VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PROTEIN STRUCTURE PREDICTION

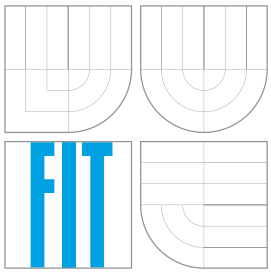
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAROSLAV TUČEK

BRNO 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PŘEDPOVÍDÁNÍ STRUKTURY BÍLKOVIN

PROTEIN STRUCTURE PREDICTION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAROSLAV TUČEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. IVANA RUDOLFOVÁ

BRNO 2009

Abstrakt

Práce popisuje prostorovou strukturu molekul bílkovin a databází uchovávajících representace této struktury, či její hierarchické klasifikace. Je poskytnut přehled současných metod výpočetní predikce struktury bílkovin, přičemž největší pozornost je soustředěna na komparativní modelování. Tato metoda je rovněž v základní podobě implementována a na závěr její implementace analyzována.

Abstract

This work describes the three dimensional structure of protein molecules and biological databases used to store information about this structure or its hierarchical classification. Current methods of computational structure prediction are overviewed with an emphasis on comparative modeling. This particular method is also implemented in a proof-of-concept program and finally, the implementation is analysed.

Klíčová slova

Bílkovina, struktura, klasifikace, PDB, predikce, komparativní modelování, ab initio modelování, threading, fragmentová metoda, zarovnání

Keywords

Protein, structure, classification, PDB, prediction, comparative modeling, ab initio modeling, threading, fragment methods, alignment

Citace

Jaroslav Tuček: Protein Structure Prediction, diplomová práce, Brno, FIT VUT v Brně, 2009

Protein Structure Prediction

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Ivany Rudolfové. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jaroslav Tuček
May 25, 2009

Contents

Introduction

In the chemistry of life, proteins play a principal role. Besides other important functions, they form the structural support of cells (keratin, myosin), function as biochemical catalysts (the enzymes), provide transportation (haemoglobin) and storage (ferritin) of other needed molecules, regulate cellular processes (hormones) and work as an integral part of the immune system (antibodies). Proteins are able to carry out this plethora of functions by being able to discriminate even among minutely similar molecules in their surroundings and to choose only particular types to operate on. In all other cases, atomic collisions would prevent the molecules from approaching the protein close enough for a reaction to start. This discriminatory ability is due to a complicated three dimensional structure which flawlessly matches only the designated target.

Although many functional properties of proteins can be inferred from their local sequence properties, the complete picture cannot be drawn without determining the spatial structure of the molecule. The reason for this is that proteins are often able to carry out their roles by attaching other molecules to themselves. This attachment happens at a region called the binding site. And binding sites can be formed by several very distant protein regions. Moreover, even very minor alterations to the protein's spatial structure can greatly impede or completely eliminate binding.

It is very hard to determine protein function experimentally as the molecule's activity is characteristic to its native environment which may be nearly impossible to prepare artificially in a laboratory. It is much easier to infer the function from structure. Unfortunately, the structural experimental discovery is a complicated, lengthy and expensive process. To illustrate the difficulties involved, let us briefly describe the experimental prediction methods here. There are three widely used ones – X-ray crystallography, nuclear magnetic resonance spectroscopy and cryo-electron microscopy.

X-ray crystallography, which currently amounts to about 90% structures stored in PDB, first produces a crystal from the examined molecule and then infers its structure from X-ray diffraction patterns produced on electrons of the crystal. This is quite an involved process because the experiment provides information only about intensities of the patterns, but phases of the rays that produced them need to be known in order to construct a model of the structure. Nevertheless, after considerable laboratory work the phases can be produced and subsequently, a model built by least-squares fitting the observed data. X-ray crystallography can produce very accurate measurements, with resolutions up to 0.5 Å under favorable circumstances, but producing the required crystals is a difficult process, especially for large or trans-membrane proteins. It is also not clear if crystallised proteins reflect their native state faithfully.

The second method, NMR spectroscopy, measures the magnetic field of atomic nuclei. Magnetic interactions decay rapidly with distance and so, the method can determine a set of spatially neighbouring residues by measuring magnetic spin alterations. The protein struc-

ture itself can be determined computationally from this set by constraint satisfaction, where constraints are lower and upper bounds on inter-atomic distances. If enough constraints are gathered during the measurement phase, only a finite number of models will satisfy them[1]. Unfortunately, there is no way of deciding, which is the best one of these. The method can examine proteins in their native state, however, large molecules pose especial problems and ambiguous results can be produced.

The third method, cryo-electron microscopy, uses conventional electron microscopy on molecules at liquid nitrogen temperatures. It can be used to examine very large molecules (1500 Å), however, its accuracy is fairly low (5 Å).

Another illustration of the problem can be found, for example, in comparing the current size and rate of growth of Genbank and PDB. Genbank is an NCBI database which collects all publicly available DNA sequences (and their protein sequence translations). Its current release, 169.0 (December, 2008), holds approximately 100 million sequences totaling 200 billion basepairs (407 GB data). PDB is an RCSB database of experimentally determined macromolecular (mostly protein) structures. As of October 28, 2008, PDB holds the atomic coordinates of 49,770 proteins only. Coming up with a computational method of protein structure prediction would help reduce the gap between these databases and greatly speed up the process of our understanding the workings of living cells. Although the step of inferring function from structure is by no means trivial, having available structures for most of the known proteins would nevertheless revolutionise the fields of biotechnology, pharmacology and medicine. Today, the computational protein folding problem remains unsolved but much progress has been made. The account of the current state follows in subsequent chapters.

The first chapter will begin with a description of the elements of protein structure. While substantial knowledge of structural biochemistry is certainly not a prerequisite for understanding the prediction methods, one has to, at least, grasp the scientific concepts the representations of which the computational methods operate upon. Additionally, the biological sciences have amassed a great body of knowledge which can be profitably directly used in the methods – for example, as rules limiting a state of solutions to be searched. We have included facts which we felt are most useful in this respect.

The second chapter contains a brief description of biological databases storing representations and hierarchical classifications of protein structure. These databases offer an amazing amount of crossreferenced information, which is, however, mostly interesting to the biologist, not the computer scientist. As, especially for the purposes of this work, most of the databases are accessible in a very simple, and trivial to parse, plain text format (e.g. the popular FASTA). We, nevertheless, include an overview of the databases and a somewhat more elaborate account of the PDB database in particular.

Chapter three broadly outlines the existing structure prediction methods. Both inductive and deductive approaches are covered, although only comparative modeling in detail. This is not to say that the other methods are unimportant, but comparative modeling is the only one able to give very accurate results in favorable cases, Moreover, comparative modeling builds a structure model for a molecule from a known structure believed to be similar – with the rising number of experimentally determined structures available, the method will become applicable to a larger portion of the data. Eventually, it might become the only method biologists will have to use.

Chapter four discusses our implementation of a comparative modeling procedure, while

[1] Usually, about 20

chapter five assesses its obtained results. The first two chapters were previously completed as a part of the term project.

Chapter 1

Protein Structure

1.1 Overview

In this chapter, we will describe the basic properties of protein molecules with an emphasis on their spatial conformation. A concise overview can be found in [CB00] or [Hun93], a much more detailed account is given in [Les01]. We will base the chapter on these.

Proteins are macromolecular unbranched sequences formed by twenty types of amino acids[1]. Each amino acid is of the same fundamental structure made by a central carbon atom, called the α-carbon, to which a single hydrogen atom, a carboxyl group COOH– and an amino group NH_2– attach (see figure 1.1). The fourth bond of the α-carbon binds a sidechain (R) group unique to each amino acid. This group determines the amino acid's chemical properties.

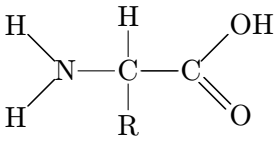


Figure 1.1: General amino acid structure

The properties of amino acids place constraints on the property of the protein molecule formed from them. For example, amino acids with large sidechains cannot be packed too densely on the backbone. Polar amino acids have an uneven distribution of electric charge over the molecule causing them to be attracted to or repelled from other charged residues. Amino acids with side chains containing atoms with large nuclei are less flexible in folding and make the local area of protein backbone rigid and so on.

For illustration, glycine, the simplest amino acid with only a single hydrogen atom as a sidechain, is non-polar and cannot ionise[2]. Another small amino acid, alanine, has a

[1] More exactly, proteins are encoded in DNA as strings of twenty standard amino acids. Others may be introduced after protein synthesis, or during translation – for example selenocysteine, see later paragraphs.

[2] That is, cannot become fully charged, as opposed to partially charged polar molecules; charged residues will mostly be found on the surface of the protein, they are also likely to form bonds, salt bridges, between each other, stabilising the fold.

methyl group attached as a side chain. Both these amino acids' sidechains are aliphatic[3]. Other amino acids with aliphatic sidechains, longer but still relatively small, are valine, leucine and isoleucine, and all three of them are hydrophobic. Hydrophobicity (as opposed to hydrophilicity) causes a molecule to be repelled from water molecules. As proteins in their native state usually exist in an aqueous environment, hydrophobicity is the chief driving force in the folding process, causing hydrophobic residues to cluster in the center of the folded protein and pulling the hydrophilic residues to the surface area. In contrast, sidechains of phenylalanine, tyrosine and tryptophan are all large (making them hard to pack onto the backbone) and aromatic. Phenylalanine and tryptophan are hydrophobic, whereas tyrosine contains a hydroxyl OH– group making it more reactive and hydrophilic. Other residues with a hydroxyl group are serine and threonine, although these do not have aromatic rings. Cysteine and methionine are two very important residues. Their sidechains contain sulphur atoms making them very reactive and prone to form disulphide bridges between each other strongly holding distant parts of the folded protein together[4]. See table 1.1 for a summary of the properties of standard (DNA encoded) twenty amino acids. The listed information includes standard one and three letter abbreviation codes, linear side chain sequence[5], hydrophobicity and polarity (partial charge)/charge. Some proteins contain other than the listed canonical amino acids but these are introduced only during the post-translational modification process[6].

When the carboxyl group of one amino acid reacts with the amino group of another, the two acids are chained together by a peptide bond (see figure 1.2) and a water molecule is released. The remainders of the amino acids (after having lost a water molecule) are called residues. Proteins are created by an arbitrary number of such reactions although most common lengths are between 50 and 1000 residue units with a current estimate of the approximate average length being 300[7]. The molecular ends are called the N-terminus and the C-terminus corresponding to the unreacted amino and carboxyl groups respectively. The sequence $\mathrm{N}_1, \mathrm{C}\alpha_1, \mathrm{C}'_1, \mathrm{N}_2, \ldots, \mathrm{C}'_\mathrm{n}$ where C' carbons belong to the carboxyl group and index i denotes that the atom X_i belongs to the i-th residue in the chain, is called the protein backbone.

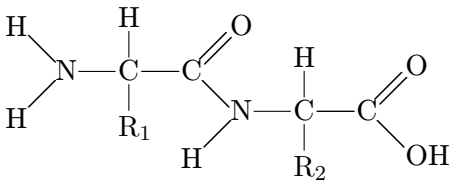


Figure 1.2: A peptide bond formed between two amino acids

[3]Aliphatic compounds consist of carbon and hydrogen atoms only and do not form aromatic rings – carbon rings with alternating single and double bonds.

[4]The formation of disulphide bridges requires oxidation of sulphide groups. This cannot happen in case of intracellular proteins, as the cellular environment is reducing.

[5]Note the multiple bond connectivity for proline's sidechain. Proline is exceptional in forming a second bond with the protein backbone at the nitrogen atom (in place of one hydrogen).

[6]Selenocysteine is one exception to this rule. Certain sequences in mRNA can cause an UGA codon, which is normally a stop codon, to code for selenocysteine instead. Selenocysteine is found, for example, in the powerful anti-oxidation enzyme glutathione peroxidase.

[7]The longest known proteins, the titins, have backbone length of around 25,000 residues.

Table 1.1: Amino acid properties

Amino acid	**Code**	**Side chain**	**Hydrophobic**	**Polar**	**Charged**
Alanine	Ala (A)	$-CH_3$	Yes		
Arginine	Arg (R)	$-(CH_2)_3NH-C(NH)NH_2$			+
Asparagine	Asn (N)	$-CH_2CONH_2$		Yes	
Aspartic acid	Asp (D)	$-CH_2COOH$			-
Cysteine	Cys (C)	$-CH_2SH$		Yes	
Glutamic acid	Glu (E)	$-CH_2CH_2COOH$			-
Glutamine	Gln (Q)	$-CH_2CH_2CONH_2$		Yes	
Glycine	Gly (G)	$-H$	Yes		
Histidine	His (H)	$-CH_2-C_3H_3N_2$			+
Isoleucine	Ile (I)	$-CH(CH_3)CH_2CH_3$	Yes		
Leucine	Leu (L)	$-CH_2CH(CH_3)_2$	Yes		
Lysine	Lys (K)	$-(CH_2)_4NH_2$			+
Methionine	Met (M)	$-CH_2CH_2SCH_3$	Yes		
Phenylalanine	Phe (F)	$-CH_2C_6H_5$	Yes		
Proline	Pro (P)	$-CH_2CH_2CH_2-$	Yes		
Serine	Ser (S)	$-CH_2OH$		Yes	
Threonine	Thr (T)	$-CH(OH)CH_3$		Yes	
Tryptophan	Trp (W)	$-CH_2C_8H_6N$	Yes		
Tyrosine	Tyr (Y)	$-CH_2-C_6H_4OH$		Yes	
Valine	Val (V)	$-CH(CH_3)_2$	Yes		

The peptide bond $-C(=O)NH-$ itself is partially a double bond and therefore planar[8] but there is a level of rotational freedom around the $N-C\alpha$ and $C\alpha-C'$ bonds, usually described by torsion angles ψ and ϕ respectively. These are single bonds and the only constraints on sizes of their torsion angles are placed by possible collisions of bonded atoms with other atoms nearby. The carbon atoms of the residue sidechains are labeled alphabetically starting from β according to their distance from the attached α-carbon. The torsion angles of bonds between these carbons are labeled χ_1, χ_2, etc. The three dimensional protein structure can be uniquely specified by either giving coordinates of every atom in the molecule or the sequence of all torsion bond angles on the backbone and bond lengths which are approximately constant in the entire molecule, see table 1.2. Angles between every three successive atoms on the backbone are constant as well. The dihedral[9] torsion bond angles then are the only freedoms of rotation.

Not all combinations of torsion bond angles are sterically allowed, however. A lot of information about protein structure can be gathered by plotting measured torsion bond angles on a graph with ϕ and ψ as x and y axes, the so called Sasisekharan-Ramakrishnan-Ramachandran diagram (see figure 1.3 for the Ramachandran plot of the crambin protein).

[8] All natural amino acids are of L-form chirality. Therefore, only the trans isomer is possible. Proline is an exception. The energy difference between its cis and trans conformations is very small and proline frequently occurs in both.

[9] A dihedral angle is given by four consecutive atoms. The first three atoms are held fixed, defining a plane, and the fourth is rotated around the bond between the second and third.

Table 1.2: Lengths of important protein bonds

Backbone bond	**Average length**	**Hydrogen bond**	**Average length**
Cα — C$'$	153 pm	OH — OH	280 pm
C$'$ — N	133 pm	NH — OC	290 pm
Cα — N	146 pm	OH — OC	280 pm

Residue sidechain structures can be represented analogously by giving the sizes of bond angles and similar regularities as on the Ramachandran plot can be observed. Possible rotational conformations are called rotamers and databases storing the regularly occurring rotamers are called rotamer libraries. When proteins (point) mutate, the substituted residue in a mutant must fit into a space given to it in the original molecule by a relatively rigid fold and thus tends to form an identical rotamer.

When placed in a specific environment with appropriate conditions (usually in an aqueous solution or a lipid and a narrow pH and temperature range), protein molecules rotate through the torsion bond angles and always adopt one particular spatial conformation, the fold. The fold is uniquely determined by the residue sequence and forms spontaneously and almost immediately[10] (even on the order of milliseconds[11]). Thus, although it is generally believed the conformation adopted is the one minimising the free energy of the protein, it cannot be found this rapidly through a blind search of possible conformations. This so-called Levinthal paradox led some scientists to hypothesise that the folding process is not driven by energetic but kinetic concerns, following a directed folding pathway. The nature of the folding pathway, however, remains unknown and we can expect little help from experimental methods in its understanding as isolating the intermediate folding stages is extremely difficult. Computational methods do not offer much help either. Interresidue interactions even between residues very far apart on the backbone can have a considerable effect on the folding process. The final fold is thus dependent on the global state of the molecule, presenting a significant obstacle to protein folding simulations. This is really unsettling, especially in light of the success of inductive structure prediction methods. Even if these could correctly predict structures for all undetermined proteins, we still would not understand the folding mechanism at all.

1.2 Levels of Protein Structure

The protein's residue sequence is called its primary structure. In the molecule's spatial conformation, certain regular motifs, the secondary structure elements, can be identified. Two most important amongst them are α-helices and β-sheets (see figure 1.4 for a display of a protein with two α-helices and one anti-parallel β-sheet clearly visible). An α-helix is a cylindrical structure with its i-th residue aligned with the (i + 4)-th residue and their amide nitrogens and carbonyl carbons bonded through hydrogen bonds. A β-sheet is a linear conformation of backbone strands running either in parallel or antiparallel to each other (the direction given from the N-terminus to the C-terminus) and supported by

[10] Although some proteins require the presence of other proteins, the chaperons, before they can fold properly. This is mostly necessary to avoid interference from neighbouring molecules in a crowded cellular environment or to prevent misfoldings due to fluctuations of the surrounding conditions, e.g. temperature.

[11] Certain proteins may take much longer, even hours, to fold, though.

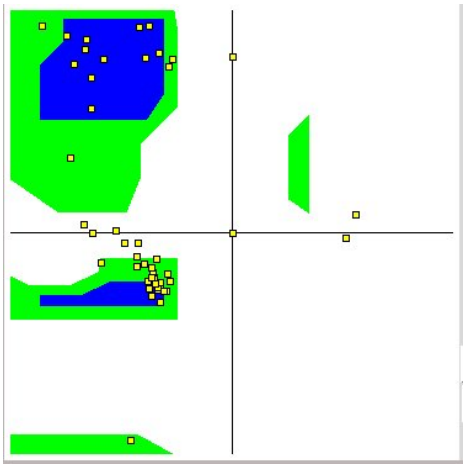


Figure 1.3: The Sasisekharan-Ramakrishnan-Ramachandran diagram of a simple protein, crambin (1JXU PDB identifier), calculated by the VMD software (http://www.ks.uiuc.edu/Research/vmd/). The diagram plots points given by the backbone consecutive ϕ and ψ torsion angle sizes, axes range from $-180°$ to $180°$ (the bond angle around the peptide bond is assumed to be $180°$). Notice how the observed angle values cluster in two regions. The color highlighted areas correspond to all sterically "possible" angle sizes – blue areas for an assumed standard van der Waals atomic radius, green for an estimate of a smallest radius possible. Residues with torsion angles in the third quadrant color area will likely form a right handed α-helix motif, whereas the second quadrant residues will form β-sheets. For a description of these structures, see below. The occurrence of different angle sizes in crambin, regardless of their "impossibility", shows that biological rules are not formal necessities. Their violation in this case might be extremely energetically disadvantageous, however, it might allow other regions of the protein to adopt more favorable conformations, producing a total enthalpy decrease. The Ramachandran exceptions are usually mostly glycines – due to their lack of a sidechain.

hydrogen bonds. The fold itself as a whole is called the tertiary protein structure. Note that secondary structures are not stable on their own as the hydrogen bonds between residues and water are energetically more favorable than between two residues. Stable secondary structures only exist in fully folded molecules in regions sufficiently shielded from water. Residues in these regions cannot satisfy their hydrogen bonding potential by bonding with water molecules and form bonds between each other instead – resulting in characteristic helices and sheets.

Sometimes, it is useful to recognise common clusters of secondary structure elements. For example, the helix-turn-helix motif occurs often in proteins (including crambin as can be seen in figure 1.4). These motifs are called the supersecondary structure. Also, many proteins need to be combined with other organic (possibly other proteins) or inorganic molecules before being able to fulfill their role in the cell. The structure of the entire resulting molecular complex is called the quaternary protein structure.

But other forces beside the hydrophobic effect and hydrogen bonding help stabilise the protein structure as well. Disulphide bridges, covalent bonds between sulphur atoms in residue sidechains, hold distant regions of the protein backbone or backbones of two distinct proteins together (see figure 1.5 for a display of the same protein, crambin, this time with disulphide bridges shown). Other covalent and coordinate bonds also hold various

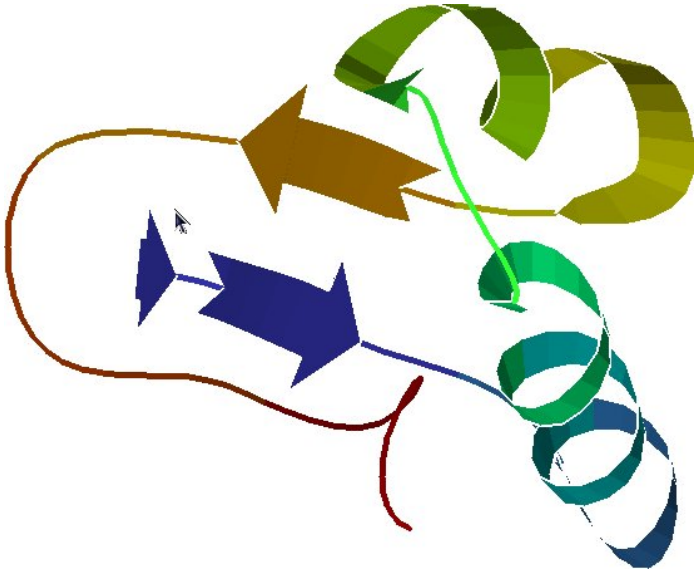


Figure 1.4: A cartoon-style view of a very simple protein, crambin (1JXU PDB identifier), drawn with the KiNG (Kinemage, Next Generation) software (http://kinemage.biochem.duke.edu/software/king.php). Only the backbone shown, thick banded helices model α-helices, aligned thick arrows β-sheets.

ligands and metal ions in the protein structure (e.g. the iron containing heme group in haemoglobin).

It is also useful to note how densely packed the protein molecule is. Our last display of crambin, figure 1.6, shows a space filling model of the molecule. In the protein core, residues are packed together so closely that no solvent molecule is able to interact – effectively shielding the hydrophobic residues from water. Additionally, because of the small interresidue distance, attractive van der Waals forces contribute significantly towards the total free energy of the system (the spheres displayed on the picture model only 75% of the van der Waals forces diameter).

Proteins can be roughly divided into three classes – globular proteins, fibrous proteins (scleroproteins) and membrane proteins – correlating with their tertiary structure. Globular proteins are mostly water soluble, compactly coiled (indeed, the name historically refers to a globe like structure) and function as enzymes. Fibrous proteins tend to be insoluble, filament-like and form cellular structural support and protection. Membrane proteins function either as receptors attaching to the lipid cell membrane or ion channels. In the latter case their structures have hydrophobic centers traversing the lipid bilayer and hydrophilic extremities protruding into the cell and to its surrounding area.

1.3 Structural Changes

The protein synthesis process begins with DNA transcription. RNA polymerase unwinds the DNA molecule and copies a region coding for a protein, a coding region – identified by a promoter-like pattern, the TATA box – into its RNA complement (transcript). In eucaryotes, the open reading frame corresponding to a protein is not continuous in the DNA but is interrupted by frequent non-coding regions (as much as 97% percent of DNA is non-coding in humans) and the transcript must subsequently be spliced, the non-coding regions

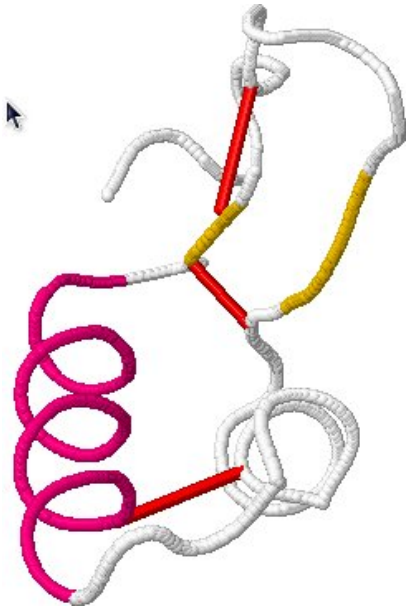


Figure 1.5: A trace-style view of a very simple protein, crambin (1JXU PDB identifier), drawn with the jmol package (http://jmol.sourceforge.net/). Only the backbone shown, stressed bonds between two distant backbone regions are the covalent disulphide bridges.

removed. The resulting molecule, mRNA, may undergo an editing process controlled by gRNA after which it is transported to ribosomes (made of rRNA) for translation into protein. Every triplet of nucleotides, a codon, codes for one amino acid. tRNA molecules with a complementary codon, an anticodon, transport amino-acids to the ribosomal translation sites and by the virtue of their reverse complementarity to particular codons ensure that only the correct amino acid is added to the growing protein strand.

But the newly synthesised protein molecules need by no means be of the final structure required for its function. Many proteins undergo a series of post-translational modifications. For example, carbohydrates or phosphates can be linked to certain residues (glycosylation[12] or phosphorylation[13] sites) making property inference from protein's DNA blueprint only very inaccurate. There are more than 200 different known post-translation modification types and the changes to protein structure they introduce may be either temporary or permanent. Post translational modifications can be much more involved than ligation, modifying the protein structure or chemical nature of the constituent amino acids. In insulin, for example, disulphide bridges are formed after translation and the backbone is subsequently cleaved into two strands held together by the newly formed disulphide bridges. During citrullination, arginine residues are deaminated creating citrulline.

Ligation always produces structural changes but these may range from minor to significant even in very similarly functioning proteins. For example, the oxy and deoxy forms of myoglobin differ only very slightly except for three last residues on the backbone, while the oxy and deoxy forms of haemoglobin are very different. Conformational changes through ligation may themselves be the main function of many proteins, including motors, muscle fibre proteins and ion-pumps. In result, care must be taken when using experimentally

[12] Most often asparagine.
[13] Most often tyrosine or threonine.

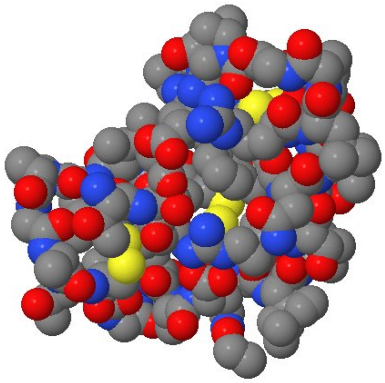


Figure 1.6: A space-filling view of a very simple protein, crambin (1JXU PDB identifier), drawn with the jmol package. Displayed spheres model 75% of the van der Waals forces diameter. Also visible in yellow, how the folded structure pairs sulphur atoms in the sidechains – for completeness, but not important for the current discussion, black spheres denote carbon, red oxygen and blue nitrogen atoms. Hydrogen atoms are not displayed for concision.

determined protein structures for model construction. The represented molecules are not static unchanging structures – large sidechains on the outer ranges of the molecule are especially mobile in almost all proteins, even those without a variable backbone.

One type of conformation change, the "hinge motion", involves a rigid movement of protein parts with respect to each other. Lactoferrin, for example, is two-domain protein of approximately 700 residues. The domains are thinly interconnected by the backbone twice at almost the same place. Upon release of the bound iron the two domains rotate through 54° around these connecting sites (residues 89-92 and 249-252). The RMS deviation of both domains from their state before the conformation change is only 0.82 Å and 0.58 Å respectively while the RMS deviation of the whole protein is 6.1 Å.

Another conformation change, the "helix shear", used in insulin, causes a displacement of the backbone atoms within helices which is made possible by a small change in the helix interfaces. The torsion bond angles shift a little allowing the side-chains to move. But the shift is minor enough to prevent a rotamer change. A database of protein conformational changes is maintained at Yale[14].

1.4 Secondary Structure

To summarise the previous sections, the protein backbone is driven (largely by the the hydrophobicity forces) to adopt a particular compact conformation. The resulting spatial arrangement prevents residues in the inner protein region from satisfying their hydrogen bonding potential by forming hydrogen bridges with water molecules in the surrounding environment. These residues therefore form the less energetically advantageous bonds with each other creating the secondary structure elements – helical or sheet-like regions connected by turns in the backbone. In a typical globular protein about 2/3 of the residues will be found in the secondary structure elements and the remaining 1/3 in the connecting turns.

[14] http://molmovdb.mbb.yale.edu/molmovdb/

Turns are much more flexible than helices and sheets and thus usually have functional roles in the protein, e.g. reacting to changes in ligation state.

By far the most common helical protein structure is the α-helix formed by sequences of residues in which the N−H group of the i-th residue is hydrogen bonded to the O=C group of the (i + 4)-th residue. Many α-helices are located on the surface of the protein presenting a hydrophobic side inwards and a hydrophilic side outwards. Recognising this pattern in the residue sequence can strongly indicate an α-helix region. A different pattern is characteristic of membrane traversing α-helices. In this case the lipid-buried helices are formed wholly by hydrophobic residues and connected by hydrophilic turn regions.

If the helical region is wound more tightly, an alternative helical structure forms, the 3_{10}-helix[15]. In a 3_{10} helix, the N−H group of the i-th residue is hydrogen bonded to the O = C group of the (i + 3)-th residue. Rarely, the helix winds less tightly forming the π-helix, in which the N−H group of the i-th residue is hydrogen bonded to the O=C group of the (i + 5)-th residue.

The exceptional residue, proline, disrupts helical regions because proline lacks the N−H group. Proline-rich regions tend to form a unique left-handed helical conformation, called the polyproline II helix.

Separate hydrogen-bonded strands form β-sheets. Adjacent strands may be oriented in the same or opposite direction. All strands of the same direction form a parallel β-sheet. Antiparallel sheets have every adjacent strand pair of an opposite orientation. All other sheets are mixed. The N−H and O=C hydrogen bonding groups are located in the plane of the sheet and groups of successive residues point in opposite directions. Therefore, they form hydrogen bonds to different strands. In antiparallel strands, each residue is able to form two hydrogen bonds with a single opposite residue, whereas in parallel strands, each residue bonds to two separate residues (two residues apart) on the opposite strand (see figure 1.7).

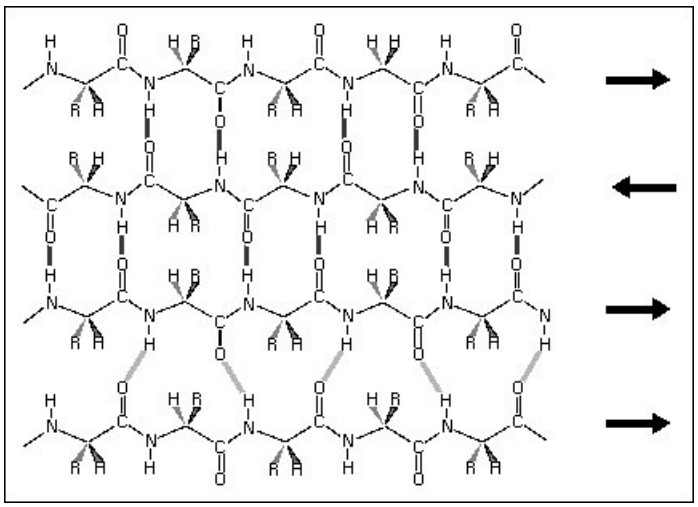


Figure 1.7: Hydrogen bonds formed in a mixed β-sheet. Picture owned by the School of Crystallography at Birkbeck (http://www.cryst.bbk.ac.uk/).

Commonly, antiparallel β-sheets form successively on the backbone, connected by short

[15]The designation indicates that there are 3 residues per turn and 10 atoms enclosed by each hydrogen bond ring. Similarly, the α-helix is sometimes described as 3.6_{13} helix.

turn-regions called β-hairpins; parallel sheets require longer bridging segments – often α-helices forming the so-called β–α–β units. Two notable features may form in β-sheets – the β-bulges and β-barrels. β-bulges form as irregularities in the structure when non-hydrogen bonding residues (usually a small number) protrude from the regular sheet-like pattern. β-barrels are created when the first strand of the sheet forms hydrogen bonds to the last strand. Table 1.3 summarises the properties of different secondary structure types.

Table 1.3: Secondary structure types parameters. ϕ and ψ are backbone torsion bond angles (ω assumed to be 180°). n gives the number of residues per structure element (turns in helices; same strands bonding residues in sheets) and d the distance between successive residues along the structure axis

Structure	ϕ	ψ	n	d
α-helix	$-57°$	$-47°$	3.6	1.5 Å
3_{10}-helix	$-49°$	$-26°$	3.0	2.0 Å
π-helix	$-57°$	$-70°$	4.4	1.1 Å
Polyproline II helix	$-79°$	149°	3.0	3.1 Å
Parallel β-strand	$-117°$	113°	2.0	3.2 Å
Antiparallel β-strand	$-139°$	135°	2.0	3.4 Å

The secondary structure elements are extremely evolutionarily conserved, although their lengths may change. On the other hand, the tertiary structure, the structural interaction of sheets and helices can undergo extensive evolutionary modification without violating the constraints presented by the function of the protein (for example, the structural haem pocket and the ability to bind oxygen in case of the globins).

Secondary structure elements occur frequently in typical super-secondary patterns. The most frequent of them include the already mentioned β-hairpins, $\beta-\alpha-\beta$ units and helix-turn-helix units, also called α-hairpins. Together, all the units sometimes form compact substructures of independent stability called domains. Domains are hard to define formally but are nonetheless important in systematic classification of protein structure (see figure 1.8 for an example of a two-domain protein).

Protein domains have independent spatial structure and protein structures are therefore classified according to the structure of domains (mostly secondary). Common structure classes include:

α-helical Domains composed exclusively or almost exclusively of α-helical secondary structure elements. Found for example in myoglobin or cytochrome c.

β-sheet Domains composed exclusively or almost exclusively of β-sheet secondary structure elements. Found for example in chymotrypsin.

$\alpha + \beta$ Domains composed of both α-helices and β-sheets but in separate regions. $\beta-\alpha-\beta$ motifs are absent. Found for example in papain.

α/β Domains composed of both α-helices and β-sheets frequently forming the $\beta-\alpha-\beta$ motifs. Centers of the sheet strands can lie almost in a line, for example in alcohol dehydrogenase, or in a circle (the β-barrel motifs), for example in glycolate oxidase.

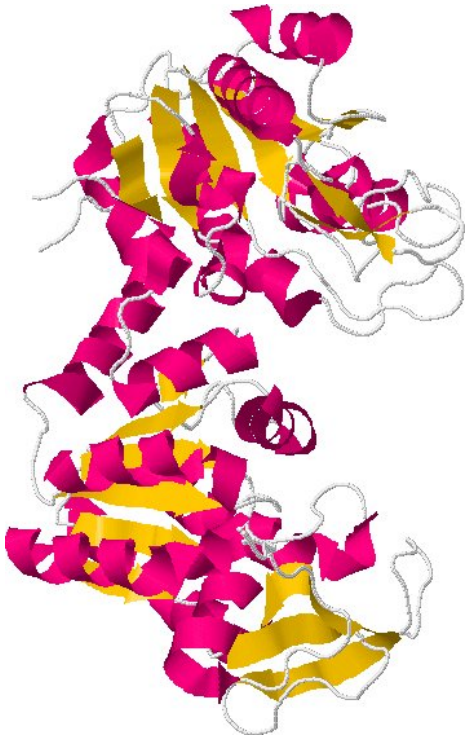


Figure 1.8: Pig muscle 3-phosphoglycerate kinase (1HDI PDB identifier), drawn with the jmol package. Two compact domains on the extremities thinly connected. The substrate binds to the area between domains.

1.5 Structure Classification

Known protein structure types are cataloged at three main sites which can be used to easily explore possible protein foldings. It is conjectured that among all the existing proteins in nature only about a thousand folds occur. As SCOP already catalogs more than a thousand folds, if true, the conjecture would imply the structure databanks have grown to a point where they contain representatives of almost every possible protein structure – making inductive structure prediction methods quite appealing. The classification sites are:

- SCOP (Structural Classification of Proteins)
- CATH (Class, Architecture, Topology, Homologous superfamily)
- FSSP (Families of Structurally Similar Proteins)

1.5.1 SCOP

SCOP[16] is a manually created dabatase which organises the recorded structures hierarchically according to evolutionary relatedness and structural similarity [MBHC95]. The lowest level in the hierarchy is occupied by individual protein domains as extracted from PDB. These are grouped into (clearly) homologue families – this is generally the case for proteins with more than 30% sequence identity. Families without sufficient sequence similarity (thus

[16] http://scop.mrc-lmb.cam.ac.uk/scop/

possibly without common evolutionary ancestry) are grouped into superfamilies. Superfamilies with a common topology, secondary structure arrangement and connection, are called folds. Each fold belongs to a structure class. SCOP recognises the already mentioned α, β, $\alpha + \beta$ and α/β classes, a "membrane proteins" class, a "small protein" class which is largely without formed secondary structure elements, held together by disulphide bridges and a "coiled coil" class. A coiled coil is a motif created by several α-helices coiling together like the strands in a rope, facing each other through hydrophobic sides as in muscle protein tropomyosin. For example, the SCOP classification of sperm whale myoglobin is: myoglobin (protein domain) – heme binding globin (family) – globin-like (super-family) – globin-like, partly opened fold with six helices (fold) – α class (class). Proteins are cataloged manually with human experts deciding whether two proteins are homologues or merely convergent – usually giving better results than the semi-automatic CATH.

The SCOP release 1.73 (September 26, 2007) catalogs 34,494 PDB entries (with a total of 97,178 domains) into 3,464 families and 1,086 distinct folds.

1.5.2 CATH

CATH[17] classification is similar to that presented by SCOP. Proteins are grouped into common class, architecture, topology and homologous superfamily [CSL+08]. Superfamilies consist of proteins with both sequence and structure similarities implying evolutionary relatedness. Each superfamily is subdivided into four families according to the degree of sequence similarity – "identical domain" family groups 100% identical domains, "like domain" family groups the remaining domains with more than 95% sequence similarity, "orthologous domain" family and "sequence" family similarly group domains with 60% and 35% sequence similarity respectively. A topology is a set of superfamilies with a common spatial arrangement of secondary structure elements connected in the same way. An architecture groups all families with a common spatial arrangement of secondary structure elements regardless of their connectivity pattern. CATH classes are α, β, $\alpha-\beta$ and a "sparse secondary structure" class. The $\alpha - \beta$ class is a union of $\alpha + \beta$ and α/β classes.

The classification proceeds semi-automatically. First, protein domains are manually extracted from PDB entries with the aid of a combination of structure[18] and sequence (e.g. HMMs) methods. Next, superfamilies are defined and domains assigned to them automatically by sequence clustering methods; topologies are defined and assigned by structure comparison of superfamilies. The domain class is also determined automatically from secondary structure composition. Human intervention is required for assigning topologies to architectures. Architectures get simple descriptive names according to the secondary structure. For example, myoglobin's architecture in CATH is "orthogonal bundle".

The current CATH version 3.2 (released July, 2008) classifies 114,215 domains into 2,178 homologous superfamilies, 1,110 topologies, 40 architectures and 4 classes.

[17] http://www.cathdb.info/

[18] PDB entries are scanned against a representative selection of CATH entries. CATHEDRAL algorithm identifies putative domains by rapid secondary structure comparison which are then more accurately aligned by dynamic programming and the alignments ranked by an SVM.

1.5.3 FSSP

FSSP[19] is a protein classification created by applying DALI[20], one of the best performing structure alignment programs developed today [HKRS08], to the entire PDB. The FSSP database is maintained by EBI[21]. DALI first eliminates structural redundancy by selecting only one representative for each set of proteins with more than 90% sequence similarity to the selected one (PDB90), reducing the size of the database by about 4/5. Common structural patterns are then detected by structure alignment and a structural similarity graph is constructed from the results. FSSP is a database created by walking the structural similarity graph from each vertex and selecting all connected (the edges in the graph represent alignments) vertices with similarity Z-scores higher than 2.0. In this way, FSSP neither requires human classification nor relies on possibly inaccurate automatic methods but merely displays all currently known structurally similar proteins and allows the user to interpret the results himself.

[19] ftp://ftp.ebi.ac.uk/pub/databases/fssp/

[20] http://ekhidna.biocenter.helsinki.fi/dali_server/start

[21] The European Bioinformatics Institute (http://www.ebi.ac.uk/), a part of the European Molecular Biology Laboratory, EMBL. EBI hosts a vast array of computational biology resources and tools including the EMBL-Bank nucleotide database, the UniProt protein database, Ensemble genome database and MSD protein structure database.

Chapter 2

Structure Databases

The major protein primary structure (amino acid sequence) database is UniProt, the universal protein database, created by joining the databases

- Swiss-Prot
- TrEMBL
- PIR

Swiss-Prot is s database of protein residue sequences maintained at the Swiss Institute of Bioinformatics. Swiss-Prot is manually curated to ensure low redundancy, high level of expert annotation and integration with other biological databases. Current UniProt release 14.6 contains Swiss-Prot release 56.6 with 405,506 protein entries (almost 150 million residues). TrEMBL (Translated EMBL), on the other hand, is a computer generated supplement to Swiss-Prot using the same database format – entries in TrEMBL are generated by translating all the coding sequences from EMBL-Bank, GenBank[1] and NSD[2] into their hypothetically corresponding proteins. Current UniProt contains TrEMBL release 39.6 with almost 7 million protein entries and over 2 billion residues. PIR, the Protein Information Resource, was a Georgetown University Medical Center database of protein sequences, finalised and merged into UniProt on December 31, 2004.

One special protein database worth noting for completeness of its data is the Human Protein Reference Database[3]. A manually curated database of human protein information maintained jointly by the Indian Institute of Bioinformatics and Johns Hopkins University. HPRD release 7 contains (as of December 27, 2008) 25,561 protein entries, 38,167 protein interaction records and 16,972 post-translational modification records. Other information stored include the corresponding DNA sequence, determined function and disease involvement, substrate binding sites and references to other biological databases.

The experimentally determined three dimensional protein structure is stored in the RCSB PDB database. In this chapter, we will describe the database, the information stored within, file formats used and methods of accessing individual records.

[1] GenBank is a nucleotide sequence and their protein translations database maintained by the National Center for Biotechnology Information (NCBI), a part of the National Institute of Health.

[2] NSD, the Nucleotide Sequence Databank, is a sequence database maintained by the DNA Databank of Japan.

[3] http://www.hprd.org/

2.1 The Protein Data Bank

The Protein Data Bank[4] is a database of publicly available molecular structures, originally at Brookhaven National Laboratory but currently maintained by RCSB[5]. PDB entries may include:

- Protein structures including ligands
- Nucleic acid polymer structures
- Carbohydrate structures
- Historically, PDB also stored hypothetical models of protein structures. But these have been removed from the database in July, 2002 and no models are accepted for entry anymore

The PDB current holdings statistics (as of December 20, 2008) report a total of 54,559 stored structures, 50,377 of which are proteins. However, it is hard to estimate the number of unique entries stored. Protein structures may have been determined independently by different methods, using different crystallisation or resolution. Also, the described molecules may differ by ligation only. Various purely bibliographic information is recorded as well. A complementary BMCD database, the Biological Macromolecule Crystallization Database[6], contains data about crystals and crystallisation conditions.

The PDB database is synchronised once a week with other protein structure databases that are part of the wwPDB group[7]. This ensures worldwide integration of structural bioinformatics resources. Beside data storage, search and retrieval facilities, these databases together offer a vast array of additional functions – various levels of annotation and crossreferencing with sequence, structure classification and biological literature databases, visualisation services, motifs, ligands and active sites information, local residue interaction patterns extraction and pairwise structure comparison functions (secondary structure matching), data extraction, format conversion and validation tools

One more thing worth noting about PDB is its uneven representation of different protein structure topologies. The majority of the database was determined using X-ray crystallography and it is notoriously hard to produce quality crystals for some kinds of proteins while being relatively simple for others like globular proteins which are consequently overrepresented in the database. PDB_Select was created to address this issue and contains a subset of PDB with no homologues. It can be used to quickly extract representative selection of different protein architectures currently known. The latest (October, 2008) PDB_Select25[8], a selection of only one representative for a set of proteins with more than 25% sequence similarity to the selected protein, contains 4,018 protein chains.

The PDB repository and the PDB file format are acknowledged standards usable by many standalone applications. We will limit our discussion of structure databases to these

[4] http://www.rcsb.org

[5] The Research Collaboratory for Structural Bioinformatics is a collaboration between The Rutgers University, The University of Wisconsin-Madison, The University of California and The San Diego Supercomputer Center founded for the study of biological macromolecular structures.

[6] http://xpdb.nist.gov:8060/BMCD4

[7] The world wide PDB with current members being the RCSB PDB, the EBI PDBe (formerly the Macromolecular Structure Database), the PDBj of Japan and the University of Wisconsin-Madison's BMRM (Biological Magnetic Resonance Bank).

[8] http://bioinfo.tg.fh-giessen.de/pdbselect/recent.pdb_select25

alone. The most important part of PDB entries are the 3D coordinates for every atom of the protein molecule, belonging both to the backbone and the sidechains, with the exception of hydrogens as these cannot be experimentally determined with sufficient accuracy. Summarisation information, literature references, experimental method details and the residue sequence may also be recorded.

2.1.1 PDB Format

Every PDB entry (currently at version 3.2) is a plain non-control ASCII file with each line 80 columns wide. First six columns of a line contain a record identifier, the remainder of the line contains a variable number of fields – left justified and blank separated. If the number of required fields exceeds the space available on a single line, the record is continued on the subsequent line with the same record identifier and the first field a right justified sequence continuation number. The entry is divided into twelve sections, each with a prescribed record order. Not all records are mandatory. In case data is unavailable for a mandatory record, a NULL value must be explicitly stored. We will now provide a brief description of the database format. For the full format specification, see the wwPDB consortium documentation [wwP08]. Note, that PDB currently makes all its entries available in mmCIF and PDBML formats as well. mmCIF is a macromolecular Crystallographic Information File format with PDB extensions to accommodate NMR and Cryo-EM data as well. PDBML is an XML representation of PDB data entries. The XML schema and mmCIF dictionaries are fully specified at PDB Exchange (`http://mmcif.rcsb.org/`).

The *Title* section provides a description of the macromolecule referred to and details of the experiment used in structure determination. It contains a *HEADER* record with a unique PDB identifier, the date of deposition and the chemical name, classification or functional description of the molecule. The identifier is a four character string, the first character is an integer indicating a revision number of the experimental structure (re)determination, only entries with this number non-zero store atomic coordinates[9]. The remaining characters are alpha-numeric. If a protein structure is revised, *OBSLTE* record is added to the section indicating the identifier of the newer entry. *SPRSDE* record indicates which entries were made obsolete by the entry. *REVDAT* record stores the history of modifications to the entry (not related to the structure itself). Complementing the *HEADER* record, the *TITLE* and *EXPDTA* records provide experiment details. For molecules that are parts of larger macromolecular complexes the *SPLIT* record provides the PDB identifiers of the remaining molecules. The *CAVEAT* record lists known limitations of the entry.

The *COMPND* record stores a list of token:value pairs holding various information about the studied macromolecule including chain and fragment identifiers, chemical synonyms and mutations present during the experiment. Detailed information about the biological or chemical sources of the molecule is stored in the *SOURCE* record. Arbitrary and model specific annotation can be specified in the *KEYWDS* and *MDLTYP* records. For molecules with multiple models in entry, the *NUMMDL* record states their number. *JRNL* and *AUTHOR* records store information about the biologists responsible for the structure determination and primary literature with published details of the experiment.

The *Remark* section stores all the information, annotation and comments not fitting into other sections and records. The section consists of *REMARK* records with the first field being an integer specifier ranging from 0 to 999 indicating the kind of the ensuing

[9] PDB used to store bibliographic entries with identifiers beginning with zero. These are no longer part of the database though.

remark. Many specifiers are already predefined in the format definition.

The *Primary Structure* section contains the protein's residue sequence information for all chains in the *SEQRES* record; crossreferences to a corresponding GenBank or UniProt databases are also specified in the *DBREF* record. In case the stored and crossreferenced sequences differ, the conflicts are listed in the *SEQADV* record while *MODRES* lists post-translational modifications.

Non-standard residues and groups of the molecule are described in the *Heterogen* section in the *HET* record. Each non-standard compound is identified by a unique (for the entry) up to three letters long alpha-numeric identifier which is subsequently defined in *HETNAM*, *HETSYN* and *FORMUL* records. These indicate the chemical name, synonyms and empirical formula of the group.

The secondary structure elements and turns recognised during the structure determination are listed in the *Secondary Structure* section in the *HELIX* and *SHEET* records. Helices are numbered and their type, length, first and last residues and the belonging chain are recorded for each. Sheet records store a number of strands and each strand is defined by the chain, number of its first and last residue and direction (parallel/anti-parallel) indication with respect to the previous strand (except for the very first one).

Determined covalent bonds linking distant regions of the protein (e.g. disulphide bridges) can be stored in the *Connectivity Annotation* section. The *SSBOND* record represents disulphide bridges. The parent chain and sequence number of linked residues as well as a bond length are stored. Other non-standard covalent bonds not implied in the primary structure are given in the *LINK* record. The very rare cis conformations of the peptide bond (mostly for proline) are recorded in *CISPEP* records.

The *Miscellaneous* section contains a single record *SITE* which is used to list important protein regions like active, regulatory, catalytic or co-factor sites.

For entries determined by the X-ray crystallography method, the *Crystallographic* section records the geometry of the crystal units used in the experiment.

The *Coordinate Transformation* section indicates the transformation from the entry's atomic orthogonal coordinates to the coordinates determined in the experiment. The *ORIGX* record specifies a 3x3 matrix **O** and 3x1 matrix **T**, such that every orthogonal atomic coordinate **c** = (x,y,z) corresponds to an experimental coordinate **cc** = (xx, yy, zz) after a transformation $\mathbf{cc} = \mathbf{O} * \mathbf{c} + \mathbf{T}$.

For our purposes the most important section is the *Coordinate* section with the orthogonal coordinates of every (usually excluding hydrogens) atom in the molecule. The coordinates of standard polymer atoms are nested between *MODEL*, *ENDMDL* records for each determined model[10] and stored in *ATOM* records. Non-standard atom coordinates are recorded in the analogous *HETATM* record. The record stores a lot of information including the serial number and chemical symbol for the atom, the parent residue name, number and chain identifier, the charge, occupancy and temperature factors[11] (the anisotropic temperature factor can be specified in the *ANISOU* record) and atomic coordinates in Å. The atom listing for a chain is terminated with a *TER* record. For a snippet of a PDB model see figure 2.1.

The connectivity within non-standard groups and between non-standard and standard

[10] Multiple models are usual only for NMR experiments.

[11] The factors are only recorded for X-ray crystallography structures. The temperature factor gives the amount of electron density spread of the atom in the diffraction pattern – accounting to the extent of the atomic thermal motion. The occupancy factor gives the fraction of observed to expected electron density for the atom in the experiment.

Figure 2.1: A snippet of a PDB model. The *ATOM* fields correspond to atom's serial number, standard chemical name, parent residue, parent chain, residue serial number, x, y, z coordinates, occupancy and temperature factors and chemical symbol respectively. You can see the first chain, A, spans residues 16 (the structure of residues 1-15 was not determined in the experiment) to 245 composed of determined atoms 1-1665. The second determined atom is a carbon atom, the α-carbon, belongs to isoleucine, the sixteenth sequential residue, on chain A and has orthogonal coordinates (-18.974, -84.925, -1.727) Å.

```
MODEL        1
ATOM      1  N   ILE A  16     -18.856 -85.087  -3.210  1.00 13.89        N
ATOM      2  CA  ILE A  16     -18.974 -84.925  -1.727  1.00 16.47        C
...
ATOM   1665  OXT ASN A 245      10.744 -96.898 -11.063  1.00 39.25        O
TER    1666      ASN A 245
...
ENDMDL
```

groups is specified in the *Connectivity Annotation* section. The record *CONECT* lists atom serial numbers of bonded atoms for each atom whose connectivity is recorded.

The final *Bookkeeping* section provides information about the entry itself. The *MASTER* record stores statistics information e.g. the number of helix and sheet records, remarks or site annotations. The *END* record closes the entry.

2.1.2 PDB Processing

The bioinformatics python module BioPython [CCF+08] implements a useful PDB file parser. *Bio.PDB.PDBParser* method *get_structure()* processes a PDB file and creates a *Structure* object which stands at the top of the protein structure representation hierarchy. The other levels consist of *Model*, *Chain*, *Residue* and *Atom* objects. Every object has a unique identifier and is accessible by direct indexing of the parent object, for example:

```
model = structure[0]; chain = model[''A'']
```

The reference back to the parent object and a list of all children (if applicable) can be retrieved with the *get_parent()* and *get_list()* methods. A structure identifier is the PDB four-letter string, a model identifier an integer and a chain identifier an arbitrary string but conventionally, chains get assigned single alphabetic characters consecutively for the chains starting from 'A'. Residues are identified with a triplet (*het*, *seq*, *alt*), where *het* is an empty string for standard residues, "W" for water molecules and concatenation of "H_" and a standard chemical name for other non-standard protein groups, *seq* is the residue sequence number on the backbone and *alt* is used to assign unique identifiers to insertion mutants (rare), blank otherwise. *get_full_id()* residue method returns a complete residue identifier, for examples see figure 2.2. *get_resname()* method returns a three-level abbreviation of the residue name.

Finally, the atom object identifier is a string up to four characters long unique to the parent residue, e.g. "CA" for the α-carbon. The object encapsulates all the PDB records (*ATOM*, *ANISOU*, etc.) describing the atom. The information is accessible through the various *get_* methods, e.g. *get_id()* or *get_coord()*. Coordinates are stored in a numPy array.

Figure 2.2: An example of full residue identifiers. All three examples of fictive PDB entries – the first model in the structure (0), the first chain in the model (A) and the tenth residue on the backbone. The first two residues are non-standard, water and glucose respectively. The third case is a standard residue.

```
(‘‘1ABC’’, 0, ‘‘A’’, (‘‘W’’, 10, ‘‘’’))
(‘‘1DEF’’, 0, ‘‘A’’, (‘‘H_GLC’’, 10, ‘‘’’))
(‘‘1GEH’’, 0, ‘‘A’’, (‘‘’’, 10, ‘‘’’))
```

BioPython can be used for comfortable processing of the PDB entries. For example, to print the coordinates of all α-carbons of crambin, we type:

```
from Bio.PDB.PDBParser import PDBParser
parser=PDBParser()
structure=parser.get_structure(‘‘1JXU’’, ‘‘1JXU.pdb’’)
for model in structure.get_list():
   for chain in model.get_list():
      for residue in chain.get_list():
         if residue.has_id(‘‘CA’’):
            ca=residue[‘‘CA’’]
            print ca.get_coord()
```

Chapter 3

Structure Prediction

In this chapter, we will provide an overview of existing computational protein structure prediction methods. A very good account of the entire field of structure prediction can be found in [Tra06], we will follow her classification here. Other useful information, especially on the lattice HP-model, is contained in [Isa06]. The area of protein structure prediction might sometimes be considered as limited to certain aspects, e.g. assigning a secondary structure type to each residue. However, we will here discuss only methods aiming at full tertiary structure prediction – assigning a spatial coordinate to each residue.

The prediction methods can be broadly divided into two classes, deductive and inductive. Deductive methods attempt to determine protein structure from the corresponding protein residue sequence and laws of nature alone. On the other hand, inductive methods attempt to discover the pattern of correspondence between residue sequences and their respective experimentally determined structures from biological databanks and use their knowledge in structure prediction.

The focus of the chapter will be on the latter, namely on comparative modeling. It is also in the context of this method that we will discuss evaluation of the predicted structure model, although it is obviously an issue common to all of them.

3.1 Deductive Methods

Deductive (ab initio – from first principles) methods try to predict protein structure from the knowledge of the residue sequence alone, given a model of physical laws. No use of databases of already determined structures is made. The theoretical foundation behind ab initio methods lies in the assumption that the native, folded, protein conformation forms a system of the lowest free energy possible. The problem of determining this conformation then reduces to discretising the continuous space of possible conformations, calculating the free energy of each conformation and searching through the discrete energy space for its global minimum.

3.1.1 Lattice HP-Model

Perhaps the simplest and crudest structure prediction model is the lattice HP-model [DBY+95]. As a simplification, all residues are classed as either hydrophobic or hydrophilic (polar) and the space is discretised into a three-dimensional unit lattice (see figure 3.1). The modeled backbone (of two kinds of residues) is then fitted into the lattice in such a way that each lattice vertex can be occupied by at most one residue, thus disallowing steric collisions.

The energy function for this model can be taken simply as a negative count of adjacent hydrophobic residues in the lattice[1] in order to favour the dense packing of hydrophobic residues usually found in the cores of naturally folded proteins.

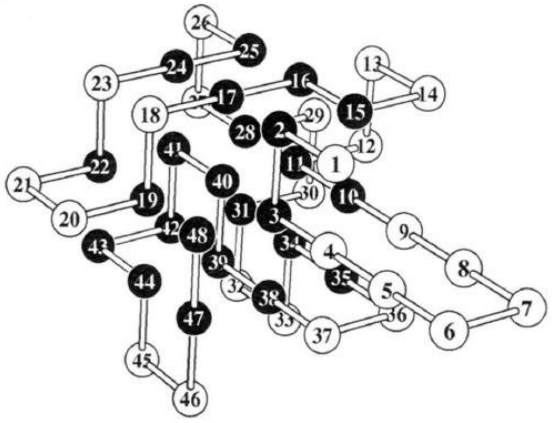


Figure 3.1: An example Lattice HP-Model (source: [DBY+95]), black beads represent hydrophobic, white hydrophilic residues. The free-energy function is a sum of all contact energies, where a contact is formed by adjacent residues non-consecutive on the backbone and the (HH, PP, HP) contact energies are (-1, 0, 0).

The model is clearly a vast simplification which forfeits almost all molecular detail and resolution in exchange for the ability to explore many conformations and conformational changes over long periods of time – something the more exact methods in later sections have considerable problems with. Also, the model presents an alternative to the more traditional theory assuming the protein folding pathway to proceed from the local formation of secondary structure elements towards the stabilisation of tertiary structure. The HP lattice on the other hand assumes that non-local hydrophobic forces are the principal protein folding driving force initiating the whole process, whereas secondary structures evolve only as a consequence.

Although the lattice model is very simple and the calculation of its corresponding energy function very fast, finding the function's global minimum is nevertheless an NP-complete problem. For short sequences (length less than 10), exhaustive search is usually used, while heuristics are employed for somewhat larger problems (length less than 50) – genetic algorithms, Monte Carlo or random sampling and exhaustive search of the samples might be used. A linear time approximate solution applicable to even larger instances of the problem exists with a performance guarantee of 0.375 [HI96].

3.1.2 Biophysically Realistic Models

The most exact results of any ab initio modeling would come from quantum mechanical principles and a direct solution of the Schrödinger equation, which is, for example, for a single particle system given by:

$$i\hbar\frac{\partial\Psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\Psi(x,t)}{\partial x^2} + V(x)\Psi(x,t) \tag{3.1}$$

[1] Alternatively, we may limit the function to counting only hydrophobic residues adjacent in the lattice but not consecutive in the sequence.

where $\hbar$ is the reduced Planck constant, $V(x)$ is a potential energy function and m the particle mass. The solution of this partial differential equation, the wavefunction $\Psi(x,t)$, can be interpreted as giving the probability of the particle occupying position x at time t. That is, $|\Psi(x,t)|^2$ is the probability density of the particle occupying position x at time t.

Unfortunately, the analytical solution to the equation for molecules larger than hydrogen is not obtainable, and a numerical solution is very expensive to compute. We will have to content ourselves with approximate solutions. Besides, even if we could compute the equation for large systems, modeling the entire protein molecule on quantum scale would be horrendously impractical. Instead, we will employ various classical models and use quantum simulation only on very simple atomic systems to obtain values for some of the parameters occurring in these models. For example, a physically reasonable, yet still computationally relatively simple, classical model represents the energy of the system as [BMRW01]:

$$E_{total} = E_B + E_{BA} + E_{DA} + E_{EL} + E_{VDW} \tag{3.2}$$

The first part represents chemical bond energy. The model considers atoms as rigid spheres and covalent bonds holding them as harmonic oscillators giving a Hooke's law energy approximation:

$$E_B = \sum_i \frac{1}{2} K_{B_i} (r_i - r_i^0)^2 \tag{3.3}$$

where K_{B_i} is a spring constant, r_i a bond length and r_i^0 an equilibrium bond length. The sum ranges over all covalent bonds in the molecule.

Bond angles can be modeled as harmonic oscillators as well giving an analogous bond angle energy contribution:

$$E_{BA} = \sum_i \frac{1}{2} K_{A_i} (\theta_i - \theta_i^0)^2 \tag{3.4}$$

where K_{A_i} is a spring constant, θ_i a bond angle size and θ_i^0 an equilibrium bond angle size. The sum ranges over angles between every consecutive pair of covalent bonds in the molecule.

The energy contributed by dihedral angles is more complex to model because there might be multiple potential barriers encountered when rotating the molecule around the corresponding bond. Usually, a multiple-minima goniometric function is used to represent the dihedral bond free energy contribution:

$$E_{DA} = \sum_i \sum_{j=1}^{N_i} \frac{1}{2} K_{D_i} (1 + cos(j * \chi_i - \chi_i^0)) \tag{3.5}$$

where K_{D_i} is a dihedral energy barrier, χ_i a dihedral angle size and χ_i^0 a phase angle. The first sum ranges over all dihedral angles in the molecule. The second sum ranges over all energy minima of the given angle.

An important contribution to the free energy of the molecule lies in electrostatic interactions between charged (or partially charged) atoms. For proteins, the electrostatic forces are especially important as hydrogen bonds form an important stabilising element of the secondary structures. In order to avoid quantum mechanical computations, we will ignore

the charge distribution within an atom and consider only interactions of uniformly charged rigid spheres[2]. Coulomb's law then gives:

$$E_{EL} = \sum_{i<j} \frac{q_i q_j}{4\pi\varepsilon_0 \varepsilon_r(r_{ij}) r_{ij}} \tag{3.6}$$

where q_i, q_j are the atomic charges, r_{ij} their distance, ε_0 the vacuum permittivity and $\varepsilon_r(r_{ij})$ the dielectric "constant" of the medium. The sum ranges over all atom pairs of both the molecule and the surrounding environment. The dielectric constant is not constant on the molecular scale, however. It is a macroscopic quantity arising from the molecules of the medium aligning themselves along the electric field and their dipoles reducing its strength. The constant's value presumes a large number of uniformly distributed medium molecules between the charges which clearly does not hold in our case. One solution would be to explicitly include the medium molecules in the energy calculation. A computationally less expensive solution presented here regards the dielectric constant as an increasing function of the distance between charges to model the effect of an increasing number of medium molecules opposing the electric field with an increasing separation of the charges.

Even uncharged atoms are affected by electromagnetic fields – they vibrate producing dipole moments resulting in attractive forces with nearby atoms. Quantum principles forbidding two electrons from having the same quantum state prevent electron orbitals from overlapping and produce strong repulsive forces between atoms too close to each other. These two effects can be summed in the van der Waals interaction, their energy given by:

$$E_{VDW} = \sum_{i<j} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right) \tag{3.7}$$

where A_{ij} and B_{ij} are empirically determined constants. The constants in the remaining equations can in some cases be determined experimentally, but are usually produced by computational means by quantum mechanics simulation of small model molecules. For example, in case of the bond energy contribution, the equilibrium bond length can be measured from X-ray diffraction patterns, the spring constant estimated from quantum simulations.

We have thus formulated a reasonable approximation to the free energy function. The conformation space can be discretised simply by choosing a space granularity in each dimension, e.g. 0.01 Å. The last component to consider is the method of exploring the vast solution space. Molecular dynamics simulation – a numerical solution to the n-body problem with extremely small time steps (on the order of $10^{-15}s$) during which the acceleration of atoms can be considered constant is one possibility. Other frequently used approaches include genetic algorithms and simulated annealing/Monte Carlo sampling.

3.2 Inductive Methods

As stated previously, inductive methods attempt to predict protein structures with the help of databases of already determined structures. The methods can be, although somewhat artificially, divided into three types – comparative modeling, fold recognition and new fold prediction – depending on the level of relatedness between the target protein and the most similar protein with an already known structure in the used database.

[2] Alternatively, we may, of course, calculate partial charges by quantum simulations of simple models.

3.2.1 Comparative Modeling

Comparative modeling ([Gin06], [Cha03]) is based on the fact that among related proteins, the spatial conformation shows strong evolutionary conservation (especially at the active sites) and on our ability to detect evolutionary relatedness by a good quality sequence alignment. Broadly speaking, we are looking for a family of proteins containing the query. By analysing the family, we may discover sequence features conserved across all the members. It is a very good guess to suppose that the reason for their conservation is their bearing a function selected for during evolution – which is directly determined by spatial structure. Simply taking a known structure of a protein from the family corresponding to the conserved regions of the backbone can serve as a reasonably good model for the query. Frequently, this is all that a biologist needs to evaluate the query's function, but refining and extending the model to cover the diverging regions as well is also possible. Comparative modeling then consists of these steps:

1. Identification of proteins related to the query
2. Alignment of these proteins with the query
3. Construction of a preliminary structure model
4. Extension of the model to the entire backbone
5. Sidechain modeling
6. Final optimisation

Evolutionary relatedness can be reliably determined by producing a sequence alignment (e.g. BLAST or FASTA algorithms) of the protein in question with sequences of all known proteins and selecting a subset of well scored alignments as a family of the target's relatives. Evaluating the obtained scores of the alignments is not an entirely trivial issue, however. One possible approach adopted in principle by both FASTA and BLAST is to estimate the statistical significance of the alignment by obtaining the probability distribution of alignment scores between completely independent sequences. We can then compute the probability of a score belonging to the distribution and therefore not being significant. The background probability distribution can be computed by reshuffling a given sequence many times to produce sequences of identical composition yet random order and consequently no evolutionary relation and sampling their alignment scores. For example, FASTA reshuffles the target sequence and repeats the alignment on subsets of the database, while BLAST constructs a set of random sequences with an average length and composition similar to sequences of the databanks and aligns the original target with these. It turns out that the expected score distribution is not Gaussian but resembles the extreme value distribution instead:

$$p(x) = \frac{1}{\beta} e^{\frac{x-\mu}{\beta}} e^{-e^{\frac{x-\mu}{\beta}}} \tag{3.8}$$

with β a positive real number and μ the center of the distribution – determined by fitting the distribution to measured data.

In the theory of extreme value distribution, the expected number of alignments attaining the score S by chance is given by:

$$E(S) = Kmne^{-\lambda S} \tag{3.9}$$

where K,λ are constants chosen depending on the substitution matrix used during the alignment and m,n are the lengths of the target and the sequence it is compared with respectively. The equation can be proved to hold for ungapped alignments only but experiments show it works reasonably well for gapped alignments as well. We can use the equation to set the alignment score threshold for two proteins to be considered related high enough to get an acceptable number of false positives. Alternatively, the algorithm can just report the E values with the results and let the user make her own decision as to their significance.

There is another issue, though, concerning not only the database search but the whole modeling procedure – what should properly constitute a query sequence? The units of evolutionary relatedness are protein domains, not the entire proteins and we may be able to get much additional information by searching the databases against each domain of a multidomain target. One immediate benefit is getting more hits than would be found in the multidomain case. More importantly, the extreme compactness and independence of domains enable the method to produce more accurate results if limited to modeling a single domain. Combining the domain models into one structure is a complex problem, though. It is not obvious how to fix the connecting regions in space and these are usually quite long making attempts at their de novo modeling inaccurate, ruining the precision of the resulting structure. If a complete model is desired then, it is better to use the entire multidomain target as a query, because any hit of sufficient similarity will produce a good structure template fixing all the domains in a reasonable mutual conformation.

Nevertheless, even if one should wish to limit the modeling to a single domain for an increase in accuracy, the domain prediction problem is very hard and still largely unsolved. Possible approaches include sophisticated machine learning methods which attempt to recognise patterns characteristic to interdomain connecting regions. However, one of the best methods is probably also the simplest one, albeit a bit brute force. It is based on estimating the length of one of the target's domains and doing the sequence alignment of overlapping segments of this length against the entire database. If a very good quality alignment is found with multiple different proteins, the segment is probably a domain or its part – which can be determined by extending the segment and retrying the search. Once a domain is identified, it is removed from the sequence and the procedure continues searching the remainder for other possible domains.

After selecting a set of similar sequences, we need to identify their conserved regions – the protein cores. The cores can be most efficiently identified by constructing a multiple sequence alignment (e.g. CLUSTAL algorithm) of all the proteins in the found family of the target's relatives. The protein cores, regions conserved during evolution by virtue of their functional and therefore structural properties will be aligned in the majority of the proteins, whereas random matches with the target will be different in each member of the family. This is the reason why the previous search for related proteins was not limited to proteins of known structure only, although these alone can be used in the structure model construction. With more proteins in the family, we are able to compute a better quality multiple alignment and identify protein cores more reliably.

Consecutive stretches of conserved positions are good core indicators. Sometimes, however, structures corresponding to even such similar sequences can differ and whenever possible, it is advisable to verify the genuine nature of the identified cores by consulting other sources of information. For example, secondary structure elements are probably safe to presume identical in the entire family while residues adjacent to insertions/deletions have certainly changed their position. If structures are available for multiple proteins of the fam-

ily, their structural alignment can be profitably used to identify truly conserved regions.

With the cores identified, we select one member of the family with its structure already known as a template. If multiple proteins with known structures exist, we choose the one with the highest sequence similarity to the target. Some methods might try to use multiple templates, superpose their structures and average the coordinates of corresponding atoms, or assign a different template to each domain according to their similarity but using a single template is a sure and safe solution.

Next, we align the template with the target to find out which residues correspond to the identified cores. This step is necessary because the previously constructed multiple alignment has likely not produced a good pair-wise alignment of these two sequences - especially if an iterative algorithm was used and the sequences were not the first two to be aligned. The optimal dynamic programming algorithm should be used. And even so, the alignment should be manually inspected for obvious errors to be removed. For example, insertions of gaps inside secondary structure elements are extremely unlikely and these should be moved to the closest loop. The modification is necessary because alignment algorithms such as the Needleman-Wunsch dynamic programming method produce a generic alignment of a guaranteed maximal score given a set scoring matrix and an affine gap penalty function. Such an alignment need not be optimal in light of additional information available to us. Besides moving gaps outside the boundaries of conserved secondary structures, other modifications are advised. For example, heavy mutations or insertions/deletions on the solvent surface of the protein are less likely than inside the buried region shielded from the surrounding environment.

We can determine which residues correspond to each other in the target and the template from this alignment. The structure model is constructed by assigning coordinates of each core atom of the template to the corresponding atom in the target. Such models constructed from highly similar templates can be very accurate, comparable to low resolution experimental models – this makes comparative modeling very useful especially if applied on globins, the cores of which will usually cover almost the entire length of the backbone. The procedure may end at this point or extending the model outside of the core – to the divergent regions – may be attempted, although no satisfactory solution is currently known.

For a start, if the divergent region is very short, say a beta hairpin, only a small number of conformations exists. These are well studied and possible hairpin types cataloged according to the backbone dihedral angles. We can try modeling the short divergent region with each type, evaluate the resulting stereochemistry and select the most reasonable one. Unfortunately, this is less useful than it may appear because such short loops with sharp turns are constructed by very specific hydrogen bonding patterns and allowed only by the flexibility of glycines, it is probable that they will be conserved in the protein cores anyway.

Models of medium-sized divergent regions can be built by constructing all stereochemically possible models and selecting the best one on energy considerations. A similar approach is usually used to model sidechains. Core sidechains belonging to residues identical in the target and template proteins are simply copied from the the template, the others are chosen from rotamer libraries each time selecting the energetically most favorable sidechain. The computational complexity of this method steeply rises with length and it is not applicable to longer divergent regions. These might be predicted by fragment-based methods (see later sections) instead.

The potential energy function used for conformation evaluation is usually not a physical force field but a knowledge-based potential. This approximation abstracts away most of the physical atomic-level entities and attributes all the interaction to a single point per residue.

The magnitude of the interaction is estimated from statistical analysis of experimentally determined structures by applying the Boltzmann distribution – that is, treating the frequency of a feature as exponentially related to the negatively-taken energy of the feature. A popular approach examines the structure databases and for each pair of residue types counts the number of contacts at different distance ranges (e.g. the number of glycine-valine contacts at less than 3 Å, 3 - 4 Å, etc.) and estimates the energy of such a contact from a comparison with the contact number expected by chance alone.

$$E(a,b,d) = -KTln\left(\frac{p(a,b,d)}{p_0(a,b,d)}\right) \tag{3.10}$$

where p, p_0 are frequencies of occurrence of contacts between residues a and b at some distance d in the database, respectively in a random structure. Random structures can be easily generated by taking an existing structure and permuting its amino acid sequence.

Finally, we should, at least in theory, be able to optimise this model by molecular dynamics refinements. In practice, such attempts have so far failed. Our current force fields are not accurate enough for the task – they will, for example, modify even an experimental model if applied to it.

Let us now discuss means of evaluating the model's quality against an experimentally determined structure. A traditional metric of their difference is the root-mean-square deviation – an rms distance of corresponding atoms after optimal superposition of the structures. That is, the $rmsd$ of structures X and X', each of them consisting of N atoms is defined as:

$$rmsd(X,X') = min_{(T,R)}\sqrt{\frac{1}{N}\sum_{i=1}^{N}[(x_i - R_x x'_i + T_x)^2 + (y_i - R_y y'_i + T_y)^2 + (z_i - R_z z'_i + T_z)^2]}$$

where $T = (T_x, T_y, T_z)$ is some translation and $R = (R_x, R_y, R_z)$ some rotation in euclidean space.

The rmsd is however a quadratic measure penalising larger deviations more severely which is not always desirable. We may wish to regard two models predicting a wrong structure for a loop in the same way, regardless of how wrong the prediction was. In addition, we mostly care only about the overall structure of the protein given by the backbone conformation and can safely ignore the positioning of the sidechains. An alternative measure would just count the fraction of residues that are within some threshold from their counterparts after optimal superposition. In practice, we average the fractions at different distance thresholds as in the popular GDT-TS measure:

$$GTT\text{-}TS(X,Y) = \frac{1}{4}(f(X,Y,1) + f(X,Y,2) + f(X,Y,4) + f(X,Y,8)) \tag{3.11}$$

where $f(X,Y,d)$ is the fraction of alfa-carbons in the structure X found within a distance d of their counterparts in the structure Y after an optimal superposition. Also, another plus – the GTT-TS measure is, unlike the rmsd, applicable to models which chose not to cover the divergent regions.

3.2.2 Fold Recognition

With sizes of our protein structures knowledge bases steadily increasing, the success rate of comparative modeling methods continues to rise as well. Nevertheless, the task of determining the conformation of a protein with no evolutionary relative of known structure

currently in the biological database still needs to be faced. Fortunately, the failure in locating a homologue of a known structure is not a complete catastrophe, as even proteins with completely unrelated sequences can adopt a similar fold. Fold recognition methods try to select a structure template for the query sequence – lack of a known homologue notwithstanding. After the template is identified, the structure model construction proceeds in the same way as in comparative modeling. This was discussed previously, hence, this section will focus on the template identification problem exclusively.

The fold recognition methods attempt to assign a structure template to the target protein by evaluating how well the given structure serves as a model for the query. A very good match at this point might indicate a newly found evolutionary relationship with another protein of known structure, eventhough mere sequence comparison was unable to establish the relationship. The distinction between fold recognition and comparative modeling is therefore rather blurred – fold recognition might be regarded as a more sensitive and less accurate type of comparative modeling. There are two principal ways of identifying a good template – profile based methods and threading.

Profile based fold recognition methods approach the problem by comparing the known physical and chemical propensities of the target protein with the actual conditions assigned to it by the structure model. These might include checking whether hydrophobic amino acids are found mostly in the inner parts of the molecule, whether charged residues are exposed to the solvent, whether secondary structure elements are formed by residues likely to be found in them and so on.

A straightforward method of performing this comparison presents itself again in sequence alignment. The primary structure of the target protein can be re-encoded to represent only aminoacid properties, not their exact types (e.g. both alanine and valine become "nonpolar residue"). Similarly, a string encoding environmental conditions imposed by the structure model being evaluated can be constructed from the model (e.g. residues on the external areas of the molecule get represented as "solvent exposed residue"). The environment profile and the property string need subsequently be optimally aligned, for which the already available tools like FASTA can be profitably used, albeit with a custom scoring matrix.

For example, the original profile method [ZE94] classed the environment imposed on each residue by the template into 18 types according to three properties: the buried sidechain area size, the fraction of the sidechain covered by polar atoms, the secondary structure type. The residues are divided into six classes according to the first two properties and a helix/sheet/coil type is assigned to each of them.

The alignment score of a residue and an environment condition encoded in the profile is derived from information values computed on well-refined protein structures. The preference of residue i to be found in a secondary structure j, with a buried area and polar fraction classes b and p respectively is defined as an information value $S_i(j,b,p)$:

$$S_i(j,b,p) = ln\left(\frac{P(i|j,b,p)}{P(i)}\right) = ln\left(\frac{P(i,j,b,p)}{P(i)P(j,b,p)}\right) \tag{3.12}$$

where P is the standardly defined conditional or joint probability.

A different approach is adopted by the so-called threading methods [JTT92]. In protein threading, the atomic model of the backbone of the evaluated structure is fitted with the sequence of the target protein and the quality of the resulting residue sidechain packing is estimated. This includes both calculations of sidechain collisions and free energy contributions. In order to avoid explicit simulation in the latter, knowledge-based and solvation

potentials are usually used.

A solvation potential is a function of residue accessibility defined in order to avoid explicit calculations of energy contributions of many solvent molecules. It can be computed by multiplying the difference in solvent accessible area before and after folding by an experimentally estimated solvation free energy per area constant. Several different sets of these constants exist and it is not obvious which are preferable. An alternative definition avoids empirical constants by analysing feature frequencies in structure databases. The knowledge-based version of a solvation potential for residue a is then defined as:

$$\Delta E^a_{solv}(r) = -RTln\left[\frac{f^a(r)}{f(r)}\right] \tag{3.13}$$

where r is the percent residue accessibility relative to the fully extended pentapeptide GGXGG, $f^a(r)$ the frequency of observing residue a with accessibility r and $f(r)$ the total occurrence of all residues with accessibility r.

3.2.3 Fold Prediction

Should it happen that sequence comparison methods fail to establish an evolutionary relationship with a protein of known structure and fold recognition proves unable to present a putative structure model, the target protein is very probably of previously unencountered structure and its model will have to be created de novo.

This is the youngest of protein structure prediction methods and the most difficult one. However, a successful solution would be especially useful even for the previously discussed methods – for example, in obtaining models of long divergent regions for comparative modeling approaches.

Current fold prediction methods, as exemplified by the Rosetta algorithm, approach the problem by breaking the protein sequence into short fragments, assigning a set of possible structure models to the fragments by searching structure databases for fragments of similar sequence and then selecting one structure from each set such that merging these structures together will produce an energetically reasonable protein conformation.

Unfortunately, because long range interactions have a considerable effect upon the final protein fold, we cannot simply define a mapping from local sequence fragments into structure fragments. Rosetta, for example, breaks the target sequence into fragments of nine residues – this length shows the greatest correlation between sequence and structure of all lengths less than fifteen residues – and then uses multiple alignment to select twenty-five most likely structure models. Rosetta measures the quality of the model as a reciprocal of distance defined as:

$$dist = \sum_i \sum_j S(j,i) - X(j,i) \tag{3.14}$$

where the first sum ranges over all positions in the fragment, the second sum ranges over all residue types and $S(j,i)$ resp. $X(j,i)$ give the frequency of occurrence of the residue j on position i in the multiple alignment of the target protein resp. of the template.

Selecting one fragment from each set of these twenty-five candidates may be performed according to the knowledge-based energy function or using Bayesian probability calculus:

$$P(structure|sequence) = P(structure)xP(sequence|structure) \tag{3.15}$$

where $P(sequence)$ does not figure in the equation because we are always searching for a fixed sequence and the probability is therefore 1. $P(structure)$ can be either set to $1/(allstructurescount)$ or to a custom function of its structural features (e.g. setting the probability higher for compact structure fragments because real proteins tend to be compact), Rosetta's approach is to set the probability to the radius of gyration of the fragment. The radius of gyration is defined as:

$$g = \left(\frac{\sum_i m_i r_i^2}{\sum_i m_i} \right)^2 \tag{3.16}$$

where the sums range over all rigid mass elements of the system, each of them of mass m_i located at a distance of r_i from the center of mass.

After merging the fragments into a backbone, simulated annealing is used to optimise the model by randomly selecting a different fragment's dihedral angle and substituting it for the one currently in the conformation. Both knowledge-based potentials and Bayesian logic are used to judge the quality of the conformation.

Chapter 4

Comparative Modeling

This chapter will detail the implementation of the components of a simple comparative modeling method. We will entirely ignore the problem of protein domains. Multidomain targets will be modeled as they are, that is by searching for a corresponding multidomain template. The search itself will be performed by the popular BLAST algorithm. BLAST's popularity is founded on its speed and a well developed statistical analysis of the results' significance. However, BLAST is not very sensitive and methods have been developed to increase the sensitivity of the search – for example the PSI-BLAST which uses BLAST to find close relatives and then combines information from their multiple alignment to construct a sequence profile. The profile is then used as a query to repeat the BLAST search, iteratively improving the profile. The downside is, with each iteration, some specificity of the search is sacrificed. We have felt that the sensitivity of the search is not really required for a typical comparative modeling task when close relatives are available in the database and the inclusion of PSI-BLAST would move the method more into the realm of fold recognition.

CLUSTALW is used for identification of the conserved cores by multiple alignment. CLUSTALW includes many improvements over CLUSTAL which allow the algorithm to produce better alignments at great evolutionary distances of the input sequences. This may not really benefit comparative modeling and the original CLUSTAL would have probably done just as well, for similar reasons as those for not including PSI-BLAST. We have nevertheless chosen CLUSTALW, as some of the improvements include more intelligent calculation of both residue dependent and residue independent gap penalties which might help in avoiding the placement of gaps in secondary structure elements.

For the task of aligning protein structures to provide additional support for distinguishing conserved protein cores, we have chosen to implement the K2 algorithm. K2 is based on evolutionary computation combined with rapid vector based prealignment of secondary structures. This serves two important ends. First, most of the time, the algorithm works with large backbone regions (the whole helices and sheets) vastly increasing the processing speed compared to a direct alignment on the level of residues. Second, the genetic algorithm which optimises the initial vector based prealignment begins with a reasonably optimal solution and the danger of falling into a locally optimal alignment is decreased.

Our implementation will remain limited to finding a structure template and identifying core regions withing which coordinates from the template may be simply assigning to the query. However, to illustrate the capabilities of comparative modeling at its best, we will use the module Modeller based on constraint satisfaction modeling to predict the structure of divergent regions and sidechains.

4.1 Database Search

In order to identify protein cores within the determined domain, we need to select homologous sequences from the sequence database. Ideally, we would perform an optimal Smith-Waterman alignment of the query with every database sequence to find out the degree of similarity between them. This is, however, not possible due to the size of sequence databases. Linear time heuristic solutions are usually employed instead – we will use the popular Basic Local Alignment Search Tool (BLAST) [AGM+90].

BLAST is based on the fact that statistically significant alignments often feature a high scoring segment pair – an ungapped aligned subsequence whose score cannot be increase by extending the segment pair in either direction. The algorithm identifies sequence homologs by evaluating the statistical significance of these high scoring segment pairs.

In the first phase, a list of all consecutive substrings of length w is build from the query sequence. A similar list is build for every sequence in the database. Obviously, online BLAST servers need to build the database list only once before processing queries, we will have to do the computation every program launch, which can take quite a long time for larger values of w. The starting positions of the segments are stored in a hash table indexed by the segment sequences.

Second, each segment of the query sequence is aligned (without gaps) against every possible segment. We are interested only in high scoring alignments and thus, select of all the possible segments only those that achieved a score above T given a chosen scoring matrix. For each database sequence, we look in the hash table, if the sequence contains one of these high scoring segments. If it does, we attempt to extend the alignment seed by extending the pair in both directions as long as the alignment score is increasing. We allow the score to decrease down to 75% of its previous highest value before terminating the process (in which case the segment is shrunk back to the length attaining the previous best score). This of course does not guarantee finding the true highest scoring segment pairs, however it is a very fast and reasonably accurate heuristic. The optimal solution would require resorting to quadratic time dynamic programming which would defeat the very purpose of BLAST.

The extension does not introduce gaps into the sequences, as this would require dynamic programming approach slowing the algorithm down. Also, the authors note that although gapped alignments increase the sensitivity slightly, specificity suffers from the modification, which makes it a doubtful improvement.

For evaluating the level of similarity between sequences we will only consider the found high scoring segment of the highest score and use the bit score [Xia07] to distinguish homologs from chance matches. The bit score bs is defined as:

$$bs = \frac{\lambda R - ln(K)}{ln(2)} \tag{4.1}$$

where R is the alignment score given a scoring matrix (we use Blosum62 by default). Tabular values for λ and K parameters will be used.

The bit score is compared with a cut off value $log_2(mn/E)$, where m and n are effective lengths of the query and the entire database respectively. E is a user specified critical E-value (we will use 0.01 by default). The effective lengths give the number of possible positionings of the alignment window within a sequence (or the sum of these numbers over all database sequences respectively). That is:

$$effective\_length(seq) = length(seq) - length(window) + 1 \tag{4.2}$$

If the bit score exceeds the cut off bit score, the sequence is reported as a match and its E value calculated to indicate the degree of significance.

$$E = mn2^{-bs} \tag{4.3}$$

To make the E value less cryptic, consider the problem of searching for an exact match of a segment of length s in a nucleotide sequence. The probability of finding a match is $0.25^s = 2^{-2s}$ assuming each nucleotide is equally likely in the database sequence. An s long match between two sequences can occur at m positions in one and n positions in the other (the effective lengths) giving an expected number of matches as $mn2^{-2s}$. This is exactly the form of the E value reported by BLAST, except we do not just take the length of an exact alignment but use the alignment score. The score is further normalised to take the statistical properties of the scoring system into account giving the bit score (plus, we are dealing with aminoacid sequences).

4.2 Multiple Alignment

In order to determine which residues belong to conserved protein cores, we will have to compute a multiple alignment of the sequences selected in the previous stage. This will be done using the popular CLUSTALW algorithm [THG94]. Determination of the optimal multiple alignment is computationally prohibitive as the dynamic programming method guaranteed to produce optimal results is a polynomial time algorithm of the order of the number of sequences to be aligned. CLUSTALW, on the other hand, is a progressive approximation algorithm which relies on the assumption that evolutionarily related sequences exhibit a very strong similarity. It is therefore possible to produce a very good alignment by progressive pair-wise alignments following the evolutionary tree[1] order – sequences and alignments are combined from the most closely related to the most distant. In consequence, the similar sequences first to be aligned will produce very precise results which will not be lost by the time more distant sequences are added to the alignment. This is especially true for gap placement, as this may be quite inaccurate for dissimilar sequences; attempting to align these in the opening stages of the algorithm would be disastrous and produce biologically meaningless results. This is a shortcoming of all progressive methods, their greedy nature makes it impossible to recover from early alignment mistakes. Thus, producing a good evolutionary tree to guide the algorithm is of critical importance. The general outline of the algorithm is then to first compute mutual similarity scores of all input sequences; second, to produce a guide tree from the scores and lastly, to produce the multiple alignment itself following the branching order of the guide tree.

The original CLUSTAL performed fast approximate pair-wise alignment by the k-tuple counting method. In addition to it, CLUSTALW supports optimal alignment by dynamic programming. This is the method we will use in our implementation. We will use affine gap penalties with CLUSTALW default values of 10.0 for gap opening and 0.1 for gap extension. Block substitution scoring matrices, computing from hand-made alignments of proteins of known evolutionary relationship, seem to produce better results compared to accepted mutations matrices in homology modeling tasks [HH92], therefore, we will use the Blosum family of matrices and arbitrarily choose the Blosum62 matrix for the pair-wise alignment stage. This matrix was calculated from protein blocks exhibiting over 60%

[1] Almost a phylogenetic tree, except phylogenetic trees are usually produced from complete multiple alignments. The tree computed by CLUSTALW will be much less exact and we will refer to it as a guide tree instead.

sequence similarity and it will serve well as a good compromise considering that comparative modeling is generally unsuitable for structure prediction of proteins with only poor relatives found in the sequence database anyway.

For each pair-wise alignment, the sequence distance is defined as a percent residue mismatch including gaps. That is, the distance between two sequences of length n and m with an optimal pair-wise alignment score s is:

$$distance = \frac{s}{max(n,m)} \tag{4.4}$$

The "phylogenetic" guide tree is constructed from these distances. Previously, CLUSTAL used the UPGMA algorithm for the guide tree construction, but that is based on the discredited molecular clock theory, CLUSTALW uses the Neighbour Joining algorithm instead [SN87]. With this method, we are able to predict different evolution rates in different tree branches, something which will come useful in the multiple alignment stage.

The Neighbour Joining algorithm is an iterative polynomial approximate[2] solution to constructing a tree with a minimal sum of branch lengths, as mentioned, far superior the the UPGMA method and less computationally prohibitive than better algorithms, e.g. methods based on maximum parsimony.

In each iteration, the Neighbour Joining algorithm searches for a node pair minimising a function M defined as:

$$M(i,j) = (n-2)D(i,j) - \sum_{k=1}^{n} D(i,k) - \sum_{l=1}^{n} D(l,k) \tag{4.5}$$

where $D(i,j)$ is the distance between sequences i and j found by the pair-wise alignments and n the total number of sequences.

These nodes are joined to a common parent in the tree and are removed from the set of unprocessed nodes, while the parent is added in their place. The distances between this newly added node and the remaining ones are calculated and the algorithm repeats the whole processes, this time with one less node to consider. The new distances are found as follows:

$$D(i,p) = \frac{1}{2}D(i,j) + \frac{1}{2(n-2)}\left[\sum_{k=1}^{n} D(i,k) - \sum_{l=1}^{n} D(j,l)\right] \tag{4.6}$$

$$D(p,x) = \frac{1}{2}[D(i,x) - D(i,p)] + \frac{1}{2}[D(j,x) - D(j,p)] \tag{4.7}$$

where i, j are the joined sequences, p their new parent and x an unrelated, still unprocessed, node.

The tree root is placed in the middle of the longest branch in the tree.

Following the branching of the guide tree, CLUSTALW progressively aligns ever more distant sequences/alignments proceeding from the leaves towards the root. Optimal dynamic programming is used for the individual alignments. The score of aligning two partial alignments is calculated by averaging the alignments of every pair of sequences where each of them belongs to a different input alignment. When introducing (or extending) new gaps, affine gap penalties are used just as in the first stage of the algorithm. For existing gaps in the partial alignments though, plain gap penalisation is used. That is, aligning any residue

[2] For an additive tree, the algorithm is optimal.

against a gap or two gaps against each other earns a zero score. For this reason, the scoring matrix is readjusted to have only non-negative values by subtracting the lowest score from each matrix value. In result, the gap score is the worst possible score.

CLUSTALW adds several enhancements to this simple scheme. Firstly, sequences are weighted to limit the consequences of unequal representation of different evolutionary distances in the alignment. Sets of very similar sequences are downweighted to offset the occurrence of nearly identical information multiple times in the data set, unevenly biasing the results. These weights are derived from the guide tree by adding branch lengths between the sequence and the root proportionally to the size of the subtree under each branch, then normalised so that the largest weight equals 1.0. See figure 4.1 for an example from the Higgins' paper. The weights act as multiplication factors when scoring each stage of the multiple alignment.

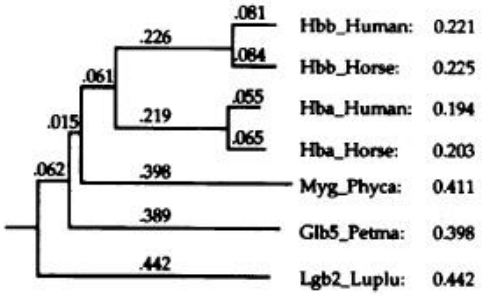


Figure 4.1: A guide tree constructed by the Neighbour Joining method with branch lengths and sequence weights displayed. The Lgb2_Luplu sequence gets a weight equal to the branch length to the root as it is the only sequence attached. On the other hand, the Hbb_Human sequence gets the sum of 0.081 as its unshared branch length, 0.226/2 because the branch is shared with another sequence, 0.061/4, 0.015/5 and 0.062/6 for similar reasons. The sum adds to a total of 0.221.

Secondly, CLUSTALW varies the gap penalties in a dynamic fashion according to several criteria. The gap opening penalty is modified depending on the severity of mismatch penalty of the scoring matrix and the similarity and length of the compared sequences. The gap extension penalty is increased for sequences of largely different length to inhibit formation of many long gaps. Formally:

$$GOP = (GOP_{Orig} + log(min(N, M))) * rm * is \quad (4.8)$$

$$GEP = GEP_{Orig} * (1.0 + |log(\frac{N}{M})|) \quad (4.9)$$

where GOP and GEP stand for gap opening, respectively extension penalties, GOP_{Orig} and GEP_{Orig} are default affine gap penalties (same as in the sequence distance determination case) and N, M are sequence lengths; rm is the residue mismatch factor, the average penalty for aligning different residues, i.e. the mean of non-diagonal values in the scoring matrix. The identity scaling factor, is, does not seem to play an important role in Conway Institute's CLUSTALW implementation. It is set to 0.75 for all sequences with more than 40% identity and only varies in less similar cases (0.5 for similarity in the range 30% to 40%, 0.6 otherwise), we will use this definition, although it is unlikely to affect homology modeling in any way other than scaling the gap opening penalty by a constant factor.

Thirdly, further gap penalty modifications are introduced in position-dependent manner. To encourage as little gaps in the multiple alignment as possible, the gap penalties are reduced for positions where a gap already occurs (and the next rules do not apply) – GEP is halved and the GOP is multiplied by 0.3 times the ratio of the number of gaps to the total number of sequences in the alignment.

Also, the gap opening penalty is increased 8 residues after the last gap as it has been observed from manual alignments of proteins of known structure that gaps do not tend to be closer than this. The adjustment factor is $(2 + (8 - d) * 2)/8$. Lastly, the gap opening penalty is reduced for consecutive runs of hydrophilic residues (by a third) as these indicate a probable loop or turn which are less evolutionarily conserved than protein cores; otherwise a residue specific modifier is applied according to observed relative frequency of gaps adjacent to different residue types, see table 4.1.

Table 4.1: Pascarella and Argos residue dependent gap factors [THG94]

Residue	**Factor**	**Residue**	**Factor**
A	1.13	M	1.29
C	1.13	N	0.63
D	0.96	P	0.74
E	1.31	Q	1.07
F	1.20	R	0.72
G	0.61	S	0.76
H	1.00	T	0.89
I	1.32	V	1.25
K	0.96	Y	1.00
L	1.21	W	1.23

Lastly, CLUSTALW varies the scoring matrix used depending on the evolutionary similarity of the sequences/partial alignments. We will take the similarity measure from the guide tree and switch between Blosum matrices in this way – 80%-100% similarity: Blosum80, 60%-80% similarity: Blosum62, 30%-60% similarity: Blosum45, otherwise: Blosum30.

4.3 Structural Alignment

To support the indication of protein cores given by the multiple sequence alignment, we will produce a structure alignment of the family members with experimental structures available. To build the alignment, we will implement K2, a hybrid genetic algorithm combined with a vector-based prealignment computation ([SW00], [SW03]). K2 is a pairwise structural alignment algorithm. It first aligns the most conserved portions of the two proteins, their individual secondary structure elements (SSEs). To rapidly determine a biologically meaningful initial alignment and spare the genetic algorithm the need of exploring a vast suboptimal space of solutions, the SSEs are modeled as vectors. We then search the set of these vectors for equivalent pairs (one in each structure) as "seeds" of the alignment. The resulting initial alignment is subsequently optimised by a genetic algorithm according to an elastic similarity score. Finally, the algorithm attempts to extend the alignment over the entire length of protein backbones, outside of the SSEs.

To sum up, the workings of K2 can be divided into three distinct stages. First, with the aid of DSSP [KS83], K2 determines the best alignment of protein SSEs represented as vectors. This operation can be performed relatively quickly and provides the genetic algorithm with a reasonably optimal initial population. Second then, a genetic algorithm optimises the alignment of constituent amino acids within the paired secondary structures. And finally, the protein backbones are searched for additional possible amino acid alignments. We will now describe the individual stages in detail.

Secondary structures will be extracted from the atomic models by DSSP. DSSP looks for hydrogen bonding patterns typical for each secondary structure type in the atomic model. Hydrogen bonds are predicted using an electrostatic model – partial charges q_1, q_2 are assumed at the C, O, respectively N, H atoms and the interaction energy between two potentially hydrogen bonding atoms is calculated as:

$$E = q_1 q_2 f \left(\frac{1}{r(O,N)} + \frac{1}{r(C,H)} - \frac{1}{r(O,H)} - \frac{1}{r(C,N)} \right) \tag{4.10}$$

where the two partial charges are set to $0.42e$ and $0.20e$, f is a geometry scaling factor equal to 332 and $r(X,Y)$ gives the distance between atoms X and Y. The hydrogen bond is predicted to form whenever the energy is lower than some cutoff value.

We will use the standalone DSSP implementation provided by the Nijmegen Centre for Molecular Life Sciences[3]. DSSP determines, among other things, what kind of, if any, secondary structure each amino acid in the protein belongs to. We subsequently smooth the DSSP output by accepting only α-helix and β-sheet type structures[4] of a certain consecutive length (4 residues). These structures are then modeled as vectors by simply considering their first and last α-carbon atom as endpoints, then normalised to unit length.

We now have to identify a set of equivalent vectors from the two structures. In each molecule, we choose a pair of vectors and use them to construct a coordinate system. The vectors form the x and y axis respectively, their cross-product becomes z. The system's point of origin is placed at the beginning of the SSE represented by the first vector of the pair. In this way, the coordinate systems define a sort of a spatial rotation and the moved origin a translation of the molecules. In case that we have chosen vector pairs in the two molecules that are actually equivalent to each other, the spatial transformation will be such, that many vector pairs will be found close together – giving a clue to a good superposition, we will use the alignment as an initial approximation, to be subsequently optimised.

That is, we construct a distance matrix D such that $D_{i,j}$ gives the distance between the centers of the i-th SSE in the first molecule and the j-th SSE in the second molecule. Their coordinates are each in a different basis and computing the distance is mathematically speaking not possible. However, for good prealignment choices, the bases should be quite similar and we will treat them as equal. Next, we solve the best-match problem on the matrix. The result is a set of assignments $A = \{(i,j)\}$ of the i-th SSE from the first molecule and j-th SSE from the second molecule such that i and j are both of the same type (helix or sheet) and $\sum_{(i,j)\in A} D_{i,j}$ is minimal. For the assignment of an equivalent element, we conduct a non-sequentially constrained search. That is, if we determine that $(i,j) \in A \wedge (k,l) \in A \wedge i < k$ we do not demand that also $j < l$ holds. Finally, vector pairs that are more distant than or form an angle greater than certain limit (10 Å and 90° respectively) are rejected. This process of constructing a coordinate system and solving the assignment problem is performed for every pair of vectors in each molecule and the

[3] http://swift.cmbi.kun.nl/gv/dssp/HTML/dsspcmbi

[4] DSSP distinguishes a total of nine different structure types.

best (according to the described minimalisation score) alignments are used to generate the initial population for the subsequent genetic algorithm stage.

To emphasise, searching for aligned pairs among every possible atom pair in the macromolecules for every possible spatial transformation of the system is too computationally expensive. Therefore, the genetic algorithm will try to optimise only the alignment of much larger blocks, the protein secondary structures. We will interpret aligned secondary structure blocks as to mean that residues belonging to one block are equivalent to residues belonging to the other - pairwise, beginning at the start of both blocks and continuing up to the end of the shorter of the pair.

The first stage of the algorithm leaves us with a set of equivalent secondary structures per coordinate system. We will construct the initial population by selecting the best aligned sets, randomly shifting the boundaries of the aligned SSEs by a small margin and copying the sets with a bias towards the better individuals. By default, we generate a population of 50 individuals. 40% of which are generated by making small random changes to the 5 best alignments, 30% are created from the next 5 best, 20% from the next 5 and the last 10% from the next quintuple, if enough alignments are predetermined. See figure 4.2 for an example of an individual from the referenced paper.

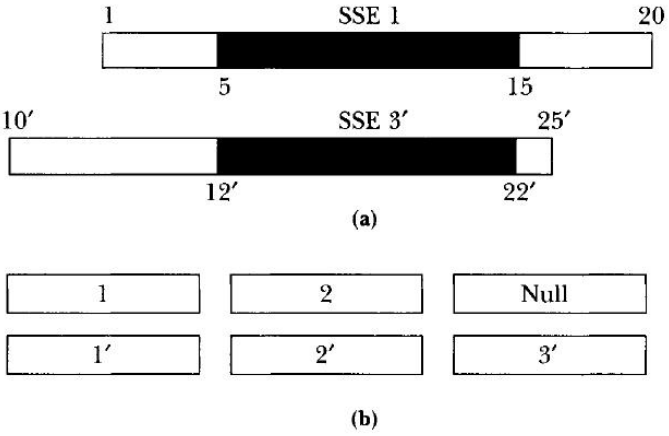


Figure 4.2: **a)** The first SSE of molecule 1 (residues 1 - 20) and the third SSE of molecule 2 (residues 10 - 25) have been shortened and now define residue equivalence between residues 5-15 and 12-22. **b)** An example of an encoded individual. The first two SSEs in both molecules correspond to (are aligned with) their same-numbered counterparts. The third SSE in molecule 2 is unaligned.

The population then undergoes evolution, which lasts until the average fitness of the population stops increasing or the 50th generation is reached, whichever comes first. In each generation, every individual has a random chance of undergoing two types of mutation. Afterwards, every pair of mutants randomly conducts one of two types of sexual reproduction. In both modifications, the individuals with the highest fitness survive - one in case of mutations, two in case of reproductions so the total population size stays constant.

The fitness function is based on the elastic similarity score. The rationale behind the function is founded on the presumption that the distance between two residues in a molecule should be very similar to the distance between residues in the other molecule equivalent to them. Formally, if the assignment set implies the residue equivalence $i \sim j \wedge k \sim l$ then $d^A_{i,k} - d^B_{j,l}$ should be minimal, $d^A_{i,j}$ denotes the distance between residue i and j in molecule

A. The fitness function is then a sum of scores for every residue. A score attained by residue i in molecule A is defined as:

$$\text{score} = \begin{cases} \sum_{k=1}^{L} \left(\theta - \frac{d^A_{i,k} - d^B_{j,l}}{d} \right) e^{-(\frac{d}{a})^2} & i \neq k \wedge i \sim j \wedge k \sim l \\ \theta & i = k \wedge i \sim j \wedge k \sim l \\ 0 & \text{otherwise} \end{cases}$$

where k ranges over all residues of molecule A, d is the average of $d^A_{i,k}$ and $d^B_{j,l}$, $\theta = 0.2$ and $a = 20$ Å. The first term in the function is meant to reward aligned residue pairs whose distances do not differ by more than 20% of their average distance, the second term merely introduces an exponential attenuation to the influence of distant residues on the contribution of the currently considered residue. Note that unaligned residues (residues not belonging to an aligned SSE) do not affect the fitness of the individual at all.

Of the two mutation operators, the first behaves exactly as the procedure which we used to introduce random changes into the initial population. It simply shifts the boundaries of every SSE in an individual by a small random margin, making sure no two SSEs ever overlap. The second one, the hop operator, chooses an aligned SSE with the least contribution towards the final fitness score and exchanges it with a random other SSE (or the NULL SSE, effectively unaligning the element or introducing a gap in the alignment). This operator is introduced in case the first phase of the algorithm missed some optimal alignments or erroneously introduced suboptimal ones. Both operators are applied with a 15% probability.

The reproduction operators used in the algorithm are the swap and the crossover operators. In the swap case, all aligned SSEs of one type (helix or sheet) in the first parent are swapped with all aligned SSEs of the other type (sheet or helix respectively) in the second parent, unaligning all conflicting SSEs in the process. The idea is to improve the situation in which one of the parents evolved to have well aligned α-helices while the other possesses well aligned β-sheets. The crossover operator is very similar. SSEs in an alignment are ordered, a random point is chosen in the ordering and all aligned structures lesser than the point in one parent are exchanged with structures greater than the selected crossover point in the second parent, irrespective of structure type but again making sure to unalign all conflicting SSEs. Exactly one of the reproduction operators is applied to every pair of mutants each generation. Which one is determined randomly with the same 50% probability for both of them.

After the evolution finishes, the best individual undergoes final modifications. In case any aligned secondary structures with negative contribution to the fitness score survived, they are removed now. The remaining aligned SSE pairs are clipped to be of equal length and the individual residue equivalencies are determined. No residue pair is allowed to the result set if formed by amino acids more than 5 Å apart or not forming a consecutive run of at least length 3. Finally, the entire backbone is searched for additional equivalent pairs. A pair is deemed equivalent if the residues are each other's nearest neighbour and satisfy the maximal distance requirement.

4.4 Modeling Divergent Regions and Sidechains

For this task, we will make use of an existing implementation of the Sali lab's Modeller – a constraint satisfaction based modeling library [SB93].

Chapter 5

Experiments And Analysis

This chapter will assess the properties of our implementation and perform a demonstration of the structure prediction capabilities offered by comparative modeling.

5.1 BLAST

BLAST is a straightforward algorithm, however, evaluating the statistical significance of hits is a different matter altogether. One serious limitation of our implementation in this respect is the lack of ability to mask regions of low complexity [Jus05] in the amino acid sequence. Low complexity regions are various ranges of biased amino acid composition, pattern repeats and amino acid over-representations. They defy the statistical assumptions behind BLAST and similar algorithms. We would assume that the observed frequency of a segment of a given composition in a database reflects the possible number of such segments. Formally, the frequency of a segment of length L is expected to be:

$$F = \frac{L!}{\Pi a_i!} \tag{5.1}$$

when the segment is composed of a_i residues of the type i.

Naturally, we would expect the relative frequency of two segments of the same length to be a function of their composition a_i, b_i given by:

$$f = \frac{\Pi a_i!}{\Pi b_i!} \tag{5.2}$$

This expectation breaks down for low complexity segments – they are much more abundant in the database than the statistics suggest they should be. The approach adopted by BLAST is to mask such regions in the query sequence by overwriting their residues with the 'x' symbol, which is ignored by BLAST. Wootton's program SEG performs the filtering operation. Unfortunately, neither the paper describing SEG nor a formal description of a low complexity segment is freely available and we have not treated the low complexity regions in any way in our implementation. This will produce spurious hits and wrongly deflate the E-values for hits produced on queries with low complexity regions.

To illustrate the possible effect, we have performed a BLAST search with the NCBI implementation of the Zein-alpha PZ19.1 protein containing the low complexity region "LPLSPLFLQQSSALLQQLPLVHLLAQ". Without low-complexity filtering, 48 hits are found against the nr database with E-values below 1×10^{-55}. In contrast, after applying SEG to the query, only 4 hits are produced. We can expect similarly inflated results from our implementation for low complexity queries.

5.2 CLUSTALW

The improvements introduced into CLUSTALW compared to CLUSTAL are rather amazing, especially those concerned with position specific gap penalties. CLUSTALW is now able to produce high quality multiple alignments even for cases with very low sequence similarity over the entire set of sequences to be aligned. We will demonstrate the qualities of CLUSTALW by aligning a random 20% selection of SH2 domains [LJR+06] from all currently known proteins containing them. We have chosen SH2 because it is well studied and a large body of human curated information is available for the domain – including a published manually revised CLUSTALW reference multiple alignment[1]. The alignment produced by our implementation is shown below.

```
SOCS2_HUMAN     |  YW------GSMTV-NEA----KEKLKEAP---EGTFLIRDSSHSDYL---------LTISVKTSAGPTNLRIEYQDGK-------FRLDSIIC---VKSKLKQFDSVVHLIDY--YV
SOCS2_HUMAN     |  YW------GSMTV-NEA----KEKLKEAP---EGTFLIRDSSHSDYL---------LTISVKTSAGPTNLRIEYQDGK-------FRLDSIIC---VKSKLKQFDSVVHLIDY--YV
CISH_RAT        |  YW------GSITA-SEA----RQHLQKMP---EGTFLVRDSTHPSYL---------FTLSVKTTRGPTNVRIEYADSS-------FRLDSNCL---SRPRILAFPDVVSLVQH--YV
SOCS3_RAT       |  YW------SAVTG-GEA----NLLLSAEP---AGTFLIRDSSDQRHF---------FTLSVETQSGTKNLRIQCEGGS-------FSLQSDPR---STQPVPRFDCVLKLVHH--YM
SOCS3_HUMAN     |  YW------SAVTG-GEA----NLLLSAEP---AGTFLIRDSSDQRHF---------FTLSVKTQSGTKNLRIQCEGGS-------FSLQSDPR---STQPVPRFDCVLKLVHH--YM
SOCS1_HUMAN     |  RI------TRASA-LLD----ACGFYWGP---LSVHGAHERLRAEPV---------GTFLVRDSRQ------RNCFFA-------LSVKMASG---PTSIRVHFQAGRFHLDG--SR
STAT2_HUMAN     |  MG------F-VSR-SQE----RRLLKKTM---SGTFLLRFSESSEGG-----------------------------------------------------------ITCSW--VE
STAT2_PIG       |  IM------GFVSR-SEE----RRLLKKTI---SGTFLLRFSETLEGG-----------------------------------------------------------ITCSW--VE
STAT3_MOUSE     |  IM------GFISK-ERE----RAILSTKP---PGTFLLRFSESSKEGGVTFTWVEKDISGKTQIQSVEPYTKQQLNNM-------SFAEIIMG---YKIMDATNILVSPLVYL--YP
STAT3_HUMAN     |  IM------GFISK-ERE----RAILSTKP---PGTFLLRFSESSKEGGVTFTWVEKDISGKTQIQSVEPYTKQQLNNM-------SFAEIIMG---YKIMDATNILVSPLVYL--YP
STA5A_MOUSE     |  ND------GAILG-FVN----KQQAHDLL---INKPDGTFLLRFSDS---------EIGGITIAWKFDSPDRNLWNLK-------PFTTRDFS---IRSLADRLGDLNYLIYV--FP
STA5A_BOVIN     |  ND------GAILG-FVN----KQQAHDLL---INKPDGTFLLRFSDS---------EIGGITIAWKFDSPDRNLWNLK-------PFTTRDFS---IRSLADRLGDLNYLIYV--FP
STA5A_SHEEP     |  ND------GAILG-FVN----KQQAHDLL---INKPDGTFLLRFSDS---------EIGGITIAWKFDSPDRNLWNLK-------PFTTREGS---IRSLADRLGDLNYLIYV--FP
O93378_CHICK    |  ND------GAILG-FVN----KQQAHDLL---ISKPDGTFLLRFSDS---------EIGG-----------------------------------------------ITIAWKFD--SA
Q23554_CAEEL    |  YH------GLLPR-ADI----NTLLENDG---DFLVRTSHIVGQDSA---------KTVLS-----VKWK-GKCHHWQ-------LQEKEDGS---IVIEERKFESVLDMVTT--LR
FES_FELCA       |  YH------GALPR-AEV----AELLTHSG---DFLVRES----QGKQ---------EYVLS-----VLWD-GQPRHFI-------IESADNL----YRLEGDGFASIPLLVDH--LL
Q18142_CAEEL    |  YH------GLLPR-EDI----KAMLRKNG---DFLVRSTEPKAGEPR---------QYVLSAMQSEELED-AGVKHYV-------MRLNPSNQ---IFLEAKGFETIASLVNY--YM
SHIP1_HUMAN     |  YR------NCVY---TY----KFTVQASEGVSMRFF---------TKLDQLIEYFKKENMGLVTHLQ-------YPVPLEEE---DTGDDPEEDTVESVVSPPELP
ITK_HUMAN       |  YN------KSISR-DKA----EKLLLDTG-K-EGAFMVR--DSRTAGT--------YTVSVFTKAVVSENNPCIKHYH-------IKETNDNPKRYYVAEKYVFDSIPLLINY--HQ
BMX_HUMAN       |  FA------GNISR-SQS----EQLLRQKG-K-EGAFMVR--NSSQVGM--------YTVSLFSKAVNDKKG-TVKHYH-------VHTNAEN--KLYLAENYCFDSIPKLIHY--HQ
BTK_MOUSE       |  YS------KHMTR-SQA----EQLLKQEG-K-EGGFIVR--DSSKAGK--------YTVSVFAKSTGEPQG-VIRHYV-------VCSTPQS--QYYLAEKHLFSTIPELINY--HQ
TXK_MOUSE       |  YH------KNITR-NQT----ERLLRQEA-K-EGAFIVR--DSRHLGS--------YTISVFTRARRHTQS-SIKHYQ-------IKKNDSG--QWYITERHLFPSVPELIQY--HQ
Q24284_DROME    |  FH------PNTTK-EQA----EQGLYRLE---IGSFLVR--PS----V--------QSINAFVISFTINRK--IKHCR-------IMQEGCL----YGIDTMNFESLVSLINY--YT
VAV2_MOUSE      |  FA------GNMER--QQ----TDNLLKSH-A-SGTYLIRERPAEAER---------FAISIKFNDEVKHIK-VVEKDS-------WIHITEAK---------KFESLLELVEY--YQ
SPK1_DUGTI      |  AW------REIQR-WEA----EKSLMKIGLQ-KGTYIIR--PSRKENS--------YALSVRDFDEKKKIC-IVKHFQ-------IKTLQDEKGISYSVNIRNFPNILTLIQF--YE
KSYK_MOUSE      |  FF------GNITR-EEA----EDYLVQGGMT-DGLYLLRQ-SRNYLGG--------FALSVAHNR----KA---HHYT-------IERELNGT-YAISG-GRAHASPADLCHY--HS
RASA1_HUMAN     |  YH------GKLDR-TIA----EERLRQAG-K-SGSYLIRE-SDRRPGS---------FVLSFLSQM----NV---VNHF-------RIIAMCGD-YYIGG--RRFSSLSDLIGY--YS
JAK2_HUMAN      |  GP------ISMDF---A----ISKLKKAGNQ-TGLYVLRC-SPKDFNK--------YFLTFAVERENVIEY---KHCL-------ITKNENEE-YNLSGTKKNFSSLKDLLNC--YQ
PTN11_MOUSE     |  FH------PNITG-VEA----ENLLLTRG-V-DGSFLAR--PSKSNPG---------DFTLSVRRN---GA--VTHIK-------IQNTGDY---YDLYGGEKFATLAELVQY--YM
PTN11_MOUSE     |  FH------PNITG-VEA----ENLLLTRG-V-DGSFLAR--PSKSNPG---------DFTLSVRRN---GA--VTHIK-------IQNTGDY---YDLYGGEKFATLAELVQY--YM
GRAP2_MOUSE     |  FH------EGLSR-HQA----ENLLMGKD---IGFFIIR--ASQSSPG---------DFSISVRHE---DD--VQHFK-------VMRDTKG---NYFLWTEKFPSLNKLVDY--YR
O02064_CAEEL    |  YH------GIIPR-VDA----ELSLRYVG-D-YLLRRAE--PRENAGG--------QFLIISVKRE---AG--FVHVP-------ISQDKEGAKQFYFMPTLKENTVLDLIDV--HL
ZAP70_HUMAN     |  YH------SSLTR-EEA----ERKLYSGAQT-DGKFLLRP-RK-EQGT---------YALSLIYGKTVYHYLISQDK---------------AGKYCIPEGTKFDTLWQLVEY--LK
KSYK_HUMAN      |  FH------GKISR-EES----EQIVLIGSKT-NGKFLIRA-RD-NNGS--------YALCLLHEGKVLHYRIDKDK---------------TGKLSIPEGKKFDTLWQLVEH--YS
O77440_HYDAT    |  FH------GVIPR-TEA----ERRLKSLSET--GMFLIRE-RGSPRGS--------YVIGLFYQGGVYHYLFESNN---------------QGQLSIKSGRSFDNLMAVVNF--YS
CSK_RAT         |  FH------GKITR-EQA----ERLLYPPETG---LFLVRE-STNYPGD--------YTLCVSCEGKVEHYRIMYHA---------------SK-LSIDEEVYFENLMQLVEH--YT
CSK_MOUSE       |  FH------GKITR-EQA----ERLLYPPETG---LFLVRE-STNYPGD--------YTLCVSCEGKVEHYRIMYHA---------------SK-LSIDEEVYFENLMQLVEH--YT
Q64103_9MURI    |  FH------GKISG-QEA----IQQLQPPEDG---LFLVRE-SARHPGD--------YVLCVSFGRDVIHYRVLHRD---------------GH-LTIDEAVCFCNLMDMVEH--YT
P70223_MOUSE    |  FH------GKISG-QEA----IQQLQPPEDG---LFLVRE-SARHPGD--------YVLCVSFGRDVIHYRVLHRD---------------GH-LTIDEAVCFCNLMDMVEH--YT
CRK_MOUSE       |  YW------GRLSR-QEA----VALLQGQRHG---VFLVRD-SSTSPGD--------YVLSVSENSRVSHYIINSSGPRPPVPPSPAQPPPGVSPSRLRIGDQEFDSLPALLEF--YK
CRKL_HUMAN      |  YM------GPVSR-QEA----QTRLQGQRHG---MFLVRD-SSTCPGD--------YVLSVSENSRVSHYIINSLP----------------NRRFKIGDQEFDHLPALLEF--YK
GRB10_HUMAN     |  HRTQHWFHGRISR-EES----HRIIKQQGLV-DGLFLLRD-SQSNPKA--------FVLTLCHHQKIKNFQ-ILPCED----------DGQTFFSLDDGNTKFSDLIQ--------
GRB14_HUMAN     |  FH------HKISR-DEA----QRLIIQQGLV-DGVFLVRD-SQSNPKT--------FVLSMSHGQKIKHFQ-IIPVED----------DGEMFHTLDDGHTRFTDLIQLVEF--YQ
SH2B1_RAT       |  FH------GMLSR-LKA----AQLVLEGGTSSHGVFLVRQ-SETRRGE--------YVLTFNFQGKAKHLR-LSLNEE----------GQCRVQHLWFQSIFDMLEHF-------R
SH2B2_HUMAN     |  FH------GTLSR-VKA----AQLVLAGGPRNHGLFVIRQ-SETRPGE--------YVLTFNFQG-----------------------------------------KAKH-------L
Q24218_DROME    |  YY------GAITR-SQC----DTVLNGHGHD--GDFLIRD-SETNMGD--------YSVSLKAPG-------RNKHFR----------VHVEQNMYCIGQRKFHSLDQLVDH--YQ
SHARK_DROME     |  YH------GNLSR-EAA----DELLKQGYED--GTFLVRE-SSTAAGD--------FVLSLLCQGEVCHYQVRRHGGE----------DAFFSIDDKVQTKILHGLDTLVDY--YQ
SH21A_HUMAN     |  HG------K-ISR--ET----GEKLLLATGL-DGSYLLRDSESVPGV---------YCLCVLYHGYIYTYR-------------------------------VSQTET--GS
SHIP2_HUMAN     |  YH------RDLSR--AA----AEELLARAGR-DGSFLVRDSESVAGA---------FALCVLYQKHVHTYRILPDGED----------FLAVQTSQGVPVRRFQTLGELIGL--YA
ABL1_MOUSE      |  YH------G-PVS--RN----AAEYLLSSGI-NGSFLVRESESSPGQ---------RSISLRYEGRVYHYRINTASDG----------KLYV-----SSESRFNTLAELVHH--HS
ABL1_HUMAN      |  YH------G-PVS--RN----AAEYLLSSGI-NGSFLVRESESSPGQ---------RSISLRYEGRVYHYRINTASDG----------KLYV-----SSESRFNTLAELVHH--HS
ABL1_HUMAN      |  GS------F-LVR--ES----ESSPGQRSIS-LRYEGRVYHYRINTA---------SDGKLYVSSESRFNTLAELVHH----------HSTV-----ADGLITTLHYPAPKR--NK
FRK_RAT         |  FF------GAIKR-ADA----EKQLLYSENQ-TGAFLIRESESQKGD---------FSLSVLD-----EGVVKHYRIR----------RLDEGGFFLTRRKTFSTLNEFVNY--YT
FRK_HUMAN       |  FF------GAIGR-SDA----EKQLLYSENK-TGSFLIRESESQKGE---------FSLSVLD-----GAVVKHYRIK----------RLDEGGFFLTRRIFSTLNEFVSH--YT
SRC42_DROME     |  YF------RKIKR-IEA----EKKLLLPENE-HGAFLIRDSESRHND---------YSLSVRD-----GDTVKHYRIR----------QLDEGGFFIARRTTFRTLQELVEH--YS
SRK4_SPOLA      |  FF------GQVKR-VDA----EKQLMMPFNN-LGSFLIRDSDTTPGD---------FSLSVRD-----IDRVRHYRIK----------KLENGTYFVTRRLTFQSIQELVAY--YT
LCK_HUMAN       |  WF------FKNLSRKDA----ERQLLAPGNT-HGSFLIRESESTAGS---------FSLSVRDFDQNQGEVVKHYKIR----------NLDNGGFYISPRITFPGLHELVRH--Y-
BLK_HUMAN       |  WF------FRSQGRKEA----ERQLLAPINK-AGSFLIRESETNKGA---------FSLSVKDVTTQ-GELIKHYKIR----------CLDEGGYYISPRITFPSLQALVQH--Y-
HCK_RAT         |  KE------GYIPS-NYV----ARVNSLETEE-WFFKGISRKDAERHL---------LAPGNMLGSFMIRDSETTKGSY----------SLSVRDFDPQHGDTVKHYKIRTLD--SG
HCK_MOUSE       |  FF------KGISR-KDA----ERHLLAPGNM-LGSFMIRDSETTKGS---------YSLSVRDFDPQHGDTVKHYKIR----------TLDSGGFYISPRSTFSSLQELVLH--YK
FGR_HUMAN       |  YF------GKIGR-KDA----ERQLLSPGNP-QGAFLIRESETTKGA---------YSLSIRDWDQTRGDHVKHYKIR----------KLDMGGYYITTRVQFNSVQELVQH--YM
FGR_FSVGR       |  YF------GKIGR-KDA----ERQLLSPGNA-RGAFLVRESETTKGA---------YSLSIRDWDEARGDHVKHYKIR----------KLDTGGYYITTRAQFNSVQELVQH--YV
FYN_RAT         |  YF------GKLGR-KDA----ERQLLSFGNP-RGTFLIRESETTKGA---------YSLSIRDWDDMKGDHVKHYKIR----------KLDNGGYYITTRAQFETLQQLVQH--YS
YES_HUMAN       |  YF------GKMGR-KDA----ERLLLNPGNQ-RGIFLVRESETTKGA---------YSLSIRDWDEIRGDNVKHYKIR----------KLDNGGYYITTRAQFDTLQKLVKH--YT
YES_AVISY       |  YF------GKMGR-KDA----ERLLLNPGNQ-RGIFLVRESETTKGA---------YSLSIRDWDEVRGDNVKHYKIR----------KLDNGG-YYITTRAQFESLQKLVK--HS
SRC_RSVP        |  YF------GKITR-RES----ERLLLNPENP-RGTFLVRKSETAKGA---------YCLSVSDFDNAKGPNVKHYKIY----------KLYSGGFYITSRTQFGSLQQLVAY--YS
SRC_RSVH1       |  YF------GKITR-RES----ERLLLNPENP-RGTFLVRKSETAKGA---------YCLSVSDFDNAKGPNVKHYKIC----------KLYSGGFYITSRTQFGSLQQLVAY--YS
Q85730_9RETR    |  YF------GKITR-RES----ERLLLNPEN-----------------------------------PRGTFLVRKSE----------TTKGAYCLSVSDFDNAKGPNVKHY--KI
```

[1] You can review it here: http://sh2.uchicago.edu/download/SH2_Alignment.pdf

```
SRC_AVIS2       |  YF------GKITR-RES----ERLLLNPENP-RGTFLVRESETTKGA---------YCLSVSDFDNAKGLNVKHYKIR-----------KLDSGGFYITSRTQFSSLQQLVAY--YS
Q86363_9RETR    |  YF------GKITR-RES----GRLLLNPENP-RGTFLVRESETTKGA---------YCLSVSDFDNAKGLNVKHYKIR-----------KLDSGGFYITSRTQFSSLQQLVAY--YS
Q64817_AVIS2    |  YF------GKTTR-RES----ERLLLNPENP-RGTFLVRESETTKGA---------YCLSVSDFDNAKGLNVKHYKIR-----------KLDSGGFYITSRTQFSSLQQLVAY--YS
P85B_MOUSE      |  YV------GKINRT-------QAEEMLSGKR-DGTFLIRESSQRGC----------YACSVVVDGDTKHCVIYRTATG--------------FG--FAEPYNLYGSLKELVLH--YQ
O18683_DROME    |  LL------KDAKRR-------NAEEMLKGAP-SGTFLIRARD-AGH----------YALSIACKNIVQHCLIYETSTG--------------FG--FAAPYNIYATLKSLVEH--YA
PLCG2_RAT       |  FH------KKVESRTSAEKLLQEYCAETGAK-DGTFLVRESETFPND---------YTLSFWRSGRVQHCRIRSTMEG-----------GVMKY--YLTDNLTFNSIYALIQH--YR
P55G_BOVIN      |  YW------GDISRE-------EVNDKLRDMP-DGTFLVRDASTKMQGD--------YTLTLRKGGNNKLIKIYHRDGK--------------YG--FSDPLT-FNSVVELISH--YH
```

Note the formation of several distinct blocks (corresponding roughly to the actual secondary structure elements) and the positioning of gaps mainly in between them (the loop regions). This effect disappears when the position specific gap penalties are not calculated in the algorithm, producing a biologically meaningless result.

Several obvious errors are visible in the alignment, for example the second gap opening in the STAT2_HUMAN sequence instead of moving the phenylalanine one position to the right. But these are easily corrected by manual inspection.

5.3 K2

The K2 structure alignment algorithm produces rather mixed results and we were not able to duplicate the authors' excellent alignments reported in the cited paper with our implementation. Unfortunately, the authors do not provide a standalone executable program and the alignment server using K2 has been removed and replaced with K2's successor which is based on simulated annealing instead of genetic algorithms. Therefore, we will assess our implementation against the Boston University's algorithm FAST[2] as a reference.

Our implementation seems to have serious problems with getting stuck in very suboptimal local optima. This is especially noticeable when aligning strongly dissimilar proteins. The sequentially unconstrained alignment makes matters worse by allowing the genetic algorithm to find many similar regions dispersed all over the molecule. This might be desirable for methods looking for similar structural fragments but it is definitely not desirable for core identification purposes.

The situation can be improved by using a much larger population and higher mutation rates in the evolution process and running the algorithm several times, then selecting the result with the highest overall fitness for the final alignment. This however increases the program run time substantially.

We will demonstrate that our version of K2 can nevertheless produce good alignments by aligning two similar structures of the Tudor domain – 1MHN and 1GV5 PDB identifiers. FAST produces a single contiguous aligned region including the residues 2-55 of the first molecule and residues 3-56 of the second molecule. K2 aligns residues 2-15 and 17-42 in the first molecule with residues 0-13 and 14-46 in the second molecule.

The first difference is the gap in the alignment, unpairing residue 16 in the first molecule. This happens because there are sheet strands in the protein formed by residues 8-12 and 19-28 which form secondary structure elements on which the genetic algorithm operates. When they are being extended, their ends meet first in the second molecule, one residue short of meeting in the first molecule also. The algorithm does not seem to be able to recover from such a situation.

Secondly, the alignment is shifted by three residues in the second molecule. This might be a reasonable alignment and is anyway not such a big problem for core identification purposes. Worse however, the aligned portions of the molecules are not extended as much towards the C-terminus as in the reference alignment by FAST, falling short by some ten residues.

[2] http://biowulf.bu.edu/FAST/

5.4 Modeller

Our approach to the structure prediction task will consist of these steps. First, we will BLAST the query against the entire nr database, the most complete sequence database accessible through NCBI. The program displays found hits and lets the user input an upper limit on the E value for sequences to be considered in the multiple alignment phase. Next, we BLAST the query against the PDB database and display the results. The user is requested to identify the best hit for a template. This will most probably be the structure with the lowest E valued sequence, however other criteria like model resolution or R-factor for X-ray crystallography models might play a role.

In the next step, the multiple alignment produced by CLUSTALW is displayed along with the structure alignments of the template with other found structures with corresponding sequences of low E values, if applicable. Additionally, the multiple alignment is converted to the PIR format usable by Modeller. The user can inspect the displayed information and modify the PIR alignment in case it contains obvious errors. Finally, Modeller [S+09] is used to complete the model construction by predicting structures for divergent regions and sidechains.

As a first example, we will choose a particularly easy task. We will predict the structure of the third chain of the bovine cytochrome BC1 complex (PDB identifier 1BGY). The nearest hit with a structure found by BLAST is the 1QCR PDB record with its third chain practically identical to the third chain of the query. The multiple alignment is not problematic and Modeller produces an accurate model with 1.1 Å RMSD from the experimentally determined structure. That is a resolution comparable to X-ray crystallography models. You can see the superposition of the predicted and experimental models on figure 5.1.

The accuracy of comparative modeling decreases with increasing evolutionary distance between the query and the template, the limit of usefulness being somewhere around 30%. For example, let us pretend that no better matches are found for the cytochrome BC1 than the first chain of the protein complex 2QJP, containing a portion of cytochrome B from rhodobacter sphaeroides at approximately 50% sequence identity with the query. The structure prediction program is still able to provide a model with 1.8 Å RMSD from the experimental structure, by all means an excellent result.

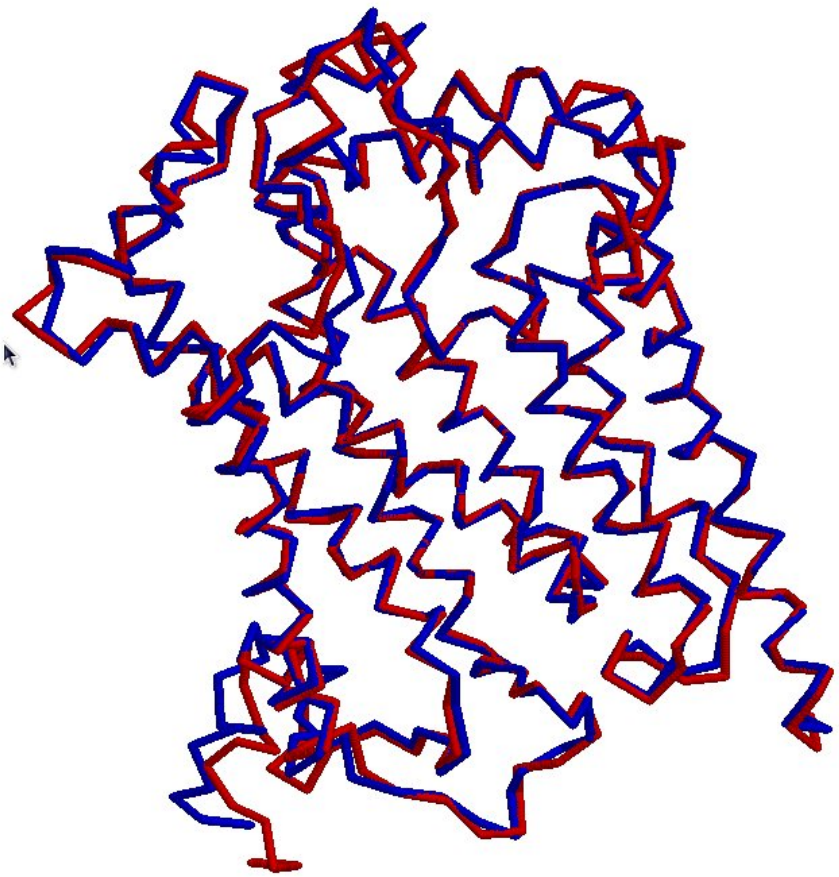


Figure 5.1: Cytochrome BC1 complex – experimental (blue) and predicted (magenta) models displayed by Rasmol after an optimal superposition by FAST.

Conclusion

We have described the fundamentals of protein structure, structure classification and prediction. We hope the inclusion and the extent of the first chapter on largely biochemical matters was not meaningless. Understanding these concepts is often useful for understanding the structure prediction algorithms; crucial for understanding the various optimisations made to them. Biology is not a science with laws but rules – each of them with known exceptions of various levels of frequency. These can profitably be incorporated into the prediction methods and whenever it is the case, it is now, hopefully, obvious why.

The second topic, structure classification, is not so crucial for understanding the prediction methods as it is useful in quickly exploring all currently known fold types and easily retrieving a group of proteins structurally similar to the one being examined. Likewise, understanding the methods of experimental structure determination and databases of their results can help us in working with the already known protein structures e.g. expecting the NMR structures to contain multiple models.

The main focus of the work was then on structure prediction methods themselves, mainly on comparative modeling. We have implemented a simple comparative modeler, making use of classical algorithms like CLUSTAL and BLAST, cornerstones of bioinformatics. We have found them, as expected, to perform admirably. While the structure alignment algorithm K2, or at least our implementation, was found to be somewhat lacking.

Two main directions of extending the work are obvious. First, implementing the functionality offered by Modeller, that is divergent regions and sidechains modeling, either by constraint satisfaction means or otherwise. Second, exploring the possibilities of optimising the produced model – for example, on energetic considerations. This is a difficult problem, both computationally and theoretically. Our current force fields are not accurate enough to perform positive changes to the model, even experimental structures would be perturbed by their application. This makes approaches like molecular dynamics simulation very challenging. Different methods appear more promising however, e.g. limiting the energetic optimisation only along principal components of structure variation. These might be interesting to explore further.

Bibliography

[AGM+90] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[BMRW01] Oren M. Becker, Alexander D. Jr. MacKerell, Benoit Roux, and Masakatsu Watanabe, editors. *Computational Biochemistry And Biophysics*. Marcel Dekker, New York, 2001.

[CB00] Peter Clote and Rolf Backofen. *Computational Molecular Biology: An Introduction*. Wiley, 1st edition, September 2000.

[CCF+08] Jeff Chang, Brad Chapman, Iddo Friedberg, Thomas Hamelryck, Michiel de Hoon, Peter Cock, and Tiago Antao. *Biopython Tutorial and Cookbook*, November 2008.

[Cha03] D. Chasman. Comparative protein structure modeling. In *Protein Structure*, pages 167–206. Marcel Dekker, 2003.

[CSL+08] Alison L. Cuff, Ian Sillitoe, Tony Lewis, Oliver C. Redfern, Richard Garratt, Janet Thornton, and Christine A. Orengo. The cath classification revisited-architectures reviewed and new ways to characterize structural divergence in superfamilies. *Nucleic Acids Research*, 37:310–314, November 2008.

[DBY+95] Ken A. Dill, Sarina Bromberg, Kaizhi Yue, Klaus M. Fiebig, David P. Yee, Paul D. Thomas, and Hue Sun Chan. Principles of protein folding - a perspective from simple exact models. *Protein Science*, (4):561–602, 1995.

[Gin06] Krzysztof Ginalski. Comparative modeling for protein structure prediction. *Current Opinion in Structural Biology*, 16:172–177, 2006.

[HH92] Steven Henikoff and Jorja G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of The National Academy of Sciences of The United States of America*, 89:10915–10919, November 1992.

[HI96] William E. Hart and Sorin Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal. *Journal of Computational Biology*, 3:157–168, 1996.

[HKRS08] L. Holm, S. Kaariainen, P. Rosenstrom, and A. Schenkel. Searching protein structure databases with dalilite v.3. *Structural bioinformatics*, 24(23):2780–2781, June 2008.

[Hun93] Lawrence Hunter, editor. *Artificial Intelligence And Molecular Biology*. AAAI Press, 1993.

[Isa06] Alexander Isaev. *Introduction to Mathematical Methods in Bioinformatics*. Springer-Verlag, Heidelberg, 2nd edition, 2006.

[JTT92] D. T. Jones, W. R. Taylor, and J. M. Thornton. A new approach to protein fold recognition. *Nature*, 358:86–89, July 1992.

[Jus05] Winfried Just. Low complexity regions. Lecture notes available at: http://www.math.ohiou.edu/~just/bioinfo05/, 2005.

[KS83] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.

[Les01] Arthur M. Lesk. *Introduction To Protein Architecture: The Structural Biology of Proteins*. Oxford University Press, 1st edition, 2001.

[LJR+06] Bernard A. Lie, Karl Jablonowski, Monica Raina, Michael Arce, Tony Pawson, and Piers D. Nash. The human and mouse complement of sh2 domain proteins - establishing the boundaries of phosphotyrosine signaling. *Molecular Cell*, 22:851–868, June 2006.

[MBHC95] Alexey G. Murzin, Steven E. Brenner, Tim Hubbard, and Cyrus Chothia. Scop: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247(4):536–540, April 1995.

[S+09] Andrej Sali et al. Modeller: A program for protein structure modeling release 9v6, r6593. Manual available from: http://salilab.org/modeller/9v6/manual.pdf, 2009.

[SB93] Andrej Sali and Tom L. Blundell. Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology*, 234:779–815, 1993.

[SN87] Naruya Saitou and Masatoshi Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology And Evolution*, 4(4):406–425, 1987.

[SW00] Joseph D. Szustakowski and Zhiping Weng. Protein structure alignment using a genetic algorithm. *Proteins: Structure, Function, and Genetics*, 4(38):428–440, March 2000.

[SW03] Joseph Szustakowski and Zhiping Weng. Protein structure alignment using evolutionary computation. In Gary B. Fogel and David W. Corne, editors, *Evolutionary Computation in Bioinformatics*, pages 59–86. Morgan Kaufmann, 2003.

[THG94] Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. Clustal w: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680, 1994.

[Tra06] Anna Tramontano. *Protein Structure Prediction: Concepts and Applications.* Wiley-VCH, Weinheim, January 2006.

[wwP08] wwPDB. Protein data bank contents guide: Atomic coordinate entry format description version 3.20. Available from: http://www.wwpdb.org/documentation/changesv3.20.pdf, 2008.

[Xia07] Xuhua Xia. *Bioinformatics And The Cell.* Springer, 2007.

[ZE94] Kam Y. J. Zhang and David Eisenberg. The three-dimensional profile method using residue preference as a continuous function of residue environment. *Protein Science*, 3:687–695, 1994.

Appendix A

CD Contents

- Directory Prerequisites
  - chemtex.zip: LaTeXpackage used by the report
  - dsspcmbi: DSSP secondary structure definition program, Linux x86 binary executable
  - fast-linux-32.tar.gz: FAST structure alignment program, source package
  - modeller-9v6.tar.gz: Modeller comparative modeling python module, installation package. Registration required, but free of charge for academic users. Note, Sali Lab do not automatically recognise VUTBR as an academic institution
  - RasMol_2.7.4.2_10Apr08.tar.gz: Rasmol molecular visualisation package (optional), source code
- Directory Text
  - Source code and figures of this report
- Directory Src
  - blast.py: Module exporting the BLAST routine and a BLAST results displaying procedure
  - clustalw.py: Module exporting the CLUSTALW algorithm, alignment output and PIR conversion routines
  - prediction_utils.py: Module exporting various auxiliary procedures used internally by the other modules
  - k2.py: Script implementing the K2 algorithm. Usage: ./k2.py PDB1 PDB2 GENS MUT POP, where PDBx are PDB four letter codes of structures to be aligned, GENS, MUT and POP specify the number of generations, mutation rate and population size during evolution
  - get_sh2.py: Script used for testing CLUSTALW. Creates a local FASTA formatted file containing SH2 domains from all the currently known proteins containing them. Usage: ./get_sh2.py list, where list can be found at http://biochem.wustl.edu/~sh2-domains/sh2_seq_list_all.html
  - get_pir_structure_header.py: Script which creates a proper PIR formatted structure header. Usage: ./get_pir_structure_header PDB START STOP, where PDB

is a PDB record file and START, STOP the first and last chain to include in the structure

- cores.py: Script which interactively guides the user through the multiple alignment and template selection processes. Usage: ./cores INPUT SEQ PDB, where INPUT is a FASTA formatted file with a single sequence – the query, SEQ and PDB are FASTA formatted databases of all sequences, sequences with experimentally determined structures available respectively. The databases can be downloaded with a Perl script provided by NCBI found here: http://www.ncbi.nlm.nih.gov/blast/docs/update_blastdb.pl. Be advised, the nr database amount to almost 5 GB
- modeller.py: Script which uses Modeller to produce a final structure model. Usage: ./modeller ALG TMPL SEQ, where ALG is a PIR formatted alignment file, and TMPL and SEQ are the alignment identifiers of the template and query respectively