Modeling protein evolution using secondary structures

By
Zia Mohaddes

Bioinformatics Program
Faculty of Graduate Studies

Thesis submitted to the Faculty of Graduate Studies
for obtaining the degree of MSc in Bioinformatics

August, 2010

Université de Montréal
Faculty of Graduate Studies

Université de Montréal

Faculty of Graduate Studies

This thesis is entitled:

Modeling protein evolution using secondary structures

Presented by: Zia Mohaddes

Evaluated by a jury composed of the following people:

Dr. Nadia El-Mabrouk, Chairperson
Dr. Sylvie Hamel, Research Supervisor
Dr. Andreea-Ruxandra Schmitzer, Co-Director
Dr. Nicolas Lartillot, Member of jury

Abstract

Protein evolution is an important field of research in bioinformatics and catalyzes the requirement of finding alignment tools that can be used to reliably and accurately model the evolution of a protein family. TM-Align (Zhang and Skolnick, 2005) is considered to be the ideal tool for such a task, in terms of both speed and accuracy. Therefore in this study, TM-Align has been used as a point of reference to facilitate the detection of other alignment tools that are able to accurately model protein evolution. In parallel, we expand the existing protein secondary structure explorer tool, Helix Explorer (Marrakchi, 2006), so that it can also be used as a tool to model protein evolution.

Keywords: Protein evolution, tools, comparison of tools, sequence based alignments, and structure based alignments.

Résumé

L’évolution des protéines est un domaine important de la recherche en bioinformatique et catalyse l'intérêt de trouver des outils d'alignement qui peuvent être utilisés de manière fiable et modéliser avec précision l'évolution d'une famille de protéines. TM-Align (Zhang and Skolnick, 2005) est considéré comme l'outil idéal pour une telle tâche, en termes de rapidité et de précision. Par conséquent, dans cette étude, TM-Align a été utilisé comme point de référence pour faciliter la détection des autres outils d'alignement qui sont en mesure de préciser l'évolution des protéines. En parallèle, nous avons élargi l'actuel outil d'exploration de structures secondaires de protéines, Helix Explorer (Marrakchi, 2006), afin qu'il puisse également être utilisé comme un outil pour la modélisation de l'évolution des protéines.

Mots-clés : L’évolution des protéines, des outils, comparaison des outils, des alignements de séquences, des alignements de la structure.

Table of Contents

List of Tables

List of Figures

Acknowledgement

First and foremost I would like to thank my supervisor, Sylvie Hamel, for been extremely supportive, understanding and approachable throughout my study and for giving me the privilege to be her student. I would love to also thank my co-director, Andreea Schmitzer for her ongoing support and encouragement.

I would like thank the responsible of the Bioinformatics courses, Marie Pageau, Gertraud Burger, Hervé Philippe and Nicolas Lartillot for giving me the opportunity to take part in their courses. I would like to also thank everyone in lbit lab for creating a very comfortable and friendly ambiance.

At last, not least, I would like to thank my mother for believing in me and for giving me all the supports I needed throughout my study. Also I would like to thank my best friend, Eli, who provided me the motivation and support that I needed to move to Montreal and to continue my master.

Abbreviations

DNA	Deoxyribonucleic Acid
RNA	Ribonucleic Acid
HTML	HyperText Markup Language
mmCIF	MacroMolecular Crystallographic Information File
NCBI	National Center for Biotechnology Information
OpenMMS	Open MacroMolecular Structures
PDB	Protein Data Bank
PDBj	Protein Data Bank Japan
PDBML	Protein Data Bank Markup Language
SQL	Structured Query Language

Introduction

Protein comparison has been used extensively in bioinformatics on topics ranging from protein structure modeling to protein evolution. Several alignment tools, including but not limited to, DALI (Holm and Sanderand, 1993), CE (Shindyalov and Bourne, 1998b), Lovoalign (Martínez et al., 2007b), LOCK2 (Singh and Brutlag, 1997), and TM-Align (Zhang and Skolnick, 2005) have been developed, all of which incorporate different structural alignment algorithms. These alignment tools aim to compare structures quickly and accurately, in order to facilitate modeling of the protein structure and overcome other bioinformatics obstacles. There are several steps involved in the comparison of protein (Eidhammer et al., 2004). First, a specific characteristic (or feature) such as the distance between secondary structures is identified upon which the comparisons will be made. Next, an appropriate algorithm is selected, and used to detect the optimal alignment. Finally, the alignment is validated using different criteria to measure the accuracy and quality of the given alignment as discussed further in the article written by Hitomi Hasegawa and Liisa Holm (Hasegawa and Holm, 2009).

A number of methods for comparing protein alignment tools have been developed and tested, the result of which is that TM-Align is considered to be faster and more accurate than other alignment tools due primarily to its optimal TM-score function. TM-Align performs a pairwise structural alignment using a dynamic programming algorithm and TM-score rotation matrix (Zhang and Skolnick, 2005; Teichert et al., 2007; Madhusudhan et al., 2009).

Protein evolution is an important field of study that addresses questions of how proteins evolve and change over a period of time. Protein evolution has diverse applications in bioinformatics, such as, probing the method by which two binding proteins co-evolve through complementary changes in each other (Goh et al., 2000); improving homology modeling techniques, which involves modeling the 3D structure of a specific protein sequence (Pál et al., 2006) and helping us understand the relationship between the evolutionary distance and number of protein-protein interactions of a protein (Fraser et al., 2003; Martínez et al., 2007a). The importance of using 3D information in modeling the evolution of proteins has been emphasized in multiple articles due to the fact that structures of proteins are more conserved than protein sequences (Lesk and Chotia, 1980; Chothia and Lesk, 1986; Holm and Sander, 1997; Bromham and Leys, 2005).

In fact, we agree that the choice of the alignment tool does affect the accuracy of the modeling of the evolution of a given protein family, and as a result this project intends to achieve three goals. Firstly, we are hoping to objectively determine the alignment tool most suited to the task of accurately modeling protein evolution when used in concert with TM-Align as a point of reference, due to its proven accuracy and speed. This will be accomplished by computing the distance between the phylogenetic trees resulting from each of the alignment tools chosen for comparison: CE, Lovoalign and LOCK2. Secondly, the performance of each of the alignment tools will be tested by applying them to protein families that have poor correlation between structure-based and sequence-based phylogenetic trees. This will be done by comparing the phylogenetic trees obtained from each of the four alignment tools (TM-Align, CE, Lovoalign and LOCK2) with the phylogenetic trees obtained by PhyML, which is a fast and accurate algorithm to estimate large phylogenetic trees (Guindon and Gascuel, 2003). The motivation for this idea comes from the PALI database study (Balaji et al., 2001). Finally we will aim to determine if the HE database can be used as a tool for modeling the evolution of proteins using secondary structures.

In Chapter 1, I will provide a brief introduction to proteins including their structure and composition. In Section 1.2, the different levels of structures of protein structures, such as alpha helices, beta sheets and turns, are discussed. The concept and importance of protein evolution, as well as the methods used to infer it, including structured-based and sequence-based methods, are discussed in Section 1.3.

The six different structural comparison tools chosen for comparison in this paper will be introduced in Chapter2. These are: DALI (2.1.1)[1], LOCK (2.1.2), CE (2.1.3), TM-align (2.1.4), STRAP (2.1.5) and Lovoalign (2.1.6), as well as the scoring function and algorithms used by each of these tools.

[1] The numbers refer to the section in which each tool is discussed.

In Chapter 3, the techniques that can be used to compare the previously described structural alignment tools will be presented and discussed. Section 3.1, will provide an introduction to phylogenetic tree inference and comparison, while Section 3.2 will introduce the various protein databases that are used to extract the necessary data. In Section 3.3, I will describe the accepted techniques used to compare and validate the alignment tools and Section 3.4 will detail the phylogenetic-based method that I have implemented in order to compare the different structural alignment tools. Finally, in Section 3.5, I will present the results obtained from these comparisons, and provide relevant discussions and conclusions.

The Helix Explorer (HE), a web-based tool that has been designed to centralize secondary-structure-based information aimed at facilitating Protein Data Bank (PDB) querying, will be introduced in Chapter 4. In Section 4.1, I will discuss the development and initial functionalities of HE. Additional functionalities that have since been added to HE will be outlined in Section 4.2, and the algorithm developed that provided us with the ability to model protein evolution, will be presented and discussed. Finally, in Section 4.3 I will discuss the results, of the comparison between the phylogenetic trees obtained using HE and the ones obtained using the four alignment tools, and suggested future improvements.

Finally, Chapter 5 presents a summary conclusion and discussion of the entire thesis.

Chapter 1. Protein Structures and Evolution

In this chapter, an introduction to proteins is provided, along with a description of their composition. I will then describe the different levels of structures (primary, secondary, tertiary and quaternary) for a given protein in Section 1.2. In Section 1.3, the concept and importance of protein evolution will be discussed, as well as the methods used to infer the protein evolution (such as structured-based and sequence-based methods). The information in this chapter is strongly inspired by "Introduction to Protein Structure" (Branden and Tooze, 1998).

1.1 Introduction to Proteins

The word protein is derived from the Greek word "proteios", which means 'primary of the first rank'. Proteins, polymers of amino acids, are created through a process termed 'translation'. Twenty natural different amino acids are commonly found in different proteins. Among these amino acids, a similar structure is conserved: with the exception of proline, they all have a hydrogen atom (H), an amino group (NH_2), and a carboxyl group (COOH), all of which are attached to a central atom (Cα), as shown in Figure 1.1. However, what makes these amino acids unique is the R-group, commonly referred to as the "side chain", attached to the central Cα. Each of the twenty natural amino acids has a different R-group, which is specified by its genetic code.

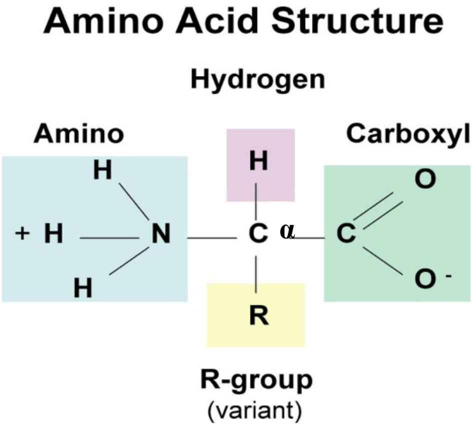


Figure 1.1. Amino acid structure, consisting of hydrogen atom (H), an amino group (NH_2), a carboxyl group (COOH) and a central carbon atom (Cα) (Moniz, 2007).

As shown in Figure 1.2, individual amino acids are joined together via peptide bonds through a condensation reaction between the carboxylic group of one amino acid and the amino group of the second, a process that releases one water (H_2O) molecule. As additional amino acids are linked together, this process repeats, elongating the chain. However, in each amino acid chain, two amino acids will remain intact: the amino group of the first amino acid and the carboxyl group of the last amino acid, highlighted in green and blue in Figure 1.2.

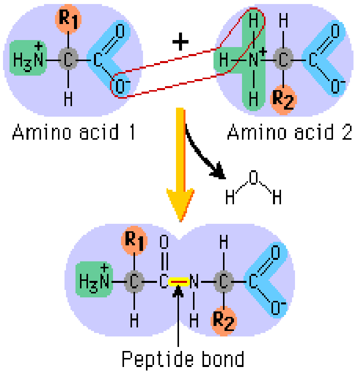


Figure 1.2. Formation of peptide bonds by condensation reaction between the carboxylic group and amino group (Pearson, 2009).

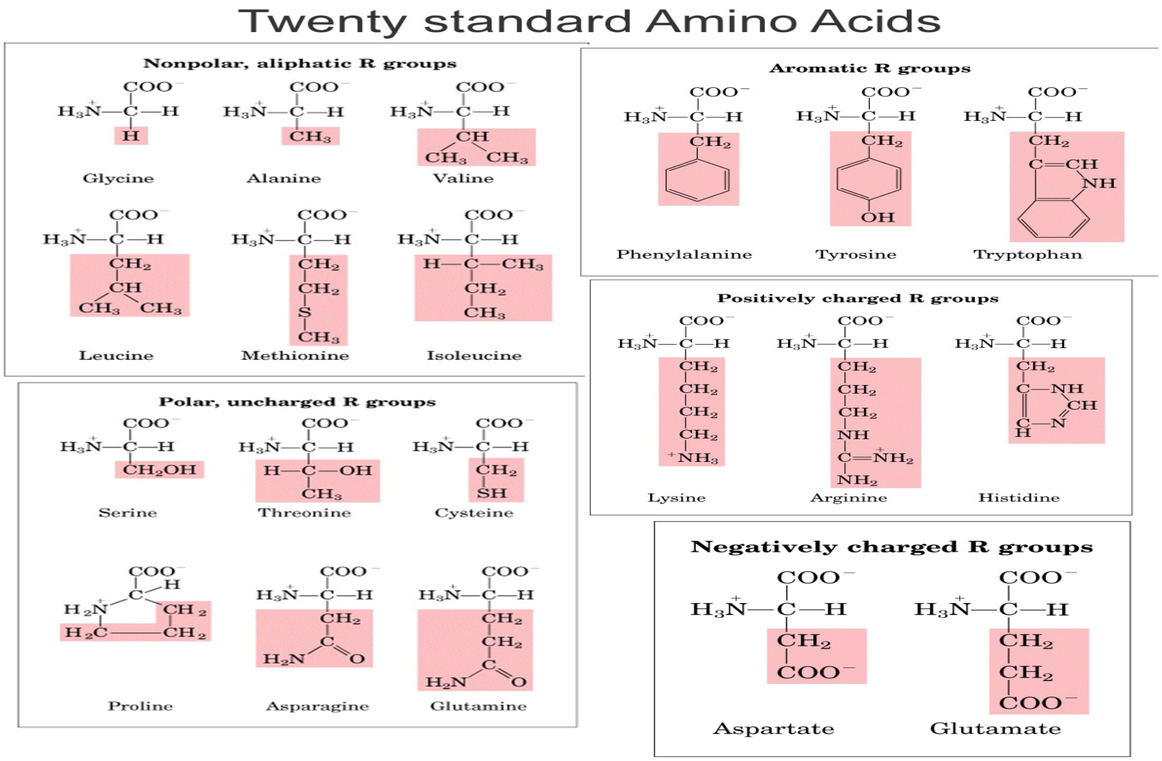


Figure 1.3. There are twenty standard amino acids that occur in proteins (McDarby, 2003).

As mentioned previously, there are twenty different standard amino acids, shown in Figure 1.3, that are used to synthesize proteins and other biomolecules. These amino acids can be divided into three different classes, according to the biochemical nature of their side chain. These classes are as follows:

a) Amino acids with strictly non polar chains: includes Glycine (G), Alanine (A), Valine (V), Leucine (L), Isoleucine (I), Phenylalanine (F), Proline (P), and Methionine (M).

b) Amino acids with charged polar side chains: chains can be either positively or negatively charged. Includes Asparatic acid (D), Glutamic acid (E), Lysine (K), and Arginine (R).

c) Amino acids with uncharged polar side chains: includes Serin (S), Threonine (T), Asparagine (N), Glutamine (Q), Tyrosine (Y), Cysteine (C), Tryptophan (W) and Histidine (H).

With the exception of glycine, which has two hydrogen atoms attached to the central Cα, each of the remaining nineteen amino acids is a chiral molecule, since each has four chemically different groups attached to Cα. Glycine residues are usually considered to be the most flexible amino acid, and are able to integrate easily into both hydrophobic and hydrophilic environments, due specifically to their single hydrogen atom side chains.

1.2 Proteins and their Structures

Any given protein has four levels of structure as shown in Figure 1.4. These are: the primary structure, which is a linear chain of amino acids; the secondary structure, which consists of highly regular structures defined locally; the tertiary structure, which is formed through attractions between secondary structures; and, finally, the quaternary structure, which is composed of more than one polypeptide chain. Due to the fact that some of the structural software used in this project incorporate secondary structure information, the different types of secondary structures will now be described in more detail.

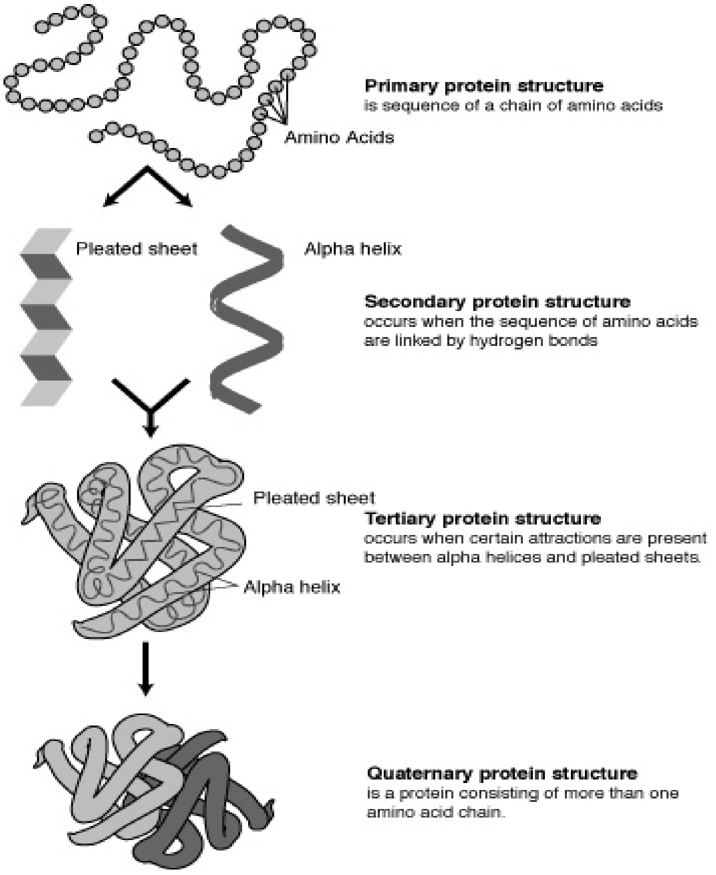

Primary protein structure
is sequence of a chain of amino acids
Amino Acids
Pleated sheet
Alpha helix
Secondary protein structure
occurs when the sequence of amino acids
are linked by hydrogen bonds
Pleated sheet
Tertiary protein structure
occurs when certain attractions are present
between alpha helices and pleated sheets.
Alpha helix
Quaternary protein structure
is a protein consisting of more than one
amino acid chain.

Figure 1.4. The different levels of protein structures: primary, secondary, tertiary and quaternary (Huskey, accessed 2010)[2](Huskey, 2010).

There are three types of secondary structures found in proteins: the alpha helix (Section 1.2.1), the beta sheet (Section 1.2.2) and the turns (Section 1.2.3).

1.2.1 Alpha Helices

Alpha helices, important elements of secondary protein structures, were firstly described by Linus Pauling at the California Institute of Technology during the 1990s (Pauling, 1996).

An alpha helix occurs when a chain of consecutive amino acids, formed through peptide bonds, all have their phi (φ) and psi angle (Ψ) in the range between -60^o and -50^o, as shown in Figure 1.5. In addition, both phi and psi angles can be visualized using the Ramachandran map

[2] The site can be accessed at http://andromeda.rutgers.edu/~huskey/

(Ramachandran and Sasisekharan, 1968). Since most of the residues in a helix are bonded in this manner, the helix is usually a rigid structure with very little internal space (Maccallum, 1997). Within an alpha helix, there are typically 3.6 residues per turn, and hydrogen bonding between the CO and NH groups of residues i and i+4 causes the formation of a right handed coil, as shown in Figure 1.6. Different variations of the alpha helix, such as the Pi helix and the 3_{10} helix are created when i+5 or i+4 residues form hydrogen bonds with the residue i. The average length of an alpha helix, typically measured in globular proteins, averages ten residues and corresponds to three turns.

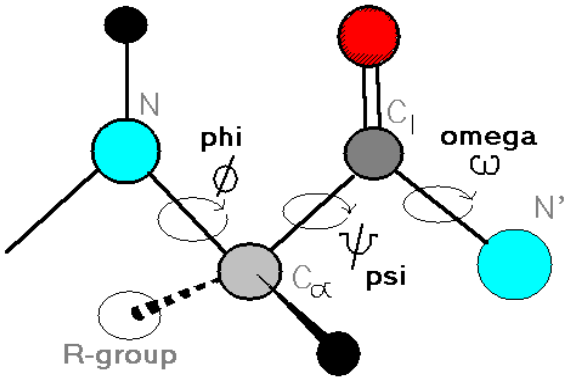


Figure 1.5. The three repeating torsion angles in the polypeptide chain are called phi (φ), psi (Ψ) and omega (ω). Black circles represent hydrogen atoms, grey circles represent carbon atoms, blue circles represent nitrogen atoms and red circles represent oxygen atoms (Wampler, 1996).

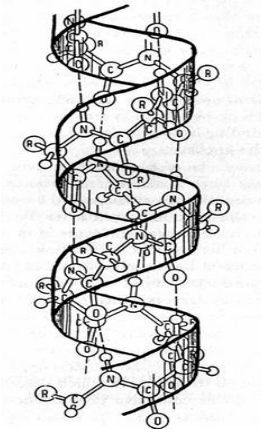


Figure 1.6. The alpha helix is one of the major elements of secondary structure in proteins. The O and N atoms are linked together by hydrogen bonds in the main chain (Hameroff, 1987).

1.2.2 Beta Sheets

The beta sheet is another major structural element of globular proteins and is, in fact, a combination of several different regions of a polypeptide chain, called beta strands. Beta strands are typically made up of five to ten amino acids, all of which have their backbones in a fully extended conformation. In this formation, their side chains alternate directions, pointing upward, then downward, and so on. The beta strands must run parallel to each other so that a hydrogen bond can form between the N-H group in the backbone of one strand and the C=O group in the backbone of another strand. The beta sheet, more complex structures composed of these strands, are usually "pleated", meaning that their Cα is slightly above or below the plane of the beta sheet, as shown in Figure 1.7.

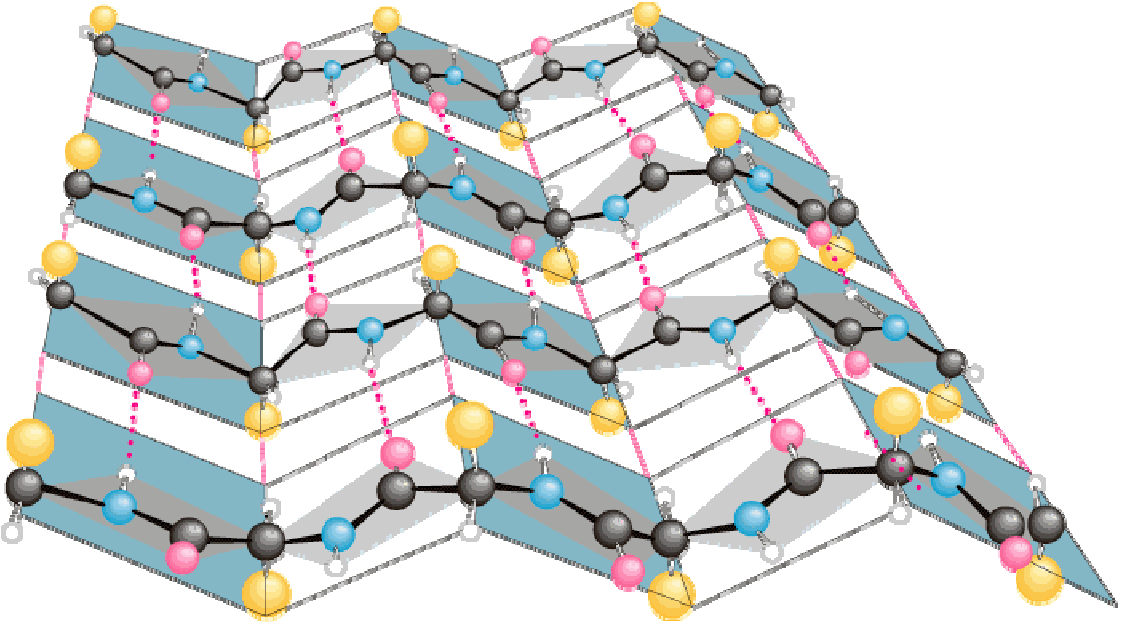


Figure 1.7. An example of a pleated beta sheet where the oxygen (O) atoms are in purple, nitrogen (N) atoms in blue, hydrogen (H) atoms in white, Cα atoms in black, and side chains in orange (Preston, 2009).

Beta sheets can be structured in two very different ways, as parallel beta sheets or as anti-parallel beta sheets, as shown in Figure 1.8. In Parallel beta sheets, the amino acids of adjacent beta strands are coordinate in the same biochemical direction, meaning that the amino terminal and carboxyl terminal of each strand are adjacent. In anti-parallel configuration, individual beta strands alternate in direction, and the amino terminal of one strand is adjacent to the carboxyl terminal of the strand on either side.

The two different forms are distinguished by their specific patterns of hydrogen-bonding. Anti-parallel beta sheets have narrowly spaced hydrogen bond pairs alternating with widely spaced pairs while parallel beta sheets have evenly spaced hydrogen bonds (Keates, 1988).

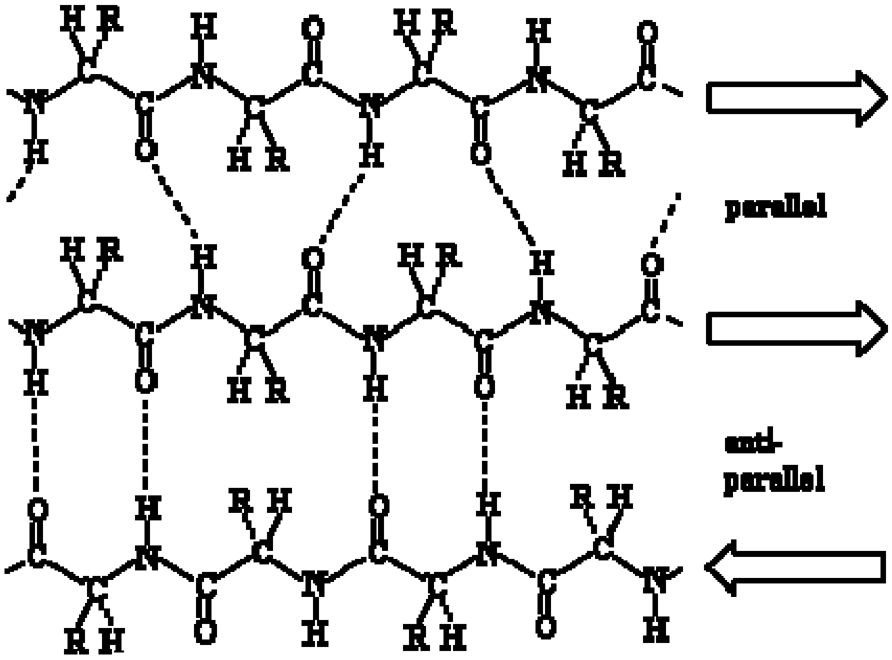


Figure 1.8. Anti-parallel and parallel beta sheets, distinguished by different amino to carboxyl terminal alignment and hydrogen-bonding pattern (Keates, 1988).

1.2.3 Turns

The two different secondary structure elements previously discussed, alpha helices and beta sheets, are usually connected in a globular protein by an irregularly shaped loop region called a turn. Loop regions, unlike other secondary structures, are usually found at the surface of the protein and as a result often form hydrogen bonds with nearby water molecules. Turns are typically rich in charged and polar hydrophilic residues, a feature that makes their presence easier to detect using prediction algorithms, unlike the other two secondary structures, which are usually embedded below the surface level of the protein, and lack the charged residues characteristic of turns.

1.3 Protein Evolution

1.3.1 Importance of studying protein evolution and protein structure

Protein evolution is a branch of biological study that focuses on the processes and mechanisms through which proteins change over time, while also addressing the question of why proteins evolve at different rates. Studying protein evolution also contributes to the reconstruction of past events that have given rise to the large variety of proteins in existence today (Doolittle, 1981).

Understanding the causes of observed variations in the evolutionary rate of proteins is essential for diverse and numerous fields, including molecular evolution, comparative genomics, and structural biology. The evolutionary rate of proteins can also be used to highlight the importance of genetic drift and selection, as well as facilitate the identification of selective forces from genomic data. Analyzing protein evolution provides a unique method for understanding complex issues such as the evolution of speciation, due to rapid genetic evolution (Webster et al., 2003). Finally, it facilitates the discovery of functionally important sites used in protein design, peptides involved in genetic diseases, drug targets, and protein interaction partners.

The importance and benefits of studying protein evolution are well displayed through a number of studies, including protein-protein binding studies (Goh et al., 2000), homology modeling (Pál et al., 2006) , and the quantification of protein-protein interactions (Fraser et al., 2003).

1.3.1.1 Protein evolution in protein-interaction partners

Protein-protein binding plays an important role in both metabolic and signaling pathways. A pair of binding proteins must co-evolve, such that any divergent changes in one partner are complemented at the surface of the other, in order to maintain their mutual functioning. Unfortunately, using the results obtained from biochemical does not currently allow us to fully understand these interactions. Therefore, to gain a better understanding of the co-evolution of binding proteins, such as receptors and ligands, the available evolutionary information must be considered.

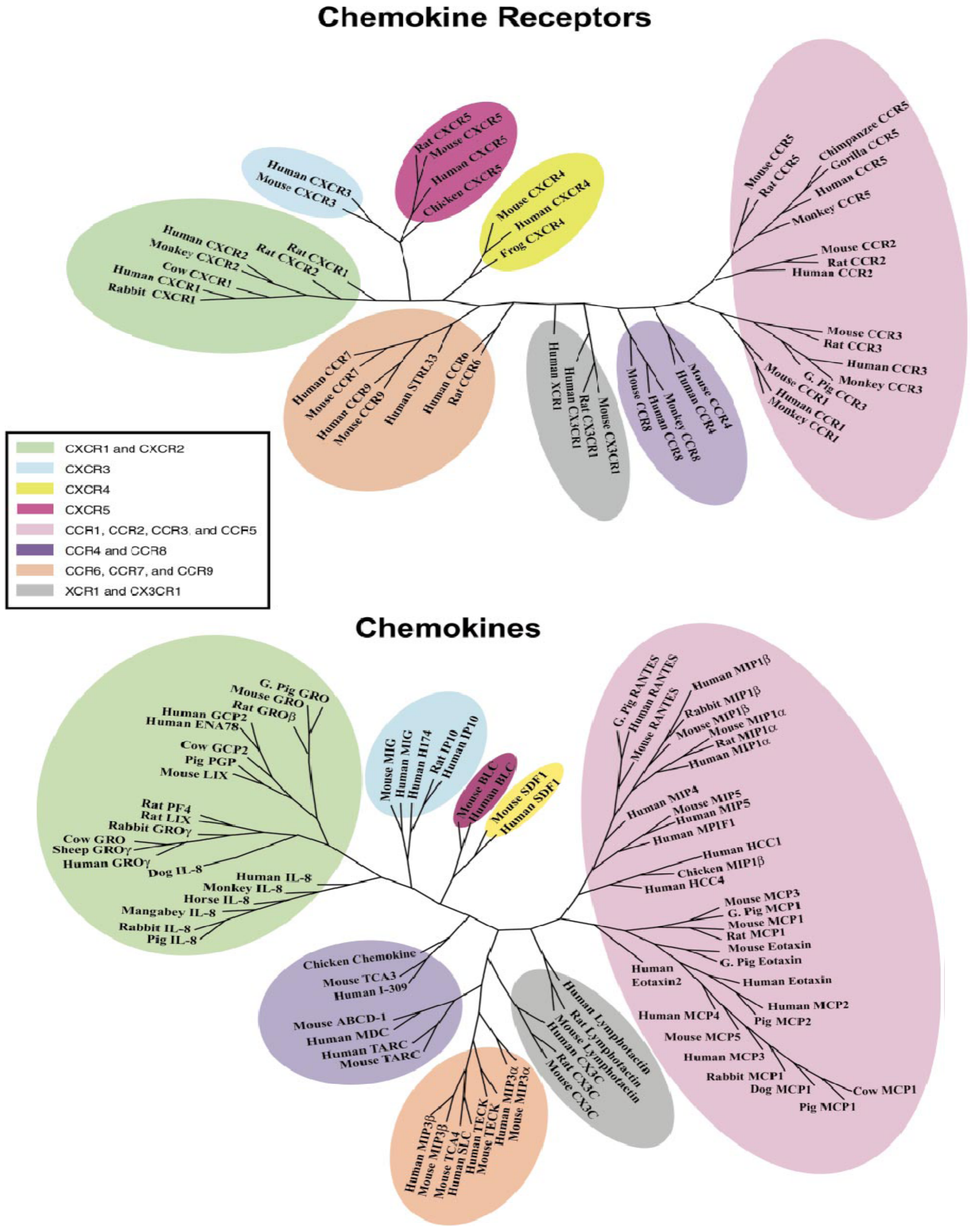


Figure 1.9. Phylogenetic trees of chemokines and chemokine receptors, where groupings among both families are shown by colored clusters. The diagrams are colored by their clustering patterns to show similar groupings among the chemokines and the receptors to which they bind (Goh et al., 2000).

Evolutionary information of proteins can be obtained using statistical comparisons between the phylogenetic trees of protein families that interact with one another. For example, the phylogenetic trees of two chemokines families (a chemokine ligand and a chemokine G-protein coupled receptor) are shown in Figure 1.9. The colored clusters show the similar groupings between the two families. Using the two phylogenetic trees, corresponding to each of the chemokine proteins (receptor and ligand), one is able to calculate the correlation coefficient, which is a quantifiable measure of their co-evolution. Further mathematical analysis of chemokine protein co-evolution can be found in a study conducted by Goh et al. (Goh et al., 2000).

1.3.1.2 Protein evolution in homology modeling

An understanding of the mechanisms and pressures that caused a protein to evolve into a different protein will improve homology modeling, also known as comparative modeling, by identifying evolutionarily related proteins. Homology modeling techniques consist of modeling the 3D structure of a specific protein sequence by comparing its homologous protein, which is a protein sharing a common ancestor, with a known 3D structure (Pál et al., 2006). There are several steps required in homology modeling. First, a homologous (evolutionary related) protein must be identified. Next, the sequence of an unknown protein structure is aligned against the chosen, known, homologous structure. Once aligned, the information in the alignment will be used to construct and indentify the structurally conserved and variable regions represented by a series of coordinates. Finally, the determined structure is assessed and validated experimentally using free web-based software package called "what check"[3].

1.3.1.3 Correlation between the evolutionary rate of proteins and the number of protein-protein interactions

It has been shown through several studies that there is a highly significant negative correlation between the number of protein-protein interactions, in which a protein is involved, and the evolutionary divergence of this protein as shown in Figure 1.10 (Fraser et al., 2003). This correlation indicates that an increase in the number of protein–protein interactions will cause the rate of evolution of a protein to slow down due to structural constraints that must be maintained

[3] 'What-check' package contains a list of software used to validate the determined structures in homology modeling is available at http://swift.cmbi.ru.nl/servers/html/index.html

to preserve all the interactions. It is important to note that this correlation can only be identified when using a large and complete set of protein-protein interaction data, as in the case of *S. Cerevisiae*. Therefore this correlation cannot be generalized in other kingdoms of life, due to the lack of complete set of protein-protein interaction data from other organisms.

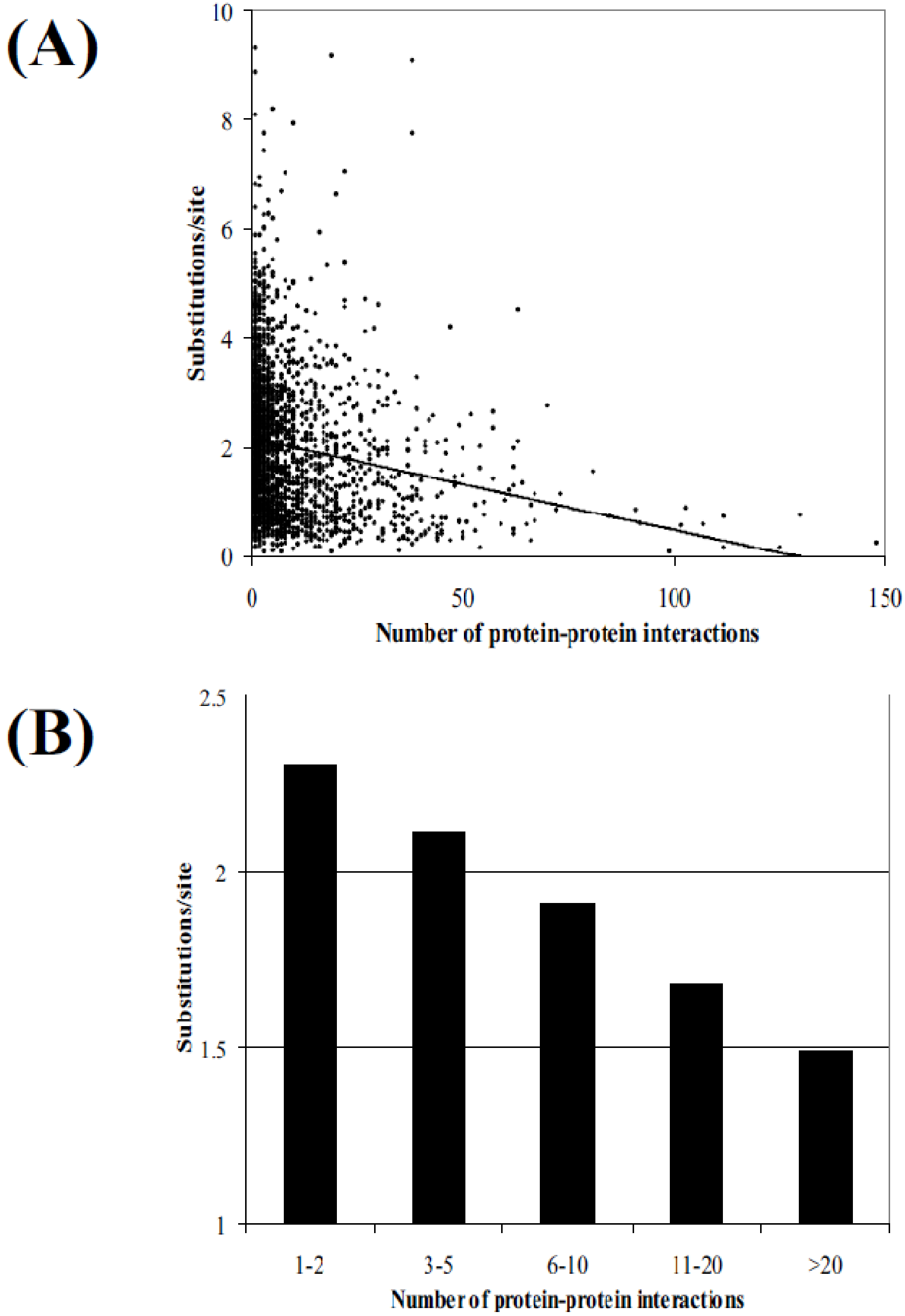


Figure 1.10. The relationship between the number of protein-protein interactions and the evolutionary rate between S. Cerevisae and S. Pombe is shown. A) The relationship between the number of protein-protein

interactions and the evolutionary rate for all interaction data. B) The average evolutionary rates of genes categorized by their number of protein-protein interactions (Fraser et al., 2003; Martínez et al., 2007b).

In conclusion, there are many fields of study and situations that demonstrate the importance of protein evolution. In all cases, either structural or sequential information is used in order to infer protein evolution. However, the question that remains is when to use either sequence or structural information to compare distantly-related proteins in preference to the other, and why. One of the goals of this project is to answer this question by comparing the results obtained from selected structure-based and sequence-based protein comparison tools.

1.3.2 Comparison of sequence-based and structure-based inference of protein evolution

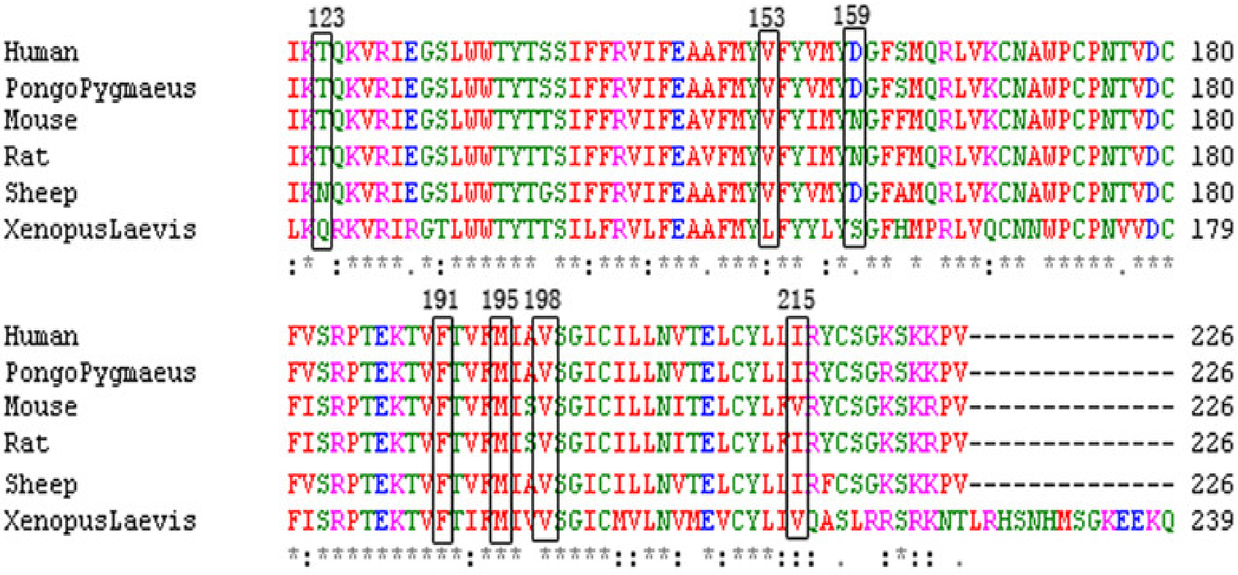
This section will provide an overview of sequence-based and structure-based protein comparisons methods, as well as describe different studies in which the structure-based alignment is preferred.

1.3.2.1 Protein alignment using sequence information

Sequence alignment is a method used to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships. In fact, the degree of similarity between amino acids at a given position indicates approximately how conserved this region is among different proteins. A pair of amino acids, one taken from each protein, is considered to be the smallest unit of comparison in sequence alignment.

Dynamic programming is widely used when performing sequence alignment. Every column in an alignment between two sequences represent an edit operation that can either be the replacement of an amino acid by another, the insertion or deletion of an amino acid or the identity; for example the amino acid in a given position stay the same between the two sequences. Each edit operation has an associated cost, and the algorithm must detect the alignment that uses the edit operations with lowest cost (Needleman and Wunsch, 1970; Eddy, 2004). The alignment can then be read and considered as a way to transform one sequence into another. It should be noted that a distinction between the aim (which is finding the optimal solution), the cost, and the algorithm to maintain the cost should be always made.

Tools such as dotlet allow for pairwise sequence alignments (Junier and Pagni, 2000), while other tools, known as Multiple Sequence Alignment (MSA) tools, are able to produce alignment between multiple proteins. MSA tools are usually more computationally expensive and complex due to the greater alignment requirements. Results produced through MSA tools, such as those shown in Figure 1.11, provide information for each column of the alignment on the mutations that occurred at one given site throughout the evolution of that protein.



```
                 123                           153   159
Human            IKTQKVRIEGSLWWTYTSSIFFRVIFEAAFMYVFYVMYDGFSMQRLVKCNAWPCPNTVDC 180
PongoPygmaeus    IKTQKVRIEGSLWWTYTSSIFFRVIFEAAFMYVFYVMYDGFSMQRLVKCNAWPCPNTVDC 180
Mouse            IKTQKVRIEGSLWWTYTTSIFFRVIFEAVFMYVFYIMYNGFFMQRLVKCNAWPCPNTVDC 180
Rat              IKTQKVRIEGSLWWTYTTSIFFRVIFEAVFMYVFYIMYNGFFMQRLVKCNAWPCPNTVDC 180
Sheep            IKNQKVRIEGSLWWTYTGSIFFRVIFEAAFMYVFYVMYDGFAMQRLVKCNAWPCPNTVDC 180
XenopusLaevis    LKQRKVRIRGTLWWTYTTSILFRVLFEAAFMYLFYYLYSGFHMPRLVQCNNWPCPNVVDC 179
                 :* :****.*:****** **:***:***.***:** :*.** * ***:** *****.***

                           191 195198          215
Human            FVSRPTEKTVFTVFMIAVSGICILLNVTELCYLLIRYCSGKSKKPV-------------- 226
PongoPygmaeus    FVSRPTEKTVFTVFMIAVSGICILLNVTELCYLLIRYCSGRSKKPV-------------- 226
Mouse            FISRPTEKTVFTVFMISVSGICILLNITELCYLFVRYCSGKSKRPV-------------- 226
Rat              FISRPTEKTVFTVFMISVSGICILLNITELCYLFIRYCSGKSKRPV-------------- 226
Sheep            FVSRPTEKTVFTVFMIAVSGICILLNVTELCYLLIRFCSGKSKKPV-------------- 226
XenopusLaevis    FISRPTEKTVFTIFMIVVSGICMVLNVMEVCYLIVQASLRRSRKNTLRHSNHMSGKEEKQ 239
                 *:**********:*** *****::**: *:***::: .  :*:: .
```

Figure 1.11. Amino acid alignment of Connexin26 in different species (Dai et al., 2009)

Multiple sequence alignment tools incorporate different algorithms depending on the tool. ClustalW, the MSA tool used to produce the results shown in Figure 1.11, makes use of a progressive alignment technique, which begins by aligning the two closest sequences to get an optimal alignment and then add the other sequences one by one in order to obtain the final multiple alignment (Thompson et al., 2002). Another MSA tool, muscle, uses iterative methods (Edgar, 2004), while hmmer uses a hidden markov model (Durbin et al., 2004). In this project, I will use ClustalW, an MSA tool using a progressive algorithm, (Thompson et al., 2002), and one of the most popular programs available for conducting MSA. Progressive algorithm consists of three steps: a) performing a pairwise alignment of all of the sequences; b) creating a phylogenetic tree using the obtained alignment scores; c) using phylogenetic relationships in the resulting tree to guide the sequential alignment of the sequences using a dynamic programming algorithm.

It is important to note that the quality of sequence alignments is unknown when there is a low sequence identity among the proteins. In such cases, the quality can be determined by comparing the obtained sequence alignment against an alternative protein alignment method, called structural alignment (Sauder et al., 2000), which is the subject of the rest of this chapter.

1.3.2.2 Proteins alignments using structural information

Since the discovery of the first proteins, the comparison of protein structures has been considered an extremely important task in structural and evolutionary biology. Establishing a correspondence between the residues of two protein structures is crucial in computational structural biology. Moreover, superimposition of similar protein structures and generation of structure-based sequence alignments will facilitate understanding of the evolutionary and thermodynamic constraints on a given fold, improve protein predictions, contribute to information about both individual proteins and protein structures, as well as help in the identification of homologous residues (Vesterstrom and Taylor, 2006; Hasegawa and Holm, 2009). Structural comparison is considered to be very efficient in providing information about common ancestry when dealing with homologous proteins, in identifying common sub-structures when dealing with non-homologous proteins, and in classifying proteins (Illergård et al., 2009).

The most natural way of comparing two objects, each represented by a collection of elements, is to determine the correspondences between them. This correspondence, or structural alignment, is usually based on the Euclidean distance of their central Cα atoms. Generally, the pairwise structural comparison of proteins can be divided into four major steps:

1) First, specific features are extracted, such as protein 3D coordinates, physiochemical properties of residues, sequential order of residues along the back bone, distance between two amino acids, and structural arrangement of secondary structures.

2) The extracted features are then used by comparison algorithms to detect a correspondence or equivalence between two proteins based on certain constraints. These algorithms are explained further in Section 2.2 (Hasegawa and Holm, 2009).

3) The detected optimal alignment (or correspondence) must be validated using a scoring

system or threshold. There are different scoring schemes available, depending on whether the structural representation is 3D, 2D (i.e. distance matrix or contact maps), or 1D (i.e. structural profile). A detailed list of these scoring schemes, according to their representation, can be found in a paper by Hasegawa and Holm (Hasegawa and Holm, 2009).

4) Finally, different evaluation tests are carried out to further measure the accuracy and quality of the alignments, as well as the ability of the alignment score to distinguish between homologous and unrelated proteins (Eidhammer et al., 1999; Eidhammer et al., 2004; Hasegawa and Holm, 2009). These properties will be discussed in more detail in Section 3.2. The schematic overview of these major steps is presented in Figure 1.12.

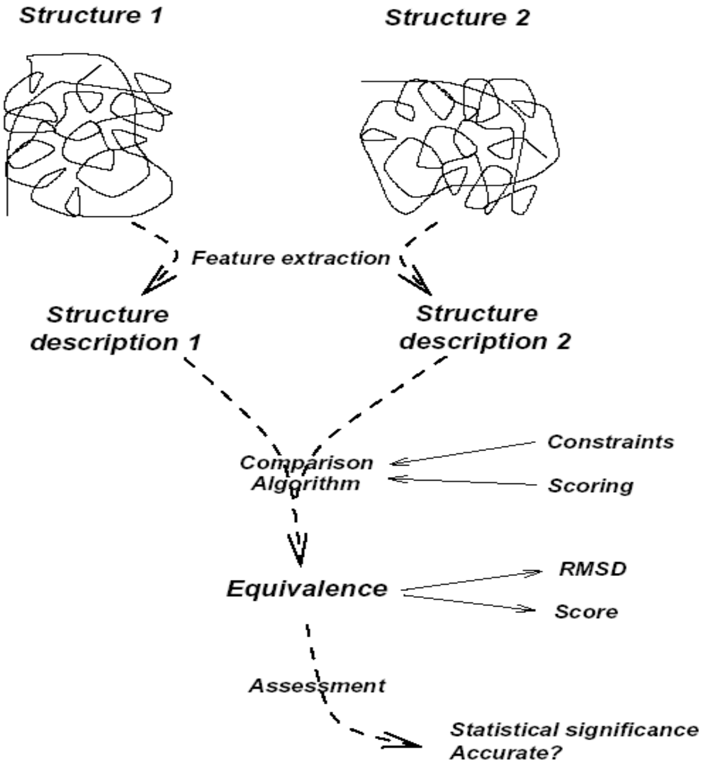


Figure 1.12. Pairwise structural comparison framework which includes: 1) feature extraction, 2) comparison algorithm 3) scoring schemes, and 4) assessment and validation (Eidhammer et al., 1999).

Despite the lack of a universally acknowledged definition of what constitutes structural similarity, there is a strong tradition of visualizing structural alignments by least square superposition, which treats the structures as rigid 3D objects (Hasegawa and Holm, 2009). This procedure involves the superposition of the 3D structure of one protein onto the 3D structure of a second protein domain such that all the atoms fit together as closely as possible. The average spatial superposition detects a correspondence between the residues, based on the 3D structures of proteins, by identifying the highest number of atoms aligned with lowest Root Mean Square Deviation (RMSD) for the two given proteins, as shown in Figure 1.13. RMSD is the sum of the distances between residues in proteins A and B at position *i* divided by the total number of amino acid residues, as described by the following formula:

$$\text{RMSD (A, B)} = \sqrt{1/N\sum_{i=1}^{N} d(a_i, b_i)^2}$$

However, RMSD has a number of pitfalls. First of all, identical RMSD in two different structural alignments (superposition) doesn't necessarily suggest the same structural divergence, since they may not have the same number of "topologically equivalent Cα atoms". Additionally, optimal alignment doesn't always guarantee the minimal RMSD. And finally, significance of RMSD is size dependant.

To overcome these issues, some structural alignment algorithms, which will be explained in Section 2.1, have proposed different structural metrics that incorporate the size of the proteins, as well as the number of equivalent Cα atoms, into the RMSD measure. Structural distance metric (SDM), which is utilized by the structural alignment program STAMP (Russel and Barton, 1992), is an example of an algorithm featuring such improvements.

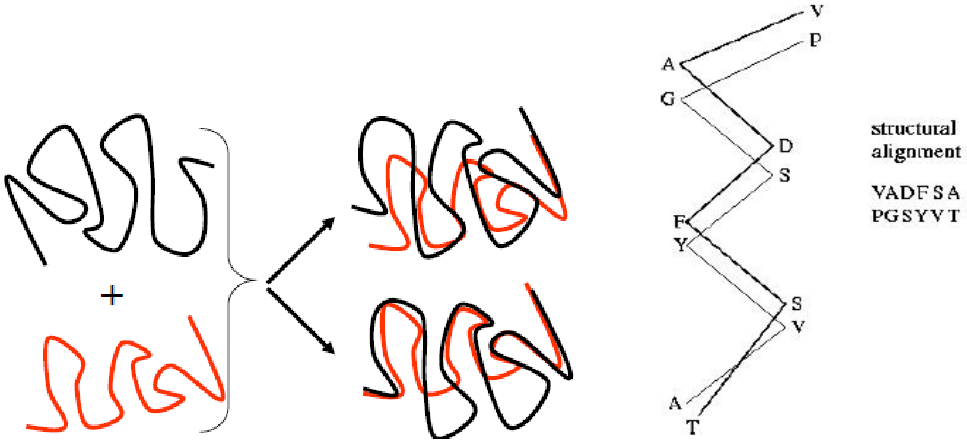


Figure 1.13. Given two protein structures, the goal is to find a transformation that superposes the two structures such that the RMSD is minimized. RMSD is the sum of the distances between residue a in the protein A at position i, and residue b in the protein B at position i divided by the number of residues (N).

The two protein alignment techniques, sequence-based and structure-based, have been described in Sections 1.3.2.1 and 1.3.2.2 respectively. The next section will describe which of these techniques can be applied when inferring protein evolution.

1.3.2.3 Inferring protein evolution using structural information

Evolution has produced homologous proteins whose sequences of amino acids have diverged significantly, but which have also maintained very similar structures. A popular example is the comparison of mouse abelson cytoplasmic tyrosine kinase to human p38 serine kinase as shown in Figure 1.14. Although they only have 28% sequence similarity, they have managed to maintain a common 3D structure.

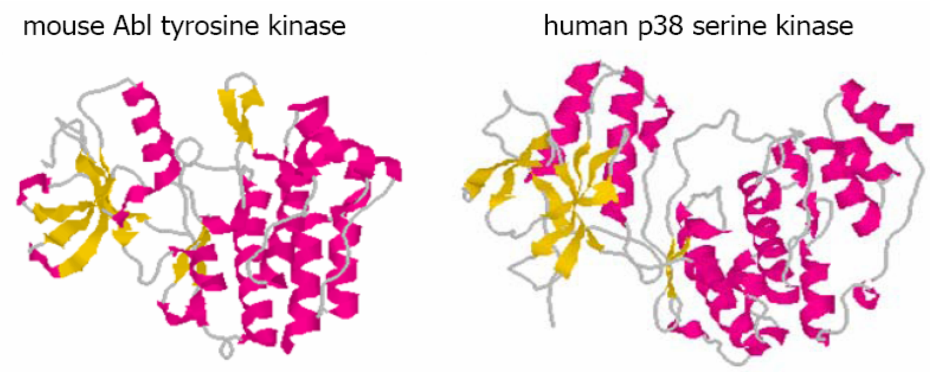


Figure 1.14. Structural comparison between two kinase proteins: mouse Abl tyrosine kinase and human p38 serine kinase. Purple indicates alpha helices and yellow beta sheets in both proteins (Thorne, 2007).

Evolutionary changes usually occur in populations due to replacement, short deletions, and insertions of single amino acid residues. The extent of such changes, or perturbations, to a 3D structure will largely depend on the type and location of the evolutionary sequence changes. For example, some sequence mutations will cause the structure to alter, whereas others, which preserve the physicochemical properties of the protein, will have no impact on the structure. While it has been shown that most sequence mutations will be structurally conservative, the challenge is to map the evolutionary changes at the sequence level to the resultant protein structural perturbations (Illergård et al., 2009). This correspondence can help emphasize the advantage of structural comparison over the sequence comparison when dealing with divergent evolution, which has been discussed in various papers (Lesk and Chotia, 1980; Holm and Sander, 1997; Bromham and Leys, 2005).

There are a number of studies that demonstrate the importance of 3D comparison over the more traditional sequence comparison when dealing with distantly evolved proteins. In the following subsections, three of these studies (Lesk and Chotia, 1980; Holm and Sander, 1997; Balaji and Srinivasan, 2007), are explained, where the last sections is more emphasized since part of my project is based on it.

Analysis of conservation patterns in 3D and identification of new enzymes families

An interesting example is the identification of ten new enzyme families based on the similarity of the architecture of three known enzymes (urease, phosphotriesterase, and adenosine deaminase). The similarity of these three enzymes was detected based solely on a structural comparison that provided evidence of their distant evolutionary relationship. It is worth mentioning that the importance of this similarity was previously unnoticed in traditional sequence comparisons(Holm and Sander, 1997). Additional details on how a structural comparison is conducted has been provided in Section 1.3.2.2.

Structural comparison of globins molecules

Similar research was conducted in which nine globin molecules were compared in order to answer the question of how different amino acid sequences maintain similar 3D structures. Figure 1.15 shows the comparison between two of these globins molecules, which have diverged widely in terms of their amino acid sequence, but have retained very similar secondary and tertiary structures. Both of these proteins have eight helices, which assemble in a common pattern. The variations in the geometrical arrangement of helices, which are produced by mutation and cause structural shift, in both globins molecules are interdependent in order to maintain similar function (Lesk and Chotia, 1980).

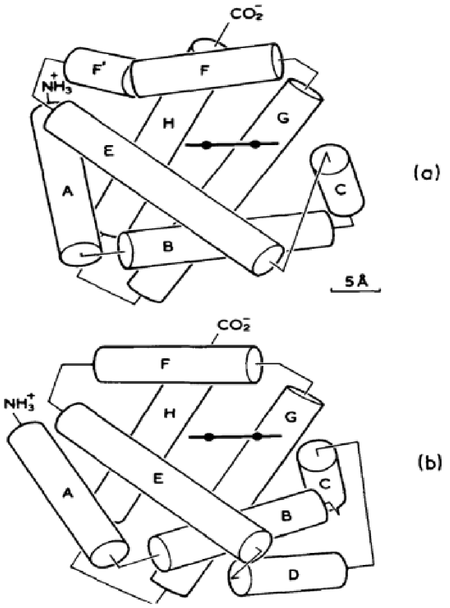


Figure 1.15. Structural patterns of two globins molecules are shown: a) the human deoxyhemoglobin; b) chiromus erythrocruorin molecules. The cylinders represent the helices in the globins molecules(Lesk and Chotia, 1980).

Correlation between sequence-based phylogenetic trees and structured-based phylogenetic tree

The final study relevant to this section, which the idea of comparing structure-based phylogentic trees was inspired from, presents an assessment conducted using 3D structure to model the evolution of homologous proteins (Balaji and Srinivasan, 2007). Using a dataset of 108 protein domain families of known structures, as well a series of constructed phylogenetic trees, a comparison of structural and sequence dissimilarities among pairs of proteins was made.

To conduct this comparison, a protein structural database called Protein ALIgnment Database (PALI), which contained structure-based and sequence-based distance matrices, as well as structure-based and sequence-based phylogenetic trees, was utilized. In order to align two proteins, a structural alignment tool called STructural AlignMent Program (STAMP) (Russel and Barton, 1992) was used. The result of this alignment was used to compute the Structural Distance Metric (SDM), which measures the extent of structural divergence, as well as the SEquence Distance Metric (SEDM) which quantifies the dissimilarity at the level of each amino acid sequence. These distances are placed into two different matrices: a structure-based and sequence-based distance matrix, respectively. A clustering algorithm, such as the Neighbour Joining Algorithm (NJ; explained further in Chapter 3) uses these matrices to generate the corresponding phylogenetic trees. Finally, as a measure of similarity between sequence-based and structure-based phylogenetic trees, a correlation coefficient is considered, where 1 indicates a perfect correspondence and 0 indicates no correlation.

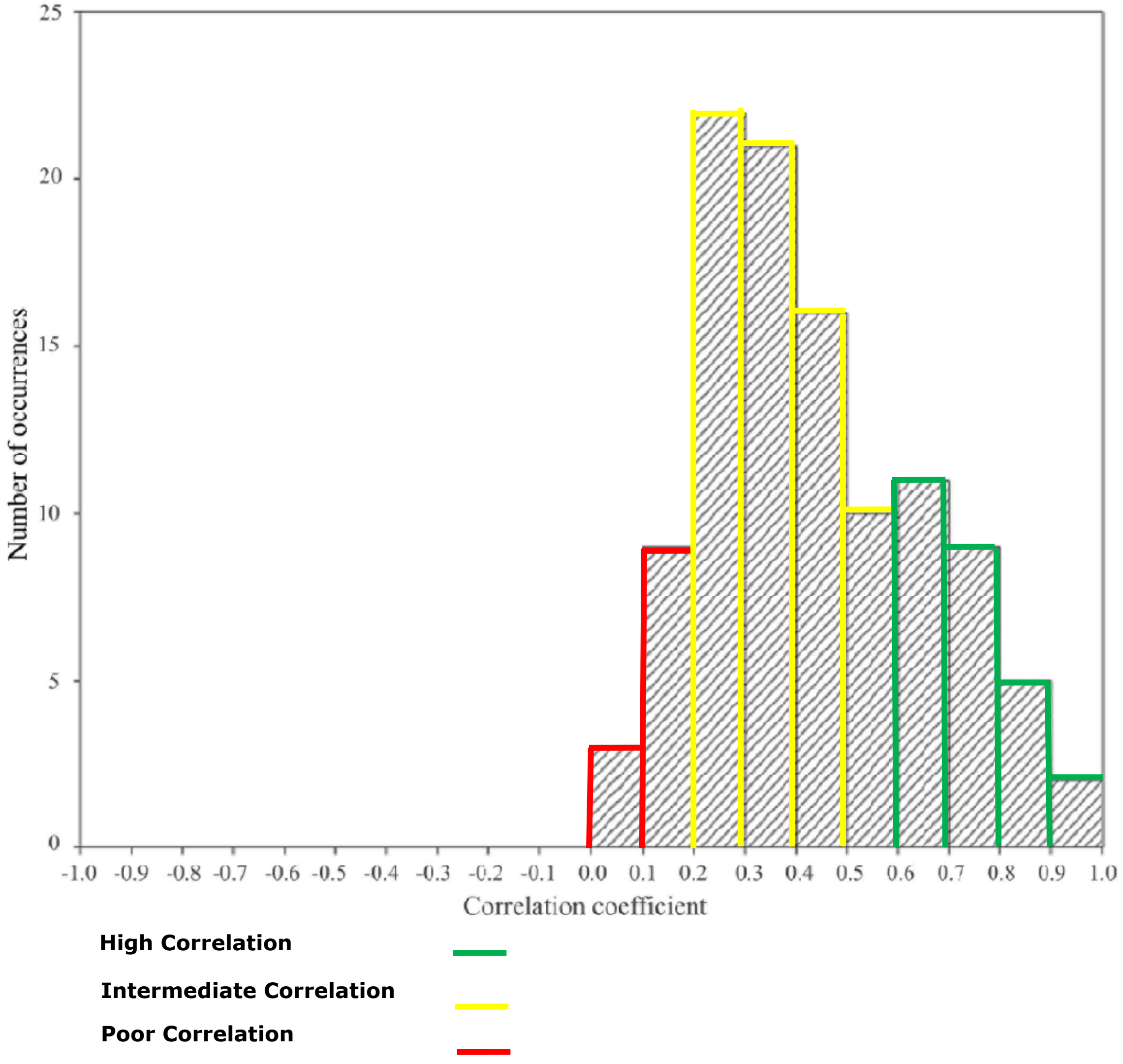


Figure 1.16. Histogram depicting the distribution of the number of occurrences for every 0.1 interval of correlation coefficient for all 108 protein families considered in the analysis (Balaji and Srinivasan, 2007).

Comparison between the phylogenetic trees results in three different types of correlation coefficient distributions, which are based on the SEDM and SDM of all 108 families, as shown in Figure 1.16: a) families with good correlation (correlation coefficient > 0.6) between sequence based and structure-based phylogenetic trees; b) families with intermediate correlation (correlation coefficient greater than or equal to 0.2 but less than 0.6); c) families with poor correlation (correlation coefficient < 0.2).

It was found that the correlation between the structure-based dissimilarity measures and the sequence-based dissimilarity measures was intermediate to high if sequence similarity among the homologous proteins was approximately 30% or greater. For protein families with low sequence similarity among the protein members, the correlation coefficient between the sequence-based and structure-based dissimilarities was poor. This was due to domain movements for the multi-domain proteins and high flexibility in the case of small proteins.

Based on the three experiments outlined above, it can be concluded that protein evolution is best modeled using 3D structure when there is a low sequence similarity amongst the homologous proteins.

Since protein structures are more conserved than sequences, it is logical to state that protein structures should be used to model the evolution of divergently related proteins. Structural comparison can be considered as a powerful “telescope”, enabling us to look back on earlier evolutionary history, as well as a more sensitive method than sequence comparison in determining protein function (Kolodny and Linial, 2004).

Chapter 2. Protein Structural Alignment

In the previous chapter, I have described the secondary structures present in proteins, the evolution of proteins, sequence-based and structure-based alignment techniques, and how protein evolution can be inferred through sequence and structural comparison. In this chapter, I will introduce six different structural comparison tools DALI, LOCK, CE, TM-align, STRAP, and Lovoalign as well as the algorithm and scoring system used by each of them.

2.1 Structural alignment tools

First, it is important to note that there is currently no universally accepted optimal way to align two structures. Choosing the right alignment tool or method depends largely on the question being asked. For example, is the goal to discover evolutionary relationships or structural relationships? Do we need to compare a single structure against the entire database, or two structures to each other? Do we need to compare the whole structure (global similarity) or only some parts (domains) of the proteins (local similarity)? The different programs considered in this chapter each possess unique strength and weaknesses, depending on the type of questions being asked (Sierk and Kleywegt, 2004).

Table 2.1. A summary of the most used tools with a brief description and the type of alignment.

Algorithm	**Tools Name and section**	**Description**	**Alignment type**
3D superposition	DALI (2.1.1)	**D**istance **M**atrix **Ali**gnment	Cα
	STRAP (2.1.5)	**ST**Ructure based **A**lignment	Cα
	TM-align (2.1.4)	Structural alignment based on **TM**-score	Cα
	Lovoalign (2.1.6)	Protein structure alignment based on convergent algorithm	Cα
Combinatorial	CE (2.1.3)	**C**ombinatorial **E**xtension of fragment alignment	Cα
Vector matching	LOCK/LOCK2 (2.1.2)	Hierarchical protein structure superposition using both secondary structure and atomic representations	SSE

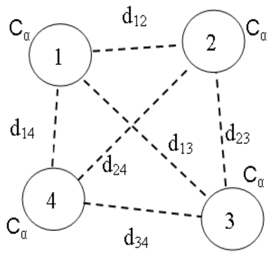
Different structural alignment tools utilize different algorithms to measure the structural similarity between two proteins (Eidhammer et al., 1999). For example, STAMP (Russel and Barton, 1992), STRAP (Gille and Frommel, 2001) and DALI (Ropodi, 2003) use the least square superposition algorithm which requires an educated guess concerning the rigid-body transformation. LOCK2, on the other hand, uses graph or vector matching algorithms (Shapiro and Brutlag, 2004). As evident by its name, CE makes use of the combinatorial extension (CE) algorithm (Shindyalov and Bourne, 1998a), while a dynamic programming alignment algorithm is used by TM-align (Zhang and Skolnick, 2005) and LOCK2 (Shapiro and Brutlag, 2004). Finally, Lovoalign uses a low order value optimization algorithm (Martínez et al., 2007b). The types of algorithms used are not limited to the above examples; rather, these examples are meant to portray the diversity of applied algorithms within structural alignment tools.

The name of each of these structural alignment tools along with their corresponding algorithm and the alignment type (Secondary structure element (SSE) or 3D structure (C_α)) are summarized in Table 2.1. I will now describe each of these tools in more details in the following sections.

2.1.1 DALI Structural alignment

The DALI algorithm was developed to achieve optimal pairwise alignment of protein structures. It assigns a one-to-one equivalence between the residues, based on the idea that similar 3D structures have similar intra-molecular distances. Each protein is represented as a 2D matrix of intra-molecular distances between Cα atoms, as shown in Figure 2.1a and 2.1b.

a) Protein A



b) Distance matrix for protein A

	1	2	3	4
1	0	d_{12}	d_{13}	d_{14}
2	d_{12}	0	d_{23}	d_{24}
3	d_{13}	d_{23}	0	d_{34}
4	d_{14}	d_{24}	d_{34}	0

Figure 2.1. Structural representation of DALI a) A graphical representation of the intra-molecular distances between the Cα atoms of protein A; b) Intra-molecular distance matrix for protein A (Can, 2007).

DALI algorithm can be explained by comparing two topologically different proteins (P1 and P2), each composed of a three stranded beta-sheet. Thus, we have strands a, b, c for P1, and strands and a’, b’, c’ for P2. The goal of this algorithm is to construct a distance matrix given the protein 3D structure and then detects the common patterns between the distance matrices as shown in Figure 2.2.

Firstly, the two sub-matrices, belonging to strand pairs (a-b) for P1 and strand pairs (a’-b’) for P2, are compared to match their common patterns, as shown in Figure 2.2, a and b. Next, the alignments obtained from comparing fragment pairs (a,b)-(a’,b’)and (b,c)-(b’,c’) will be merged into a larger alignment, called a ‘seed’, consisting of strands (a,b,c)-(a’,b’,c’), as shown in Figure 2.2c. Finally, a Monte Carlo algorithm, in combination with a branch-and-bound and/or neighbour walk algorithm, is utilized to optimize the similarity scores, obtained from the seed alignment by either accepting or rejecting basic moves (insertion or deletion), and to build up the full alignment (Holm and Sanderand, 1993).

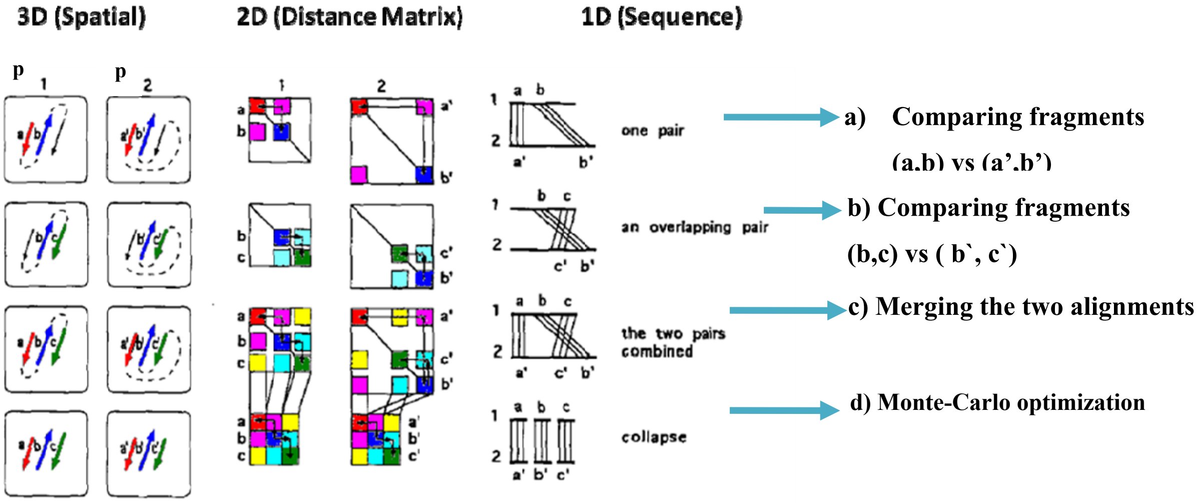

3D (Spatial)
2D (Distance Matrix)
1D (Sequence)
p 1
p 2
one pair
an overlapping pair
the two pairs combined
collapse
a) Comparing fragments (a,b) vs (a’,b’)
b) Comparing fragments (b,c) vs (b`, c`)
c) Merging the two alignments
d) Monte-Carlo optimization

Figure 2.2. Schematic view of DALI algorithm; Step1 involves the comparison of two sub-matrices by matching their common patterns (a) and b)). Step2 consists of merging the alignments, obtained from the sub-matrix comparison, into a larger alignment called a seed (c). Step3 involves optimization of the alignment scores, using a Monte-Carlo algorithm, by applying either deletion or insertion moves (d) (Holm and Sanderand, 1993).

2.1.2 LOCK algorithm: Hierarchical protein structure superposition using both secondary structure and atomic representation

The following section is largely inspired by an article written by Singh and Brutlag (Singh and Brutlag, 1997). The LOCK algorithm is based on a hierarchy of structural representation, ranging from the secondary structure level to the atomic level. This algorithm deals first with secondary structures, such as helices and strands, and then proceeds to atomic coordinate comparison. Secondary structures are processed first because they provide most of the stability and functionality of a protein, and as a result are more conserved than atoms during the evolution of a protein. The LOCK algorithm can be divided into two parts. Firstly, the alpha helices and beta strands are represented as vectors, as shown in Figure 2.3. The distance between two given vectors is computed by averaging the distances between the corresponding start, middle, and end points of the vectors. The angle is computed by taking the inverse cosine of the dot product of the two vectors.

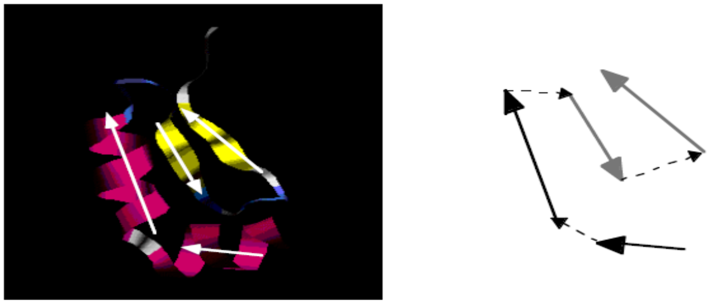
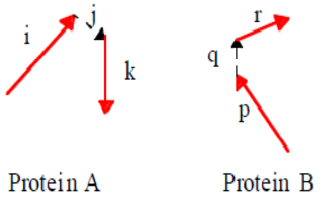


Figure 2.3. Representing secondary structure elements as vectors (Singh and Brutlag, 1997).

These comparisons are made possible through the use of a set of seven scoring functions ($S_1 – S_7$ in Figure 2.4). Five of these scoring functions are orientation independent and two of these scoring functions are orientation dependent, based on distances and angles between the vectors. In fact, orientation independent score is based on the comparison of internal angles and distances between helices or beta sheets of each of the two proteins (i.e. comparing the angle between vector i and *k* in protein A to the angle between vector *p* and *r* in protein B). Conversely, orientation dependant scoring is based on comparing the individual vector distances and angles of each protein with the other protein (i.e. comparing the orientation of vectors *i* in protein A to the vector *p* in protein B), as shown in Figure 2.4. A dynamic programming algorithm is then used to detect the best local alignment of the two sets of vectors which consist of finding the longest set of optimally matched pairs.



Orientation Independent Scores:

$S_1 = S\{|\text{angle}(i,k) - \text{angle}(p,r)|\}$
$S_2 = S\{|\text{angle}(i,j) - \text{angle}(p,q)|\}$
$S_3 = S\{|\text{angle}(j,k) - \text{angle}(q, r)|\}$
$S_4 = S\{|\text{distance}(i,k) - \text{distance}(p,r)|\}$
$S_5 = S\{|\text{length}(k) - \text{length}(r)|\}$

Orientation Dependent Scores:

$S_6 = S\{\text{angle}(k,r)\}$
$S_7 = S\{\text{distance}(k,r)\}$

The function S is defined as follows (Gerstein and Levitt, 1996):

$$S(d) = \frac{2M}{1 + \left[\frac{d}{d_0}\right]^2} - M$$

where, M = maximum score
d = attribute value
d_0 = value at which score should be 0

Figure 2.4. Alignment of secondary structure vectors (for proteins A and B) based on seven scoring functions (Singh and Brutlag, 1997).

The second part of the algorithm uses atomic coordinates to improve the vector alignment previously obtained by minimizing the RMSD (as defined in Section 1.3.2.2) between pairs of closest atoms from the two proteins. Finally, the obtained alignments are processed further by combining both the number of well aligned atoms and the RMSD to obtain the optimal alignment, known as the core alignment.

The use of secondary structure information allows the detection of both global and local similarities, as well as rapid and efficient initial superposition of the two proteins. The hierarchical method of detecting optimal alignments among secondary structures, and then among atomic level alignments, makes the overall algorithm more flexible and faster. A comparison between the alignment results obtained by LOCK and those obtained using DALI demonstrates that LOCK manages to detect structural similarities accurately and also that it is able to detect a few low structural similarities missed by DALI.

A new version of LOCK, called LOCK2, has been developed to detect distant structural similarity by placing increased emphasis on the alignment of secondary structure elements (Shapiro and Brutlag, 2004).

2.1.3 Combinatorial Extension (CE) algorithm

The following section is heavily inspired by an article written by Shindyalov and Bourne (Shindyalov and Bourne, 1998b).

Instead of using dynamic programming, which is used by LOCK, or Monte Carlo optimization, which is used by DALI, CE uses combinatorial extension of an alignment path. The alignment path is constructed by aligning short fragments from two proteins. The generated Alignment Fragment Pairs (AFPs), eight amino acids in size, are based on the orientation of secondary structures, RMSD, residue distances, and local secondary structure distances. The AFPs are placed into a similarity matrix, and an alignment path is extended by computing the distances D_{ij} between AFP_i from protein A and AFP_j from protein B for all i and j, as shown in Figure 2.5 (Mettu, 2008).

Based on the previous definitions, the optimal alignment between two protein structures A and B, of length n_A and n_B, respectively, is the longest continuous path, P, of AFPs in the similarity matrix.

A Z-score is used to evaluate the statistical significance of the longest alignment path. Z-score is computed by estimating the probability of finding an alignment path of the same length, with the same number or a smaller number of gaps, when two random structures are compared.

Comparisons made between CE and DALI, have shown that structural similarities involving small proteins are usually detected solely by CE, making CE well suited to the task of detecting similarities in very short proteins. In addition, its speed and accuracy in finding an optimal structure alignment make CE an ideal choice for database scanning and detailed analysis.

Unfortunately, the major limitation of CE is its inability to find alignments that have different topology[4].

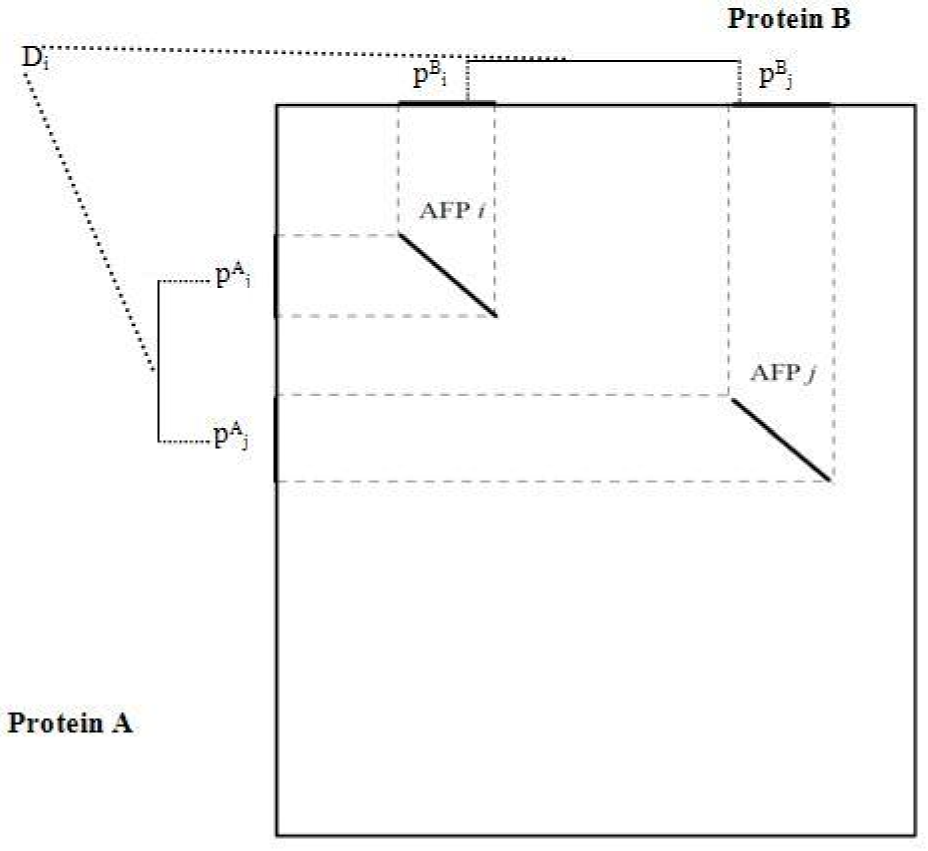


Figure 2.5. A similarity matrix showing the calculation of distance D_{ij} for alignment, represented by two AFPs (AFP_i for protein A and B at position i and AFP_j for protein A and B at position j). CE aligns structures and their sequences based on internal distances within each protein, rather than on inter-protein distance (Shindyalov and Bourne, 1998b).

2.1.4 TM-Align: A protein structure alignment algorithm based on the TM-Score

The following section is heavily inspired by an article written by Zhang and Skolnick (Zhang and Skolnick, 2005).

TM-Align uses the idea of the STRUCTAL (Levitt and Gerstein, 1998) and SAL (Kihara and Skolnick, 2003) algorithms, to speed up the process of identifying best structural alignment.

[4] Describes the sequential connectivity and spatial properties between two proteins.

Briefly, TM-Align uses three types of initial alignments, which are described in more detail in this section.

The first initial alignment is carried out using dynamic programming to align the secondary structures elements (SSE) of two proteins. A score of either 1 or 0 is given for an identical SSE, and a score of -1 is given as a penalty for a gap opening.

The second initial alignment is obtained by selecting the smaller of the two proteins and being compared and aligning it against different parts of the larger protein. In this way, the alignment with the best TM score (rather than RMSD) can be chosen. The TM score is defined as:

$$\text{TM-score} = \text{Max}\left[\frac{1}{L_{Target}}\sum_{i}^{L_{ali}}\frac{1}{1+\left(\frac{d_i}{d_0(L_{Target})}\right)^2}\right]$$

$$\text{Given} \qquad d_0(L_{Target}) = 1.24\sqrt[3]{L_{Target}-15} - 1.8$$

Where $d_0(L_{Target})$ is the distance, L_{ali} is considered to be the number of aligned residues, and d_i is the distance between the i^{th} pair of the aligned residues. It should be noted that the TM score combines three important features: alignment quality, alignment coverage and alignment accuracy. A TM-score is also normalized such that the resulting score is not dependent on the protein's size.

The final initial alignment is performed using a gap-opening penalty of -1 and two score matrices: the secondary structure matrix and the distance score matrix. Using the aligned residues from the three initial alignments, described above, a heuristic algorithm then incorporates an iterative procedure to rotate the protein structures according to a TM-rotation-matrix.

The advantage of using TM-score is that it places more weight on close matches than on distant matches, the result of which is that it is more sensitive to the global topology of the proteins than RMSD, as shown in the Figure 2.6. According to Figure 2.6, the structural superposition has the same topology, but the difference in the loop region creates a high difference in RMSD, while resulting TM-scores remain very similar.

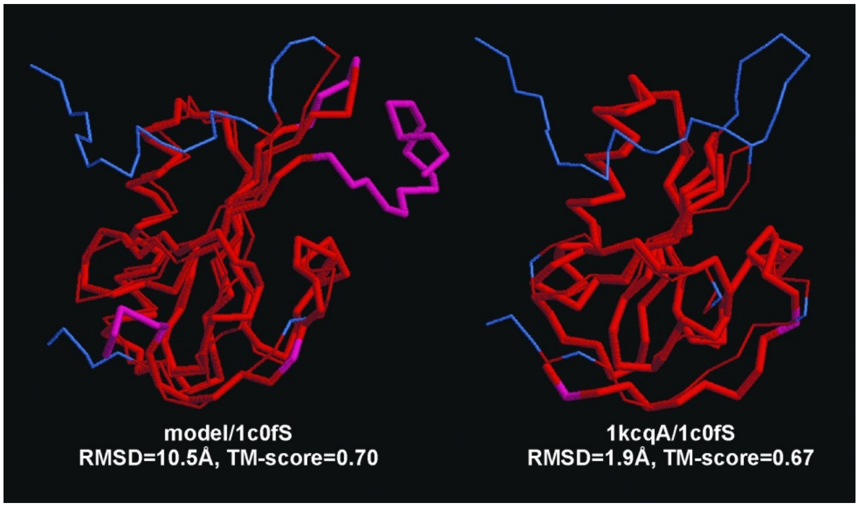


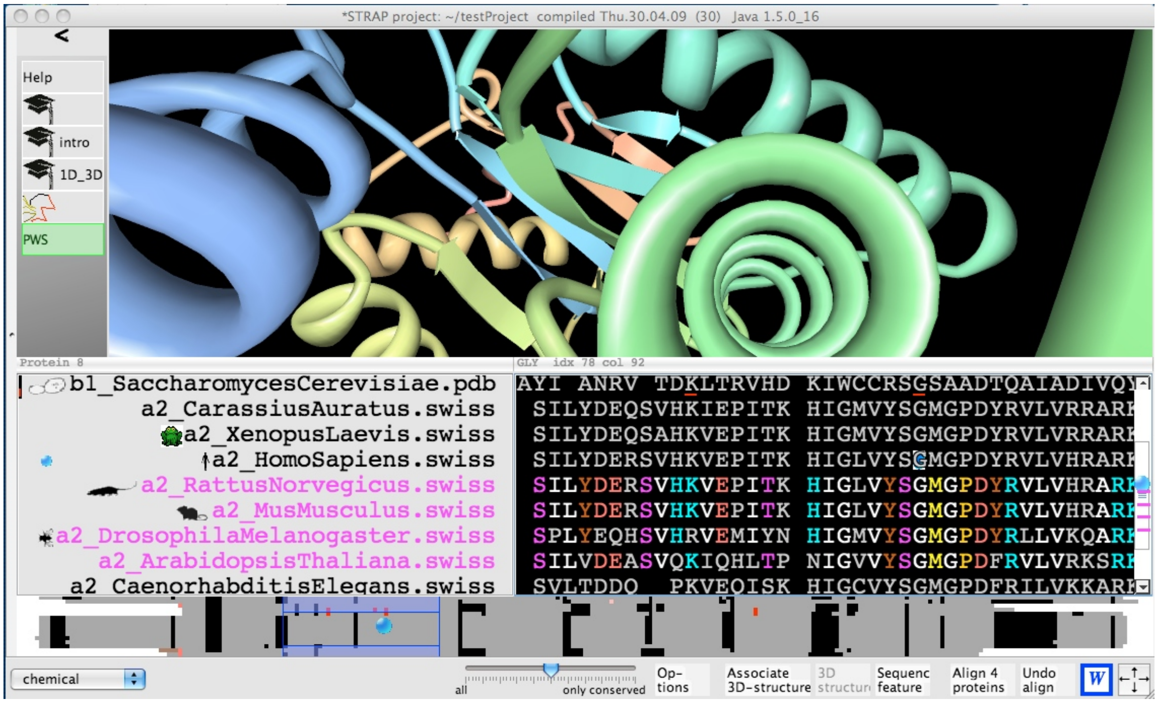
Figure 2.6. The superposition of two protein structures (1c0fS vs. model and 1c0fs vs. 1kcqA) shown to have similar topology according the TM-score measure (TM-score =0.70 and 0.67). However a loop variation can produce a rather high difference in the RMSD (10.5 and 1.9) (Zhang and Skolnick, 2005).

According to the comparison, the TM-Align algorithm is approximately four times faster than CE, and approximately twenty times faster than the DALI algorithm and has higher accuracy and coverage than the other programs. Additional studies, besides only the one discussed here, have also verified and emphasized the speed and accuracy of TM-align (Teichert et al., 2007; Madhusudhan et al., 2009). Due to the accuracy and speed of TM-align, a further comparison of this tool against other structural alignment tools will be made in Chapter 3.

2.1.5 STRAP (structural alignment program) with user friendly viewer

The following section is inspired largely by an article presented by Gille and Frommel (Gille and Frommel, 2001). **ST**ructural **A**lignment **P**rogram (STRAP) is a comprehensive tool with a user friendly interface used for the generation and refinement of multiple alignments of protein sequences using 3D coordinates. The efficiency of STRAP can be largely attributed to its simple visualization of Cα atom spatial distances within the alignment, as depicted in Figure 2.7. It is important to note that STRAP can only be used to perform alignments using either TM-align or a CE alignment program, and that it doesn't incorporate its own protein structural alignment tool.

There are a number of tools that perform multiple sequence alignment by incorporating 3D information. However, these tools are not built to handle a large number of sequences (more than 400 sequences), and the result is that the sequences do not fit in one screen and require awkward scrolling. STRAP was designed to address these issues and has the capacity to handle a high number of sequences, in addition to allowing the viewing of spatial distance in sequence alignments. STRAP is capable of detecting structural relationships between aligned sequences, handling high number of sequence alignments by stacking several proteins in one line, and viewing and editing alignments in several synchronized windows, all coordinated through a user-friendly protein viewer. In addition to its simple viewer, STRAP has reduced memory consumption, stores alignment results, and is extremely fast in loading proteins.


*STRAP project: ~/testProject compiled Thu.30.04.09 (30) Java 1.5.0_16

```
Protein 8                                       GLY  idx 78 col 92
bl_SaccharomycesCerevisiae.pdb        AYI  ANRV  TDKLTRVHD  KIWCCRSGSAADTQAIADIVQY
a2_CarassiusAuratus.swiss              SILYDEQSVHKIEPITK  HIGMVYSGMGPDYRVLVRRARK
a2_XenopusLaevis.swiss                 SILYDEQSAHKVEPITK  HIGMVYSGMGPDYRVLVRRARK
a2_HomoSapiens.swiss                   SILYDERSVHKVEPITK  HIGLVYSGMGPDYRVLVHRARK
a2_RattusNorvegicus.swiss              SILYDERSVHKVEPITK  HIGLVYSGMGPDYRVLVHRARK
a2_MusMusculus.swiss                   SILYDERSVHKVEPITK  HIGLVYSGMGPDYRVLVHRARK
a2_DrosophilaMelanogaster.swiss        SPLYEQHSVHRVEMIYN  HIGMVYSGMGPDYRLLVKQARK
a2_ArabidopsisThaliana.swiss           SILVDEASVQKIQHLTP  NIGVVYSGMGPDFRVLVRKSRK
a2_CaenorhabditisElegans.swiss         SVLTDDQ   PKVEQISK  HIGCVYSGMGPDFRILVKKARK
```

Figure 2.7. Screen-shot of STRAP showing an alignment of the β1 subunit of the proteasome colored according to charge and secondary structure types (Gille and Frommel, 2001).

2.1.6 Lovoalign: Protein alignment as a LOVO problem

Information provided in this section is inspired in large part by a detailed article written by Andreani et al. And Martinez et al. (Andreani et al., 2003; Martínez et al., 2007b). The

Lovoalign tool transforms the protein alignment problem into a Low Order Value Optimization (LOVO) problem. The LOVO problem is a generalization of the classic minimax problem, which aims to maximize the scoring functions in protein alignment, and it can be defined as follows: Given a set of real functions f1(x)….fm(x), the goal is to find x such that the maximum of f1(x)….fm(x) is maximal.

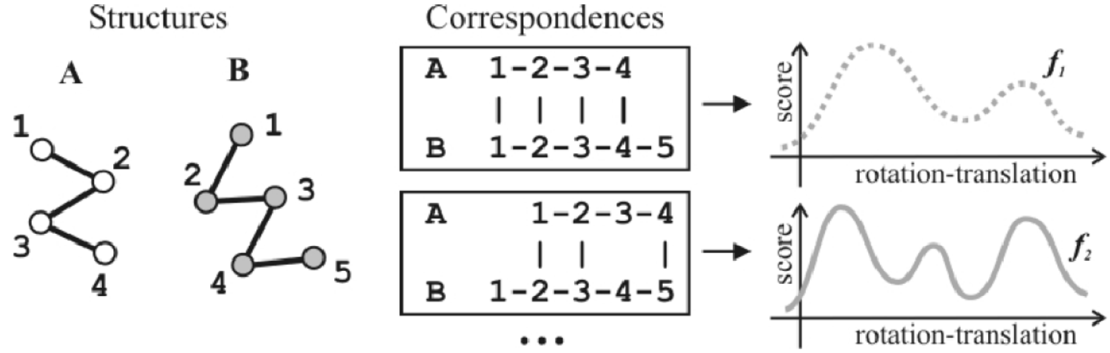


Figure 2.8. Protein alignment as a low order value optimization problem (Martínez et al., 2007b).

The LOVO method can be easily applied to the task of structural alignment. As mentioned previously, the goal of structural alignment is to find a correspondence by minimizing the RMSD between the C_α atoms through the application of transformation (translation and rotation) to one of the protein structures. Based on this definition, a function (i.e. scoring function) of rotation and translation can be defined using the obtained correspondences, as shown in Figure 2.8. Therefore, the goal of Lovoalign is to maximize the scoring function of each individual correspondence, and the correspondence with the maximal score will be selected as the optimal alignment.

The Lovoalign can be either used on the website[5] or downloaded on a Linux-based computer. Not only is Lovoalign fast, but it has also successfully maximized the STRUCTAL score (used by STRUCTAL alignment), as well as other distant-dependant scores. The detailed steps of the algorithm can be found in the article by Martinez et al. (Martínez et al., 2007b).

[5] The Lovoalign can be accessed at http://www.ime.unicamp.br/~martinez/lovoalign/home.html

Chapter 3. Comparison of structural alignment tools

In the previous chapter, six different alignment tools, including their functionalities, were described. In this chapter, I will introduce the techniques that can be used to compare the previously described structural alignment tools. In Section 3.1, I will start with the introduction to phylogenetic tree inference and comparison. In Section 3.2, I will introduce the different protein databases used to extract the necessary data. In Section 3.3, I will describe the different techniques used to compare and validate the alignment tools. In Section 3.4, I will describe the phylogenetic based method that I have utilized to compare different structural alignment tools. Finally, in Section 3.5, I will present the results obtained, as well as the relevant discussion and conclusion.

3.1 Phylogenetic tree inference and comparison

3.1.1 Definition and purpose of phylogenetic trees

This section is strongly inspired by a book on bioinformatics written by Mount (Mount, 2004).

Phylogenetics is an area of research concerned with identifying the genetic relationships between various organisms by relying on information extracted from DNA, RNA, or protein sequences (Potter, 2008).

An evolutionary tree, also known as a phylogenetic tree, depicts the evolutionary relationships between organisms. As shown in the Figure 3.1, a group of organisms, genes, or species can be used to construct a phylogenetic tree. A phylogenetic tree is composed of several different components. The leaves of the tree represent the taxa (a group of organisms), while internal nodes (also termed ‘divergence points’) represent the hypothetical ancestor of the taxa rooted at this node. Branches (or lineages) of a phylogenetic tree connect the different nodes. Finally, the ancestral node, which can be visualized as the root, represents the ancestor of all taxa presented at the leaves of the tree. Usually, the number of sequence changes that occurred prior to the point of separation or speciation can be characterized by the length of each branch. In a situation where two species have the same branch length from their respective leaf to the common ancestor, it is suggested that they have evolved at relatively the same rate.

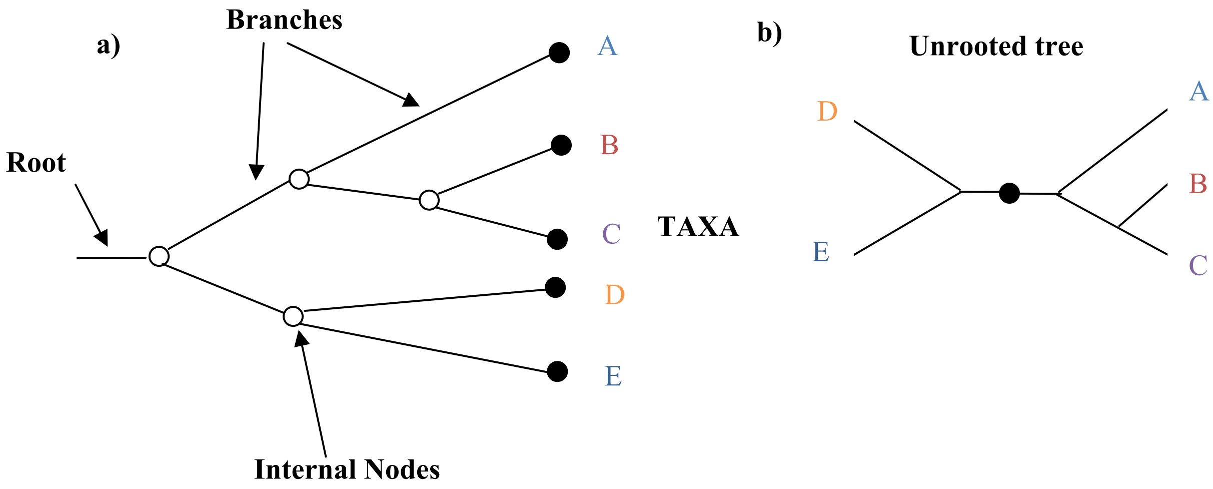


Figure 3.1. a) A simplified tree illustrating common phylogenetic tree terminology. A phylogenetic tree usually consists of the root, branches, internal nodes and leaves. b) The corresponding unrooted tree.

A phylogenetic tree is usually binary, meaning that only two branches stem from each node. A binary tree can either be rooted or unrooted; an unrooted tree is one that makes no assumptions about the position of the root, as shown in Figure 3.1b.

3.1.2 Multiple sequence alignment used to generate phylogenetic trees

To construct a phylogenetic tree, the protein sequence must first be aligned using a multiple sequence alignment tool. Once these results are obtained, a phylogenetic analysis algorithm is utilized to infer the resulting phylogenetic tree.

In this project, a specific protocol is used to infer the phylogenetic trees. As explained in Section 1.3.2.1, ClustalW can be used to perform an MSA for a series of protein sequences, and this is what has been done in this project. Gblocks (Castresana, 2000) is then used to extract poorly aligned regions or divergent regions from the MSA obtained via ClustalW (Thompson et al., 2002), while minimizing the loss of informative sites. Finally, the factored MSA can be used to infer the phylogenetic tree through the use of different tree analysis algorithms, as explained in detail in the following section.

3.1.3 Methods to infer phylogenetic trees

There are three main methods used to infer phylogenetic trees: maximum parsimony[6], distance-based method (Section 3.1.3.1), and maximum likelihood (Section 3.1.3.2). In this project, I have used both a tool called FITCH, which incorporates a distance-based method, as well as PhyML, which uses a maximum likelihood method.

3.1.3.1 Distance based method

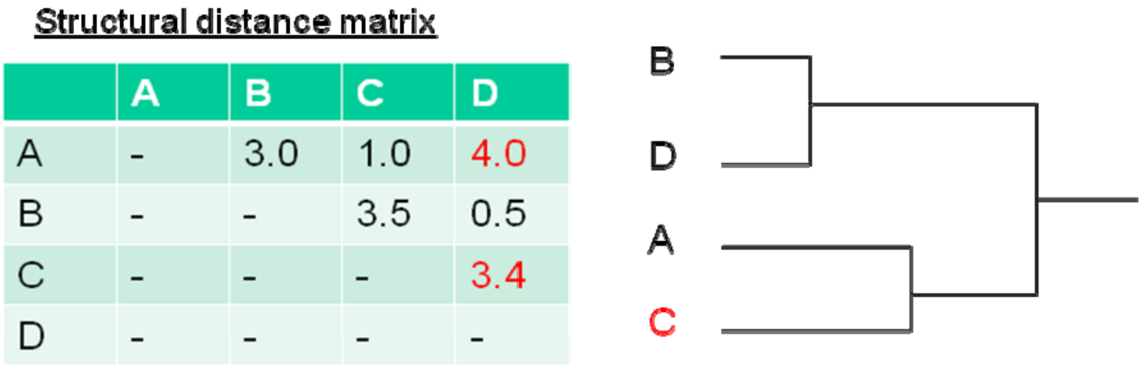
The distance based method uses genetic distances between two sequences in a MSA, which is the number of positions where the sequence has undergone mutation. Neighbours are represented by the sequences that have the smallest distances and share a common ancestor. The goal of the distance-based method is to determine the tree that correctly positions these neighbours.

The PHYLIP package (Felsenstein, 2004) contains a series of distance-based analysis programs: FTICH, KITCH and NEIGHBOUR. In this project, I have utilized the FITCH method, described below, to generate the phylogenies.

FITCH METHOD (Kuhner and Felsenstein, 1994)

This method estimates a phylogenetic tree by searching for the tree that minimizes the sum of squared differences (least square problem) between the actual distances and that of the distance matrix corresponding to the optimal tree. This method takes the distance matrix, which contains the pairwise structural alignment distances between different proteins (shown in Figure 3.2), as input, and creates the optimal consensus tree. FITCH does not consider a molecular clock, meaning that the rates of evolution along the branch will vary. This is the primary reason why FITCH is used in this project in order to infer phylogenetic trees given a distance-based matrix.

[6] A method that infers the phylogenetic tree by minimizing the number of steps required to generate the observed changes in the protein sequences. Since this method will not be used in this project, it will not be described in detail.



Structural distance matrix

	A	B	C	D
A	-	3.0	1.0	4.0
B	-	-	3.5	0.5
C	-	-	-	3.4
D	-	-	-	-

Figure 3.2. Constructing a phylogenetic tree using the pairwise comparison of 4 sequences (A, B, C, D) included in the structural distance matrix.

3.1.3.2 Maximum likelihood

Maximum likelihood (ML) is a phylogeny reconstruction method that starts with a probabilistic model for protein or nucleotide substitution. This model is adjusted until an optimal tree with the highest probability of representing the observed data[7] is found. There are several probabilistic models, such as: a) the Whelan and Goldman (WAG) model (Whelan and Goldman, 2001), which uses an ML method to estimate an amino acid replacement matrix, called a WAG[8] matrix, for each protein family; b) Dayhoff model (Dayhoff et al., 1972), which uses the maximum parsimony method to create Point Accepted Mutation (PAM) matrices to identify acceptable amino acid changes that maintain the function; c) Jonathan, Taylor and Thomas (JTT) model (Jones et al., 1992), which uses an automated procedure, based on the Dayhoff method, to produce a replacement matrix from a much larger database, faster and more efficiently.

In this project, I have used a program called Phylogenies by Maximum Likelihood (PhyML) (Guindon and Gascuel, 2003), which uses the JTT model. I have chosen to do so because it's faster than the other two probabilistic models. In addition, I have chosen to use the a-la-carte web-based tool[9] (Dereeper et al., 2008), which combines the programs PhyML, ClustalW, and GBLOCKs. The reason for using the "a-la-carte-website" lies in the fact that it provides a very convenient and user-friendly interface for executing all three programs at once.

[7] It refers to the observed sequence variations detected in the column of an MSA.

[8] WAG matrix can be downloaded from www.ebi.ac.uk/goldman/wag/wds.dat

[9] This tool can be accessed at www.phylogeny.fr

3.1.4 Phylogenetic tree comparison

This section is strongly inspired by articles written by Robinson and Foulds and Pattengale et al. (Robinson and Foulds, 1980; Pattengale et al., 2007).

There are many different methods available for phylogenetic tree comparisons. However, in this project, the Robinson-Foulds (RF) metric, which is the most common distance measure between two trees, has been used for the phylogenetic comparison. This method can be described as follows. In a tree, each edge defines a bipartition of leaves. The goal of the RF metric is to count the number of branches which define the same bi-partition. In my project, I used an RF algorithm carried out in C++[10]. The C++ program produces a correlation coefficient between zero and one, where zero indicates that two trees are identical and 1 indicates that they are completely different. This program finds the number of branches which are different in both trees and it then divides it by the total number of branches.

3.2 Protein databases

The previous chapter provided an overview of different methods used for the comparison of phylogenetic trees. Another important aspect of phylogenetic tree comparisons, either using structural or sequential information, is the choice of protein database. Therefore the next section (3.2) will be completely devoted to discussions regarding two protein databases used in this project: PDB, for structural info and PALI, for structural protein families.

3.2.1 Protein Data Bank

The Protein Database Bank (PDB) (Westbrook et al., 2002) was created at Brookhaven National Laboratory, and, since its inception, has been considered the reference database or repository of choice for 3D structural data belonging to large, biological molecules. The data stored in this database is obtained using x-ray crystallography or Nuclear Magnetic Resonance (NMR) spectroscopy. As of now, there are 55,519 protein structures stored in the database. In addition, in 2003, the World Wide Protein Data Bank (wwPDB) was established, consisting of

[10] Program is developed by Denis Bertrand

three main organizations[11]: RCB PDB (USA), PDBe (Europe), and PDBj (Japan), all of which act as storage and data processing centers for PDB (Berman et al., 2006).

The data stored in the PDB correspond to three different formats: PDB file format (the original PDB format, which is widely supported and easily interpreted), Macro-Molecular Crystallography Information File (mmCIF), which was designed to correct some discrepancies in the PDB format file, and, finally, Protein Data Bank Markup Language (PDBXML), which is based on XML schema. In this project, the Open Macro-Molecular Structure (OpenMMS)[12] (Greer et al., 2002) software has been used to store and customize the PDB data for further analysis, in order to structure the Helix Explorer database described in more details in Chapter 4.

.

3.2.2 Database of Phylogeny and Alignment of homologous protein structures

Since this project uses information from the PALI database in order to compare the secondary structure-based phylogenetic trees against the sequence based phylogenetic trees, I will briefly explain the functionality of this database (Balaji et al., 2001). The PALI database consists of structure-based sequence alignments, based on the **P**rotein **FAM**ily (PFAM) database (Bateman, 2004), as well as structure-based alignments of homologous domains derived from the Structural Classification of Proteins (SCOP)[13] database (Murzin et al., 19995). PALI also uses STAMP (Russel and Barton, 1992) to perform the structural alignments and the FITCH method to construct the associated phylogenetic trees. Finally, the PALI database provides an efficient and useful resource to detect and analyze the relationship between protein sequences and structural variations for the chosen protein family.

3.3 Existing methods and comparison of structural alignment tools

Having explained the main components of the two databases used in this project, the last, but not the least, topic to cover will be the tools and methods used to evaluate and compare the

[11] The goal of this organization is to ensure a single uniform archive of PDB data. This information was verified on May 1st , 2010.

[12] OpenMMS software can be downloaded from: openmms.sdsc.edu

[13] SCOP provides a detailed and comprehensive description of the structural and evolutionary relationships of the proteins of known structures.

results obtained from structural alignment programs. Even though this project will solely use phylogenetic trees, obtained from structural protein alignment, to evaluate the results of structural alignment tools, it is important to cover other existing methods.

This section is heavily inspired by several informative articles (Yang and Honig, 2000; Kolodny et al., 2005; Kim and B., 2007).

3.3.1 CATH as a gold standard

There are a series of widely used methods that compare similarly aligned structures against a protein structure classification database , called CATH[14] (Orengo et al., 1997), as well as for verifying if the aligned structures being classified are indeed truly similar, or homologous (Novotny et al., 2004).

CATH database (Orengo et al., 1997) contains four main levels of classifications: a) a protein class, the simplest level that describes secondary structure composition; b) architecture, which describes secondary structure shapes, such as barrels; c) topology, which describes sequential connectivity; and d) homologous superfamily.

Programs differ in their ability of accurately detect the protein homologs according to the CATH structure database. The detection ability of different structural alignments can be evaluated by, through the use of pairwise comparison. First a non-redundant subcategory of CATH is created, in which a single member of each protein family, as opposed to the entire family set, is chosen as a representative to reduce computational cost. Next, a pairwise comparison between a given protein and one of these protein representatives is performed. Finally, the results are compared against a specific threshold.

[14] CATH is a database of hierarchical classifications of protein domain structures. The reasons why this structural database is name CATH, is because the classification is divided into four different levels: Class, Architecture, Topology and Homologous Super family

3.3.2 Curves Comparison

This section is inspired by the text of a number of articles (Wikepedia; gribskov and Robinson, 1996; Sierk and Pearson, 2003; Sam et al., 2006; Qi et al., 2007).

The success of different methods in detecting the same domain, defined in the CATH database, can be assessed by comparing the Receiver Operating Characteristics (ROC) curves. ROC curves are used in various literatures to determine how a given scoring scheme agrees with a particular gold standard (i.e. CATH database) through the following steps: a) a pairwise structural alignment, for each pair of proteins, is performed in the dataset, and their corresponding alignment scores (or RMSD) are recorded; b) if the protein pairs correspond to the CAT classification, they will be considered as a true positive (TP); those that do not are deemed false positive (FP); and c) finally the fraction of FPs ("called specificity") will be plotted against the fraction of TPs (called sensitivity). The probability of correct classification is usually defined by the area under the curve.

3.4 A phylogenetic-based method to compare structural alignment tools

3.4.1 Problem statement

As explained in Section 3.3.1, CATH has been used as a gold standard classification to determine whether two aligned structures are truly homologues based on a given threshold. In, Section 3.3.2, another study computed the probability of a correct classification (using CATH as a gold standard) by plotting the TP and FP, based on a combination of the CATH classification and RMSD for each pair-wise alignment, using ROC curve. The area below the ROC curve provides the probability of a correct classification.

None of the previously mentioned protocols have taken into account the notion of protein evolution when evaluating the different structural alignment tools. Therefore we have decided to use a different approach, by evaluating the different structural alignment tools based on their ability to model the protein evolution.

To accomplish this goal a series of pairwise alignments have been carried out using different structural alignment tools (CE, Lovoalign, TM-Align and LOCK2), as described in Chapter 2.2, for protein families extracted from PALI database. The RMSDs obtained from these alignments are stored in a symmetric matrix, which will later become the input to the FITCH program, creating the corresponding phylogenetic trees. The resulting trees are then compared against each other using the RF method, as described in chapter 3.1.4.

It is worthwhile to mention the reason why structural information are used to construct the phylogenetic trees (to model protein evolution) is that structures are more conserved, providing a more accurate information specially when dealing with protein families which have divergently related proteins (as described in Section 1.3.2.3). Furthermore, the reason behind using these four alignment tools (CE, Lovoalign, TM-Align and LOCK2) is that previously many literatures have compared these tools, validating their accuracy using the previously mentioned methods mentioned. Finally this experiment can be indicative as to whether distance between phylogenetic trees is a promising method of evaluating and comparing different structural alignment tools.

According to the literature, the TMalign can almost always find close structural analogs with an average RMSD of 3 A and 87% alignment coverage, proving to be more accurate than other alignment tools (as explained in Section 2.1.4). Therefore, we have chosen TMalign as a gold standard to compare alignment tools amongst which CE, LOCK2 and Lovoalign, can be utilized to construct an accurate evolutionary model of a given protein family for a given data set using phylogenetic tree comparison (assuming that this method provides the most accurate evolutionary history for the given protein family).

To further extend our idea, we have decided to compare the structure-based phylogenetic trees produced by each of the four structural alignment tools (using RF distance), for each given protein family, against the sequence-based trees constructed by JTT (used a gold standard); determining which protein family has a higher or lower correlation between their structure-based and sequence based phylogenetic trees. For both studies, the obtained results will be checked against the PALI database to determine which alignment tool will give the similar result for the given protein family the database itself.

The overall objectives of this project are summarized below:

1) To determine the ideal alignment tool (amongst CE, Lovoalign and LOCK2) in modeling the evolution of a given protein family for a given data set using the RF distance algorithm (described in Section 3.1.3.2) in comparison to TM-Align.

2) To identify the alignment tools that will be able to infer the best structure-based phylogenetic trees in comparison to the sequence-based phylogenetic trees.

3) A stand-alone project has also been conducted aiming to construct the phylogenetic trees using merely secondary structures rather than 3D structures. This will allow us to determine if the HE database is a suitable tool (or source) to model the evolution of a given protein family. In order to link this study with the previous one, we have decided to compare the phylogentic trees created by the Helix Explorer program to those created by different structural alignment tools (TM-Align, Lovoalign, CE and LOCK2). This goal will be explained in details in Section 4.2.2.

3.4.2.1 Comparing the structure-based phylogenetic trees

In the previous section (Section 3.4.1) I have introduced the different goals that this project aims to achieve. In this section, I will introduce the methods used to achieve each of these goals. In section 3.4.2.1, I will introduce the steps taken to compare the different structural alignment tools that have been described in chapter 2. In section 3.4.2.2, I will introduce the method used to compare the sequence-based and structure-based phylogenetic trees.

All the structural alignments tools (Lovoalign, CE, LOCK2 and TM-Align) described in Chapter 2 are used to construct structure-based distance matrices. It should also be noted that instead of testing the entire protein family at once, the family is broken down into smaller subsets to facilitate the analysis (where the smallest set is subset 4). The average of results obtained from all possible combinations of each subset is then calculated. The following are the steps used to compare these alignment tools, using Perl[15] scripting.

[15] Documentation available at http://www.perl.org/

First, for each of the PDBs belonging to a given protein family (listed in Table 3.1; see Annex I), a pairwise alignment is performed using each of the four alignment tools (TM-Align, CE, Lovoalign, and LOCK2). Next, the RMSD, obtained from each of the previous alignments, is stored in a symmetrical matrix corresponding to each of the alignment tools. Then, given the RMSD-based distance matrix, FITCH is used to infer the phylogenetic tree. In order to determine the correlation between each of these tools in comparison to TM-Align, the correlation coefficient between each of their corresponding phylogenetic trees is computed, using the RF algorithm described in Section 3.1.4. The value '1' denotes no similarity and '0' indicates that the two trees are identical. Finally, the resulting RF correlation coefficients are placed in another matrix, which is then used to create a graph, as shown in Figures 3.3 to 3.10.

Families with a poor correlation between structure-based and sequence-based phylogenetic trees and a high sequence similarity

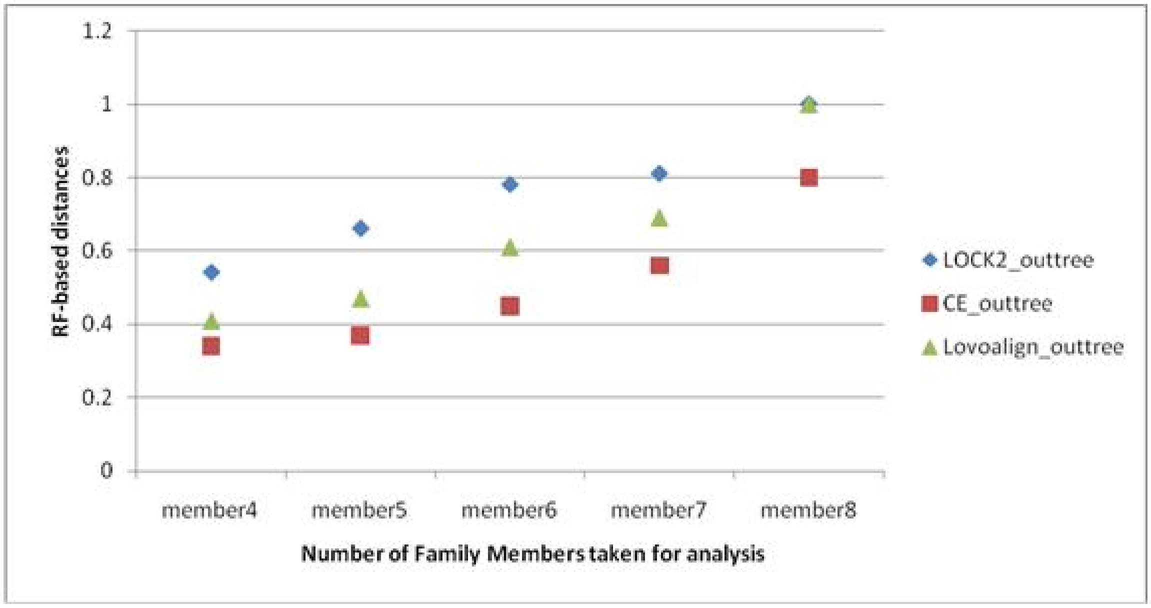


Figure 3.3. The average RF-based correlation coefficients comparing the results of LOCK2, CE, and Lovoalign against those of TM-Align for the small kunitz family.

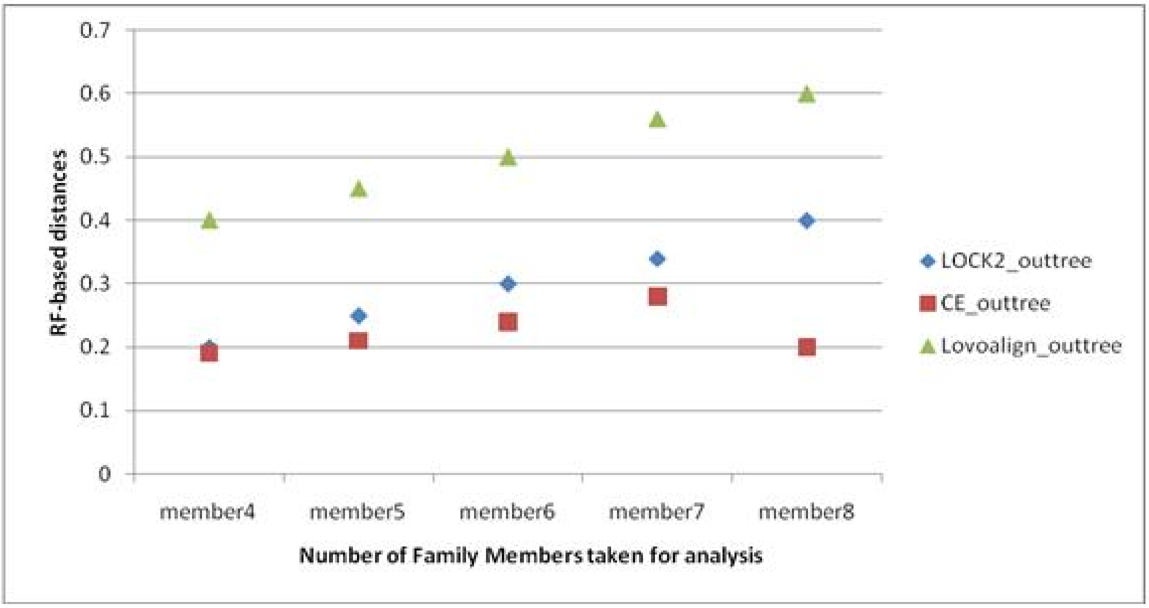


Figure 3.4. The average RF-based correlation coefficient comparing the results of LOCK2, CE, and Lovoalign against those of TM-Align for the Serpins family.

As shown in Figure 3.4 for the Serpins family (for all the family subsets), CE has obtained the lowest correlation coefficient in comparison to TM-Align. In the case of the small kunitz family (for all the family subsets), as shown in Figure 3.3, once again CE performs better than other tools. In both cases, the RF-based correlation coefficient between phylogenetic trees, obtained by CE and TM-Align respectively, is the lowest when compared to other tools. CE could possibly be the ideal alignment tool to model the evolution of protein families that have poor correlation between structure-based and sequence-based phylogentic trees.

Families with a poor correlation between structure-based and sequence-based phylogenetic trees and a low sequence similarity

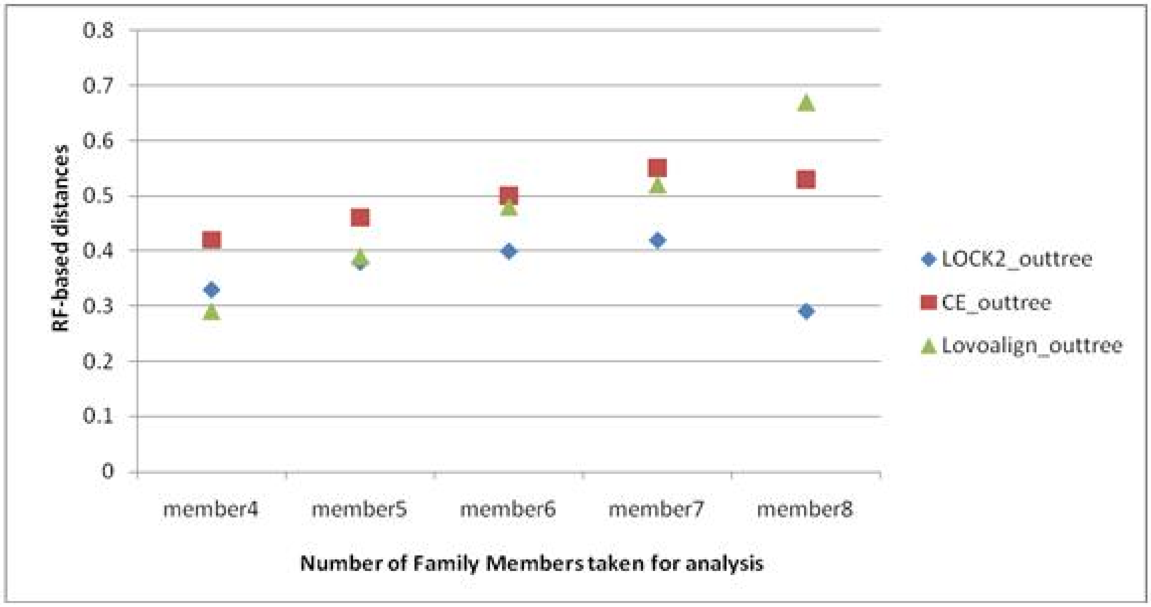


Figure 3.5. The average RF-based correlation coefficient comparing the results of LOCK2, CE, and Lovoalign against those of TM-Align for the short chain cytokines family.

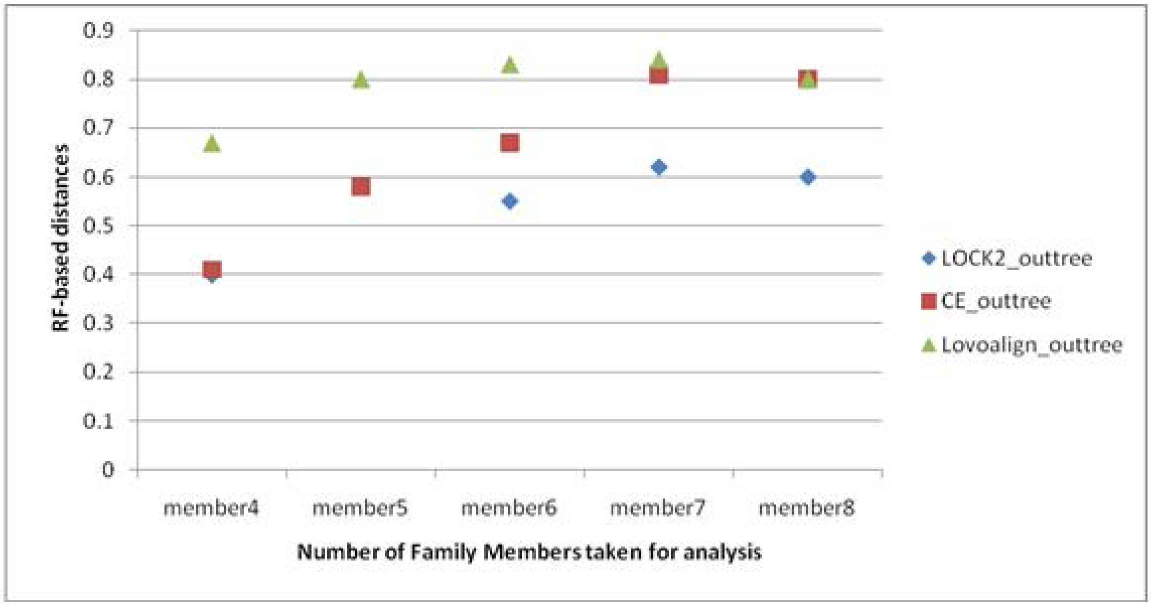


Figure 3.6. The average RF-based correlation coefficient comparing the results of LOCK2, CE, and Lovoalign against those of TM-Align for the calmoduline-like family.

As shown in Figure 3.5 for short-chain cytokines, Lovoalign has the lowest correlation coefficient for the family subset that has only four members. However, for the rest of family subsets, the performance of LOCK2 was the best. For the calmodulin-like family, as shown in Figure 3.6, CE and LOCK2 have the lowest correlation coefficient for the family subset of four members. However, for the rest of family subsets, LOCK2 again managed to maintain the lowest RF-based correlation coefficient in comparison to the other two alignment tools. Based on these two cases, it can be suggested that, overall, LOCK2 is a good candidate for modeling the evolution of protein families with poor correlation coefficient between structure-based and sequence-based phylogenetic trees in addition to low sequence similarity.

Families with a high correlation coefficient

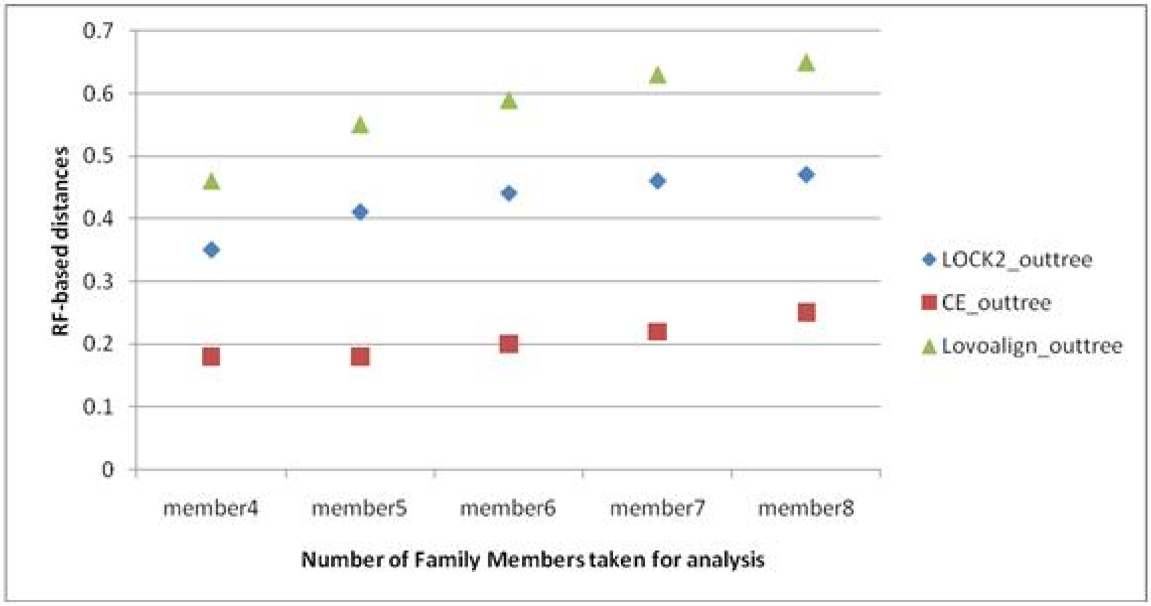


Figure 3.7. The average RF-based correlation coefficient comparing the results of LOCK2, CE, and Lovoalign against those o f TM-Align for the Fe-mn-Superoxide-dismutase family.

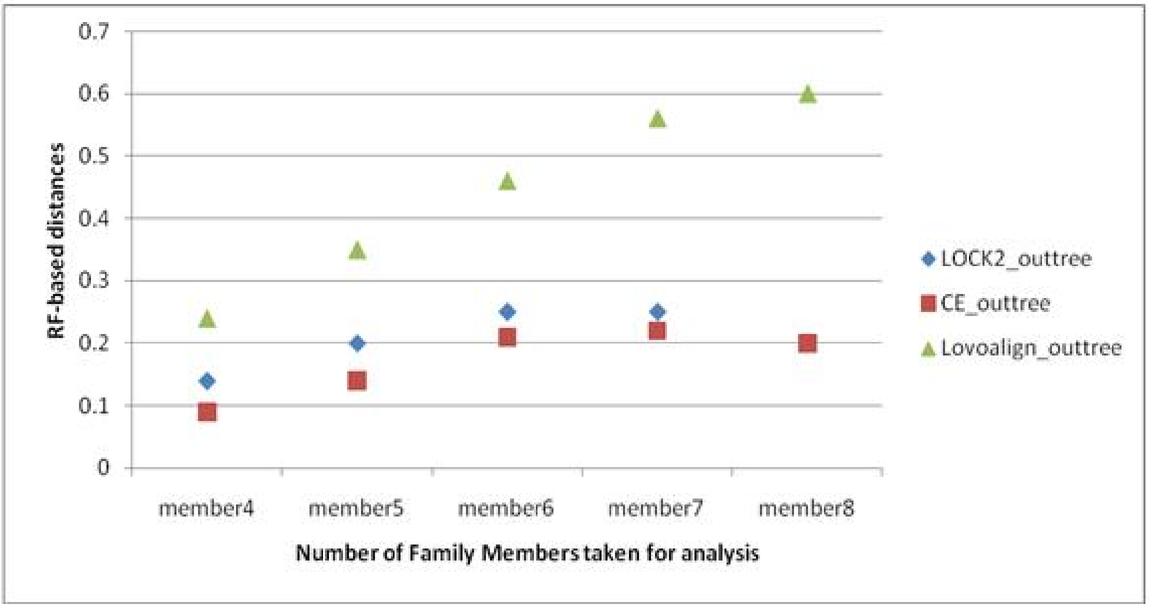


Figure 3.8. The average RF-based correlation coefficient comparing the resuls of LOCK2, CE, and Lovoalign against those of TM-Align for the TIM family.

For both Fe-Mn superoxidase and TIM (Figures 3.7 and 3.8 respectively), CE performed the best by maintaining a lower correlation coefficient in comparison to the other tools. This suggests that CE should be considered for situations that require the modeling of the evolution of protein families that have a high correlation between both sequence-based and structure-based phylogenetic trees.

3.4.2.2 Comparing the structure-based phylogenetic trees against the sequence-based phylogenetic trees

In this section the methods used to obtain the sequence-based phylogenetic trees are described. Similar to the method described in the previous section, these methods have been developed using Perl.

Firstly, the FASTA format of the protein sequence of each of the PDBs for the given protein family listed in Table 3.1 (see Annex I) is obtained from the PDB website. It should be noted that, unlike the previous method, in this method the entire family is selected for analysis. The reason for this is that an “a la carte” web-based tool (as described in Chapter 3.1.3.2) has been utilized to create a sequence-based phylogentic tree, rather than Perl scripting. Using the “a la carte” website, the phylogenetic tree is constructed in the following way: a) firstly, ClustalW is used to determine multiple sequence alignment using the default parameters; b) next, the aligned sequences are uploaded to Gblocks, also using the default parameters; c) finally, the curated sequence alignment is exported into the PhyML program to infer the phylogenetic tree, based on the JTT model.

After this has been complete, the correlation coefficient between the resulting sequence-based phylogenetic trees and the structure-based phylogenetic trees (obtained in Section 3.4.2.1) is computed using the RF distance algorithm. Finally, the same procedure is repeated for all the protein families, and the results are stored in a symmetric matrix, which is later used to construct a graph for further analysis, as shown in Figure 3.9.

The RF-based distance between the phylogenetic trees created by all four alignment tools and the ones created by the JTT turns out to be rather high, for proteins families with a poor-correlation between structure-based and sequence-based phylogenetic trees (Small kunitz, Serpins, Calmoduline-like and short-chain cytokines). For Fe-MN-superoxidase both CE and TMalign have performed better by maintaining a lower RF-based distance. Finally, for TIM protein family, Lovoalign has produced the lowest RF-based distance.

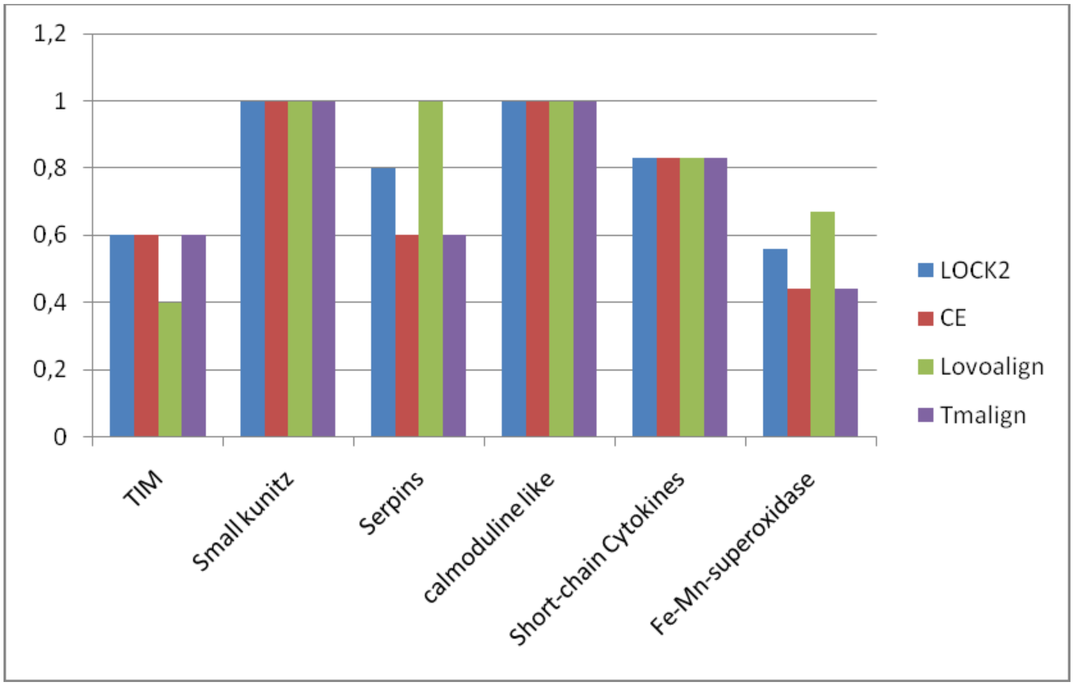


Figure 3.9. The comparison between the structure-based and sequence-based phylogenetic trees. In this example lock2, ce, lovoalign, and TM-Align are compared against JTT.

3.4.3 Improvement and limitations

Firstly, in this project, all of the distance measures used were RMSD, in order to maintain a standard score for each of the alignment tools. However, as explained in chapter 1.3.2.2, RMSD has a series of pitfalls, not least of which are the facts that it is size dependant and that a small deviation in the structure can cause the RMSD to be high. To overcome these issues, a couple of alternatives could be investigated in the future. Since TM-score is the most optimal score, as explained in Section 3.4.2, the RMSD obtained from each alignment tool could be converted into a TM-score. Consequently, the TM-score can then be used to construct the phylogenetic trees. Another alternative would be to convert RMSD to SDM (structural distance metric) according to the formula used in the PALI database study (Balaji et al., 2001). The new SDM will be size-independent and will overcome the RMSD issue.

Secondly, TM-Align has been selected as the optimal tool according to the existing literature. However, as a future improvement, a local comparison could be made between different alignment tools by considering the average RMSD, the coverage (number of aligned residues divided by protein length) and the CPU time. The obtained results could then help us to select the most accurate tool.

Finally, due to time constraints, only six protein families were analyzed in this study. In the future, a larger number of protein families with increased numbers of protein members should be considered in order to obtain a more concrete conclusion.

Chapter 4. Helix Explorer

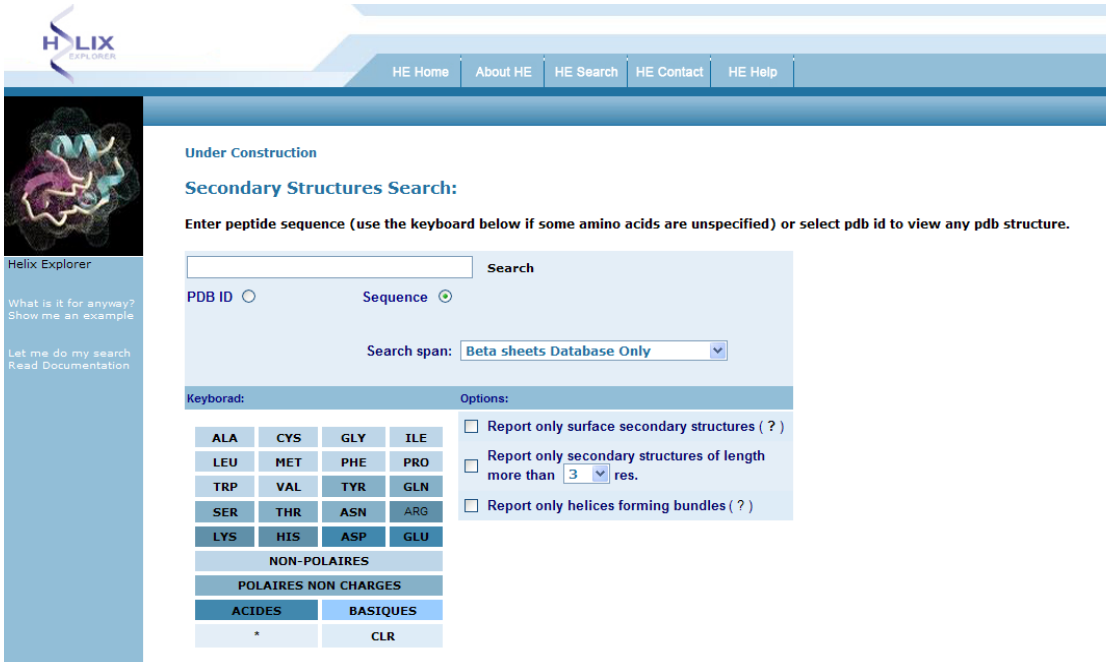
In this chapter, I will introduce the Helix Explorer database (HE), which contains different information about protein secondary structures. In Section 4.1, I will provide a brief introduction to HE, including its different functionalities and database architectures. Since one of the goals of this project is to expand HE, so that it can be used as a tool to construct phylogenetic trees, in Section 4.2 I will introduce the new functionalities added to HE. The algorithm developed to extract secondary structure data from HE in order to construct phylogenetic trees, and the results obtained, are also discussed in detail.

4.1 What is Helix Explorer?

This section is inspired by a thesis written by Mohammad Marrakchi (Marrakchi, 2006).

Helix Explorer a tool designed to extend the functionality and search capabilities in the PDB developed by Mohammad Marrakchi. This web-accessible tool allows an individual to browse through the existing secondary structures in PDB by providing the following three features:

i) A search engine, allowing one to search for a series of secondary structure elements using an amino acid sequence, a regular expression query, or the PDB identifier, as shown in the Figure 4.1.



Under Construction

Secondary Structures Search:

Enter peptide sequence (use the keyboard below if some amino acids are unspecified) or select pdb id to view any pdb structure.

PDB ID ○ Sequence ◉

Search span: Beta sheets Database Only

□ Report only surface secondary structures (?)

□ Report only secondary structures of length more than 3 res.

□ Report only helices forming bundles (?)

Figure 4.1. The secondary structure search view within Helix Explorer.

ii) Neighbour view, which displays all the distances and angles of the secondary structures for the given protein located in the closest proximity, known as neighbours, to each secondary structure in the other proteins, as shown in Figure 4.2. HE uses three different distance measures to compute the distance between two neighbouring secondary structures. These three different distance measures are explained below:

1) Minimal distance, D_{min} is the smallest euclidean distance of the distances measured between any two Cα atoms, each belonging to one of the two secondary structures, as shown in Figure 4.3a. Mathematically, this can be defined as:

$$D_{min}\ (S, P) = \min\ \{d\ (\square_{si}\ \square_{pi})\ |\ i<m,\ i<n\}$$

2) The center distance D_{center}, is defined as the distance between the geometric centers of the two secondary structures, considered as the center of a series of Cα atoms, as shown in Figure 4.3b. This distance metric is mathematically defined as:

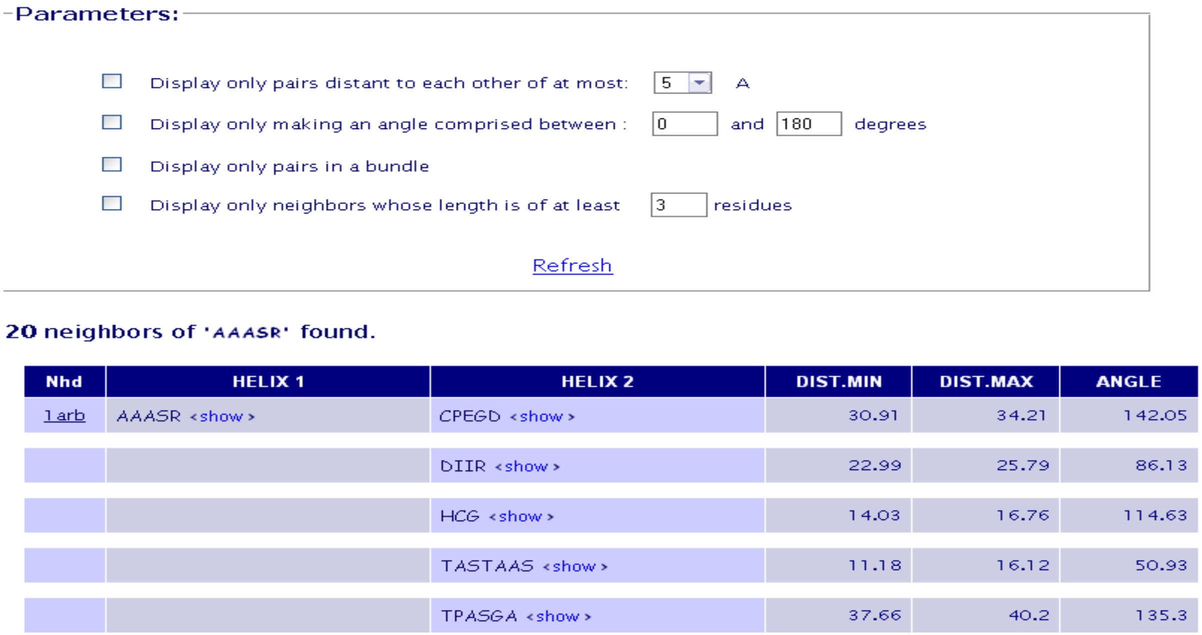
$$D_{ctr}\ (S, P) = d\ (\text{Center}(s), \text{Center}\ (P))$$

3) Average distance, D$_{avg}$, is computed by averaging the minimum distances between each of the Cα atoms in the shorter of the two secondary structures (i.e. the one with fewer amino acids in its sequence). However, when the two secondary structures are identical in length, one of them will be chosen randomly to represent the smallest distance, as shown in Figure 4.3c. The D$_{avg}$ is described by the following formula:

$$D_{avg}(S,P) = \left(\sum_{i=1}^{m} \min\{ d(\alpha_{S_i}, \alpha_{P_j}) \mid j \leq n \} \right) / m$$

where d is the cartesian distance and

$$d(\alpha_{S_i}, \alpha_{P_j}) = \sqrt{(x_{S_i} - x_{P_j})^2 + (y_{S_i} - y_{P_j})^2 + (z_{S_i} - z_{P_j})^2}$$



Parameters:

□ Display only pairs distant to each other of at most: 5 A

□ Display only making an angle comprised between : 0 and 180 degrees

□ Display only pairs in a bundle

□ Display only neighbors whose length is of at least 3 residues

Refresh

20 neighbors of 'AAASR' found.

Nhd	HELIX 1	HELIX 2	DIST.MIN	DIST.MAX	ANGLE
1arb	AAASR <show>	CPEGD <show>	30.91	34.21	142.05
		DIIR <show>	22.99	25.79	86.13
		HCG <show>	14.03	16.76	114.63
		TASTAAS <show>	11.18	16.12	50.93
		TPASGA <show>	37.66	40.2	135.3

Figure 4.2. The Helix Explorer neighbour view.

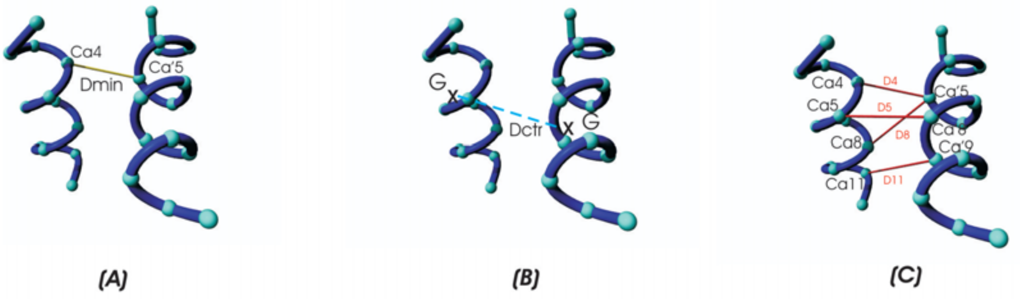


Figure 4.3. There are several different distance measures used in the Helix explorer: a) D_{min}, which is the minimum distance; b) D_{center}, which is the center distance; and c) $D_{average}$, which is the average distance.

In addition to distances, angle between two helices, found in the same protein, are another kind of metric considered by the Helix explorer. This angle is defined as an angle between two respective axes of the helices, as shown in the figure 4.3. The axes of the helices are calculated using a method called 'parametric least-square' which is explained in details in the thesis written by Mohammad Marrakchi (Marrakchi, 2006).

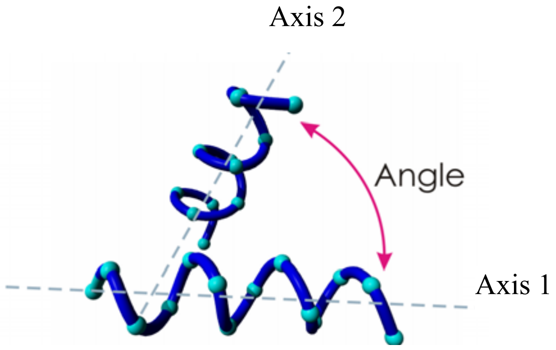
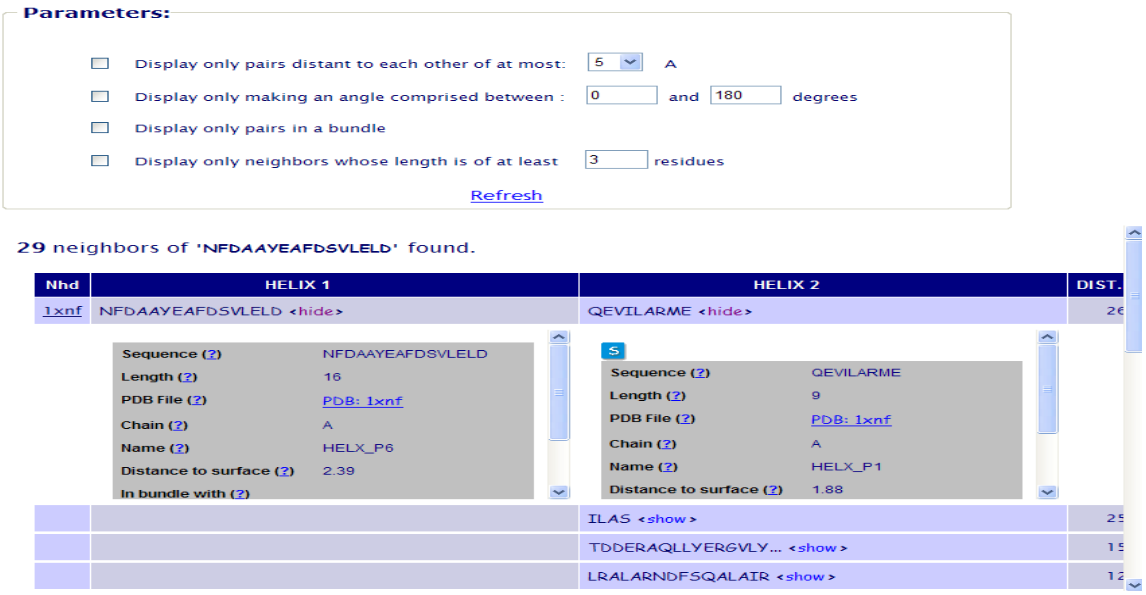


Fig.4.4. Angle between helices. Schematic of the angle between two helices in the Helix Explorer. The axis of a given helix is the one with the ideal helical structure which approaches to the helix the most. The angle between the two helices is the undirected angle (between 0 ° and 180 °) between the projections of the axes of the two helices on a plane that is parallel to the two axes. It is also the arc cosine of the scalar product of two unit vectors defining each axis of the helices.

iii) Secondary structure view displays all the properties attached to each secondary structure element (obtained using the query search), such as the type of secondary structure, its sequence, the length of its sequence, its distance to the molecular surface, and other such features as shown in Figure 4.5.



Parameters:

□ Display only pairs distant to each other of at most: 5 A

□ Display only making an angle comprised between : 0 and 180 degrees

□ Display only pairs in a bundle

□ Display only neighbors whose length is of at least 3 residues

Refresh

29 neighbors of 'NFDAAYEAFDSVLELD' found.

Nhd	HELIX 1	HELIX 2	DIST.
1xnf	NFDAAYEAFDSVLELD <hide>	QEVILARME <hide>	26
	Sequence (?) NFDAAYEAFDSVLELD Length (?) 16 PDB File (?) PDB: 1xnf Chain (?) A Name (?) HELX_P6 Distance to surface (?) 2.39 In bundle with (?)	S Sequence (?) QEVILARME Length (?) 9 PDB File (?) PDB: 1xnf Chain (?) A Name (?) HELX_P1 Distance to surface (?) 1.88	
		ILAS <show>	25
		TDDERAQLLYERGVLY... <show>	15
		LRALARNDFSQALAIR <show>	12

Figure 4.5. Secondary structure properties.

The Helix Explorer web-accessible database is based on three main components: the database, the web interface, and the database builder, as shown in Figure 4.6. The database is built using MySQL[16] and stores data into two main tables: i) secondary structure table, which includes all the secondary structure information extracted from PDB as well as distance to the surface; and ii) pair table, which includes the distances between each pair of secondary structure, as well as some additional information. The web interface is built using Java, which utilizes XML and XSLT to generate displayable HTML pages. The database itself is constructed using a database builder called Pdbase, a program included in the OpenMMS toolkit (Greer et al., 2002). More information about these three components can be found in the thesis written by Marrakchi (Marrakchi, 2006).

[16] A relational database management system which runs as a server providing access to multiple databases.

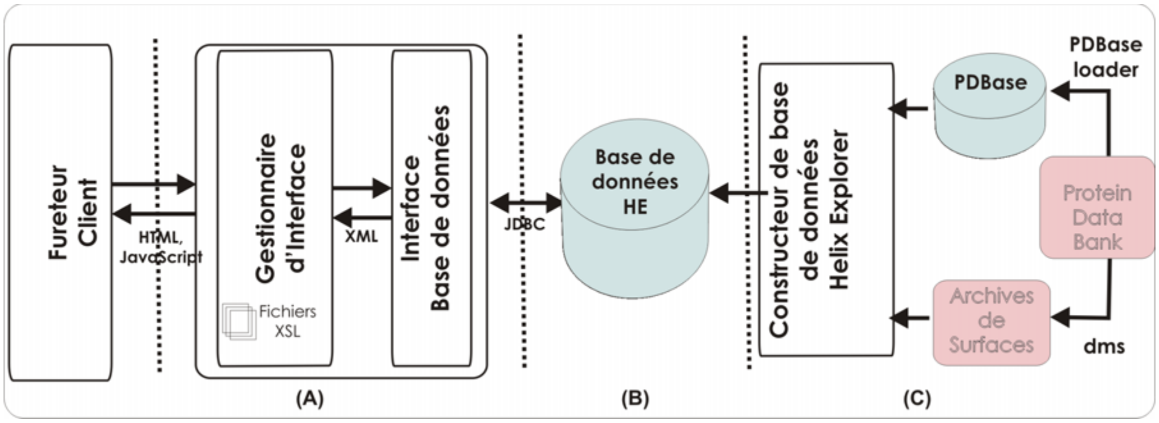


Figure 4.6. Architecture of helix explorer, which consists of three main components: a) Helix explorer web interface; b) Helix explorer data base; c) data base constructor (Marrakchi, 2006).

4.2 What has been added?

4.2.1 Additional secondary structures

The HE, when built, only contained data on protein's helices. However several features have since been added to make HE a more comprehensive database of secondary structures. Both beta sheets and turns are now stored in HE database, and neighbour distances, previously available only for helices, have been computed for turns and beta sheets. All secondary structures can now be searched at once, and the results are grouped together. Lastly, the website has been modified and made more user friendly in general.

4.2.2 Comparing the structure-based phylogenetic trees against the HE-based phylogenetic trees

In this section, I will introduce the programs and algorithms (written in Java) used to query the HE database, as described in chapter 3.2.3 and extract the distances and angles between helices. The method used is described below, and is depicted in Figure 4.7.

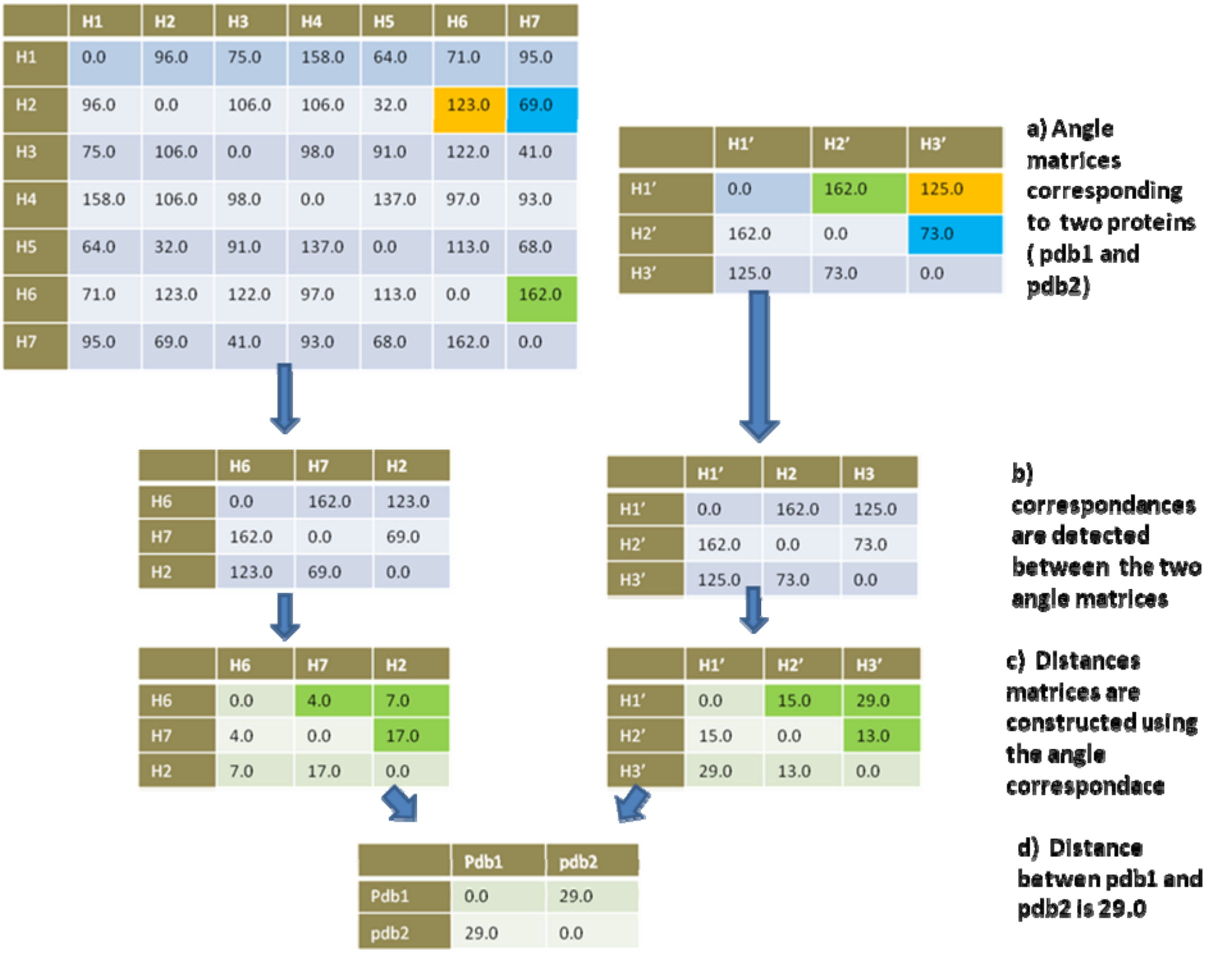
Only alpha helices are utilized, since they are the only secondary structures which have the distances and angles implemented in Helix Explorer. The goal here is to see if a naive algorithm (as described below), that will only use distances between secondary structures to construct phylogenetic trees, can still be compared, in certain special cases, to a more sophisticated algorithms that use 3D information and RMSD. In addition, protein families which only contain alpha helices are selected to do our tests and analysis. The set of alpha protein families have been extracted from SCOP database and are listed in Table 3.2 in Annex 1. The goal is to detect the correspondence between the helices of the given proteins.

For each protein listed in Table 3.2 belonging to a given protein family, a symmetrical matrix containing pairwise angles between the helices is constructed, as shown in Figure 4.7a. Often, like it is the case in this example, one protein contains more helices than the other one. If this is the case, for each helix in the smaller protein, we try to find a corresponding helix in the bigger one. These are the only helices that will be considered in our distance matrix and this is why, in part, the algorithm is considered naïve.

A backtracking algorithm has been developed to determine the correspondence between the two proteins, in two different matrices, using a threshold of ten. As shown in Figure 4.7, the algorithm takes the first pair of helices of the smaller protein, H1' and H2', creating an angle of 162º. The larger matrix (corresponding to the other protein) is then scanned to find pairs of helices with an angle between 152º and 172º, according to the given threshold. There are two possible pairs of helices (H1-H4 and H6-H7) that satisfy this constraint. The algorithm will then look for a correspondence between H1'-H2' and each of these two pairs of helices in the larger matrix. The possible correspondence between H1'-H2' and H1-H4, would result in either H1' corresponding to H1 and H2' corresponding to H4 or the other way around. If H1' corresponds to H1 another helix, which corresponds to H3', must be detected. Between H1' and H3' there is an angle of 125º, but there are no helices that have an angle between the allowed range (115º and135º) with H1. The algorithm will therefore disregard H1-H1' correspondence and backtrack until it finds the correct corresponding sets: (H1'-H2') and (H6-H7). At the end of the algorithm, the correspondence between two Helix-based protein matrices, as shown using three different colors in Figure 4.7a, are H2-H6 to H1'-H3', H6-H7 to H1'-H2' and H2-H7 to H2'-H3'. Using the detected correspondence between the matrices, a subset of three helices in a larger matrix is constructed, as shown in Figure 4.7b.

In cases where there is more than one unique correspondence, the correspondence resulting in the smallest distance is chosen. As described previously, HE computes 3 different distances between helices (minimum, average, and center) and for this study the minimum distance is utilized. Given the optimal correspondence between helices using angles, an appropriate symmetrical distance matrix, consisting of the distances between the helices already present in the HE database, is constructed, as shown in Figure 4.7c. The distance between the two distance matrices is computed by taking the sum of the differences of the corresponding distances. As shown in Figure 4.7c and 4.7d, the distance between the two proteins (PDB1 and PDB2) is ((H6-H7) + (H6-H2) + (H7-H2)) – ((H1'-H2') + (H1'-H3') + (H2'-H3'))) which is ((4.0 +7.0 + 17.0) – (15.0 +29.0 + 13.0) = 29.0.

The previous steps are repeated for each of the PDBs, and the result of all of the calculated distances between the PDBs is stored in a symmetrical distance matrix. This symmetrical distance matrix is used by the FITCH method to construct a phylogenetic tree. This method is used to construct phylogenetic trees for each of the protein families listed in the Table 3.2 (see Annex I). Finally, the obtained phylogenetic trees created from the HE database are compared against the phylogenetic trees obtained from the four structural alignment tools (TM-Align, Lovoalign, CE, and LOCK2) using the RF algorithm. The result of these comparisons is plotted on a graph, which is shown in Figure 4.8. These results will be discussed further in the Section 4.2.3.



	H1	H2	H3	H4	H5	H6	H7
H1	0.0	96.0	75.0	158.0	64.0	71.0	95.0
H2	96.0	0.0	106.0	106.0	32.0	123.0	69.0
H3	75.0	106.0	0.0	98.0	91.0	122.0	41.0
H4	158.0	106.0	98.0	0.0	137.0	97.0	93.0
H5	64.0	32.0	91.0	137.0	0.0	113.0	68.0
H6	71.0	123.0	122.0	97.0	113.0	0.0	162.0
H7	95.0	69.0	41.0	93.0	68.0	162.0	0.0

	H1'	H2'	H3'
H1'	0.0	162.0	125.0
H2'	162.0	0.0	73.0
H3'	125.0	73.0	0.0

a) Angle matrices corresponding to two proteins (pdb1 and pdb2)

	H6	H7	H2
H6	0.0	162.0	123.0
H7	162.0	0.0	69.0
H2	123.0	69.0	0.0

	H1'	H2	H3
H1'	0.0	162.0	125.0
H2'	162.0	0.0	73.0
H3'	125.0	73.0	0.0

b) correspondances are detected between the two angle matrices

	H6	H7	H2
H6	0.0	4.0	7.0
H7	4.0	0.0	17.0
H2	7.0	17.0	0.0

	H1'	H2'	H3'
H1'	0.0	15.0	29.0
H2'	15.0	0.0	13.0
H3'	29.0	13.0	0.0

c) Distances matrices are constructed using the angle correspondace

	Pdb1	pdb2
Pdb1	0.0	29.0
pdb2	29.0	0.0

d) Distance betwen pdb1 and pdb2 is 29.0

Figure 4.7. The procedure used to detect the correspondence between the helices belonging to two different proteins. a) Angle matrices corresponding to two proteins are extracted from the HE database. b) Correspondences between the two angle matrices are detected using a recursive algorithm, shown in the colors orange, blue and green. c) Using this correspondence, the distance matrix is constructed. d) The colored (green) squares are summed for each of the matrices, and then subtracted. The result is placed in a symmetric matrix, which is then used to construct phylogenetic trees.

4.2.3 Results of Comparison of structure-based phylogenetic trees against the HE-based phylogenetic tree

As explained earlier the analysis of phylogenetic trees obtained from HE is only done on the alpha protein families, since there is currently more helix data in the HE than there is for the other structures. As shown in Figure 4.8, there is no correlation between the phylogenetic trees obtained from Lovoalign, TM-Align, LOCK2, and CE in comparison to HE-based phylogentic trees for the immunoglobin set family. For the Homeo-domain family there is a rather high correlation coefficient between HE-based phylogenetic trees and phylogenetic trees obtained using other alignment tools (i.e. Lovoalign 0.4 and TM-Align 0). Finally, there is no correlation for the HMG and Cytochrome-c, but the reason for the lack of correlation cannot be provided, since these families are not listed in the PALI database study.

Unfortunately the obtained results are not satisfactory due to the simplicity of the approach. In addition, due to the limited time, this algorithm has only been tested with the minimum distances and other distances have not been tested. Possible improvements are discussed in the following section.

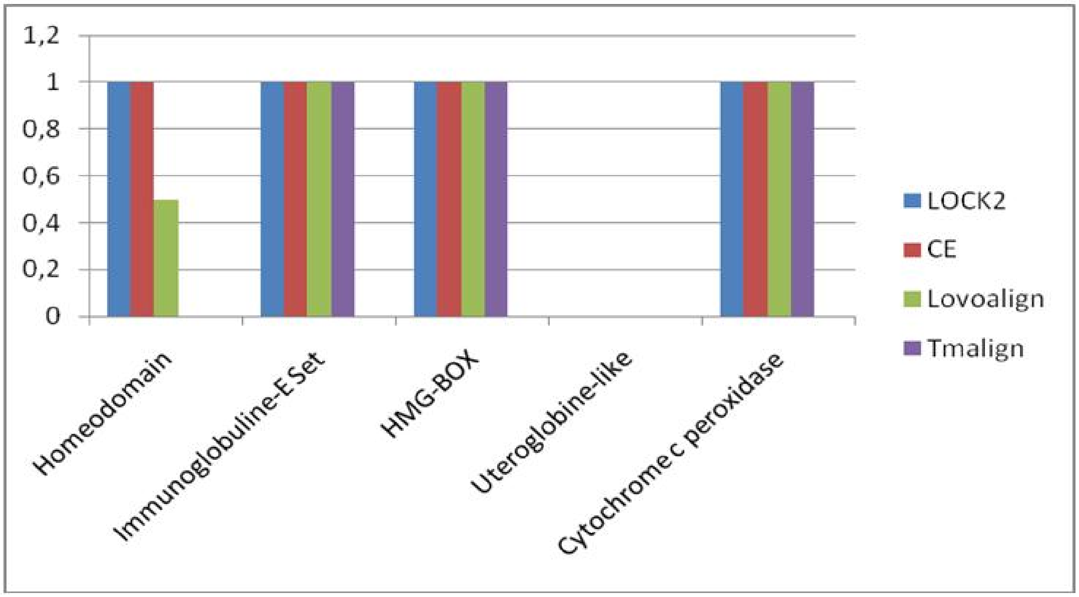


Figure 4.8. The comparison between the structure-based phylogenetic trees obtained from the four structural alignment tools (LOCK2, CE, Lovoalign, Tmalign) and the phylogenetic trees resulting from secondary structure distance matrix extracted from the HE database.

4.2.4 Limitations and Improvements

This study, like any other study, contains a number of limitations and possible future improvements, which will be discussed in this section.

For this study, only proteins belonging to alpha protein families were chosen. In the future, this study should be expanded to include other families, such as beta or alpha-beta, which contain a mixture of alpha and beta secondary structures.

More importantly, only minimum distances have been used in this study. For future development, this algorithm can be tested with the other metric distances (average and center) to further validate its accuracy.

In addition, only the corresponding helices are incorporated in our distances and no score penalty is considered for the additional or missing helices. Also the considered secondary structure distances are more generalized compared to RMSD –based distances. Moreover, the unstructured part of protein connected to the helix is not considered in the calculation.

Another potential improvement would be to include additional databases, other than PALI database, that contain information regarding the comparison between structure-based and sequence-based phylogenetic trees. Also, due to time constraints, only five protein families were analyzed. In the future, a larger number of different protein families could be explored and analyzed in order to draw a more concrete conclusion.

Finally, the current algorithm used to detect correspondence between the alpha helices (as described in the Section 4.2.2) doesn't function optimally when dealing with proteins with large difference in number of helices. The algorithm could be improved upon to address the shortcoming.

Chapter 5. Conclusion

In this project, I have accomplished two main goals, analysis of the accuracy of the structural alignment tools as described in Section 5.1, and construction of phylogenetic trees using the HE database, described in Section 5.2.

5.1. Analysis of structural alignment tools

In this project, I have analyzed the accuracy of three different structural alignment tools: CE, Lovoalign and LOCK2, by comparing obtained phylogentic trees against phylogenetic trees created by TM-Align. Based on this analysis the following conclusion can be made:

For both calmoduline-like and short-chain-cytokines family, overall, LOCK2 has managed to produce phylogenetic trees with a low RF distance in comparison to the phylogenetic trees created by TMalign. Therefore LOCK2 can be considered as the right choice for modeling the evolution of protein families belonging to the category of families with ‘poor correlation coefficient between structure-based and sequence-based phylogenetic trees. For both Fe-Mn superoxidate and TIM, CE can be considered a good candidate when modeling the evolution of protein families that have a high correlation between both sequence-based and structure-based phylogenetic. Finally for Serpins and small kunitz family, CE can be the preferred alignment tool to model the evolution of protein families which have a poor correlation between structure-based and sequence based phylogenetic trees (according to the PALI database).

Based on the given limited dataset and the obtained results, it can be concluded that CE is the suitable tool for modeling the protein evolution when dealing with protein families with a high correlation between sequence and structure-based phylogenetic trees, as well as the protein families with a poor correlation while maintaining a high sequence similarity. LOCK2 however can be a more appropriate tool when dealing with protein families with a high correlation.

Although the phylogenetic tree comparison is an appealing method to compare structural alignment tools, the obtained results are insufficient to determine the optimal tool (amongst LOCK2, CE and Lovoalign) for modeling protein evolution of any protein family. In the future, larger numbers of protein families, other type of alignment score (TM score rather than RMSD), and finally local comparisons (such as coverage and CPU time) can be considered to optimize the obtained results. In addition, the analysis was made based on the hypothesis that TMalign is the most suitable tool for modeling protein evolution. The validity of this hypothesis needs to be confirmed.

5.2. Comparison of structural alignment tools against JTT

In addition, the sequence-based phylogenetic trees, for six different protein families, have been compared against the phylogenetic trees obtained using the four alignment tools. For proteins families (Small kunitz, Serpins, Calmoduline-like, short-chain cytokines) with a poor-correlation between structure-based and sequence-based phylogenetic trees, all four alignment tools have performed poorly; producing a high RF-based distance in comparison to the phylogenetic trees created by the JTT. For TIM protein family, Lovoalign has produced the lowest RF-based distance. Finally, for Fe-MN-superoxidase both CE and TMalign have performed better by maintaining a lower RF-based distance.

In conclusion, the results obtained do not fully correspond to the results listed in the article written by Balaji S and Srinivasan N Andreani (Balaji and Srinivasan, 2007). The main reason for this discrepancy is due to the fact that instead of using the RF distance (as done in this project), PALI study has used the correlation coefficient between SDM (Structural Distance Metric) and SEDM (Sequence Dissimilarity Metric) as a measure of similarity between the structure-based and sequence-based phylogenetic trees. As a future improvement, a linear statistical correlation coefficient between the structure-based and sequence-based scores can be incorporated in this study.

5.3. Construction of phylogenetic trees using HE database

In Chapter 4, I have presented a database called HE database which was constructed in order to provide different information about protein secondary structures. However the goal was to allow HE to be used as a tool to construct phylogenetic trees merely using secondary structures distance matrices. This is enabled, in part by a) constructing a symmetrical matrix containing the distances between helices for each protein, b) constructing another protein matrix which contains the pair-wise angles between the two helices in two protein matrices and finally, c) using FITCH to construct the phylogenetic trees. However, the problem is the fact that we have no knowledge as to which helix in one protein corresponds to the helix in the other protein.

In order to resolve this issue, a naïve backtracking algorithm has been developed in JAVA, determining the correspondence between two proteins by constructing a symmetrical matrix, containing pairwise angles between the helices for each protein, as described in Section 4.2.2. Using this correspondence one of the protein matrices is re-constructed and finally the distance between the two distance matrices is computed by taking the sum of the pair-wise differences of helices. This procedure is repeated for each of the PDBs, and the result of all of the calculated distances between the PDBS is stored in a symmetrical distance matrix which is then used by FITCH to construct the phylogenetic trees. Using the RF algorithm, the obtained phylogenetic trees are compared against the phylogenetic trees obtained from the other four structural alignment tools (TM-align, Lovoalign, CE, and LOCK2). Following this experiment, the HE-based phylogenetic trees have shown to have a low correlation in comparison to those constructed by the other alignment tools.

The obtained unsatisfactory results can be due to different contributing factors. Firstly, only alpha protein families have been considered for this study. In the future other type of protein families such as beta or alpha beta can be explored. Secondly, due to the lack of time, only the minimum distance metric has been used to construct the protein matrices. However, it would be interesting to consider other distance metrics (such as average and center) to further validate the results.

Thirdly, the developed algorithm does not produce optimal results when comparing proteins having a large difference in their number of helices. This is due to the fact that this study does not take into consideration any score penalty for inserting or deleting helices. Finally, all secondary structures should be taken into account to efficiently compare proteins and not only helices.

Annex I

Table 3.1. Six different protein families are used to compare the four different structural alignments. Each protein family belongs to one of the three categories: a) high correlation between structure-based and sequence based trees; b) poor correlation with low sequence similarity; and c) poor correlation with high sequence similarity.

High correlation		***Poor correlation with low sequence similarity***		***Poor correlation with high sequence similarity***	
TIM	***Fe-Mn superoxidase***	***Small kunitz***	***Serpins***	***Short chain cytokines***	***Calmoduline-like***
1sw3	*1ixb*	*1g6x*	*1hle*	*1rhg*	*1top*
2jk2	*1mng*	*1kth*	*1uhg*	*1bgc*	*2sas*
1r2r	*1jr9*	*1dtx*	*1as4*	*1bge*	*1uhk*
1mo0	*1gv3*	*1jc6*	*1att*	*11ki*	*1nya*
1ney	*1p4k*	*1bun*	*1jjo*	*1f6f*	*2o5g*
1kv5	*1b06*	*1shp*	*1sek*		*1omr*
1o5x	*1ids*	*1dtk*	*1m93*		*1tiz*
1n55	*1coj*	*1dem*	*1mtp*		
1tre					
1hg3					

Table 3.2. Five different protein families have been used to compare the structure-based phylogenetic trees against the ones obtained from the HE database.

Homeodomain	***Immunoglobuline-E set***	***HMG_BOX***	***Uteroglobine-like***	***Cytochrome c peroxidase***
2hdd	*1lp1*	*1ckt*	*1utg*	*1b7v*
1puf	*1edi*	*1i11*	*1ccd*	*1ctj*
1ig7	*1edk*	*1j46*	*1utr*	*1dvh*
1ftz	*1h0t*	*1j3d*	*2utg*	*1c6s*
1ltz	*2spz*	*1hsm*		*1c6r*
1k61	*1g2n*	*1e7j*		*1gdv*
				1cyi

References

Andreani R, Dunder C, Martínez JM (2003) Order-value optimization: Formulation and solution by means of a primal cauchy method. Math Meth Oper Res 58:387-399.

Balaji S, Srinivasan N (2007) Comparison of sequence-based and structure-based phylogenetic trees of homologous proteins: Inferences on protein evolution. J Biosci 32:83-96.

Balaji S, Sujatha S, C. KSS, Srinivasan N (2001) PALI- a database of Phylogeny and ALIgnment of homologous protein structures. Nucleic Acids Research 29:61-65.

Bateman A (2004) The Pfam protein families database. Nucleic Acids Research 32:D138-D141.

Berman H, Henrick K, H. N, Markley JL (2006) The worldwide Protein Data Bank(wwPDB): ensuring a single uniform archive of PDB data. Nucleic Acids Research 35:D301-D303.

Branden CA, Tooze JD (1998) Introduction to protein structure. In. New York: Garland Publishing.

Bromham L, Leys R (2005) Sociality and the Rate of Molecular Evolution. Mol Biol Evol 22:1393-1402.

Can T (2007) Protein Structural Alignment(Dali Method). In. Ankara.

Castresana J (2000) Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. Mol Biol Evol 17:540-552.

Chothia C, Lesk AM (1986) The relation between the divergence of sequence and structure in proteins. EMBO J 5:823-826.

Dai P et al. (2009) GJB2 mutation spectrum in 2063 Chinese patients with nonsyndromic hearing impairment. Journal of Translational Medicine 7:26.

Dayhoff M, Schwartz R, Orcutt B (1972) A model of evolutionary change in proteins. of Protein Sequence and Structure 5:89-99.

Dereeper A, Guingon V, Blanc G, Audic S, Buffet S, Chevenet F, Dufayard JF, Guindon S, Lefort V, Lescot M, Claverie JM (2008) *Phylogeny.fr: robust phylogenetic analysis for the non-specialist*. Nucleic Acids Research 36.

Doolittle RF (1981) Similar Amino Acid Sequences:Chance or Common Ancestry? Science 214:149-159.

Durbin R, SR. E, Krogh A (2004) **Probabilistic Models of Proteins and Nucleic Acids**. Nucleic Acids Research 32:1792-1797.

Eddy SR (2004) What is dynamic programming?,. Nature Biotechnology 22:909-910.

Edgar RC (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinformatics 5:113.

Eidhammer I, Jonassen I, Taylor WR (1999) Structure comparison and structure patterns. In: REPORTS IN INFORMATICS. bergen: University of Bergen.

Eidhammer I, Jonassen I, Taylor WR (2004) Protein Bioinformatics : An algorithmic approach to sequence and structure analysis: Wiley.

Felsenstein J (2004) Inferring phylogenies: Sinauer.

Fraser HB, Wall DP, Hirsh AE (2003) A simple dependence between protein evolution rate and the number of protein-protein interactions. BMC Evolutionary Biology 3:1-6.

Gille C, Frommel C (2001) STRAP: editor for STRuctural Alignments of Proteins. Bioinformatics 17:377-378.

Goh CS, Bogan AA, Joachimiak M, Walther D, Cohen FE (2000) Co-evolution of proteins with their interaction partners. JMolBiol 299:283-293.

Greer DS, Westbrook JD, Bourne PE (2002) OpenMMS: An Ontology Driven Architecture for Macromolecular Structure. Bioinformatics 18:1280-1281.

gribskov M, Robinson NL (1996) The use of reciever operating characteristic(ROC) analysis to evaluate sequence matching. Comput Chem 20:25-33.

Guindon S, Gascuel O (2003) PhyML- A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Systematic Biology 52:696-704.

Hameroff SR (1987) Biomolecular Consciousness and Nano Technology. In. Tuscon: Elsevier Science.

Hasegawa H, Holm L (2009) Advances and pitfalls of protein structural alignment. Curr Opin Struct Biol 19:341-348.

Holm L, Sanderand C (1993) protein structure comparison by alignment of distance matrices. JMolBiol 233:123-138.

Holm L, Sander C (1997) An evolutionary treasure: unification of a broad set of amidohydrolases related to urease. Proteins 28:72-82.

Huskey P (2010) Protein Structure. In.

Illergård K, Ardell DH, Elofsson A (2009) Structure is three to ten times more conserved than sequence- a study of structural responsed in protein cores. Proteins:1-10.

Jones DT, Taylor WR, Thornton JM (1992) The rapid generation of mutation data matrices from protein sequences. Comput Appl Biosci 8:275-282.

Junier T, Pagni M (2000) Dotlet: diagonal plots in a web browser. Bioinformatics 16:178-179.

Keates W (1988) Biophysical Methods - Protein structure component. In: Lecture 1: Secondary structure of Proteins: University of Guelph.

Kihara D, Skolnick J (2003) The PDB is Covering Set of Small Protein Structures. ELSEVIER 334:793-802.

Kim C, B. L (2007) Accuracy of structure-based sequence alignment of automatic methods. BMC Bioinformatics 8.

Kolodny R, Linial N (2004) Approximate protein structural alignment in polynomial time
PNAS 101:12201-12206.

Kolodny R, Koehl P, Levitt M (2005) Comprehensive evaluation of protein structure alignment methods: Scoring by geometric measures. JMolBiol 346:1173-1188.

Kuhner MK, Felsenstein J (1994) A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Mol Biol Evol 11:459-468.

Lesk AM, Chotia CA (1980) How different amino acid sequences determine similar protein structures: the structure and evolutionary dynamics of the globins. JMolBiol 136:225-270.

Levitt M, Gerstein M (1998) A unified statistical framework for sequence comparison and structure comparison. Proc Natl Acad Sci 95:5913-5920.

Maccallum RM (1997) Computational Analysis of Protein Sequence and Structure. In: Department of Biochemistry and Molecular Biology. London: University College London.

Madhusudhan MS, Webb BM, Marti-Renom MA, Eswar N, Sali A (2009) Alignment of multiple protein structures based on sequence and structure features. Protein Eng Des Sel 22:569-574.

Marrakchi MT (2006) Helix Explorer: Une nouvelle base de donnes de structures de proteins. In: Computer science Montreal: Universite de Montreal.

Martínez L, Andreani R, Martínez JM (2007a) Convergent algorithms for protein structural alignment. 8:306-321.

Martínez L, Andreani R, Martínez JM (2007b) Convergent algorithms for protein structural alignment. BMC Bioinformatics 8:1471-2105.

McDarby M (2003) An Online Introduction to Advanced Biology. In: Chapter 5: Chemistry: Molecules of Life.

Mettu RR (2008) Overview of Protein Structure and Function In: ECE 597M: Bioinformatics. Amherst.

Moniz M (2007) Bioinformatics course. In. ottawa: University of ottawa.

Mount DW (2004) Bioinformatics: Sequence and Genome Analysis second Edition: Cold Spring Harbor Laboratory Press.

Murzin AG, Brenner SE, Hubbard T, al. e (19995) SCOP: A Structural Classification of Proteins Database for the Investigation of Sequences and Structures. JMolBiol 247:536-540.

Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48:443-453.

Novotny M, Madsen D, Kleywegt GJ (2004) Evaluation of Protein Fold Comparison Servers. Proteins 54:260-270.

Orengo CA, Michie AD, Jones S, Jones DT, Swindells MB, Thornton JM (1997) CATH--a hierarchic classification of protein domain structures. Structure 5:1093-1108.

Pál C, Papp B, Lercher MJ (2006) An integrated view of protein evolution. Nat Rev Genet 7:337-348.

Pattengale ND, Gottlieb EJ, Moret BME (2007) Efficiently computing the Robinson-Foulds metric. J Comput Biol 14:724-735.

Pauling L (1996) The discovery of the alpha helix. The chemical intelligencer 2:32-38.

Pearson (2009) From Gene to Protein: Translation. In: Pearson Eduction Inc.

Potter RM (2008) Constructing Phylogenetic Trees using Multiple Sequence Alignment. In: University of Washington.

Preston H (2009) Beta Sheet. In. Wembley Science College.

Qi Y, Sadreyev RI, Wang Y (2007) A comprehensive system for evaluation of remote sequence similarity detection. BMC Bioinformatics 8:1-19.

Ramachandran GN, Sasisekharan V (1968) Conformation of polypeptides and proteins. Adv Protein Chem 23:283-438.

Robinson DF, Foulds LR (1980) comparison of phylogenetic trees. mathematical biosciences 53:131-147.

Ropodi A (2003) Dali: An algorithm for optimal structure alignment using distance matrices. In: Trust Sanger Institute.

Russel RB, Barton GJ (1992) Multiple protein sequence alignment from tertiary structure comparison: Assignment of global and residue confidence levels. Proteins 14:309-323.

Sam V, Tai C, Garnier J (2006) ROC and confusion analysis of structure comparison methods identify the main causes of divergence from manual protein classification. BMC Bioinformatics 7:1471-2105.

Sauder JM, Arthur JW, Dunbrack RL (2000) Large-Scale Comparison of Protein Sequence Alignment Algorithms With Structure Alignments. Proteins 40:6-22.

Shapiro J, Brutlag D (2004) Foldminer and LOCK2: protein structure comparison and motif discovery on the web. Nucleic Acids Research 32:W536-W541.

Shindyalov IN, Bourne PE (1998a) Combinatorial Extension (CE) using a Composite Property Description. A New Approach to 3-D Structure Alignment and its Application to the Protein Kinase Family. In.

Shindyalov IN, Bourne PE (1998b) Protein Structure Alignment by Incremental Combinatorial Extension (CE) of the Optimal Path. Protein Engineering 11:739-747.

Sierk ML, Pearson WR (2003) Sensitvity and selectivity in protein structure comparison. Protein Science 13.

Sierk ML, Kleywegt GJ (2004) Deja Vu All Over Again: Finding and Analyzing Protein Structure Similarities. Structure 12:2103-2111.

Singh A, Brutlag D (1997) hierarchical protein structure superposition using both secondary structure and atomic superposition. In: Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology, pp 284-293: AAAI Press

Teichert F, Bastolla U, Porto M (2007) SABERTOOTH: protein structural alignment based on a vectorial structure representation. BMC Bioinformatics 8:425.

Thompson JD, Gibson TJ, Higgins DG (2002) Multiple sequence alignment using ClustalW and ClustalX. Curr Protoc Bioinformatics Chapter 2:Unit 2 3.

Thorne JL (2007) Protein evolution constraints and model-based techniques to study them. Current Opinion in Structural Biology 17:337-341.

Vesterstrom J, Taylor WR (2006) Flexible Secondary Structure Based Protein Structure Comparison Applied to the Detection of Circular Permutation. J Comput Biol 13:43-63.

Wampler JE (1996) Tutorial on peptide and protein structure. In. Athens: University of Georgia.

Webster AJ, Payne RJH, Pagel M (2003) Molecular Phylogenies Link Rates of Evolution and Speciation. Science 301:478.

Westbrook JD, Feng Z, Jain S, et al (2002) The Protein Data Bank: unifying the archive. Nucleic Acids Research 30:245-248.

Whelan S, Goldman N (2001) A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. Mol Biol Evol 18:691-699.

Wikepedia Reciever Operating Characteristic. In.

Yang A, Honig B (2000) An Integrated Approach to the Analysis and Modeling of Protein Sequences and Structures. I. Protein Structural Alignment and a Quantitative Measure for Protein Structural Distance. J Mol Biol 301:665-678.

Zhang Y, Skolnick J (2005) TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic Acids Research 33:2302-2309.