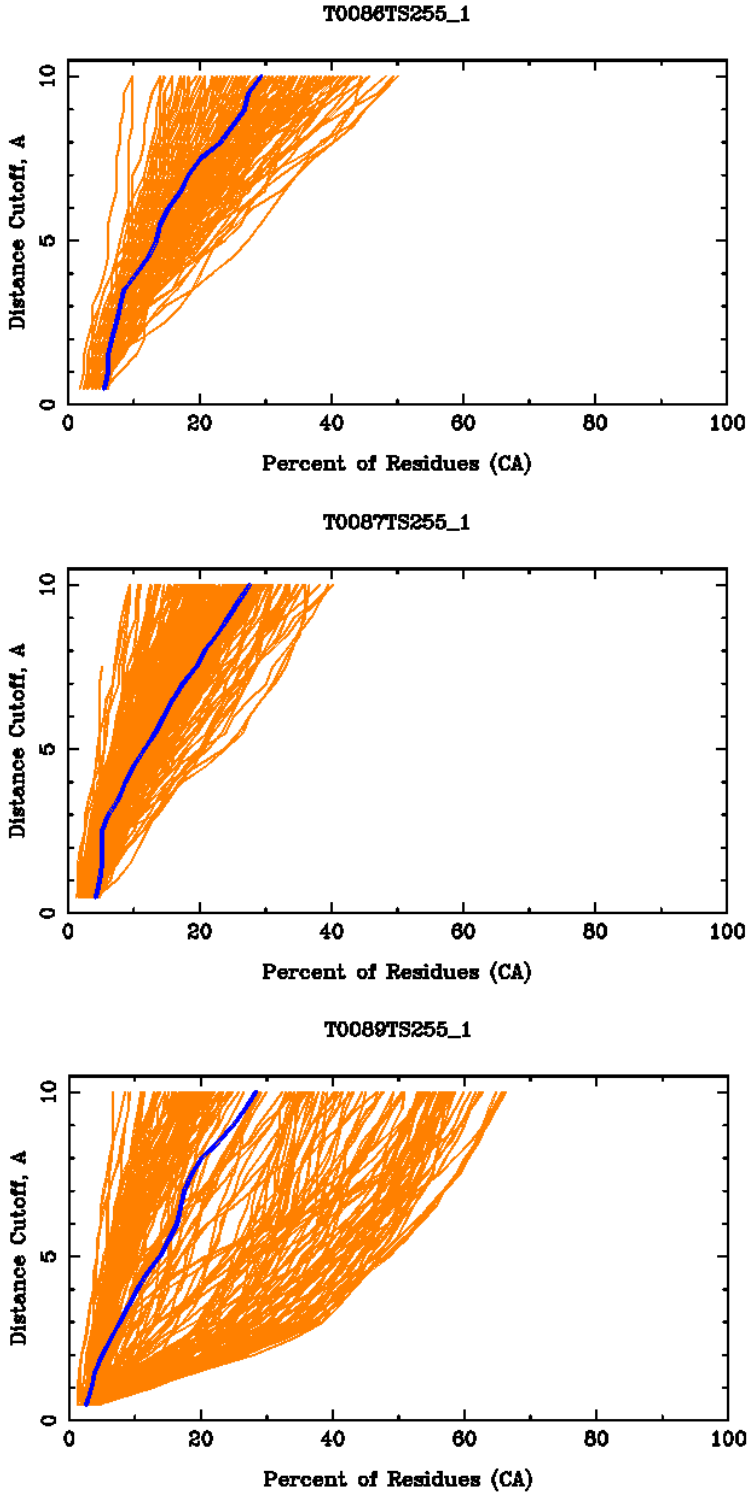

FIGURE A.1. GDT graph for T0086, T0087, T0089

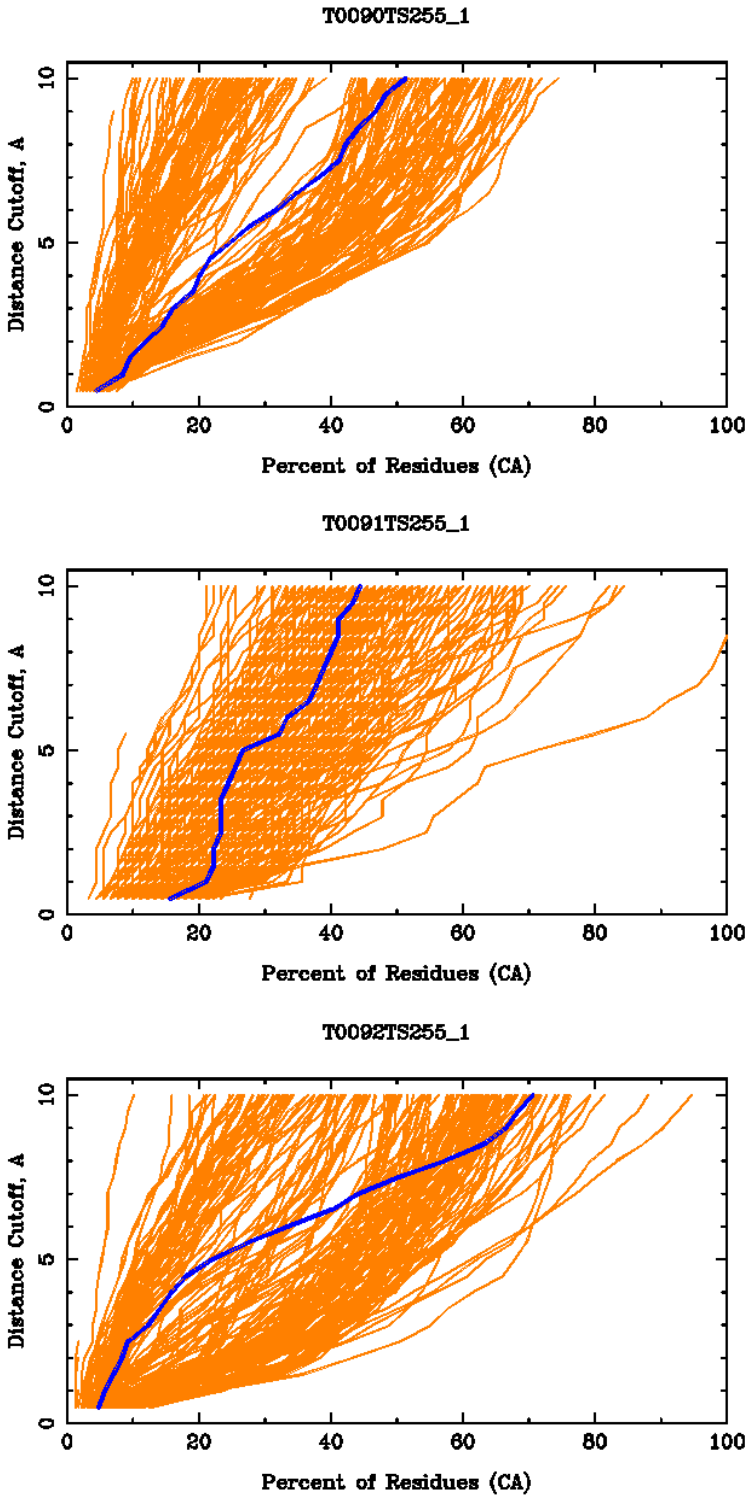

FIGURE A.2. GDT graph for T0090-T0092

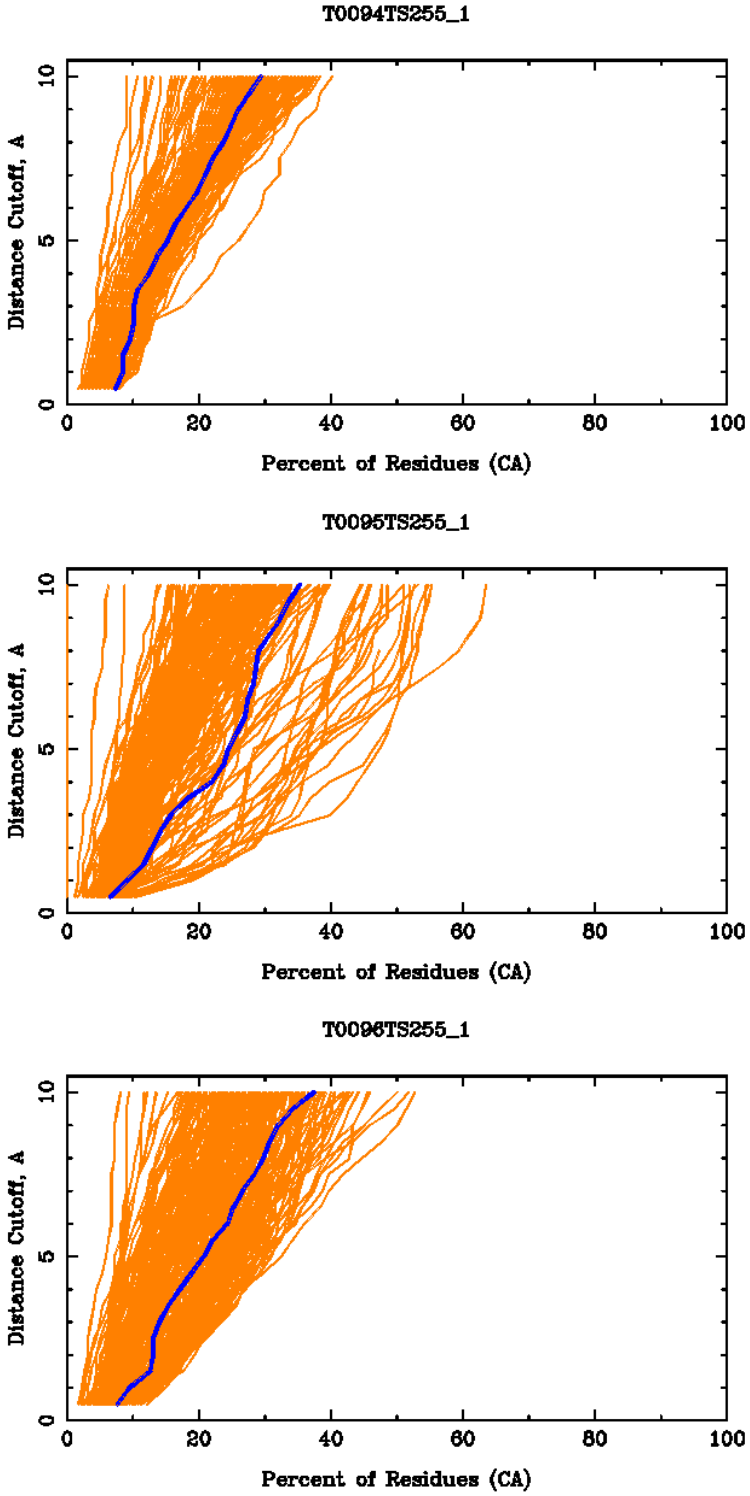

FIGURE A.3. GDT graph for T0094-96

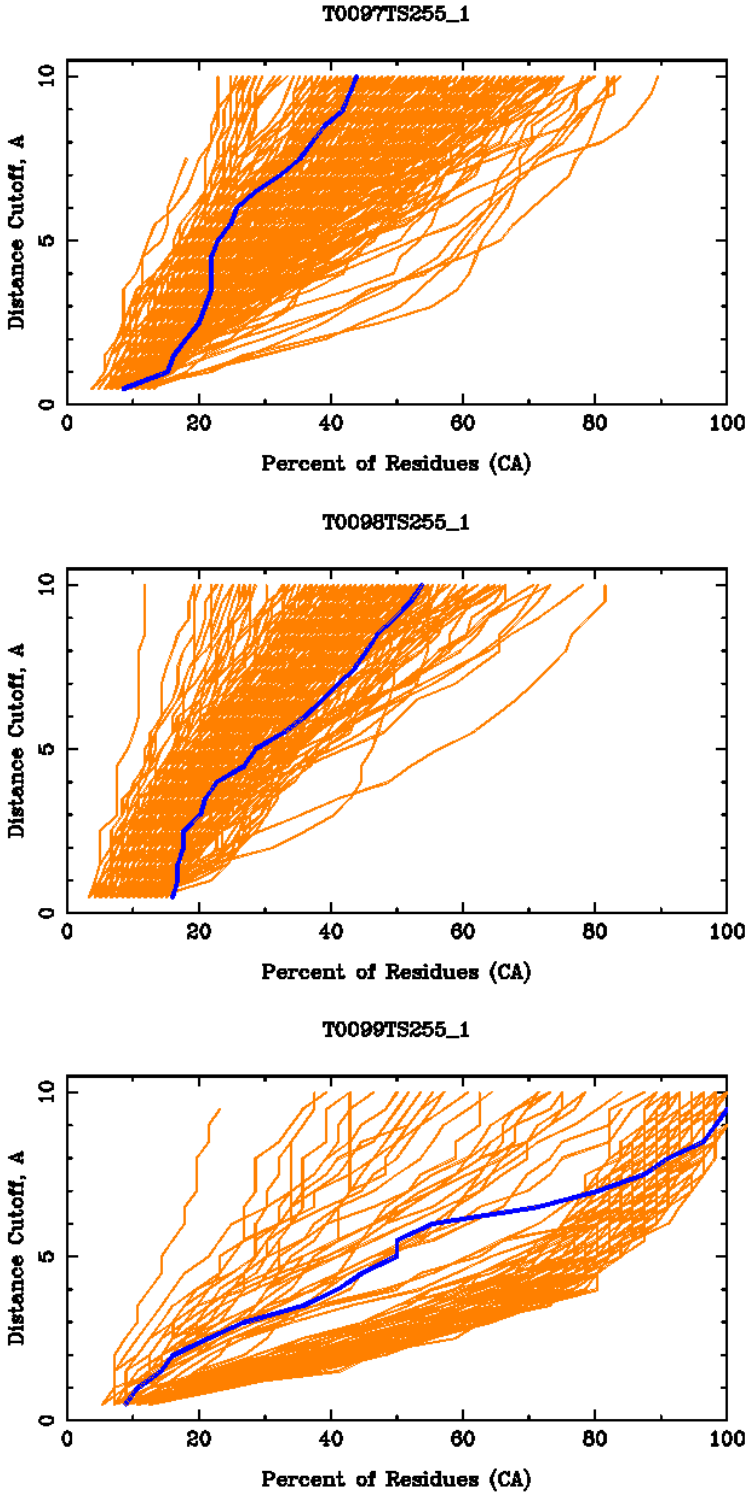

FIGURE A.4. GDT graph for T0097-T0099

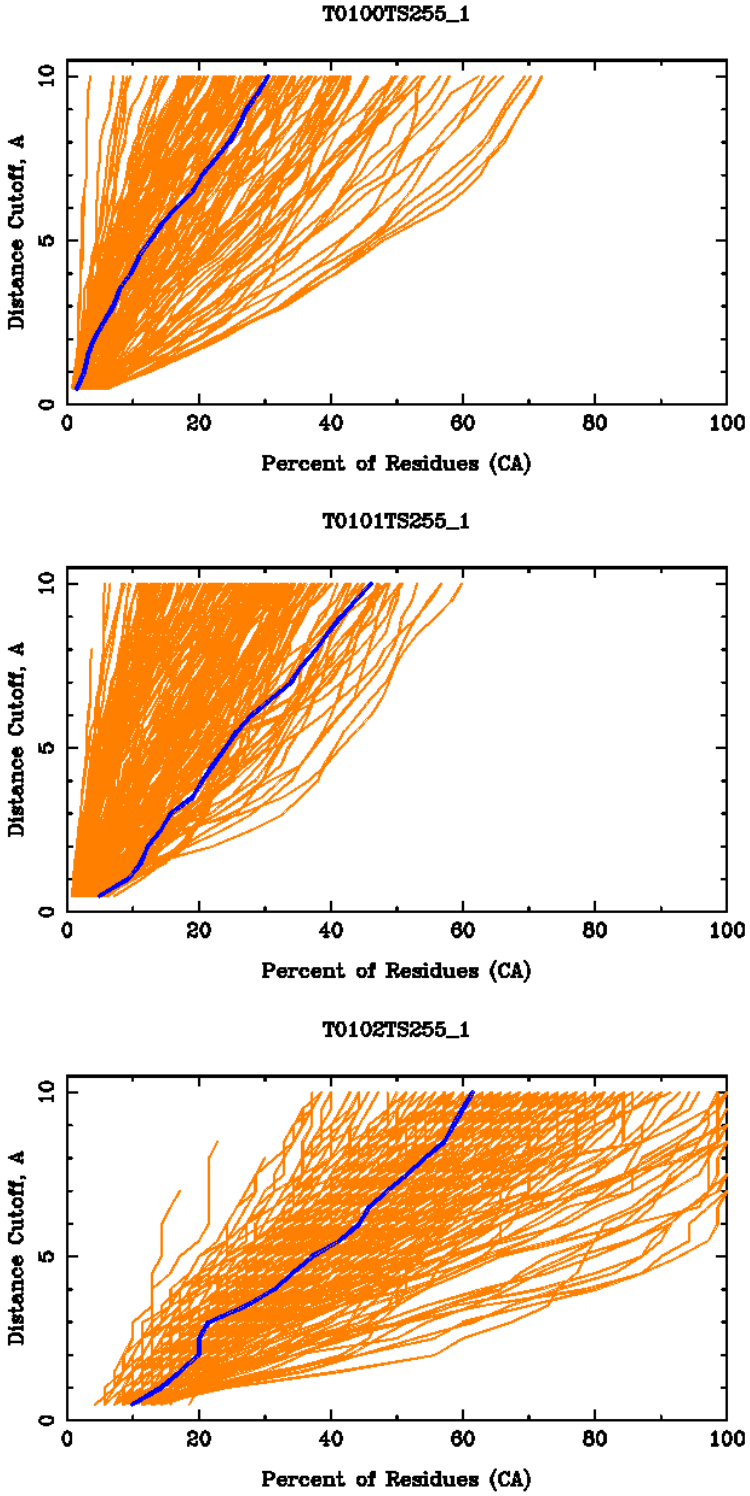

FIGURE A.5. GDT graph for T0100-T0102

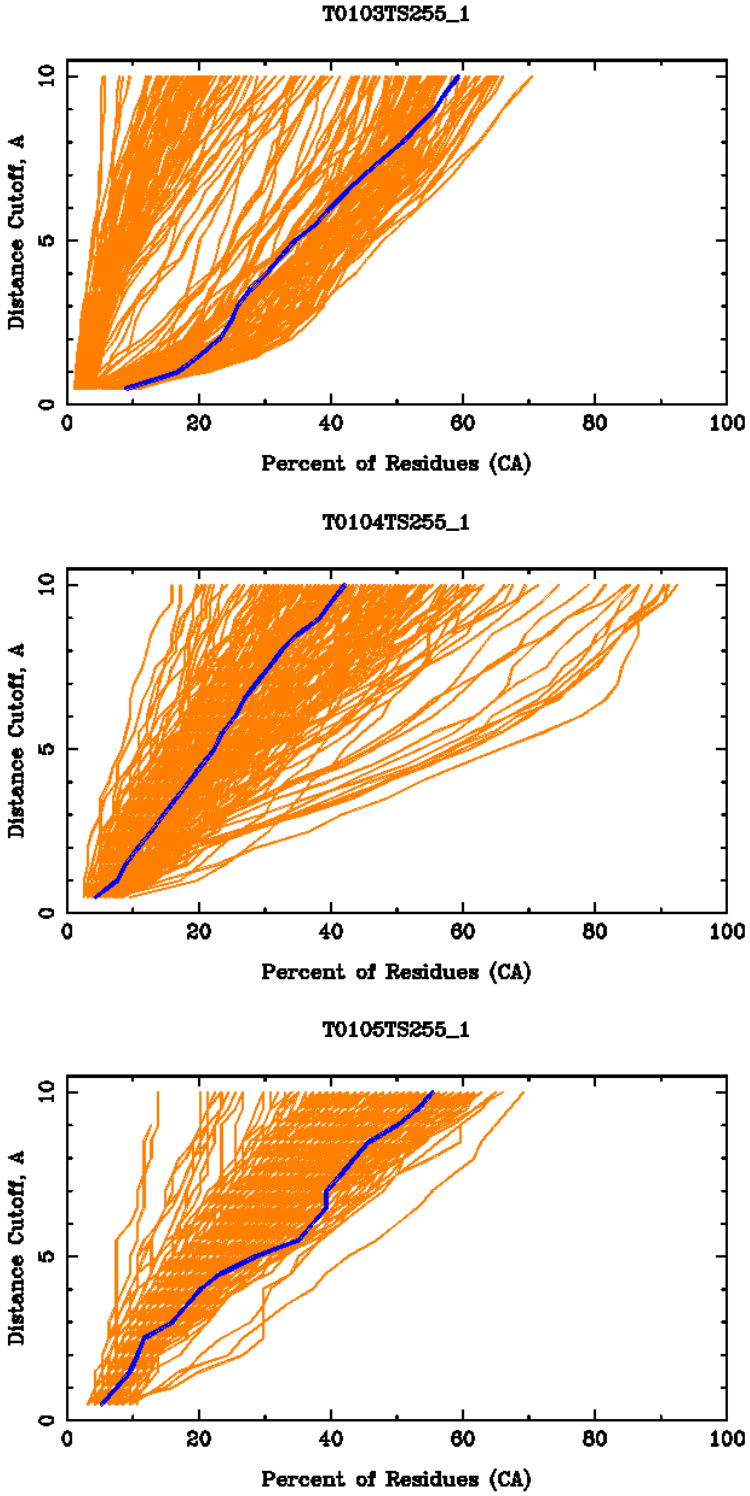

FIGURE A.6. GDT graph for T0103-T0105

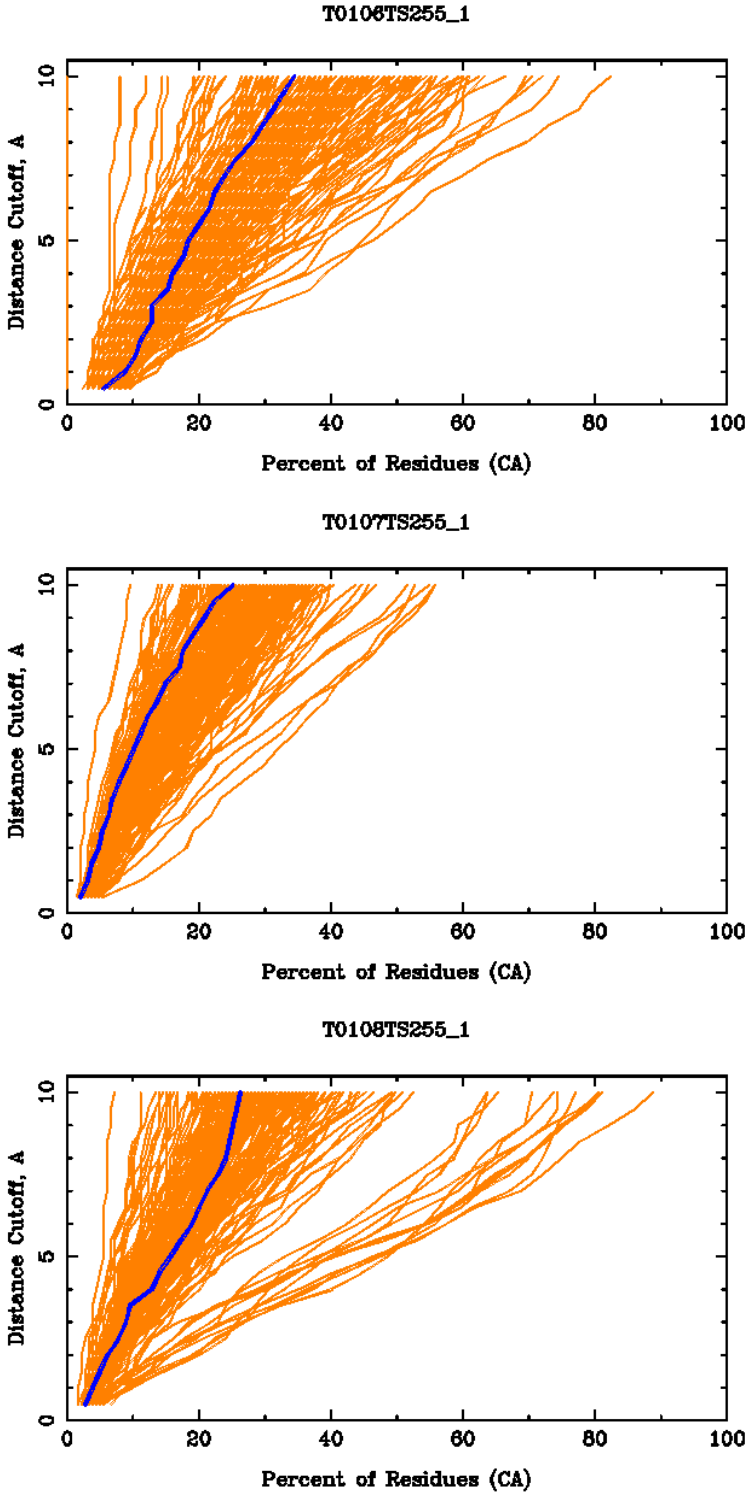

FIGURE A.7. GDT graph for T0106-T0108

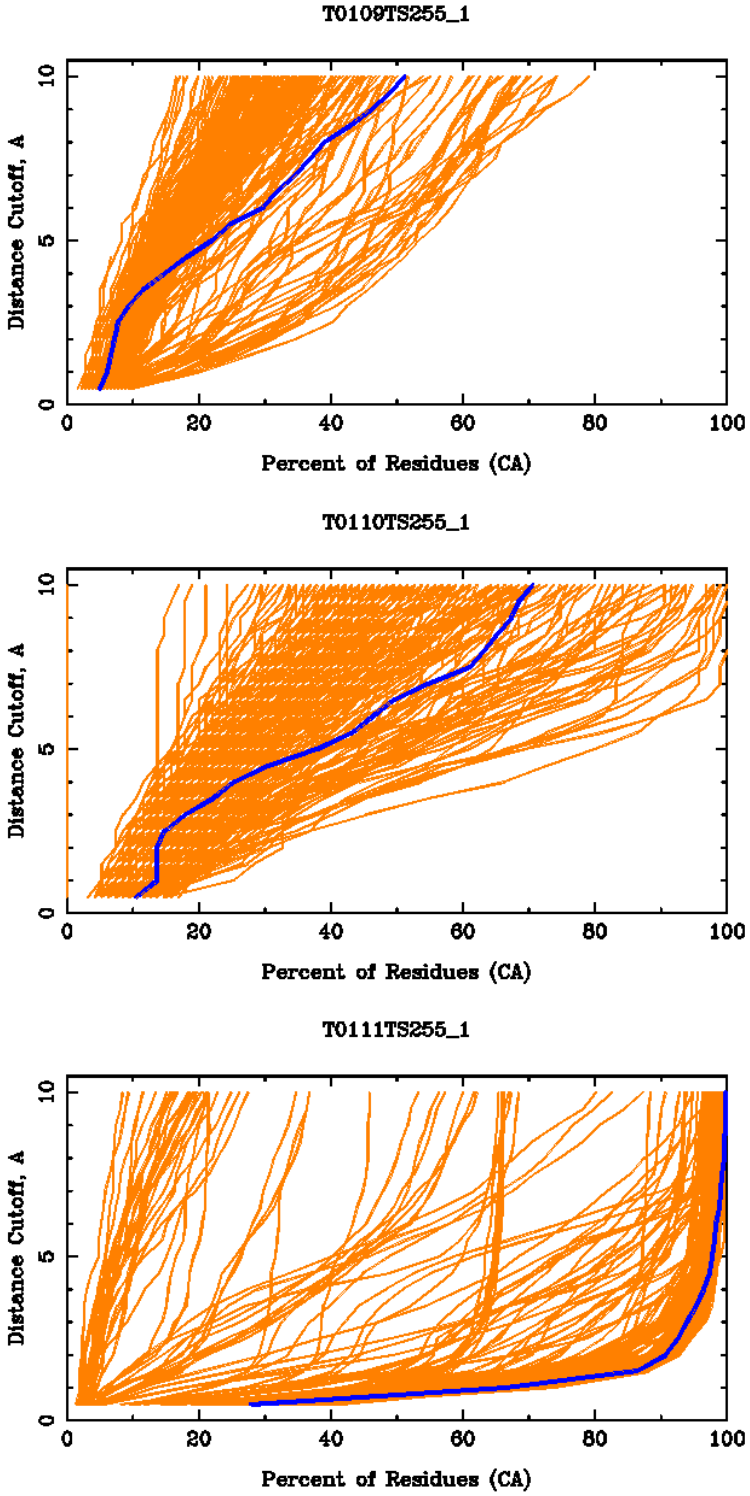

FIGURE A.8. GDT graph for T0109-T0111

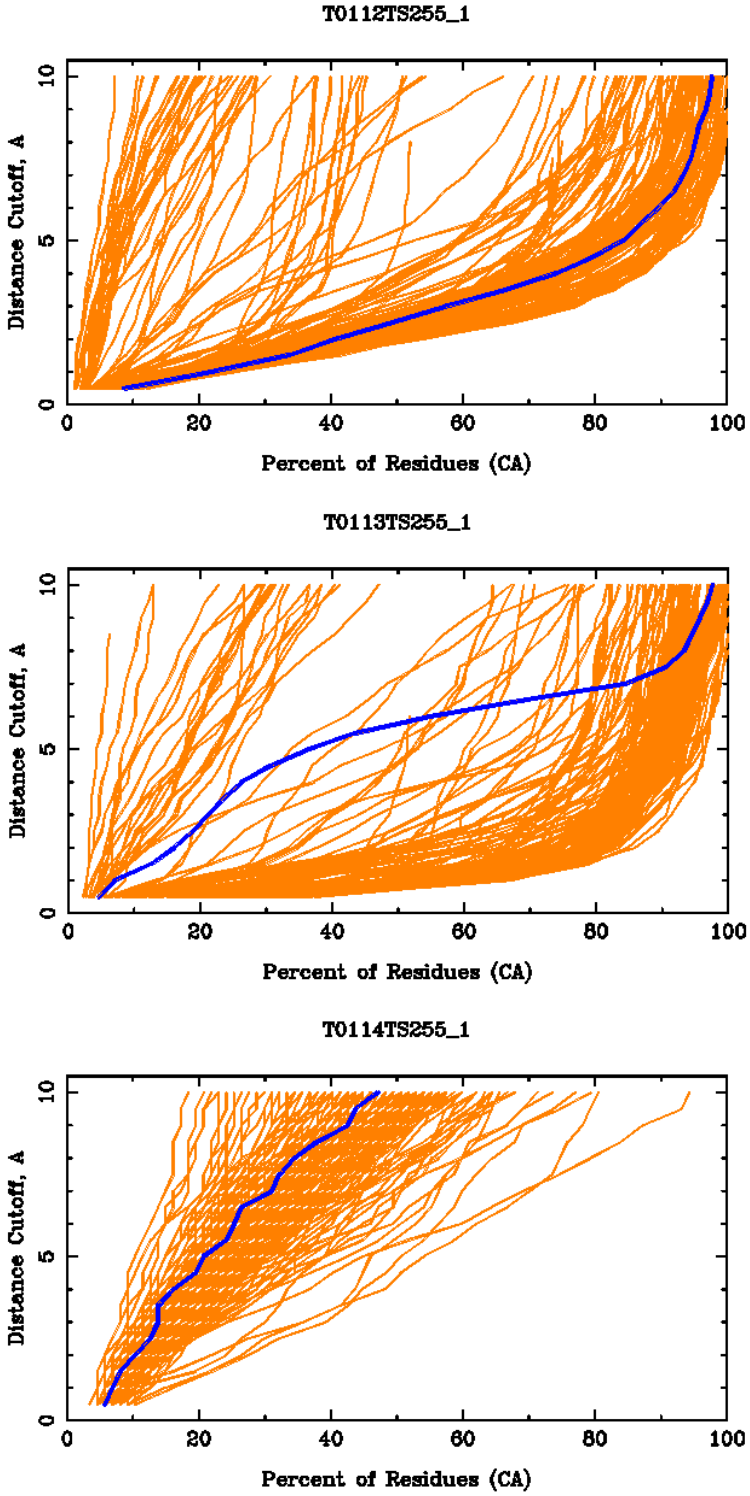

FIGURE A.9. GDT graph for T0112-T0114

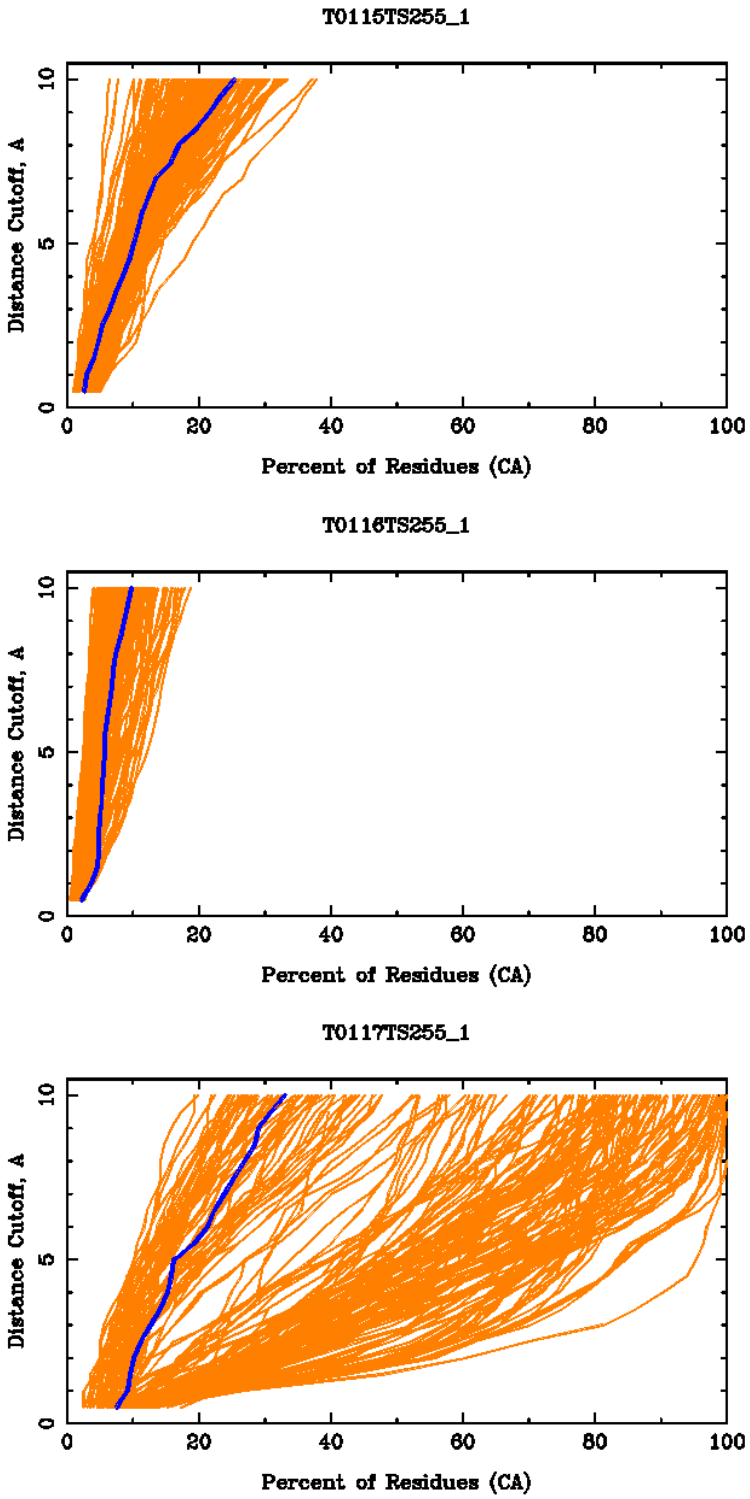

FIGURE A.10. GDT graph for T0115-T0117

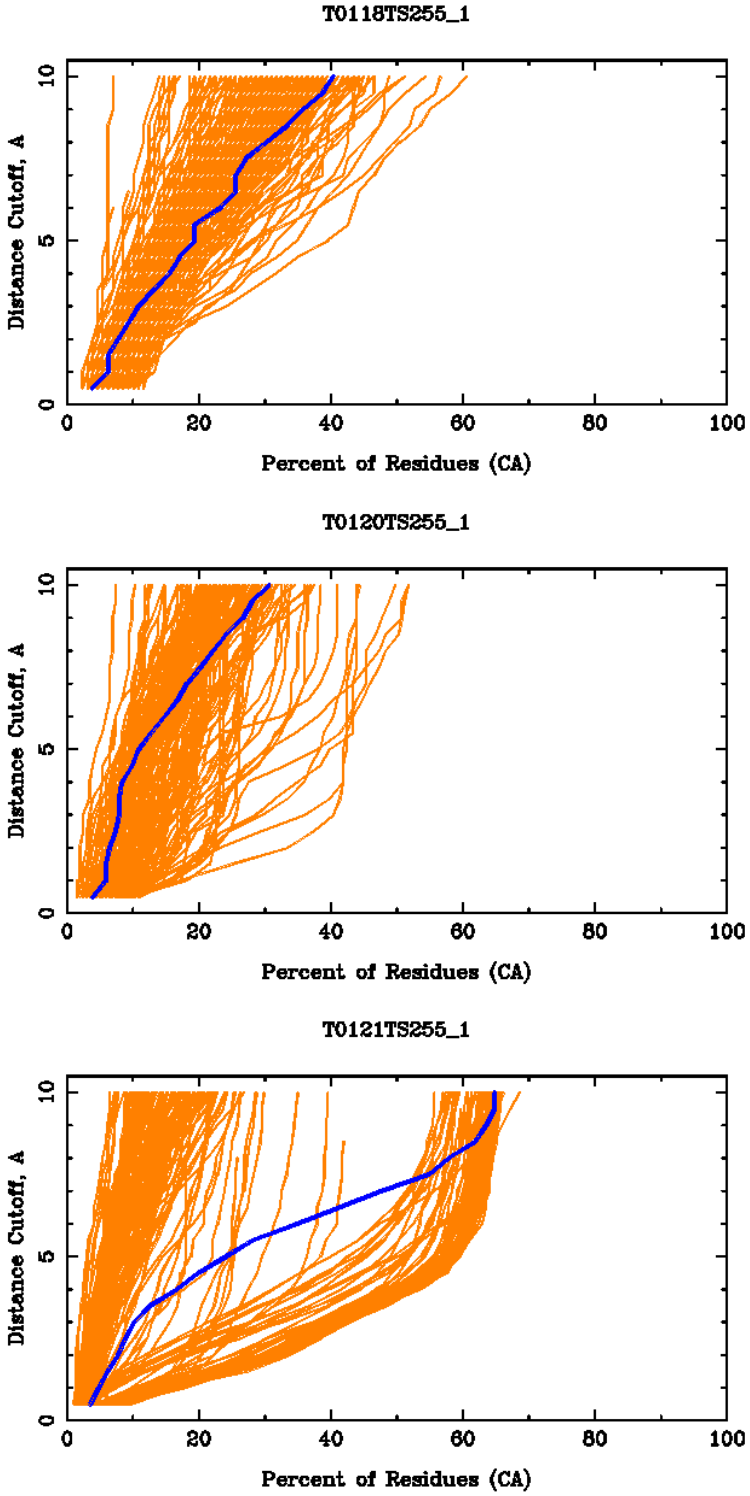

FIGURE A.11. GDT graph for T0118, T0120, T0121

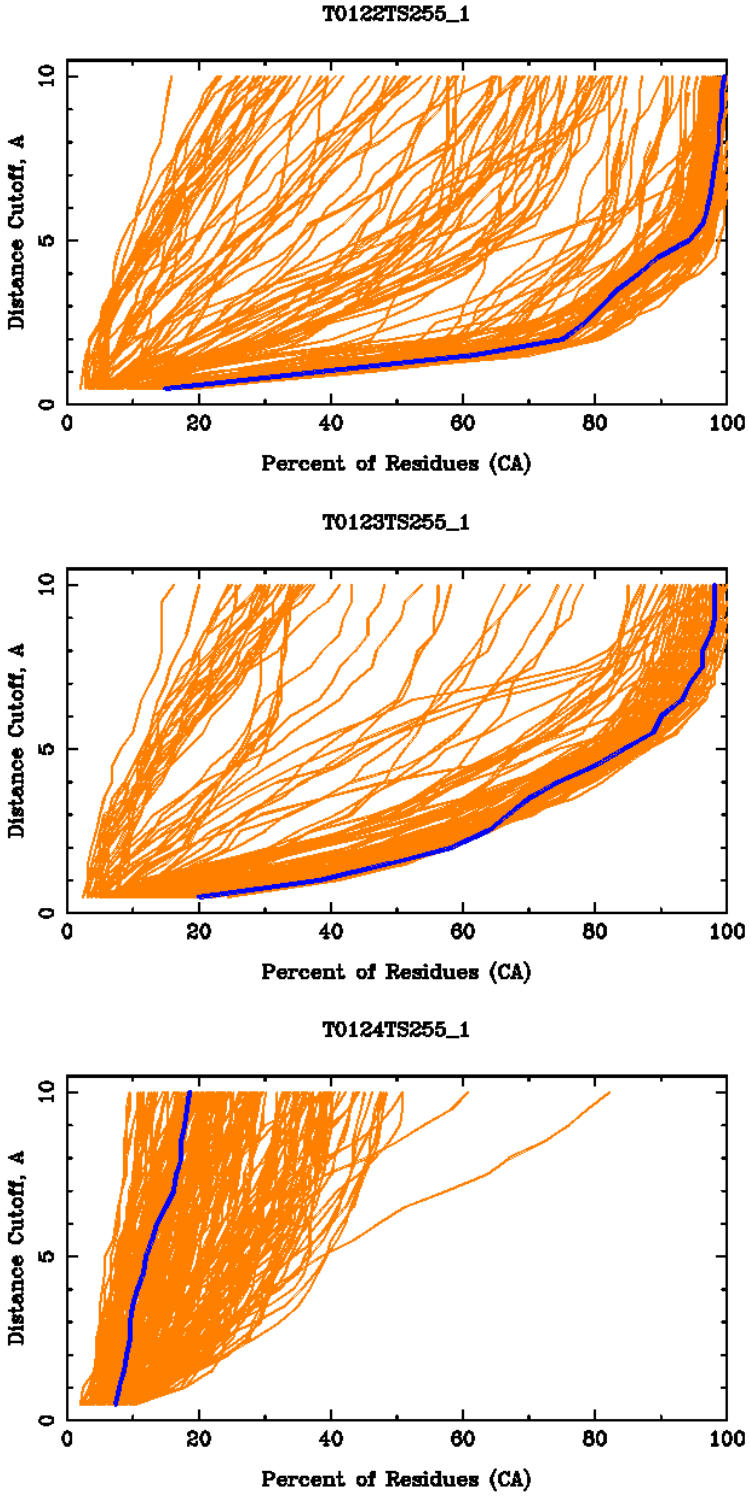

FIGURE A.12. GDT graph for T0122-T0124

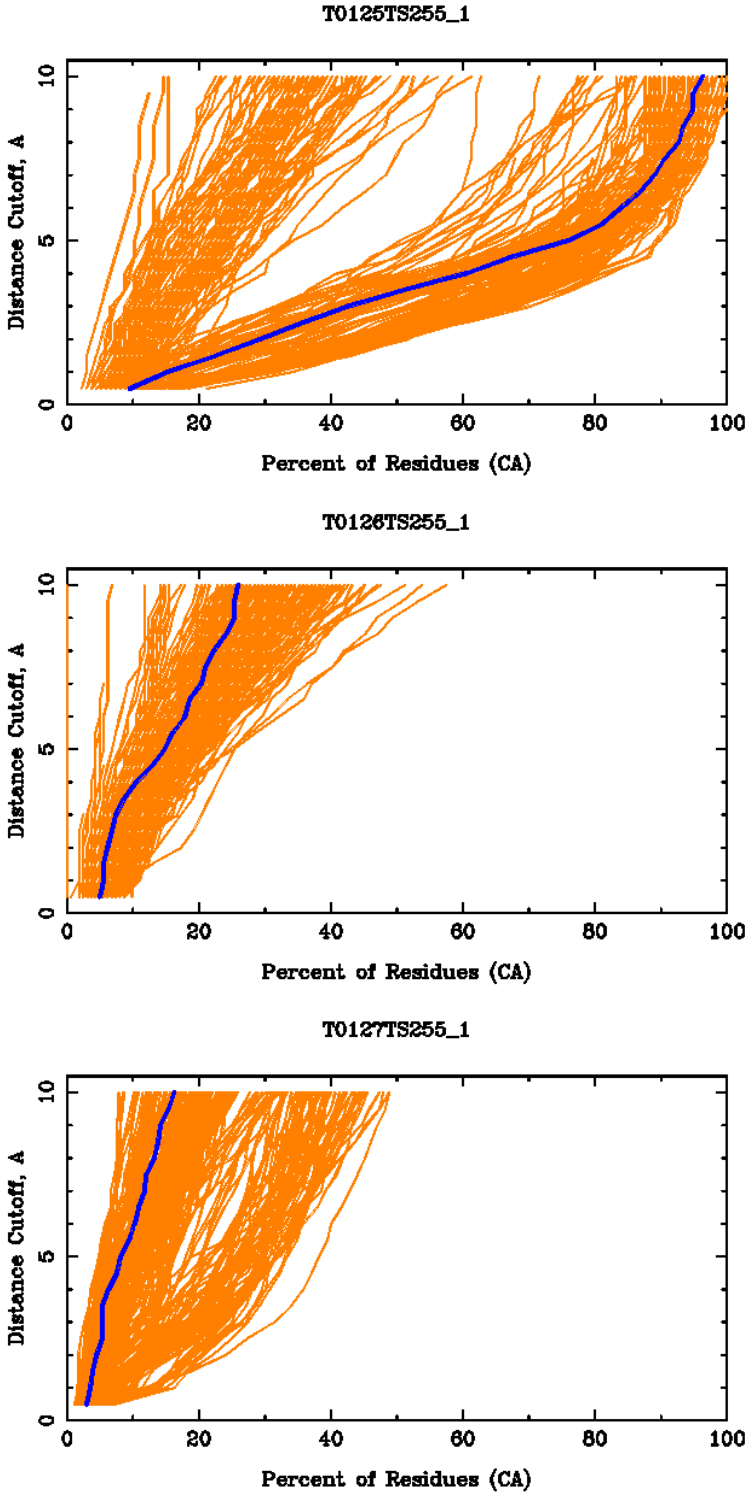

FIGURE A.13. GDT graph for T0125-T0127

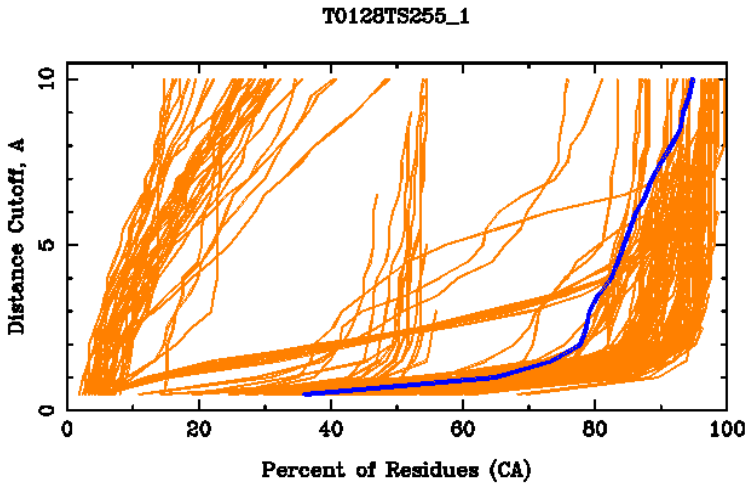

FIGURE A.14. GDT graph for T0128

Appendix B
Lists of Employed Proteins

The following list contains the PDB codes of the proteins used to train and test the loop modeling algorithm (see Section 15.1). Chain identifiers are added to the names, e.g. "`1ccw_A`" is chain "`A`" of protein "`1ccw`". The structures can be retrieved from the Protein Data Bank [127].

B.1 Loop Modeling Training Set

```
3pte    2ayh    2myr    1ccw_A    1nox    1aws_A    2a0b
  1moq    1hfc    1sgp_I    2ctc    1ben_B    1rcf    1nxb
  1ppt    1pin_A    1bsm_A    2lis_A    1utg    1plc    1bk0
  1bxa_A    1qdd_A    1flm_A    1dg6_A    1c52    1qks_A
  1ctq_A    1qau_A    1oaa    1msi    2tps_A    3chb_D
  1jet_A    2pth    2sn3    1amm    1bx7    1qq4_A    1qu9_A
  1mun    2nlr_A    1ifc    1mro_B    1mro_A    1mro_C
  1vfy_A    1rge_A    1tax_A    1a7s    1qlw_B    1bkr_A
  1bs9    1qj4_A    1c5e_A    2igd    3sil    1psr_A    1cc8_A
  1cxq_A    1qtw_A    5pti    1lkk_A    2erl    1a6m    1cex
  1ixh    1c75_A    1byi    1aho    7a3h_A    1bxo_A    2fdn
  1b0y_A    3lzt    1rb9    2pvb_A    3pyp    1gci    1ohk
  1ksa_A    1bxw_A    1vhi_B    2csn    1juk    2ay9_B
  1a0c_A    1du5_A    1bw0_A    1cp9_A    1pya_A    1ith_A
  1rvv_1    1b35_D    1b35_C    1jmc_A    1pvc_4    1cnt_2
  1eai_C    3caa_B    2bos_A    3tdt    1egp_A    1b12_A
  2psp_A    1lou_A    1ptq    1bea    1lts_A    1lts_C
```

1cqy_A 1kwa_B 1ryp_H 1ryp_1 1ryp_L 1ryp_C
1dto_A 1rss 1b66_A 1sei_A 1gr2_A 1buo_A
1stm_A 1c8z_A 1mai 2ebo_A 2hdd_A 1qj8_A
1agq_A 1otf_A 1b0n_A 1fip_A 1b0n_B 1ayo_A
1fle_I 1dxy 1lkf_A 1jdw 1nci_A 1lat_B
1yag_G 1c2a_A 1rec 1dfn_A 1alv_A 1who 1cpo
1b93_B 1reg_Y 3pro_D 1dfu_P 1ak0 1cy9_A
1mwp_A 2spc_A 1ccz_A 1vca_A 2gar 1lcl
1mml 1msk 1mug_A 1cmb_A 1vsr_A 1npk 1ayl
1nbc_A 1slu_A 1tif 1nul_B 1db1_A 1a28_A
1bdo 1c3d 1pym_B 1bg6 1vfr_A 1afw_B 1iib_A
1bbh_A 1bj7 2sak 1unk_A 1gdo_A 1wap_A
2sic_I 1guq_A 1dqs_A 1mgt_A 1kve_A 2acy
19hc_A 1dxg_A 1pgs 1kp6_A 1atl_A 1nar 1rmd
1a8r_A 1b6r_A

B.2 Loop Modeling Test Set

1wap_A 1yac_A 1dqs_A 1dxg_A 1kp6_A 1nar
1cyd_A 2cbp 1bk7_A 1cv8 1chd 6gsv_A 1kpt_A
3cla 1a2z_A 1xwl 1qst_A 1d4a_A 1ajj 1bf6_A
2bop_A 2bc2_A 1aoh_B 1vie 1vhh 1mty_G
1mty_B 1sml_A 1gof 1mol_A 1mof 1qcx_A
1ay7_B 2gdm 1knb 1d3v_A 1ttb_A 1czf_A
2ccy_A 1b4k_A 1dos_A 3chy 5hpg_A 1dhn
3std_A 1cxy_A 1qgw_D 1qgw_B 1vcc 8ruc_I
8ruc_A 1gso_A 2cpg_A 1ads 1qgx_A 2dri
1qgi_A 1a3c 1a1i_A 1edg 1phe 1b5e_A 16pk
1dps_A 1b6a 1rzl 1smd 1aru 1cvl 1nif 1lam
1ppn 1hfe_S 1qh8_A 1qh8_B 1dtj_B 1pht 1cun_A
1eay_D 1dce_A 1d3b_B 1d3b_G 2tod_B 1hav_A
1d2z_A 1a1x 1dfa_A 1got_G 1tig 1d6j_B 1c3q_A
1a8p 1btn 1bft_A 1cfb 1d9c_A 1amx 1bbp_A
1hoe 1uox 1cew_I 1taf_A 1d3y_A 1sra 1wdc_A
1vid 1byr_A 1poc 1sur 1dvr_A 1qu0_C 1mkp
1ayy_B 8ohm 1cr5_C 1c25 1g31_A 1bl0_A 1a2x_B
1fzc_A 1qdn_A 1am9_D 1iar_B 1b33_N 1nst_A
2occ_L 2occ_H 2occ_I 2occ_G 2occ_F 2occ_J
2occ_M 1tmc_A 2prg_C 1aux_A 1rl2_B 1tab_I
1qo3_D 1jsu_C 1qkj_A 1ycq_A 1auq 1c9e_A 1gal

1dco_A 1byl_A 1kit 1bjk 1fxs_A 1am7_A 1tpl_A
1oun_B 1a79_A 1mnm_C 1ign_A 1rcb 1b7a_A
1tii_D 1tii_C 1qo0_D 1jen_D 1jen_C 1p32_A
2cgp_A 1a15_B 1d9y_A 1cn3_A 1cfz_A 1noy_A
1tyf_A 1dev_D 1qla_B 2siv_B 1pfo 1fro_A 1grj
1ucw_A 1tip_A 1tup_B 1dyn_A 1aw8_B 1xxa_C
1bdy_A 1p35_C 1ecm_B 1kte 1qj2_A 1qj2_C
1dio_G 1dio_B 1am2 1del_B 1bb9 1tul 1cno_A
1aqe 1jly_A 1gpm_A 1cxa 1b4u_A