2014

Multiple Alignment of Structures using Center of Proteins

Kaushik Roy
University of Windsor

Multiple Alignment of Structures using Center Of proTeins (MASCOT)

by

Kaushik Roy

A Thesis

Submitted to the Faculty of Graduate Studies

through the School of Computer Science

in Partial Fulfillment of the Requirements for

the Degree of Master of Science at the

University of Windsor

Windsor, Ontario, Canada

2014

Multiple Alignment of Structures using Center Of proTeins

(MASCOT)

by

Kaushik Roy

APPROVED BY:

M. Hlynka

Department of Mathematics and Statistics

D. Wu

School of Computer Science

A. Mukhopadhyay, Advisor

School of Computer Science

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyones copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

There is a buzz among structural biologists about conducting a major portion of their future work in *silico*, due to progressively refined computational tools and an amazing quantity of digitized biological data. This masters thesis focusses on the area of computational methods for aligning multiple protein structures.

As the problem under consideration is known to be np-complete, several ways for coming up with good approximations have been suggested over the years. A new approach for achieving better, or at least as good results as before, is presented here. We discuss the proposed algorithm and its constituent methods. Finally, we report the widely used root mean square deviation (RMSD) as measures of structural similarity, and the execution time. Some chosen results, from our extensive experimentation, and their significance have been discussed.

A web server has also been implemented for trying out a pairwise alignment algorithm. This is hosted on the university website and the link has been provided in the contributions

DEDICATION

To my loving mom, Sikha Roy

-Of all that walk the earth, you are most precious to me.

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my advisor *Prof. Asish Mukhopadhyay.* Without his supervision and constant help this thesis would not have been possible.

I would like to thank my committee members, *Prof. Myron Hlynka* and *Prof. Dan Wu* for their, patience, objectivity, and observations.

A number of positive forces came together in the form of the following people, to help me get through some trying times, and credit goes to:

- *Sikha Roy* for being an uncomplicated risk-it-all-for-son loving mother

- *Satish Ch. Panigrahi* for his contagious thoroughness and supportive presence

- *Shantanu* for being the programming jedi that he is

- *Arijit* for his timely hand, latex skills, and constant reality check

- *Antariksha* and *Arjun* for loads of inspiration and funny conversations

- *Taha* for believing in me, for his laptop which I used when mine broke, and a roof when I had none

Kaushik Roy

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

”It is a miracle that curiosity survives formal education.”

- Albert Einstein

Specialized data banks [1][2][3] all around the world thrive to record all kinds of high quality biological data in as many meaningful ways as possible. As the dossier on genomic information gets bigger with every passing year, a new task in the bioinformatics discipline has come to surface: comparison of molecular structures. In this chapter we state the problem being worked on, the reason as to why its important, and some contributions that we have made to achieve this.

Structural comparison is the matching of three dimensional configurations of proteins. Ideally we would like to reveal the most significant similarities that is possible within the structures being compared. There are two parts to this problem first one must find a strategy to search for similarities, and second we must find a way to quantify the extent of the similarity. Unfortunately, finding an exact solution is not an option, so we resort to a heuristic that minimizes the sum-of-pairs distance between alignments.

1.1 Problem Statement

Let P = P_1, P_2 ... P_N be a set of N protein structures. Each structure is represented by the coordinates of their alpha carbon (Cα) atoms, in order from N-terminus to C-terminus. The number of residues in the each of the proteins are L_1, L_2 L_N respectively. P_{ij} denotes the j^{th} residue of the i^{th} structure, for i = 1 ... N and j = 1 ... L_i.

A multiple structural alignment of P is X = (x_{ij}) ,1 ≤ i ≤ N, 1 ≤ j ≤ L, such that:

a) Max (L_1, L_2 ... L_N) ≤ L ≤ (L_1 + L_2 + ... + L_N).

b) Each element of X is either one of the residues of P_{ij} or a special null residue called gap, denoted by the symbol '-'.

c) The i^{th} row of X contains the ordered set of C positions of structure i, possibly with gaps sprinkled in between. This also means that the alignment preserves the order of residues.

Having obtained the matrix of equivalences X, let TR (Rot_i , $Trans_i$) 1≤ i ≤ N, be a set of rigid body transformations, each having a proper rotation matrix Rot_i , where det(Rot_i)=+1, and a translation tuple $Trans_i$ act upon each protein in X, to drive an optimal superposition of the structures.

Given a set of reference 3D points of the equivalent residues, a superposition of minimum coordinate root mean square deviation (RMSD) is sought.

$$RMSD = \sqrt{\frac{1}{n}\sum_{i=1}^{N}((v_{ix} - w_{ix})^2 + (v_{iy} - w_{iy})^2 + (v_{iz} - w_{iz})^2)} \tag{1.1}$$

Here, (v_{ix}, v_iy, v_iz) are the 3D coordinates of residue i after a structure ν has been superposed on another structure ω. The distances here are the Euclidean distance between corresponding residues v_i and w_i. The number of aligned residues is denoted by n. Ideally. the aim is to minimize RMSD while maximizing aligned residues.

1.2 Motivation

There are two main classes of applications for multiple protein structure alignment:

1. Biological data mining
2. Quality of life

Biological data mining is defined as "the process of discovering meaningful correlations, patterns, and trends by digging into large amounts of data stored in data banks" [4]. In other words, it is the driving force behind some of the most unprecedented bio-molecular discoveries in the data-rich twenty-first century. Knowledge Discovery in Databases[4], which is what this topic is coined as in the scientific community, is not specific to any industry. Rather it is almost entirely contrived out of intelligent algorithms and willingness to explore the possibility of hidden knowledge that resides in the data. Aligning multiple protein structures is instrumental in this field due to the following observations:

a) Classification - With the Protein Data Bank (PDB) [1] now containing a minimum of 99000 documented structures, one feels the need to organize the protein universe. The natural question is, how? The motivation here is to have a reductionist approach that could take a subset of structures, a unique set maybe, and generate all others from it. Aligning multiple proteins to find out 'parts' (read domains) that are conserved can lead to a bottom-up approach of protein structure classification, as long as we use scoring functions that have biological relevance. Non-redundant structural classifications such as FSSP [5] have been obtained using this approach.

On the other hand we also have the top-down approach that can come up with very useful classifications. SCOP [2] and CATH [3] are two manually curated databases that contain 'clades' of protein families based on both structure and sequence. Here too, we use multiple alignment of proteins in that, an unknown protein has to be 'compared' to a representative set of proteins to find out which superfamily the former belongs to. As a result, we have to only remember the name of the family instead of the individual protein names.

b) Functionality - Structural comparison of proteins produces precise residue correspondences among a set of input proteins. Given such a set of structurally equivalent residues, the information of a well-known protein can be transferred to a still unknown protein. In the case of a significant similarity that indicates homology or analogy, we can hypothesize that both proteins share the same function. For example, proteins 1DM1 and 1ASH share the same structure, so we can guess that 1ASH has a high probability of performing the same activities as 1DM1. Not surprisingly, this turns out to be true since they are both globins. A multiple alignment can thus help us assert properties of entire groups of unknown proteins; whether they are edible, or antibiotics, or even poisonous etc.

c) Evolutionary relationships - Biologists estimate that there are about 5 to 100 million species of organisms living on Earth today. From Aristotle's precursor concept of parsimony to Darwin's 1837 notes on evolutionary 'tree' to Haeckel's coinage of phylogeny in 1866 [6], our understanding of evolution has come a long way to take the form that it has today. The notion that all life is genetically connected via a vast phylogenetic tree is one of the most romantic notions to come out of science. How wonderful to think of the common ancestor of humans and beetle. This organism most likely was some kind of a worm. "Several studies based on the known three-dimensional (3-D) structures of proteins show that two homologous proteins with insignificant sequence similarity could adopt a common fold and may perform same or similar biochemical functions. Hence, it is appropriate to use similarities in 3-D structure of proteins rather than the amino acid sequence similarities in modelling evolution of distantly related proteins" [7]. Thus a multiple alignment of proteins will help us create structure-based phylogenetic trees. Databases such as PALI [8] are already in place, although they contain small number of separate trees. The quest to conjoin these trees to form one huge 'tree of life' is still on, and this thesis is another small step towards the same.

Below is an example of how phylogeny trees based on structure can differ from those based on sequence alone [7].

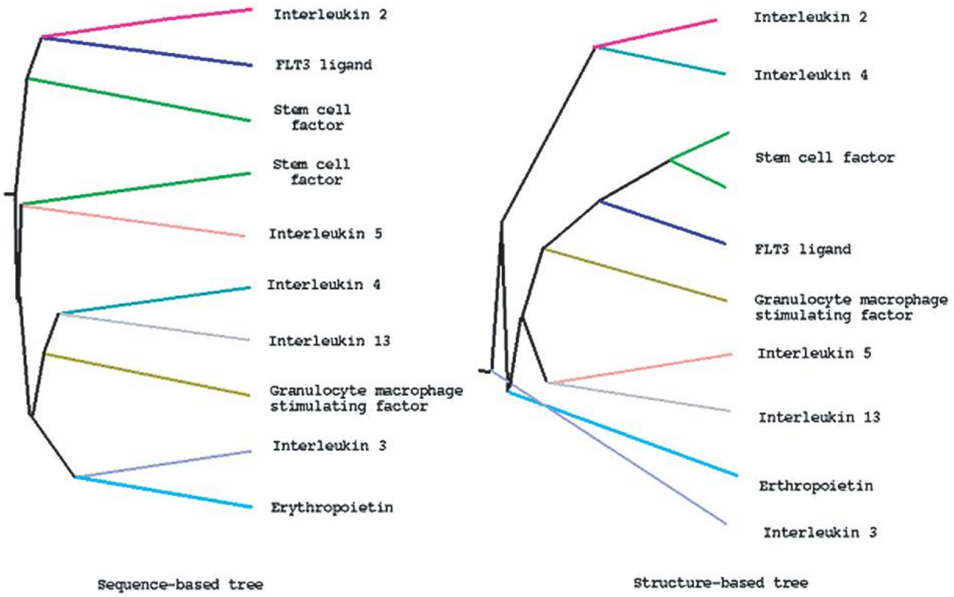


Fig. 1.2.1: Phylogeny tree based on sequence and structure

The consensus is that the dendogram on the right gives more insight into relationships among short chain cytokines than the one on the left [7].

Quality of life is a common dream that all of humanity shares whether they know it or not. A not-so-recent offshoot of genomics, called pharmacogenetics deals exclusively with pharmaceutical innovations, the latest of which is personalized medicine. The Food and Drug Administration performs extensive clinical trials on an average of 18 drugs a year [9], but when the drugs are marketed out their effect on subjects tend to diversify more than expected. The result is that the increasing investments in clinical research have not matched the development rate of ubiquitously usable drugs, and this 'efficacy-effectiveness' gap has reached alarming levels enough for structural biologists to examine how transformation in protein structures arising from genetic differences affect protein-drug interactions. This is still an open challenge, and since it begins with understanding the patient-specific protein at the molecular level, we need to come up with a profile for the underlying proteins that impact the drug response on the patient populace.

To this end, a multiple alignment of proteins is instrumental in providing a scaffold as to the physiochemical properties based on which the 'druggability' [10] of a set of proteins can be decided.

1.3 Contributions

The following are the contributions of this thesis:

1. A new algorithm for aligning more than two protein structures has been proposed. The 3D protein structures are represented in a way which makes processing faster while keeping the quality intact. A feasible heuristic has been used to come up with as many residue-residue correspondences as possible. The number of correspondences is improved by bringing residues closer in space through rigid body superposition. We then report the RMSD values for some alignments with data taken from available sources, and interesting observations have been highlighted.

2. A web server has been implemented, and deployed on a dedicated machine, to try out a new pairwise protein structure alignment algorithm developed by our professor and his team. We discuss what methodologies and technologies have been used to achieve this. Also, we present design diagrams, and screenshots explaining details.

1.4 Chapter Outline

The list below presents the organization of the chapters which make up this thesis. Also given is a brief description of the topics each chapter deals with.

- Chapter 2 delves into the background knowledge that is required to appreciate the work done, as well as a comprehensive literature review citing previous work.

- Chapter 3 describes the proposed algorithm and its inner workings, giving justifications for the chosen approach at each step.

- Chapter 4 shows the experimental results after applying our algorithm on various data sets, and presents some conclusions.

- Chapter 5 presents detailed description of a web server that hosts a pairwise alignment algorithm, its methodologies and usage.

- Bibliography declares a detailed list of references from which factlets and numbers have been used as a guide for this thesis.

Chapter 2

Background and Literature Review

2.1 Biological Aspects

What is so interesting in biological cells that cannot be seen? To the naked eye, nothing. A journey on a microscopic level must be undertaken to discover the invisible.

2.1.1 Life Origins

The molecular machinery of life is a complex system that started off about 3.5 billion years ago, approximately 10 billion years after the big bang. The biological systems that we observe today therefore needed a fourth of the time that the universe exists in order to evolve; an evolution that generated an entire tree of life in an incredible process of repeated mutation and selection. A process which finally led to our existence. We are now on a quest to decipher this molecular assembly, because it determines to a great part who we are, what we look like and feel, and whether we are healthy. Our ultimate goal is to reverse engineer the molecular machinery [11]; to specify building blocks, detect recurrences and figure out how they function.

Darwinian theories [12] promote a natural evolutionary process. Under the umbrella of evolution, the whole process can be interpreted as a race, to become the best, from which the term 'survival of the fittest' was coined.

Nucleic acids such as deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) represent the chemical carriers of a cells genetic information. DNA stores an organism's blueprint and passes it on to RNA, which reads, decodes and uses that information to make proteins. Humans have thousands of different proteins, each having a specific structure along with a unique function. As a matter of fact, proteins are the most structurally sophisticated molecules ever identified [13], which is why they deserve a section of their own, and the next section does that.

2.1.2 Protein and Protein Structures

2.1.2.1 What are proteins?

Proteins [14] are gigantic sequential molecules of smaller recurring molecules. They are made up of amino acids that are connected by peptide bonds to form polymers in the form of polypeptide chains. A protein may consist of one or more polypeptide chains. In nature over 100 different amino acids have been found. However, only 20 of them are created by ribosomes in protein synthesis.

When peptide bonds are formed between amino- and carboxyl acid groups from adjacent amino acids, a water molecule is released in the process; the remains of the amino acid is now called a *residue*. After all amino acids have bonded to become residues, the backbone of the protein come into existence. It consists of three atoms namely the nitrogen (N) atom from one amino group, the central Ca atom, and the carbon (C) atom from the carboxylic group repeated in triplets, one for each residue . The peptide bonds between residues are rigid. Thus, there are only two types of rotatable bonds along the protein backbone: The bond between the Ca atom and it's N neighbor, and the bond between the Ca atom and it's C neighbor. This means that the overall 3D structure of a protein in principle is determined simply by the rotational states of these two bonds in each residue. The angles of these two bonds are commonly denoted by phi and psi[14].

Generally, there are two main classes of protein. The globular proteins play an active role in categorizing metabolic processes, replication and expression of genes. These proteins can be thought of as the workhorses of the cell. They tend to be non-repetitive and between 100 and 300 residues in size, they are typically compact and sphere-shaped. The other class, fibrous proteins, and more passive, and often serve a structural purpose. For instance, nails and hair are comprised of fibrous proteins. Finally there are membranes, where they control and regulates traffic in an out of the cells of various atoms and molecules. Membrane proteins also serve as message passing devices between cells.

2.1.2.2 How are they structured?

There are four levels of protein structure organization: primary, secondary, tertiary and quaternary structure [13].

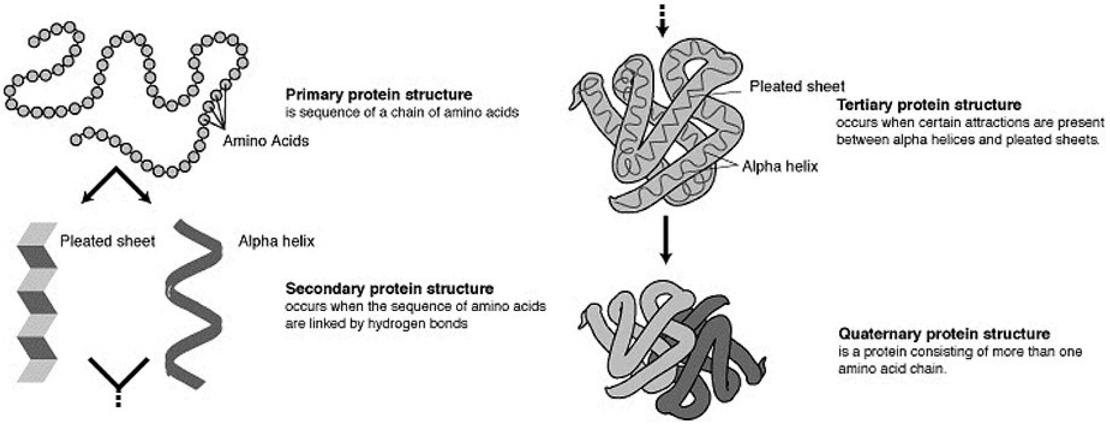

Primary protein structure
is sequence of a chain of amino acids
Amino Acids
Pleated sheet
Alpha helix
Secondary protein structure
occurs when the sequence of amino acids are linked by hydrogen bonds
Pleated sheet
Tertiary protein structure
occurs when certain attractions are present between alpha helices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

Fig. 2.1.1: Internals of a protein

The primary structure is the sequential arrangement of amino acid residues, referred to as protein sequence. The 1- or 3- letter codes that are used to denote amino acid residues are given in the table below.

Table 2.1.1: Amino acids in proteins

Amino Acid	3-letter code	1-letter code
A	Alanine	Ala
R	Arginine	Arg
N	Asparagine	Asn
D	Aspartic acid	Asp
C	Cysteine	Cys
Q	Glutamine	Gln
E	Glutamic acid	Glu
G	Glycine	Gly
H	Histidine	His
I	Isoleucine	Ile
L	Leucine	Leu
K	Lysine	Lys
M	Methionine	Met
F	Phenylalanine	Phe
P	Proline	Pro
S	Serine	Ser
T	Threonine	Thr
W	Tryptophan	Trp
Y	Tyrosine	Tyr
V	Valine	Val

The secondary structure comprises regular elements that are stabilized by hydrogen bonds between the carboxyl group (C=O) and amide group (N–H) of two peptide bonds. The most common secondary structure elements, abbreviated SSEs, are alpha-helices and beta-sheets, which are formed by stabilization of hydrogen bonding. The tertiary structure is the final three dimensional folded arrangement of a protein, which results from a large number of non-covalent interactions between amino acids.

In the quaternary structure, non-covalent interactions bind multiple polypeptides into a single, larger protein. For example, Hemoglobin has quaternary structure due to association of two alpha globin and two beta globin polyproteins [14].

2.1.2.3 Which way do we represent them?

Protein structure can be represented in many ways. Three of the most used are standard 3D Cartesian coordinates, torsion angles, and internal distances also known as distance matrices. Cartesian coordinates are simply the raw 3D coordinates of the atoms that are included in the description. Torsion angles are the previously described and angles (section 3.2) both angles must be saved for each residue. Internal distances are the distances between all pairs of C$_\alpha$ atoms in the protein. The distances are stored in a quadratic matrix, where entry (i, j) is the Euclidean distance between the C$_\alpha$ atoms of residue i and j.

Cartesian coordinates and distance matrices are widely used in protein structure applications, whereas torsion angles are more rarely used. It is possible to convert between the three types of representations. However, converting from torsion angles or internal distances to Cartesian coordinates might produce a mirror image of the true structure (known as a chirality) [14], because the distance matrix representation cannot distinguish mirror images from one another. In addition to deciding how to represent the available information, it must also be decided what information to include in the description. Some possible choices in this context are:

1. Alpha carbon atoms only (which in the most typical choice in structural comparison)

2. all backbone atoms (N, C$_\alpha$, C, N, C$_\alpha$, C, N, . . .)

3. one of the above including a description of the type and position of the side chain

4. all known atoms of the protein

2.1.2.4 Where do the residues belong?

On a broader perspective, the residues in a protein can be assigned to either the helix type, strand type, or as none of the two. An existing problem, which sounds similar but isnt, is the secondary structure prediction problem. This is not the same as assigning residues to types, since for the former we only have the knowledge of the primary sequence, whereas for the latter we know the tertiary structure too. We can call this the SSE assignment problem. However, it is not an obvious task to assign residues to SSEs. Sometimes there seems to be no distinct or perfect assignment. There are a number of ways in which this assignment can be ascertained. For example, Definition of Secondary Structures of Proteins (DSSP) scheme [15] categorizes the residues into bins of SSE elements based on analysis of hydrogen bonding angles following the backbone.

Primary sequence	ARNGDCEGHIM
DSSP letters	..HHHHHHHHH..

Another way to assign residues is STICKS by Taylor [16]. It uses the geometry from the alpha carbon atoms along the backbone, by first taking the mean of the alpha carbon atom positions, and then identifying small sub-sequences that occur on a straight line. The idea is that there is a high probability of such sequences will turn out to be SSEs. An up-side to using this scheme of representation is that it can be used even when information about hydrogen bonds is not present.

A third possible way of assigning residues to some sort of conformation is to use conformational letters [17] . However we shall look into this a little later.

2.1.3 Homology Detection

Relatedness between proteins is understandably a direct consequence of evolution which causes changes between species over time. This metamorphosis occurs due to seemingly random genetic mutations either caused by external conditions, or chemical factors, or both. It is precisely these alterations that gave rise to the daunting task of aligning biological information. However, it must be noted that the process of alignment does not aim to inverse the effects of evolution. Rather, our target is to find out how much of genetic information has been conserved with time. Following this we need to simulate agents that can help us categorize evolutionary similarities and differences tracing back to a common ancestor. This will give us the intended result of reconstructing a phylogeny based on which families of species can be grouped together.

Proteins from different species can be either closely related or far apart depending on how much change has occurred, if at all the species do have a base ancestor. Roughly speaking there are 3 kinds of relatedness than can arise between a pair (and more) of proteins:

a) Identity : proteins are said to be identical if all formations in one protein match all the formations in other proteins.

b) Similarity : proteins are said to be similar if they are nearly related without being identical.

c) Homology : This is a special case of similarity, where proteins are projected to have a common ancestor. This is used to create protein superfamilies based on the two kinds of homology that proteins portray, viz. sequence and structural.

This thesis aims to help people ascertain homologies by aligning multiple protein structures at once.

2.2 Algorithmic Aspects

2.2.1 Global Alignment

Matching patterns in the form of sequences, structures, and sequences with structures is the most basic activity in protein family analysis. When an alignment illustrates some sort of match between proteins, it can make an educated guess as to the function, and evolutionary distance of the proteins from a common ancestor. Now, since at least one of the proteins being aligned is well documented and understood, the degree of relatedness allows all the strenuously acquired biological data to be associated with the new protein.

2.2.2 Needleman-Wunsch Algorithm

Let X=X_1,.,X_m and Y=Y_1,.,Y_m be two sequences of lengths m and n respectively with the input alphabets A consisting of symbols that may represent amino acids or DNA nucleotides. Since we have already established the usefulness of an alignment let us define what an alignment is in a more formal manner. A global alignment of X and Y introduces gaps (-) at the beginning or end, or between any pair of letters or strings, such that the resulting output has the following properties:-

a) Its a 2 x L matrix where max(m, n) $\leq$ L $\leq$ m + n.

b) First row has either blank or a character from X. Second row has either blank or character from Y.

c) No column can have only blank.

One of the first global alignment methods was the Needleman-Wunsch dynamic programming algorithm to compute optimal edit distance between two strings[18]. It goes as follows:

Table 2.2.1: DP formulation of global alignment

match/mismatch	H[i-1][j-1] + score(X[i], Y[j])	
insertion	H[i][j-1] + score(-, Y[j])	H[i][j] = max
deletion	H[i-1][j] + score(X[i], -)	

Here H[0][0] = 0, H[0][j] = H[0][j-1] + score(-, Y[j]), and H[I][0] = H[i-1][0] + score(X[i], -).

This can be represented in the following diagram:

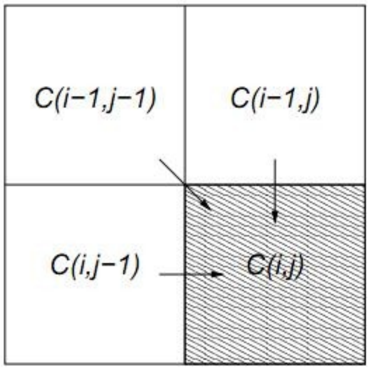
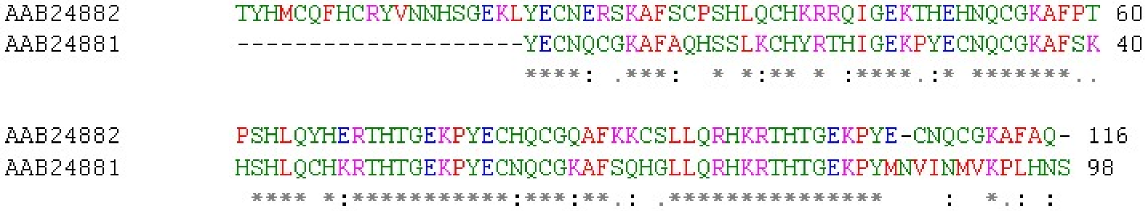


Fig. 2.2.1: DAG for NW algorithm

Earlier implementations took cubic time, but of late this can be reduced to linear space and quadratic time [19].

Below is an example of amino acid sequence of two human zinc finger proteins.



```
AAB24882        TYHMCQFHCRYVNNHSGEKLYECNERSKAFSCPSHLQCHKRRQIGEKTHEHNQCGKAFPT 60
AAB24881        --------------------YECNQCGKAFAQHSSLKCHYRTHIGEKPYECNQCGKAFSK 40
                                    ****: .***:  * *:** * :****.:* *******..

AAB24882        PSHLQYHERTHTGEKPYECHQCGQAFKKCSLLQRHKRTHTGEKPYE-CNQCGKAFAQ- 116
AAB24881        HSHLQCHKRTHTGEKPYECNQCGKAFSQHGLLQRHKRTHTGEKPYMNVINMVKPLHNS 98
                 **** *:***********:***:**.: .***************    :  *.: :
```

Fig. 2.2.2: Sequence alignment of rat and human gene

The score (p, q) function is the cost of changing one character p in any of the sequences to q in the other sequence. The simplest possible scoring matrix, where a character can only be substituted by itself will score a 1 for a match and 0 for everything else. This will be an identity matrix and is used to see if two sequences are extremely closely related, and is thus not practical for biological purposes. Fortunately, we have some widely used options such as Point Accepted Mutation (PAM) [20] and BLOck Substitution Matrix (BLOSUM) [21] both of which measure the rate at which one character in a sequence changes to other character states over time.

A less well known scoring function that is used mainly for aligning proteins structurally is CLESUM [17].

2.2.3 Gap Model

The alignment obtained from a basic form of the dynamic programming algorithm, such as one above, may look good at first glance. However, to maximize the matching we have made wanton use of the gap symbol. Now, a gap in one of the sequences signifies that one or more residues needed to be deleted from one sequence. This brings us to a number of relevant questions. To begin with, is there a cap on the number of gaps we can put? Is there a certain way the gaps should be put? Do we treat a single gap the same way as a string of gaps? Most importantly, even say we maximize the identity of two sequences, does the alignment obtained give us any insight into the biology of the subject in question? Superficially one would think raising such questions might be overthinking the problem, but a correct answer is crucial for an alignment that can be used by biologists.

Gaps often are inserted during the alignment of homologous regions of sequences and represent deletions or insertions. A gap is any maximal, consecutive run of spaces in a single sequence of a given alignment. Typically, the length of a gap is the number of indel operations in it. The idea is to treat the gap as a whole, rather than give each of its spaces the same weight. This signifies the insertion or deletion of an entire subsequence occurring as a single mutational event. To give an example, recall that an RNA molecule can be transcribed from the DNA of a gene. After the pre-mRNA is created, a splicing process takes place: each intron-exon boundary is located, introns are spliced out (hence the gaps) and exons are concatenated. The resulting molecule is the messenger RNA. These mRNA later go on to help produce proteins. Thus to reverse-engineer the process and find a good 'match' we could simply associate the run of gaps to the introns that have been spliced out.

The gap model discusses the use of values called 'gap penalties' that are incurred when a run of indels are found in an alignment. There are 3 most used axioms for introducing gap penalties in an alignment:

a) Constant gap penalty - This is the simplest form of gap penalty, where a gap of size k $>$ 0 will have a score w(k). Since we are penalizing the score we shall add -w(k) to the score. This is usually implemented using some peripheral tables and the following recursions:

$$T_m[i,j] = score(i,j) + max \begin{cases} T_m[i-1,j-1] \\ T_i[i-1,j-1] \\ T_d[i-1,j-1] \end{cases} \tag{2.1}$$

Here, T_m is the main table, T_i is the table for inserts, and T_d is the table for deletes. The other two tables are populated as follows:

$$T_i[i,j] = \begin{cases} T_m[i,j-k] - w(k) & 1 \leq k \leq j \\ T_d[i,j-k] - w(k) & 1 \leq k \leq j \end{cases} \tag{2.2}$$

$$T_d[i,j] = \begin{cases} T_m[i-k,j] - w(k) & 1 \leq k \leq i \\ T_i[i-k,j] - w(k) & 1 \leq k \leq i \end{cases} \tag{2.3}$$

Here, T_m[0,0] = 0, T_i[0,j] = - w(j), T_d[i,0] = - w(i), and other initializations are set to -∞.

The time complexity for this is O(m^2n+mn^2).

b) Affine gap penalty

In this model a gap is given two weights. One, h to 'open the gap' and the other, g to 'extend the gap'. The net gap penalty is w(k) = h + gk, k≥1, w(0) = 0. It follows that the constant gap weight model is simply the affine model with h = 0. The recursions of tables T_i and T_d are modified as follows:

$$T_i[i,j] = max \begin{cases} -(h+g) + T_m[i,j-1] \\ -g + T_m[i,j-1] \\ -(h+g) + T_d[i,j-1] \end{cases} \tag{2.4}$$

$$T_d[i,j] = max \begin{cases} -(h+g) + T_m[i-1,j] \\ -g + T_d[i-1,j] \\ -(h+g) + T_i[i-1,j] \end{cases} \tag{2.5}$$

The initializations for using this model are as follows:

$$T_m[0,0] = 0 \tag{2.6}$$

$$T_m[i,0] = -\infty, \quad 1 \leq i \leq m \tag{2.7}$$

$$T_m[0,j] = -\infty, \quad 1 \leq j \leq n \tag{2.8}$$

$$T_j[i,0] = -\infty, \quad 1 \leq i \leq m \tag{2.9}$$

$$T_j[0,j] = -(h+gj), \quad 1 \leq j \leq n \tag{2.10}$$

$$T_d[i,0] = -\infty, \quad 1 \leq i \leq m \tag{2.11}$$

$$T_d[0,j] = -(h+gj), \quad 1 \leq j \leq n \tag{2.12}$$

The time complexity is O(mn).

c) Convex gap penalty

The concept here is that each additional space in a gap contributes less to the gap weight than the previous space. Although this model is said to better describe biological behavior, affine model is more favored because of efficiency.

2.2.4 Heuristics - A singular ally

A fundamental task in computer science is coming up with algorithms that can do a particular task correctly and accurately. This generally means an algorithm that claims to solve a problem has to produce proof of correctness and upper bounds on execution time. But what happens when we encounter problems so complex that one or both of the above goals seem theoretically impossible to meet? For example, suppose we are given a sequence 1, 2, 4 and asked to extend the series. The answer is not obvious since it could be 1, 2, 4, 7, 11, 16, 22 or it could well be 1, 2, 4, 8, 16, 32, 64, , or for arguments sake something completely different. It is here that we must apply experience and domain knowledge to come up with 'an' answer that may or may not be what we are looking for. A heuristic is a 'shortcut' problem solving technique used when exhaustive search for a solution is impractical. It so happens that a large section of problems in bioinformatics cannot be solved optimally and/or in polynomial time. Some examples are, phylogeny construction [22], motif detection [23], protein docking [24] and multiple protein sequence/structure alignment [25].

In this thesis we have used our own heuristic, and it's shown to work on many occasions. Needless to say there is no heuristic that gives good solutions (read alignments) in every situation. If there was such a thing, then there would be no need for heuristics, since nobody likes them but that's the best that can be done.

2.2.5 Multiple Sequence Alignment

A multiple sequence alignment (MSA) of N (> 2) sequences, or in this case protein primary residue sequences, is a N x L matrix, where max{L_1, L_2, , L_N} $\leq$ L $\leq$ L_1+L_2+...+L_N, and each entry in the matrix is either a gap ('-') or a sequence alphabet. Also, not all entries in any column are gaps. In order to find an optimal alignment, we need to be able to measure how good an alignment is. This is where objective functions come into play. The following is a MSA of 4 sequences MQPILLLV, MLRLL, MKILLL, and MPPVLILV:

Table 2.2.2: MSA of given sequences

M	Q	P	I	L	L	L	V
M	L	R	-	L	L	-	-
M	K	-	I	L	L	L	-
M	P	P	V	L	I	L	V

MSAs are used for many reasons, including but not limited to:

a) Detect conserved regions in a family of proteins.

b) Provide more clues than pairwise similarity for structural and functional inferences.

c) Serve as a guide for phylogeny reconstruction.

Some objective functions that are used to assess the quality of alignments are:

a) Sum-of-Pairs (SP) - In this scoring scheme, the score of an MSA is the sum of the scores of all of the pairwise alignments.

b) Entropy - The goal of this scoring algorithm is to minimize the entropy, or randomness, in the in the alignment. To calculate the entropy of the alignment, first, we must calculate the probability of a column and then use that probability to calculate a score for that column. This score measures the variability observed in the aligned column. By minimizing the sum of this column score over all of the columns, we minimize the entropy and create a good alignment.

c) COFFEE - The COFFEE score reflects the level of consistency between a multiple sequence alignment and a library containing pairwise alignments of the same sequences.

The Venn diagram below depicts an overview of the various paradigms of solving an MSA, mentioning package names wherever applicable. For a comprehensive review, refer to an excellent survey by Notredame [26].

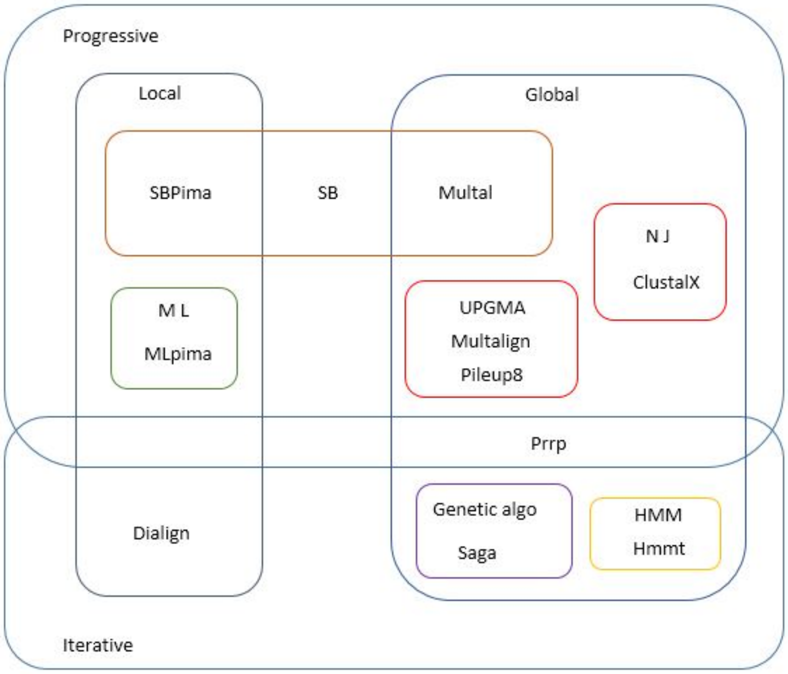


Fig. 2.2.3: Different MSA paradigms

Note:

M L Maximum Linkage

S B Sequential Branching

N J Neighbor Joining

HMM Hidden Markov Model

A few extensively used MSA algorithms with copious citations are as follows:

a) CLUSTAL [27]

b) T-COFFEE [28]

c) MUSCLE [29]

d) CENTER-STAR [25]

e) PROBCONS [30]

CLUSTAL

The CLUSTAL set of MSA programs was first developed in 1988. A study by Higgins and Sharp in the same year describes CLUSTAL as a 'quick and dirty' [27] version of the Feng and Doolittle [31] progressive alignment algorithm. The reason why it became a huge hit among researchers was that they could perform sequence alignments sitting in a lab without any actual biological equipment. The method consists of three steps:

1) Calculate all pairwise sequence similarities, and construct a similarity matrix

2) Create a dendogram, or a guide tree, from the matrix obtained above

3) Merge the alignments into a binary tree following the guide tree generated above

CLUSTAL W, an enhancement over the original CLUSTAL procedure was put forward in 1994 [32] by Thompson et al., and its comparable speed and accuracy soon made it the method of choice for biologists. The main drawback of the original CLUSTAL was that it made certain biologically unrealistic assumptions, thus producing non-optimal results in many cases. The enhanced accuracy was due to a couple of improvements incorporated into CLUSTAL W; the use of weighted sum-of-pairs, the use of revised gap penalties, and the use of neighbor-joining instead of UPGMA in generation of the phylogenetic tree.

A point of interest here is the erroneous use of single-weight matrices conceived from aligning sequence pairs that assumed the sequences in a group to be equally divergent from each other. The choice of using single-weight matrices could be upheld as long as the condition above was true. However, in reality, without *a priori* knowledge it's difficult to say if the sequences are equally different from each other. This meant some sequences could be very similar, while the other could potentially be outliers. Single-weight matrices could not account for such variety, and CLUSTAL W corrected this by assigning individual weights to sequences, i.e. weights were assigned according to the tree branch length, which is the measure of their evolutionary distance. Thus, redundant or similar sequences were given less weight while divergent sequences had more weight.

CLUSTAL W changed the way in which gap penalties were being used, so instead of fixed values, proper opening and extension penalties were incorporated. By this time it was well accepted that gaps found in related proteins were not random occurrences. Regions of conserved structures are a lot less likely to have gaps than the linkers that connect these structures. For example, the residue-specific gap penalties and locally reduced gap penalties in hydrophilic regions account for formation of gaps in potential loops rather than regular secondary structures [14]. Also, early alignments encourage opening up of new gaps, unlike the later alignments.

No significant changes have been made to CLUSTAL W since its release in 1994, but a new member CLUSTAL X was released in 1997 [33]. This version uses the same algorithm, but has more user-friendly GUI features.

To summarize:

Advantage(s) - strikes a balance between speed, accuracy, and memory

Disadvantage(s) - lack of objective function and no real way of quantifying the alignment

T-COFFEE

The T-COFFEE (Tree-based Consistency Objective Function for alignment Evaluation) alignment program was introduced in 2000 [28] by Notredame et al. The first algorithm to produce any meaningful improvement on the CLUSTAL W technique, is at its core still a progressive alignment method. However, it differs in that it is consistency-based and takes advantage of a plethora of available biological information. Moreover, it is designed to consider information from all of the sequences at every step of the alignment process, instead of just those sequences being aligned at that particular step.

Greedy heuristics such as CLUSTAL W had a major flaw that a misalignment in the first step propagated through the rest of the alignments and could not be rectified later as the remaining sequences were added in. T-COFFEE, although a greedy heuristic by definition, managed to minimize such error propagation by making better use of domain knowledge stored in the form of libraries. The algorithm has two essential features:

a) Use of heterogeneous data sources that refer to pairwise alignments obtained from both local and global alignments

b) Progressive alignment is a way that considers alignment between all of the pairs during the generation of the MSA

These features grant the speed of a conventional progressive alignment but with much less tendency for misalignment. The method consists of the following steps:

1) Generate a primary library of alignments between all of the sequences both globally and locally

2) Derive a set of weights for the library by assigning a weight to each pair of aligned residues, according to sequence identity

3) Combine the libraries by merging duplicate pairs into a single entry, and giving a weight of zero to pairs that do not occur

4) Assign a final weight to the residues in the library, by taking each aligned pair and checking with the remaining sequences (thus inducing consistency)

5) Progressively align by using the neighbor-joining method, and scoring residues (X_i, Y_j) to be the sum of the weights of the alignments in the library containing the

alignment X_i to Y_j

To summarize:

Advantage(s) - quite noise tolerant, and distinct accuracy improvement over CLUSTAL W, by using a combination of local and global pairwise alignments to generate the sequence library

Disadvantage(s) - too slow for real-time processing of large sets of sequences.

MUSCLE

Robert Edgar proposed the MUSCLE (Multiple Sequence Comparison by Log Expectation) algorithm in 2004 [29]. MUSCLE is a matrix-based algorithm, and like most MSA programs it starts off by constructing a tree guided by the rudimentary pairwise alignments. Following this initial step, is a refinement process that takes into account a number of parameters, such as kmer distance and Kimura distance, to produce the final MSA. There are two distinguishing features of MUSCLE:

a) To find out the distance measure for a pair of sequence it uses both the kmer distance (for unaligned pair) and the Kimura distance (for aligned pair). A kmer (k-tuple) is simply a contiguous sequence of letters of length k. The conjecture is that sequences that are related will have more kmers in common. Because this measure doesnt require an alignment, MUSCLE performs significantly faster than other MSA algorithms.

b) The algorithm can be terminated at the completion of any stage, since a MSA is available after each stage.

Distance matrices in MUSCLE are clustered using UPGMA instead of neighbor-joining, thus sacrificing adherence to taxonomy evolutionary tree in return for 'slightly improved results' [29]. There are three main stages to the MUSCLE algorithm:

1) Known as 'Draft progressive' [29], this stage begins by computing the kmer distance between each pair of sequences, producing a distance matrix which as mentioned above is clustered using UPGMA to create a sub-optimal MSA.

2) 'Improved progressive' [29] stage enhances the alignment in stage 1 by re-estimating the tree using the Kimura distance, which is more accurate but requires an alignment. A progressive technique is used to create a second MSA. Then the trees from stage 1 and 2 are compared to identify a set of nodes for which the branching order is different. A new MSA is built if the order or the nodes has changed. Otherwise the first MSA is kept.

3) The 'Refinement' [29] stage starts off by dividing the tree from stage 2 into two sub-trees by deleting an edge. A profile for each sub-tree is calculated and the two profiles are re-aligned to produce a new MSA. If the sum-of-pairs score has improved, the new alignment is kept. Otherwise it is discarded. These steps are repeated until convergence or until some threshold is reached.

To summarize:

Advantage(s) - fastest among the MSA algorithms discussed so far, and a unique way of calculating distance measure using a profile function called log expectation score.

Disadvantage(s) - not many, except for some distinct cases where other algorithms give better results.

CENTER-STAR

First proposed by Gusfield [25], the center-star algorithm for obtaining an MSA is an aberration in that it aims to provide provable solution qualities and run-time bounds thus falling under the category of approximation algorithms. In a seminal paper appropriately titled 'Efficient Methods For Multiple Sequence Alignment With Guaranteed Error Bounds' [25] the author showed that it is possible to come up with solutions of MSA optimal up to a constant factor 2 under the sum-of-pairs metric. Being a metric means the cost function must satisfy the following properties for sequences x, y, and z:

Cost[x , x] = 0 (reflexive)

Cost[x , y] $=$ Cost[y , x] ≥ 0 (symmetric)

Cost[x , y] $+$ Cost[y , z] $\geq$ Cost[x , z] (triangle inequality)

The steps for this algorithm is as follows:

1) Find the center sequence Sc by minimizing the SP metric

2) Iteratively align all N-1 sequences S_i, i = 1 ... N , i $\neq$ c to Sc following once a gap, always a gap policy

To summarize: Advantage(s) guaranteed worst case complexity of $O(N^2L^2)$, for N sequences having O(L) length

Disadvantage(s) there is a trade-off between optimization and practicality

PROBCONS

Probability Consistency-based MSA (ProbCons) [30] is a progressive alignment consistency-based algorithm that expresses the MSA problem in a unique way; it uses a three-state pair-hidden Markov model (HMM) as an alternative formulation of the sequence alignment problem where emissions correspond to traditional substitution scores based on the BLOSUM62 matrix and transitions correspond to gap penalties. ProbCons uses probabilistic consistency transformation to incorporate multiple sequence conversion information during pairwise alignment. This is a modification of the sum-of-scores method: the transformation is to re-estimate the probabilities using three-sequence alignments instead of pairwise alignments. A noteworthy feature of ProbCons is that it makes no use of any biological concepts such as evolutionary guide tree construction or position-specific gap scoring.

The ProbCon algorithm has five main steps:

1) Computation of posterior-probabilities matrices

- For every pair of sequences x and y, a matrix is computed where the terms of the matrix are the probabilities that letter xi and yj are paired in an alignment of x and y as generated by the model.

2) Computation of expected accuracies

- The expected accuracy of a pairwise alignment a between x and y to be the expected number of correctly aligned pairs of letters, divided by the length of the shorter sequence.

3) Probabilistic consistency transformation

- Re-estimate the matrix quality scores by applying the probabilistic consistency transformation.

4) Computation of a guide tree

- Use hierarchical clustering.

5) Compute progressive alignment

- Align sequence groups according to order specified in the guide tree.

To summarize:

Advantage(s) - highest overall accuracy among the methods mentioned, no rigorous tree construction

Disadvantage(s) - speed could be improved upon

2.3 Structure Alignment

As stated, a protein's biological function is determined by its 3D structure. The observation, that the retainability of these functions during evolution results from structures being more conserved than sequences, is best described by Holms and Sander - "comparing protein shapes rather than protein sequences is like using a bigger telescope that looks farther into the universe, and thus farther back in time, opening the door to detecting the most remote and most fascinating evolutionary relations" [34]. A structure alignment is an alignment whose residue correspondences are identified based on structural information. It can therefore only be computed for proteins for which the 3D structure is known. Formally put, protein structure alignment is a one-to-one mapping of evolutionary related residues in a set of N proteins. Residues that cannot be mapped to some other residue is said to be aligned with a gap '-'. The number of possible alignments between two proteins of length N$_B$ and N$_B$ grows exponentially. It is [35]

$$\sum_{k=0}^{min(N_A,N_B)} 2^k \binom{N_A}{k} \binom{N_B}{k}$$

If N$_A$ and N$_B$ are both greater than 106 (residues), there are more than 1080 possible alignments, which is more than the conceivable number of particles in the universe. It often happens that there is low sequence similarity, which can be measured by the percentage of identical matched amino acids in an alignment of two proteins. The region of sequence similarity between 20 and 35% is called the twilight zone and the region of sequence similarity below 20% the midnight zone. Structure alignment is especially important for protein pairs from these regions. For N=2 the problem is a pairwise alignment one, for which there are several well established algorithms Dali [36], SSAP [37], Eigenvalue Decomposition [38], Combinatorial Extension [39], etc. Things get really complicated when N>2, commonly known as the multiple structure alignment (MStA) problem, and this thesis is based on this class of problems. Some algorithms have been proposed over the years and the next section describes some approaches and an example for each of them.

2.4 State-of-the-art

Twenty years of continuous attempts to solve multiple structure alignments (MStAs) more accurately and efficiently have led to the development of numerous techniques [40][41][42][43]. Structural biologists have since had a bunch of niche applications where current methods could likely be applied. While an exhaustive enumeration of all available MStA techniques lies beyond the scope of this thesis, a reasonably sound categorization is discussed below.

2.4.1 Progressive alignment approach

Progressive alignment algorithms constructs a multiple alignment by starting with the most similar pair of structures and then incrementally adding more distant structures to the initial alignment, by following a guide tree. They may or may not follow the sequence order of the protein backbone and prefer to use either directly or some variation of the neighbor-joining method made famous by Feng and Doolittle [31]. Mustang [44], msTALI [45], mulPBA [46], Lupyan [42], and CE-MC [47] are some of the algorithms that use this technique to construct a multiple alignment. Mustang, proposed in 2006 by Konagurthu et al, has had some time to be played by structural biologists, and it's as good an example worth reviewing, as any, in this category of approach. Mustang builds up the alignment bottom-up by first finding similar fragment pairs and extending this seed pair to find more pairs that eventually add up to cover the entire length of the protein. It uses this strategy to perform an all-pair-all-fragments scoring among the input proteins, following which outliers are pruned and a guide tree is constructed. The proteins are then aligned progressively along the guide tree to produce the set of correspondences from which parameters for rigid body superpositions, for each protein are obtained.

Mustang, unlike msTALI or mulPBA, has been on researchers' radar for quite some time and, although [45] and [46] provide better preliminary results, the former has been used on many real-life occasions. For example, Zhang et al [48] has used Mustang to discover novel DENN proteins, and their effects on the evolution of eukaryotic intracellular membrane structures and human disease. A structure-based phylogeny for functional characterization of proteins with small barrel-like structures has been created by Agarwal et al [49] with the help of Mustang. Also, PepX, a structural database of non-redundant proteinpeptide complexes [50] is based on Mustang. However, Mustang does have some significant impediments. Apart from all the resident flaws of progressive techniques, such as dependence on initial alignments and error propagation due to frozen misalignments, it also suffers from high running time and relatively low accuracy. For instance, Mustang is outperformed in number of aligned residues, and RMSD, by MATT [51] and POSA [52], which use approaches mentioned later in this thesis.

2.4.2 Core optimization approach

This approach attempts to find a common core among the given set of proteins, and to maximize the core till the score does not improve any more. The stimulus towards adopting this approach is in the alignment of twilight zone proteins, and 'identification of structurally conserved active sites in them' [40], along with providing a threading template for structure prediction. To sum up, this approach starts by considering a minimal pseudo-protein created from pair-wise seed alignment of structures, and keeps adding newly found common sub-structures into the core thus producing an 'alignment' of sorts. This step is followed by iterative refinement of the core by various techniques, and finally report the core size and matched sub-structures. At this point either the core size or the derived pseudo-protein can be used as is, or rigid body superpositions are required to actually align the proteins in space to obtain an RMSD. Some noteworthy algorithms that use this approach are MultiProt [40], Deterministic annealing [53], MASS [54], MATT [51], MAPSCI [55], and Smolign [56]. Although Smolign is the latest entrée in this category, MATT is by far the most popular package for finding a suitable consensus structure from a set of proteins, with MAPSCI producing only marginally favorable results than MATT [55].

Several mechanisms are pursued during each step of this approach, by different algorithms. For instance, the first step of choosing an initial core is solved by Zhou et al [53] by choosing the longest protein in terms of length, whereas Smolign uses a transformation-invariant representation of local structures to come up with a maximum contact overlap that it labels as the initial consensus structure [56]. MAPSCI, on the other hand first plays with the idea of a median protein before settling for a better alternative, the 'maxcore protein' [55] as its consensus of choice. MATT proceeds in a top-down manner by grouping the set of entire proteins into g groups and iteratively merging one group at a time by dynamically assembling non-rigid similar fragments across all the proteins, till only one group remains, thus necessarily producing both a core and a multiple alignment. This makes MATT account for flexibility in the structures, a feature shared only by POSA and Smolign.

To actually align the multiple structures, Deterministinc annealing uses an integer linear programming approach [53], while MultiProt finds out the largest common set of points (LCP) [40] with the points representing backbone alpha carbon atom coordinates. Mass uses geometric hashing to put the proteins into 'bins' that give residue-residue correspondences to work with [54]. Most other algorithms in this category, including MAPSCI and Smolign, creates the alignment by iterative optimization of the core driven by an objective function and repeated rigid body superpositions.

This approach, however enticing due to speed and accuracy, is not without its flaws. For example, MATT needs extra pre-processing since it accepts proteins already aligned and segregated into g groups as input, and Deterministic annealing algorithm's choice of longest protein as its initial core is questionable at best. Another point of note is that the final core obtained through these methods is almost always a pseudo-structure, and although such cores are interesting experimental observations, they may or may not translate well in terms of biological relevance.

2.4.3 Graph based approach

Ye and Godziks' take on the MStA problem remains the only known example in this particular category of approach, where all structures are considered as partial order graphs, or directed acyclic graphs (DAG). Of course, their effort is thoroughly inspired by the success of similar formulation of the MSA problem [57]. This unique feature provides the user with genuinely flexible alignments, and apart from being used as a benchmark for many latter MStA algorithms [58], it is also applied to actual problems such as elastic shape analysis of RNAs and proteins [59].

As mentioned above, POSA represents each protein as a partial order graph (POG) of connected residues following the backbone of the protein. It starts off by using the FATCAT [60] program for pairwise flexible structure alignment which outputs the AFPs with the highest score (or within a certain threshold distance) for consideration to be included in the POG. Hinge detections is done using dynamic programming, and a bifurcation in the POG is introduced whenever a possibility of flexibility is seen. Following this step is a multiple structure alignment by constructing a high-dimensional non-planar POG [52] which accounts for turns and twists in the structures. The final alignment is chosen from this POG by optimizing some criterion, and merging appropriate branches in the POG to get a one-to-one correspondence of residues along with a subset of common substructures which in essence is a common core of the input set.

With features such as hinge detection and higher core detection even among extremely divergent structures, one might think this is the absolute method of choice for biologists. A study by Ferhatosmanoglu et al [56] suggests that POSA has maximum RMSD cost compared to Smolign and MASS. Quite unexpectedly, POSA could not detect alignments among Tim-barrel proteins and helix-bundle proteins [56]. This brings a couple of thoughts to mind. For one, MSA techniques might not be equally successful when applied to the MStA problem. And two there is still scope for improvement in terms of speed and accuracy, and thus fresh algorithms in this class of problems. A brand new study by M Mernberger [61] attempts to play with the idea of MStA being reduced to the maximal common subgraph (MCS) problem, but its still under much scrutiny as of now.

2.4.4 Pivot based approach

Our final category of approach towards the MStA problem is the pivot-based approach where one structure is chosen as a *pivot* and the rest of the structures are aligned to the pivot. MISTRAL [62], Janardan [63], and BLOMAPS [17] use this approach to construct a multiple alignment of protein structures. Although algorithms using this line of thought is relatively new and unexplored (21, 18, and 3 citations respectively as of 2014), their roots can be traced back to the famous center-star algorithm proposed by Gusfield in 1993 [25]. The center-star algorithm has been discussed in a previous section of this thesis, and is one of the few examples of MSA techniques being applied to MStA problems, while keeping the speed and quality of alignment intact. This is not the only reason why some new methods (including ours) are inclined to use the pivot-based approach. Recall that the center-star algorithm has provable bounds in terms of complexity, and produce results within an approximation ratio of 2, when applied to the MSA problem. Which makes one wonder can such provable bounds be derived for the more complex MStA problem? If so, then what would its implications be? That remains an open challenge as of now and enthusiasts can work on it in future. Meanwhile, let us explore a few subtleties that make the aforementioned algorithms different from each other.

A natural question arises as to what should be the pivot protein. Can it be chosen randomly? How sensitive is the final alignment to the choice of the pivot? As discussed below, different algorithms use different logic to get around these hurdles.

Ye and Janardan produced one of the first algorithms with an approach analogous to the center-star technique, calling it a 'center-star-like' method [63]. The difference was that instead of aligning alphabet characters representing amino acids, it aligned unit vectors derived from the protein backbones. Janardan argues that unlike MSA, MStA final output is not so much affected by the choice of pivot. However, latter studies [62] indicate that this may not be the case, since ideally the pivot molecule should be 'closest' in relation with all the other molecules being aligned. Further on [63] aligns the remaining proteins with the pivot and updates the pivot to form a new pivot that minimizes the sum-of-pair distance between them. [63] also provides proof that minimizing the SP distance between center and the remaining proteins also creates an optimal alignment that in turn minimizes the sum-of-pairs distance between each pair of proteins, since they are already aligned in space. BLOMAPS, on the other hand, 'simply takes the shortest protein as the pivot' to create what they call highly similar fragment blocks (HSFBs) [17]. A little research revealed these HSFBs to be miniature center-stars created around fragments of the pivot protein. [17] makes the use of conformational letters which reduce the protein structure into a string of alphabets where each letter represents a 'conformation' of the residue in the structure. These letters are brought together in the form of a BLOSUM-like substitution matrix called CLESUM [64]. CLESUM is a new measure of similarity between protein residues, as long as the correct letters are assigned to them. CLEPAPS [64] uses CLESUM for pairwise structure alignment. However, unlike BLOSUM, CLESUM is not derived from manually curated alignments, but from the FSSP (families of structurally similar proteins) database of Holm and Sander [5]. BLOSUM builds up the multiple alignment bottom-up using a unique anchoring and coloring scheme. A scaffold is created from these HSFBs after fragment pairs are superposed and evaluated for inclusion into the final alignment. The pivot is updated at this point and the process of coloring and fragment-based superposition is repeated till a certain cutoff threshold is reached and a multiple alignment is reported.

The MISTRAL method uses a 'piecewise-linear sigmoidal weight function to reward short separations of pairs of amino acids from proteins' [56]. Unlike other methods where similarity between protein structures is driven by the Euclidean residue-residue distances, MISTRAL models the optimal superposition of a given set of proteins by minimizing an energy function that represents protein-protein interaction. A simulated annealing scheme is then applied to the relative orientations of the proteins by superimposing fragments of 10-20 amino acids. The authors claim that longer fragments only affect the number of computations and not the quality of the alignment [62]. A center-star approach is undertaken by first computing all-pairwise structure alignments and then labelling one of the proteins as the pivot to which others are aligned. Smolign claims to supersede MISTRAL results in terms of residue correspondences, and attributes this difference to the 'protein-centric pairwise evaluation strategy' [56] in place of the 'motif-centric all-inclusive evaluation used in Smolign' [56].

Chapter 3

Methods

"Character, I am sure, lies in the genes."

- Taylor Caldwell

This chapter shows how the background given in the previous sections is applied to the MStA problem. The focus of this chapter is on Multiple Alignment of Structures using Center Of proTeins (MASCOT), a new algorithm for aligning more than two proteins at once. This is a major part of the contribution made in this thesis. The subsequent sections illustrates the data used in this project, the main idea and assumptions, and the algorithm in details.

3.1 Protein Data Bank (PDB)

The Protein data Bank (PDB) [65] was first conceived at Brookhaven National Laboratories in 1971. The archive initially contained only seven structures of macromolecules. The advent of technologies such as nuclear magnetic resonance imaging and X-ray crystallography for structure determination in the early eighties quickly increased the number of available structures. A huge boost to the bank's accessibility and exponential growth was provided by a change in the attitude towards sharing data, and above all the advent of the Internet.

All known protein structures are stored in the repository in PDB format. The PDB format contains data for each atom in the structure, viz. its type and (x,y,z) coordinates, residue number and the type of residue. Each atom takes up a single line in the PDB file. For instance, an entry in the pdb file for the globin FERRIC APLYSIA LIMACINA which has pdb code 2FAL is as follows:

ATOM 493 CA ARG A 66 56.089 1.103 41.810

The above line indicates that there is a carbon atom at the position (56.089, 1.103, 41.810). Moreover, the 'CA' shows that it is the central C_α atom of a residue, namely residue 66 of type 'ARG' from chain A. The value 493 is a unique atom identifier within the file. In short, a pdb file is a digitized version of the actual protein chemical.

3.2 Main idea and assumptions

There are three distinct hurdles to overcome when it comes to aligning multiple proteins. The first is to choose a representation of the protein structure that stays true to the 3D structure of the molecule and helps the processing at the same time. The second is to derive a set of residue-residue equivalences among all the proteins being aligned, while preserving whatever biological properties the proteins have. This is to make sure that the resultant alignment takes into account the evolutionary relevance of the protein residues. The final hurdle is to score the alignment so that it reflects the quality of the alignment obtained. MASCOT takes care of all these hurdles using a three-step process and a heuristic that selects one protein, which is most closely related to all the other proteins, as the center protein; hence the nomenclature.

3.3 Algorithm description

MASCOT works in three phases:

a) Phase 1: Preparation - sets up the data in the right way so that processing becomes easier without losing accuracy.

b) Phase 2: Processing - applies heuristic and yields evolutionarily equivalent residues across multiple proteins.

c) Phase 3: Product - produces a visualization of the multiple alignment and an RMSD value as outputs of the algorithm.

A flowchart of MASCOT is given below, followed by a pseudo-code listing of the same.

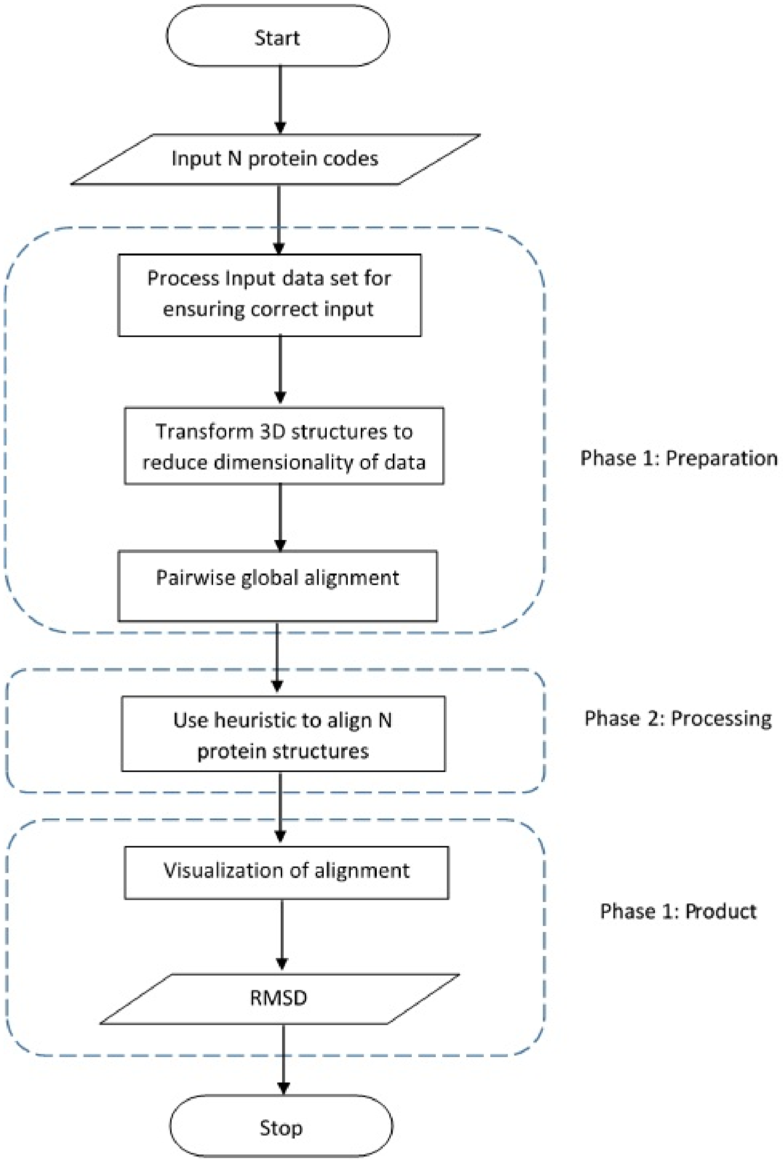

Start
Input N protein codes
Process Input data set for ensuring correct input
Transform 3D structures to reduce dimensionality of data
Pairwise global alignment
Phase 1: Preparation
Use heuristic to align N protein structures
Phase 2: Processing
Visualization of alignment
RMSD
Phase 1: Product
Stop

Fig. 3.3.1: Flowchart of MASCOT

Algorithm MASCOT

Input: PDB1(:chainid) PDB2(:chainid) .... PDBN(:chainid)

Output: Aligned PDB s, RMSD value

Process:

// phase 1

1. Extract the particular chains from PDB s and store in P_i, $1 \leq i \leq N$

2. Convert P_i to DSSP sequences S_i , $1 \leq i \leq N$

3. Align S_i and S_j using global alignment algorithm $1 \leq i \leq j \leq N, i \neq j$

// phase 2

4.Create edit distance matrix between every pair of alignments obtained above

5.Assign the protein with minimum sum-of-pairs distance, as center protein S_c

6.Iteratively align S_i to S_c, $1 \leq i \leq N$, $i \neq c$ to produce an N x L matrix called the correspondence matrix, $\max(L_j) \leq L \leq \sum L_j$, $1 \leq j \leq N$, L_j being the length of P_j

7.Assign any alignment of residues of P_i with another residue of P_j (instead of a gap) as a residue-residue equivalence between P_i and P_j , $1 \leq i \leq j \leq N, i \neq j$

// phase 3

8. Apply Kabsch's method to perform rigid body superposition of all P_i with respect to P_c

9. Apply dynamic programming using inter-residue Euclidean distance threshold to calculate 'centerRMSD' with respect to the center protein

3.3.1 Details

Step 1 - A typical input to the algorithm looks like 7API:A 8API:A 1HLE:A 1OVA:A 2ACH:A 9API:A 1PSI 1ATU 1KCT 1ATH:A 1ATT:A 1ANT:L 2ANT:L . Every entry in this list is in PDBid(:chainid) format. The chainid if present means only that particular group of atoms from the whole molecule needs to be aligned. This step begins by renumbering the residues, since at times the raw pdb files may have discrepancies regarding residue number. Then it extracts the required atoms and stores them in separate files.

Step 2 - The data till now contains individual (x,y,z) coordinates for every atom in every molecule. MASCOT reduces the dimensionality of the data by representing each residue with its role in the SSE to which it belongs. These roles represent the motif to which that residue conforms to in that protein. The residues are substituted by their DSSP elements derived from Kabsch's dssp program[15]. The following table provides the available motifs to which a residue could be assigned to:

Table 3.3.1: DSSP Motif elements

Code	Related motif
H	Alpha helix
B	Beta bridge
E	Strand
G	Helix-3
I	Helix-5
T	Turn
S	Bend
-	No motif

DSSP reliably assigns the residues to the above SSE elements [54] and the resulting one-dimensional string correctly captures the structural information of the protein. It is to be noted that the '-' dssp code is very different from the gap symbol '-' and thus residues which are labeled as '-' by the dssp program are replaced by the 'Z' symbol in the resultant string. To exemplify, a dssp sequence for a globin from a sea cucumber, with pdbid 1HLM is as follows:

...HHHHGGGZZIIIITTHHHHHHHTTSSI...

Each P_i is transformed to an S_i , $1 \leq i \leq N$, in this way and at the end of this step we have a set of sequences ready to be operated upon by powerful string matching algorithms.

Step 3 - Every pair of dssp sequences (S_i,S_j) $1 \leq i \leq j \leq N$, $i \neq j$, are aligned using a global alignment algorithm [18] and stored for later use. A custom substitution matrix such as the one below has been used to find the highest scoring alignments:

Table 3.3.2: Custom scoring matrix for residues

	H	B	E	G	I	T	S	Z
H	1	0	0	1	1	0	0	0
B	0	1	1	0	0	0	0	0
E	0	1	1	0	0	0	0	0
G	1	0	0	1	1	0	0	0
I	1	0	0	1	1	0	0	0
T	0	0	0	0	0	1	1	0
S	0	0	0	0	0	1	1	0
Z	0	0	0	0	0	0	0	1

The ’1’ entries in this matrix make sense for the following reasons:

a) H, G, and I have the same helical structure, so they could have evolved from a single structure

b) B is a similar kind of singleton residue of which E represents entire beta sheets

c) T and S are the flexible areas of a protein, thus they can be treated in the same way

d) Z is redundant

The following examples shows how this step works with three proteins 1DM1, 1MBC, 1MBA:

```
1DM1   ..ZZZHHHHHHHHHHHHHHHHTHHHHHHHHHHHHHHHSGGG-...
1MBA   ..ZZZHHHHHHHHHHHHHHHHTHHHHHHHHHHHHHHHZGGG...

1DM1   ..ZZZHHHHHHHHHHHHHHHHTHHHHHHHHHHHHHHH-SGGG...
1MBC   ..ZZZHHHHHHHHHHHHHHGGGHHHHHHHHHHHHHHHZTHHH...

1MBC   ..ZZZHHHHHHHHHHHHHHGGGHHHHHHHHHHHHHHHZTHHH...
1MBA   ..ZZZHHHHHHHHHHHHHHHHTHHHHHHHHHHHHHHHZGGGGG...
```

Step 4 - An NxN matrix is created to store the edit distances between every pair of aligned sequences(proteins). To find the edit distance we simply score the matches as 1 and rest as 0. For the above example the edit distance matrix will look like this:

Table 3.3.3: Pairwise Distance matrix

	1DM1	1MBC	1MBA
1DM1	0	37	4
1MBC	34	0	35
1MBA	7	35	0

Step 5 - From this point on we use a heuristic that minimizes the sum-of-pairs (SP) metric distance by taking entries from the matrix obtained in step 4. The formula for this is

$$\text{P}_c = \text{Protein with min } \sum_{i=1}^{N} \sum_{j=1}^{N} editdistance(P_i, P_j)$$

The protein, at index c, having the minimum SP distance is labelled as the center protein P$_c$ and its dssp sequence as S$_c$. Among the three proteins above 1DM1 is selected as the center protein since it has the lowest SP distance (equal to 41).

Step 6 - The alignment pairs (S_c,S_i) are retrieved from step 3. An empty matrix is assigned as the correspondence matrix. The first pair of alignment (S_c,S_i) is added to the matrix as is. After that the latter pairs are merged iteratively keeping with the 'once a gap, always a gap' policy to form the final correspondence matrix between N proteins, with respect to the center protein. If three sequences are HHHSGGGGGGSTTTTTVVHHHHHHVTHH, GGGHHHHHHHHHH-HHHHHVTHHHHTVTTTTTVVS, and HHVGGGGGGZTTTTTVVHHHHHTVTTHH, then the merging happens as below:

The center is HHHSGGGGGGSTTTTTVVHHHHHHVTHH

The first merge produces

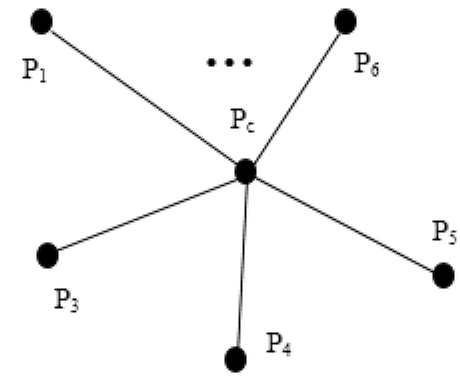
```
- - - - -HHHSGGGGGGSTTTTTVVHHHHHHVT–HH- - - - - - -
GGGHHHHHHHHHH- - - - - - -HHHHHHVTHHHHTVTTTTTVVS
```

The second merge produces

```
- - - - -HH-HSGGGGGGSTTTTTVVHHHHH-HV-T–HH- - - - - -
GGGHHHHHHHHH-H- - - - -HHHHH-HV-THHHHTVTTTTTVVS
- - - - -HHV- -GGGGGG - TTTTTVVHHHHHT-VTT- -HH- - - -
```

Notice there are no columns with gaps in all rows.

In this step MASCOT emulates the center-star method with the center protein driving the alignment. The result is an MSA of dssp sequences S_i for proteins P_i, $1 \leq i \leq N$. This is efficient since, in the case of MSA, center-star method has a defined polynomial upper bound [25].



For instance, a correspondence matrix for dssp sequences of 1DM1, 1MBC, 1MBA could be

```
ZZZHHHHHHHHHHHHHHHHTHHHHHHHHHHHHHHH-SGGGGGGS-TTTTTZZ-HHHHHHZT-HHHHHHHHHHHHHHHHHHHHT…
ZZZHHHHHHHHHHHHHHGGGHHHHHHHHHHHHHHHZTHHH---TZTTTTTZZSHHHHHHZ--HHHHHHHHHHHHHHHHHHHHT…
ZZZHHHHHHHHHHHHHHHHTHHHHHHHHHHHHHHH-ZGGGGGGZ-TTTTTZZ-HHHHHTZTTHHHHHHHHHHHHHHHHHHHHT…
```

Step 7 - The matrix obtained from the previous step is more than the sum of its parts now. It gives valuable insight into the structural similarities of the input molecules, since from this matrix we can pick any two rows i and j to get the residue-residue equivalences between proteins P_i and P_j. This step justifies the heuristic, since P_c is 'closest' in structural similarity (step 5) and any alignment driven by Pc is highly likely to induce proper residue equivalences between every other pair of proteins. An analogy can be that we can bring a group of different people together if we can identify a common friend among them. So, suppose the correspondence is as below:

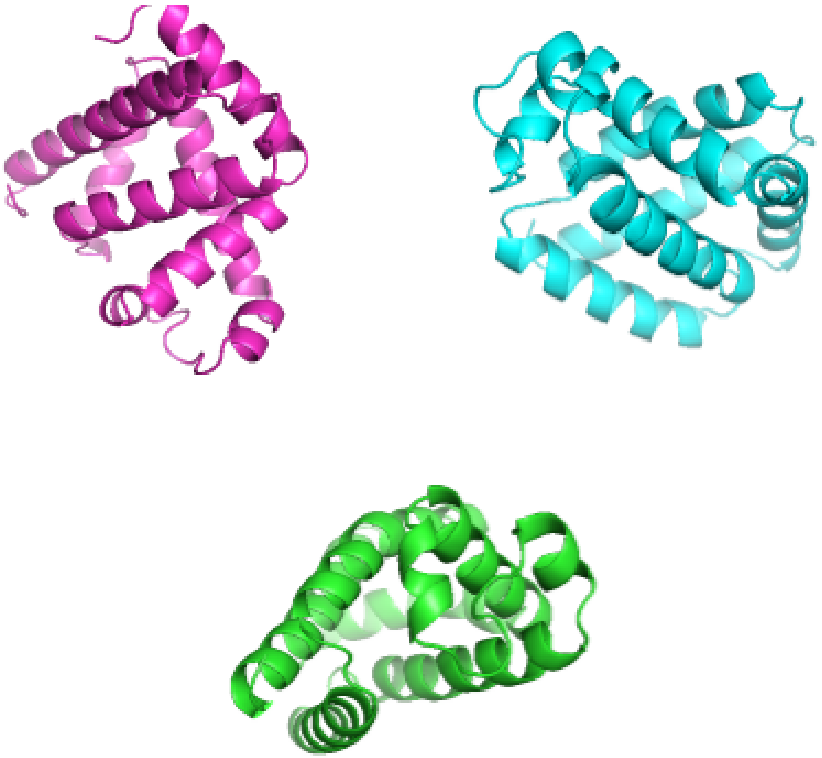
Table 3.3.4: Identifying equivalences

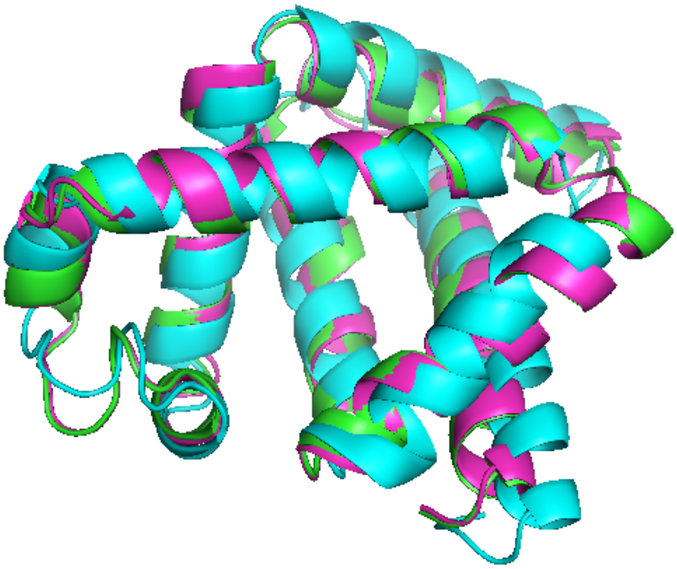
Residue no.			1	2	3	4	5		6
Center protein	-	-	H	H	T	I	E	-	G
Other protein	S	S	H	H	-	G	E	E	I
Residue no.	1	2	3	4		5	6	7	8

Some annotated equivalences between center and the other protein would be (1,3) (2,4) (4,5) (5,6) and (6,8).

Step 8 - The residue equivalences can be used to do a number of things at this stage. For example, we can use them as input to other related biological problems such as phylogeny reconstruction, druggability check etc. One of the first things to do after comparing protein structures is to visualize them in 3D space. This is done by rigid body superpositioning of one structure onto another.

MASCOT uses Kabsch's method [66] to accept these equivalences as input, and create a translation vector and a rotation matrix for the target structure, that when applied to the coordinates will bring equivalent residues close together in space. This process is iteratively applied for every protein P_i with respect to P_c. An example of such an alignment for 1DM1 (violet). 1MBC (green), and 1MBA (cyan) in 3D space is given below:





Step 9 - MASCOT reports the *centerRMSD* value as a measure of the quality of the alignment. The formula for calculating this is

$$\frac{1}{N-1}\sum_{i=1,i\neq c}^{N} RMSD(P_i, P_c)$$

Once the molecules are aligned in space, we find the corresponding residues that have Euclidean distance less than the set threshold (in our case 5Å), and calculate the root mean square deviation for each pair (P_i,P_c), $1 \leq i \leq N$, $i \neq c$. We then use the above formula to arrive at the centerRMSD.

The centerRMSD is limited by the value of the threshold taken, but a lower value (typically less than threshold/2) indicates a good multiple alignment. The centerRMSD for the above alignment is 0.443116206658 which is understandably near-perfect, given the above visualization.

Chapter 4

Results and Discussion

"However beautiful the strategy, you should occasionally look at the results."

- Winston Churchill

We implemented MASCOT in Python 2.7.5 using packages from Bio-python 2.0. In this chapter, we first present the computational results that were obtained using the algorithmic approach described in this thesis. This is followed by a discussion of the results along with some conclusions that can be drawn from them. Finally, we look into some potential limitations of MASCOT, and any possible future work that can emanate from it.

4.1 Experimental results

The following sections mention the different data sets used for investigation, and their consequent alignments. Since prioritization of case studies is not possible, the results are mentioned in the order in which the experiments were conducted. Note that T represents the time taken right from giving the input to producing the output files.

4.1.1 Globins

If one is human, chances are he/she is familiar with globins; haemoglobin and myoglobin being ubiquitous as they are. Evidently, the globin family is one of the most rigorously studied proteins in the literature [63][40][44][62]. Thus, an MStA algorithm should be able to find similarity among members of this family.

Table 4.1.1: The table below shows the globins used in this section:

Name	PDB ids	Count	T
Set 1	1HHO:A 2DHB:A 2DHB:B 1HHO:B 1MBD 1DLW 1DLY 1ECO 1IDR:A 2LH7	10	23s
Set 2	1MBC 1MBA 1DM1 1HLM 2LHB 2FAL 1HBG 1FLP 1ECA 1ASH	10	24s
Set 3	5MBN 1ECO 2HBG 2LH3 2LHB 4HHB:B 4HHB:A	7	13s
Set 4	1ASH 1ECA 1GDJ 1HLM 1MBA 1BAB:A 1EW6:A 1H97:A 1ITH:A 1SCT:A 1DLW:A 1FLP 1HBG 1LHS 1MBC 1DM1 2LHB 2FAL 1HBG 1FLP	20	1m 38s

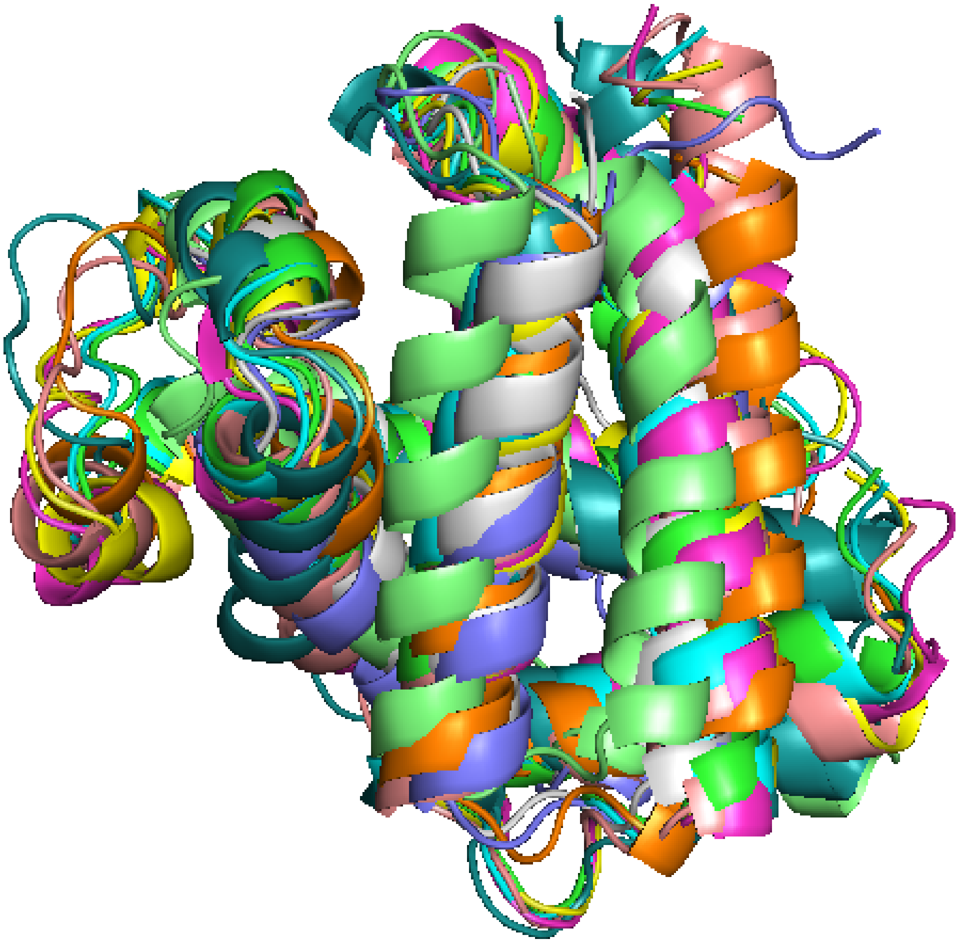


Fig. 4.1.1: Set 1

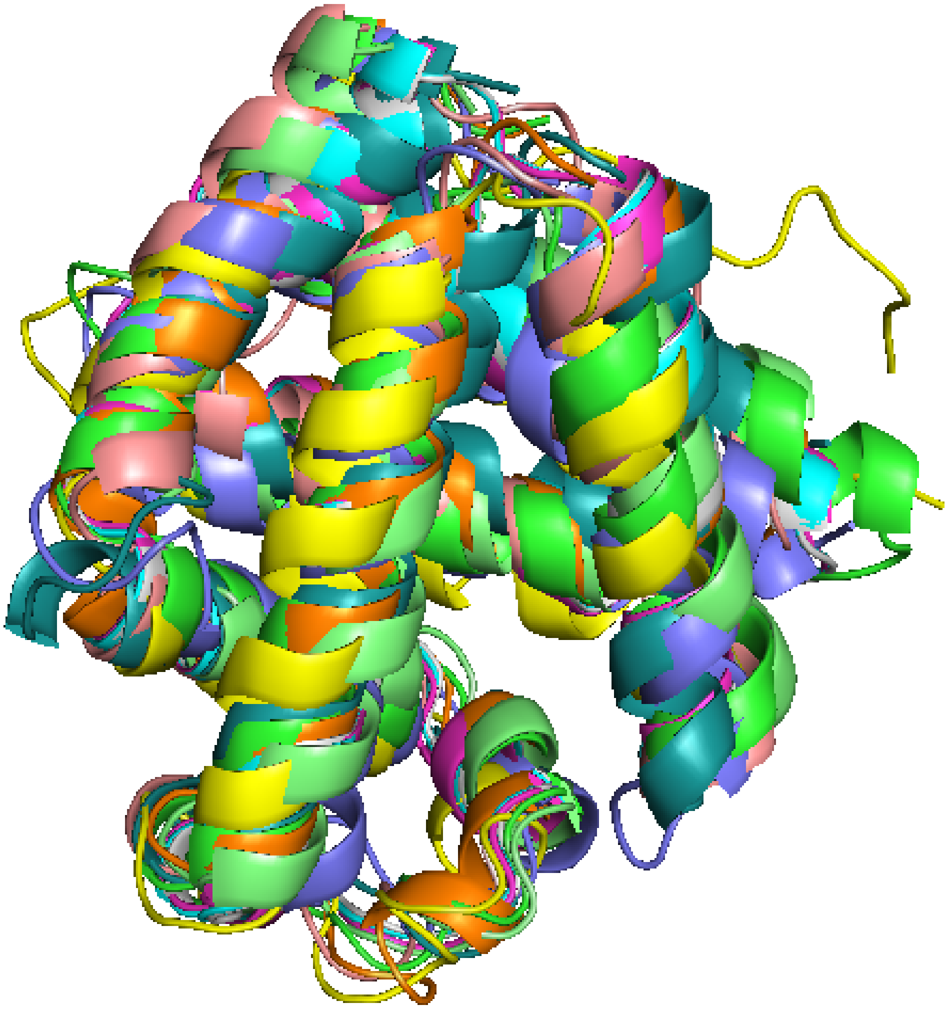


Fig. 4.1.2: Set 2

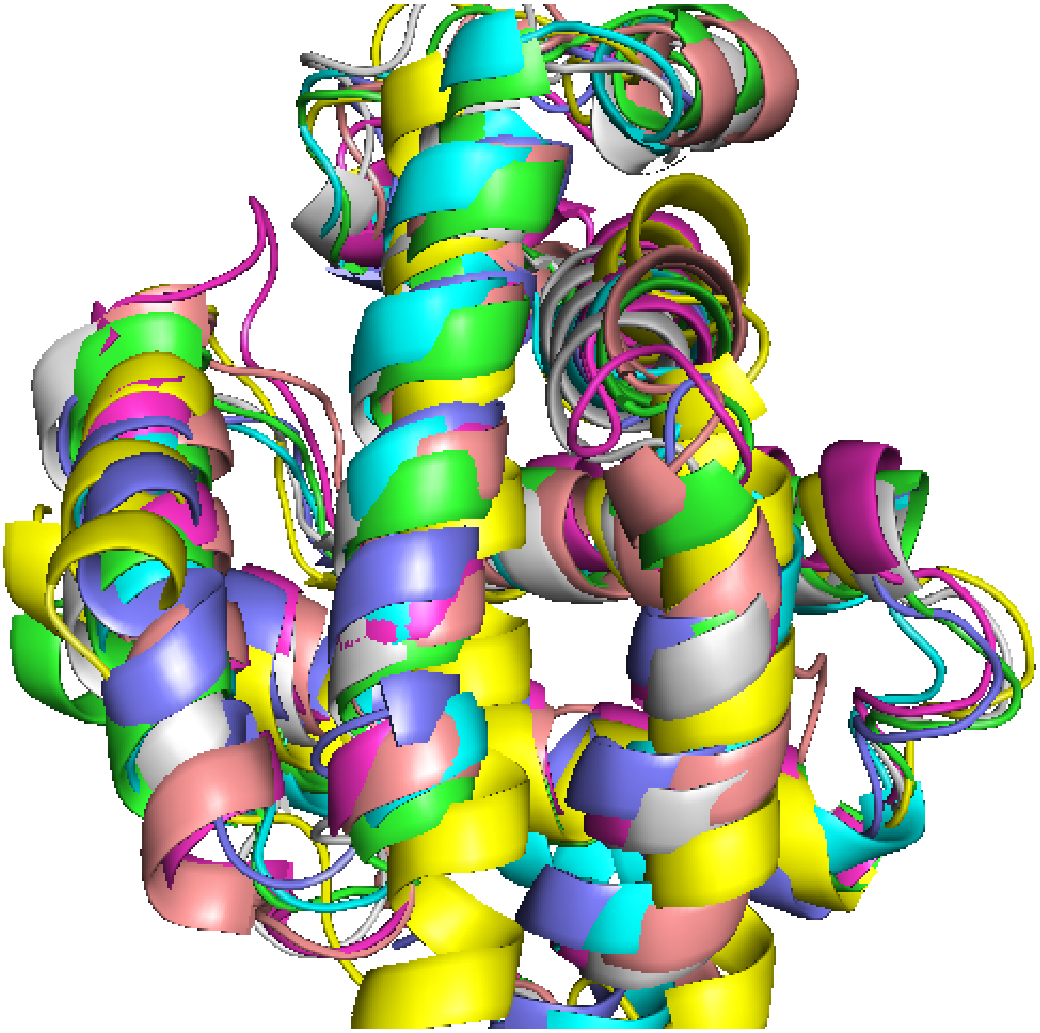


Fig. 4.1.3: Set 3

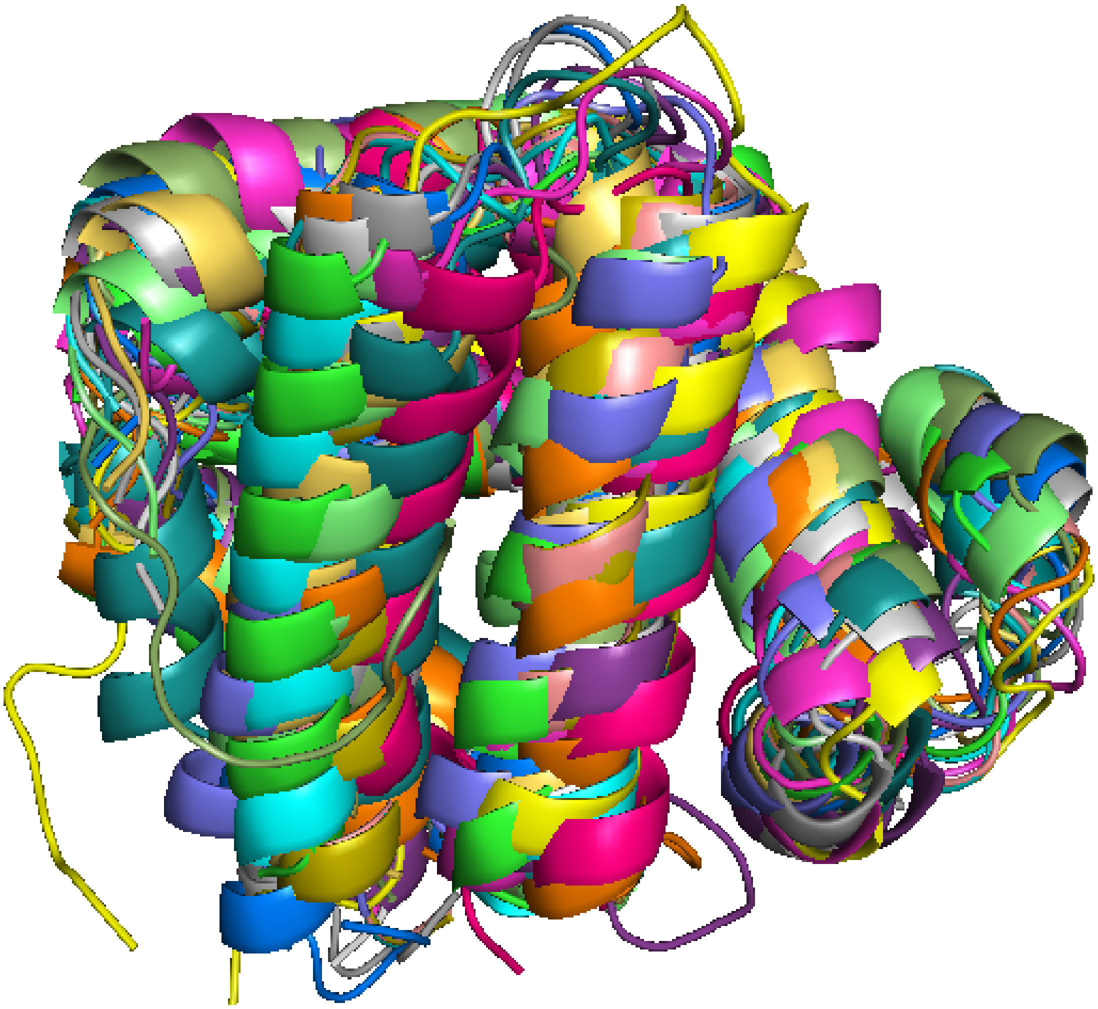


Fig. 4.1.4: Set 3

Set 1 is used by [62], [44], and [56] to show how their algorithms align globins. The rmsd for this superposition is 2.765. Set 2, [63], has been aligned with an rmsd of 2.39. Set 3 is [40]'s test data with rmsd 2.41. Set 4 is a custom assortment of 20 globins created from [63] and [42]. The purpose is to see how well they are aligned visually and with how much rmsd. As one can see, the helices and the hinges are placed within the threshold distance as much as possible, with rmsd 2.038.

4.1.2 Serpins

Serpins play an important role in the biological world. For instance, thyroxine-hinding globulin is a serpine which transports hormones to various parts of the body, and Maspin is a serpine which controls gene expression of certain tumors [67]. The name Serpin stands for Serine Protease Inhibitors. The following serpins have been aligned using MASCOT:

Table 4.1.2: The table below shows the serpins used in this section:

Name	PDB ids	Count	T
Set 5	7API:A 8API:A 1HLE:A 1OVA:A 2ACH:A 9API:A 1PSI 1ATU 1KCT 1ATH:A 1ATT:A 1ANT:L 2ANT:L	13	3m 33s

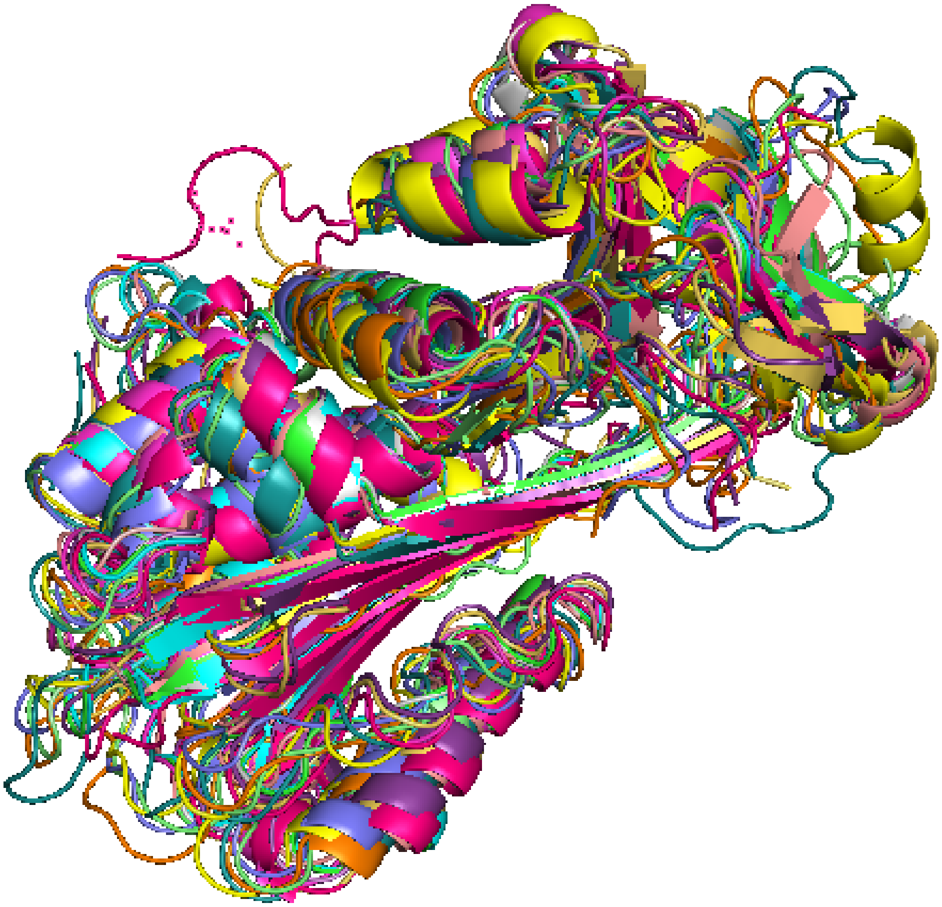


Fig. 4.1.5: Set 5

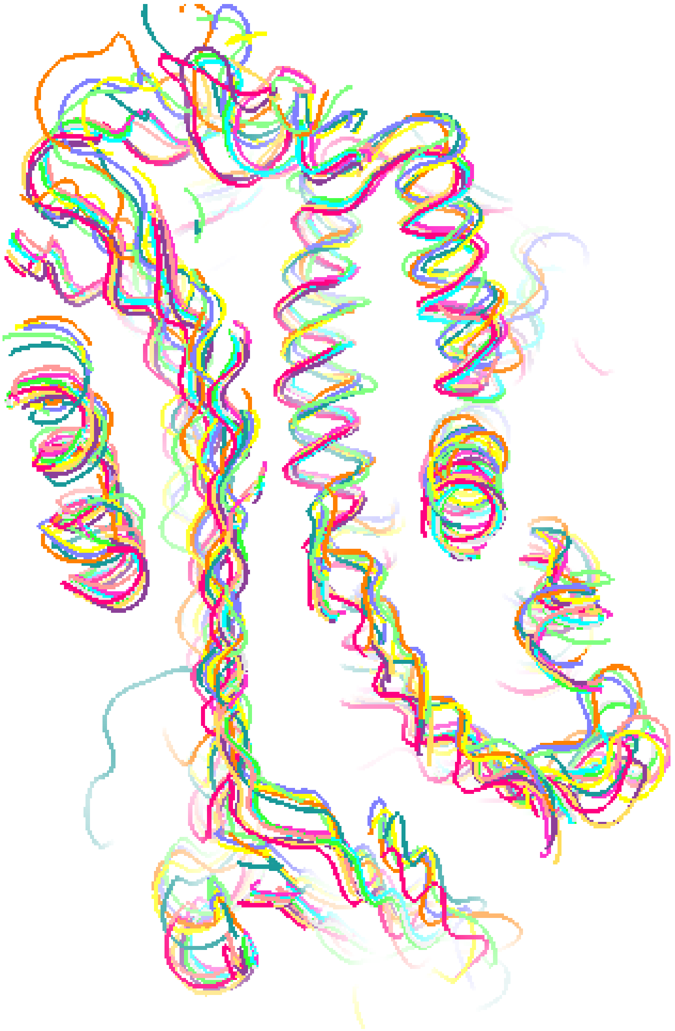


Fig. 4.1.6: Set 5 LIPR

The serpins in set 5 is the same one used by [40] and is said to be quite difficult owing to their large size and motif distribution. Unlike [40] we do not attempt to find a common core. Instead, we perform a global alignment over the length of the proteins. The first figure shows how the beta sheets, hinges, and helices are aligned together in spite of the difficulty. Also some non-alignable parts have been correctly identified and left out. The rmsd for this alignment is 2.99. The second figure is a low intensity PyMol rendition (LIPR) of the same alignment viewed from another angle. It uses a ribbon representation to condense the output and show most of the aligned portions of the proteins. The pictures suggest that all these serpins share functionality and purpose, within the body. We can club all these proteins into a single family, and keep adding to it as and when such high similarities are found.

4.1.3 Barrels

The eight-stranded TIM-barrel is found in a lot of enzymes, but the evolutionary history of this family has been the subject of rigorous debate. The ancestry of this family is still a mystery. Aligning TIM-barrel proteins will allow us to add to this ever-expanding family. The proteins aligned in this category are as follows:

Table 4.1.3: The table below shows the barrels used in this section:

Name	PDB ids	Count	T
Set 6	1A49:A 1A49:B 1A49:C 1A49:D 1A49:E 1A49:F 1A49:G 1A49:H 1A5U:A 1A5U:B 1A5U:C 1A5U:D 1A5U:E 1A5U:F 1A5U:G 1A5U:H 1AQF:A 1AQF:B 1AQF:C 1AQF:D 1AQF:E 1AQF:F 1AQF:G 1AQF:H 1F3X:A 1F3X:B 1F3X:C 1F3X:D 1F3X:E 1F3X:F 1F3X:G 1F3X:H 1PKN 1F3W:A 1F3W:B 1F3W:C 1F3W:D 1F3W:E 1F3W:F 1F3W:G 1F3W:H 1PKM 1PKL:A 1PKL:B 1PKL:C 1PKL:D 1PKL:E 1PKL:F 1PKL:G6 1PKL:H 1A3W:A 1A3W:B 1A3X:A 1A3X:B 1E0T:A 1E0T:B 1E0T:C 1E0T:D 1PKY:A 1PKY:B 1PKY:C 1PKY:D7 1E0U:A 1E0U:B 1E0U:C 1E0U:D	66	2h 25m
Set 7	1SW3:A 1SW3:B 1WYI:A 1WYI:B 2JK2:A 2JK2:B 1R2T:A 1R2T:B 1R2R:A 1R2R:B 1M5W:A 1M5W:B 1M5W:C	13	1m 22s

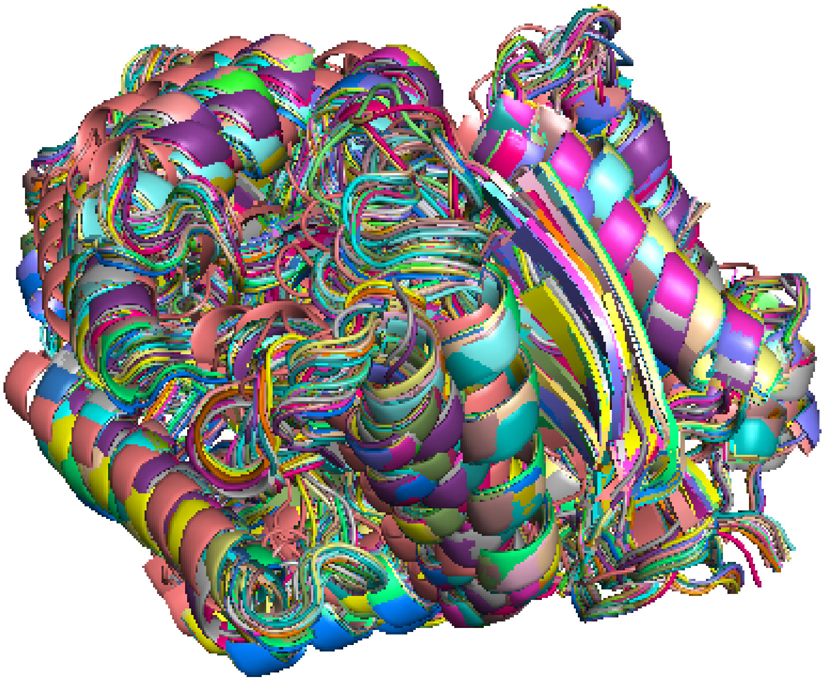


Fig. 4.1.7: Set 6

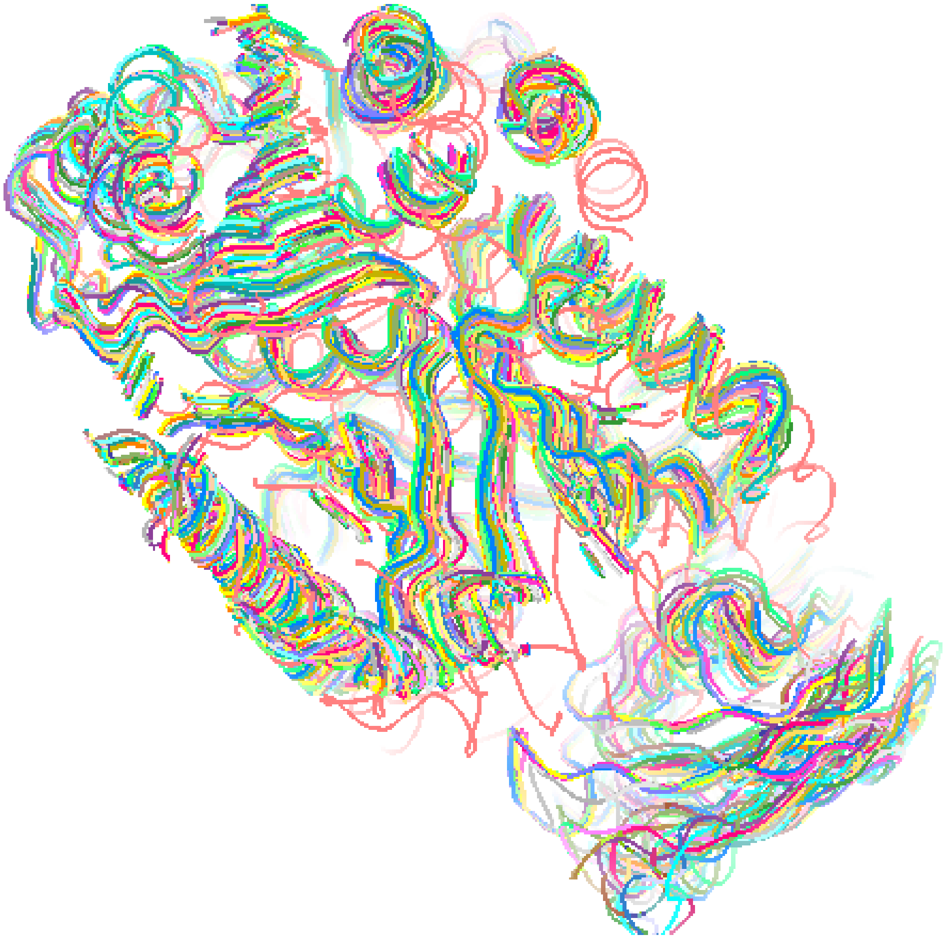


Fig. 4.1.8: Set 6 LIPR

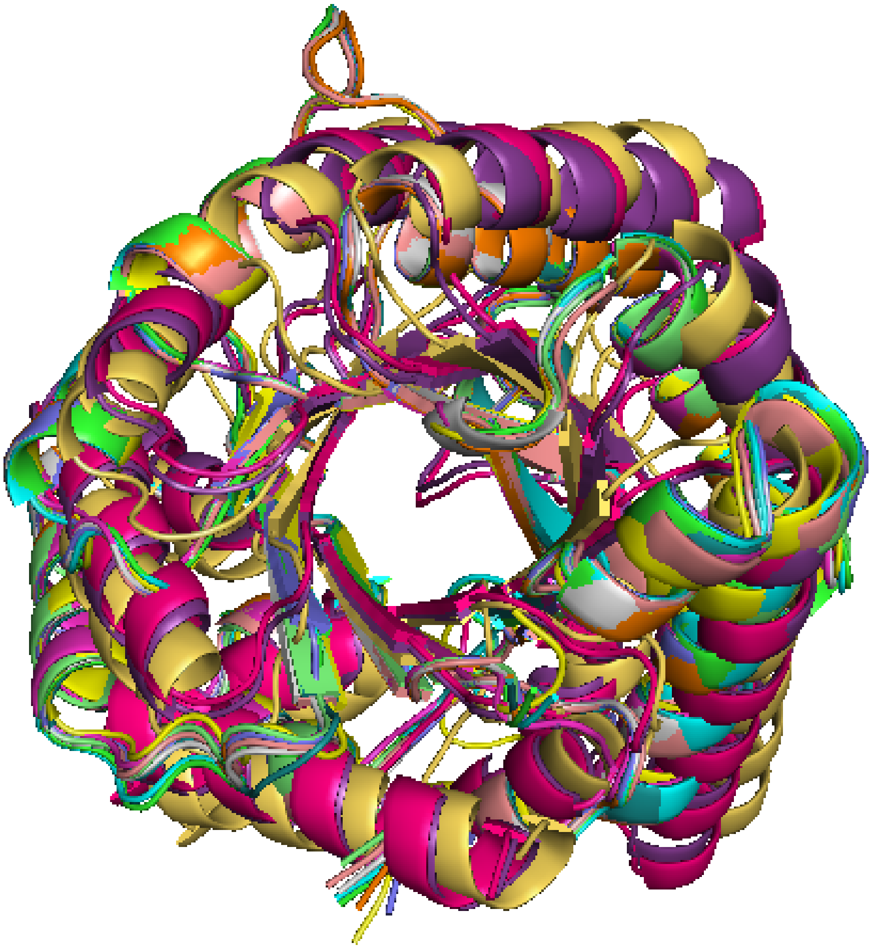


Fig. 4.1.9: Set 7

MASS [54] has used the 66 molecules in set 6 to show how it aligns proteins with barrels. MASCOT produces an rmsd of 3.4 for this alignment. The first figure shows how the new algorithm can superimpose proteins having the TIM barrel supermotifs. The second figure is an LIPR of the same alignment, for convenience. The result clearly shows these proteins have structurally highly conserved regions since all 8 helices and 8 beta sheets have been aligned. Set 7 has been taken from the gold standard manually curated SCOP database. The proteins are taken from different superfamilies but, as the third figure suggests, MASCOT is still able to align the barrel motifs on top of each other, with an rmsd of 3.76.

4.1.4 Twilight-zone proteins

Sequence alignment is still an option except when proteins have less than 30% sequence identity. The lesser the sequence similarity, the more important becomes structural comparison. Here we have taken some data sets that belong to the twilight zone.

Table 4.1.4: The table below shows the sets used in this section:

Name	PDB ids	S.I	T
Set 8	1STF:I 1MOL:A 1CEW:I	<8%	1m 55s
Set 9	1BGE:A 1BGE:B 2GMF:A 2GMF:B	<12%	5s
Set 10	1NSB 2SIM 1F8E 4DGR	<20%	19s

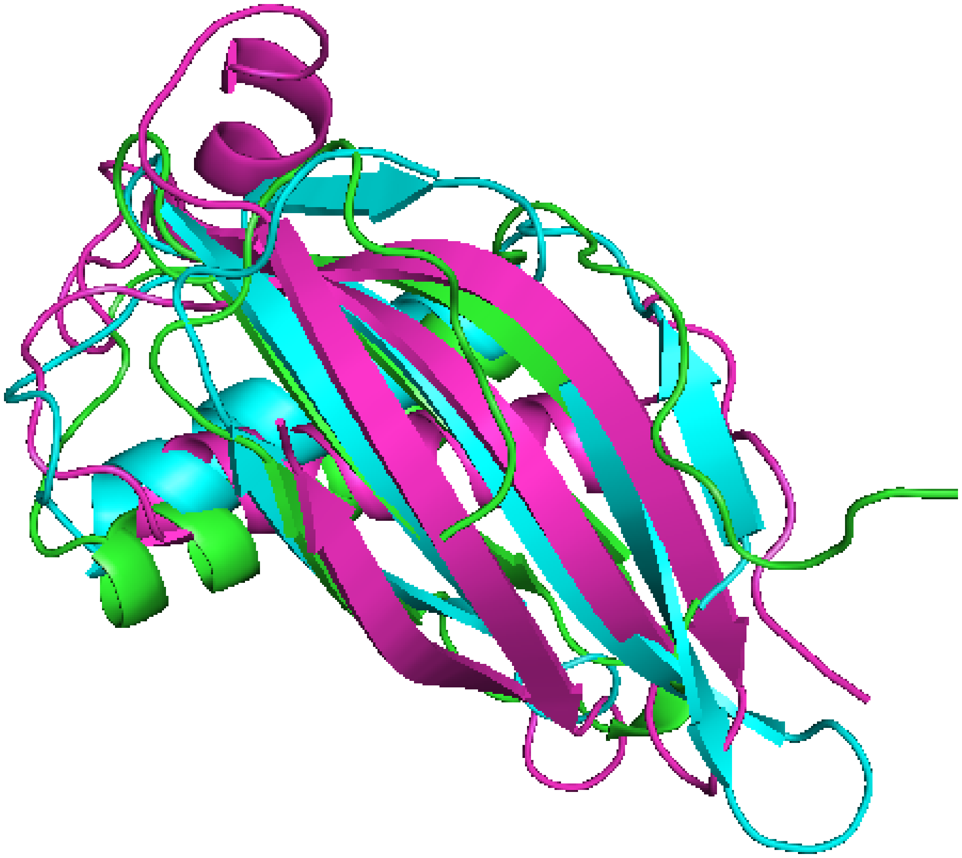


Fig. 4.1.10: Set 8

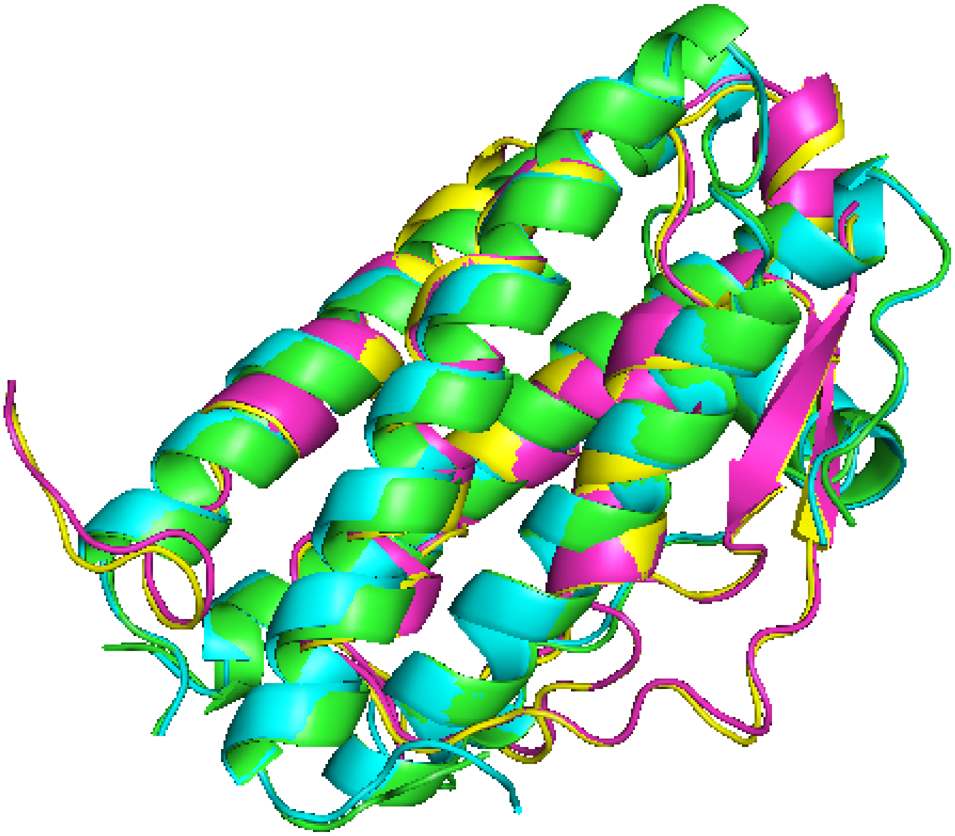


Fig. 4.1.11: Set 9

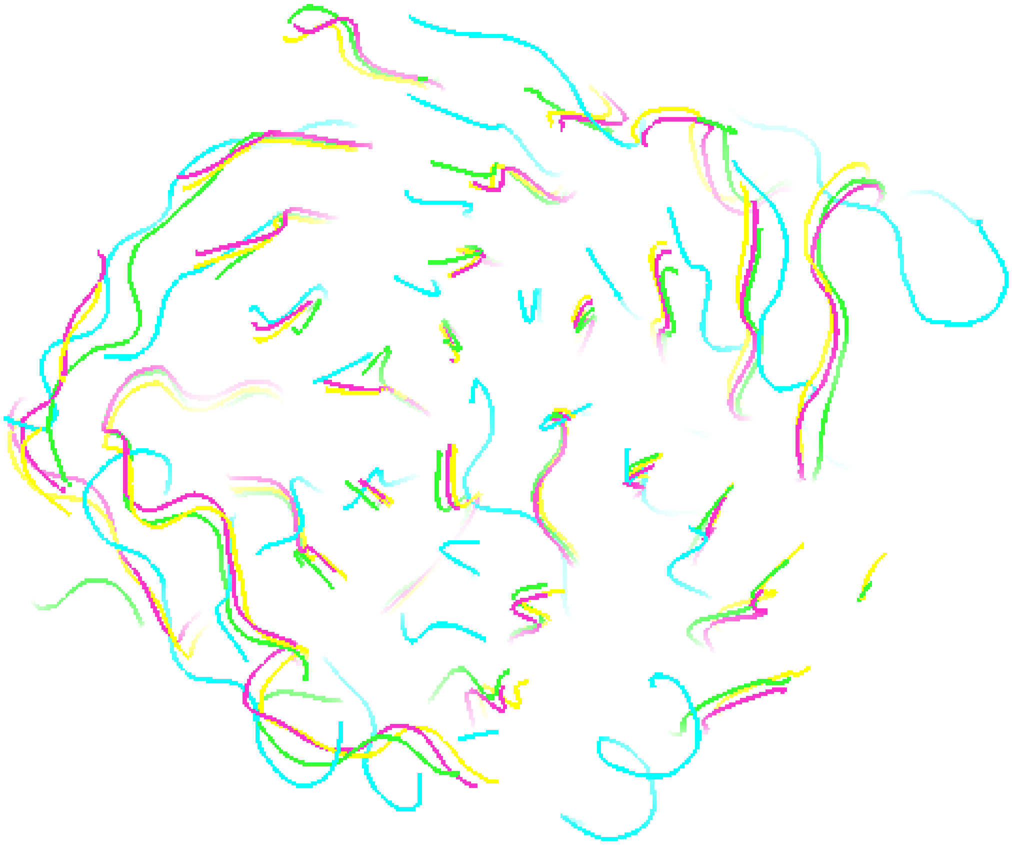


Fig. 4.1.12: Set 10 LIPR

The above 3 sets have been chosen, after numerous trials, for their significantly low sequence similarity. The motive is to show that proteins that would never have been labeled as similar, even by the most powerful MSA techniques, can be aligned using MASCOT. This is possible because the sequential representation used here consists of SSE elements and not primary residues. Set 8, 9, and 10 represent three bands of sequence identity within the twilight zone. They have rmsd of 3.61, 0.1, and 3.15 respectively.

4.1.5 Pig, Malaria, Human, and Dogfish - connected?

The 'Tree of life' has sprung many branches over millennia. Could the branches for pigs, malarial parasites, humans, and dogfish have had a common root at some point of time? The structures below have been taken from these species and an alignment is sought to gain more insight:

Table 4.1.5: The table below shows the sets used in this section:

Name	PDB ids	Count	T
Set 11	1MLD:A 1MLD:B 1MLD:C 1MLD:D 1T2D:A 1I0Z:A 1I0Z:B 1LDM:A	8	49s

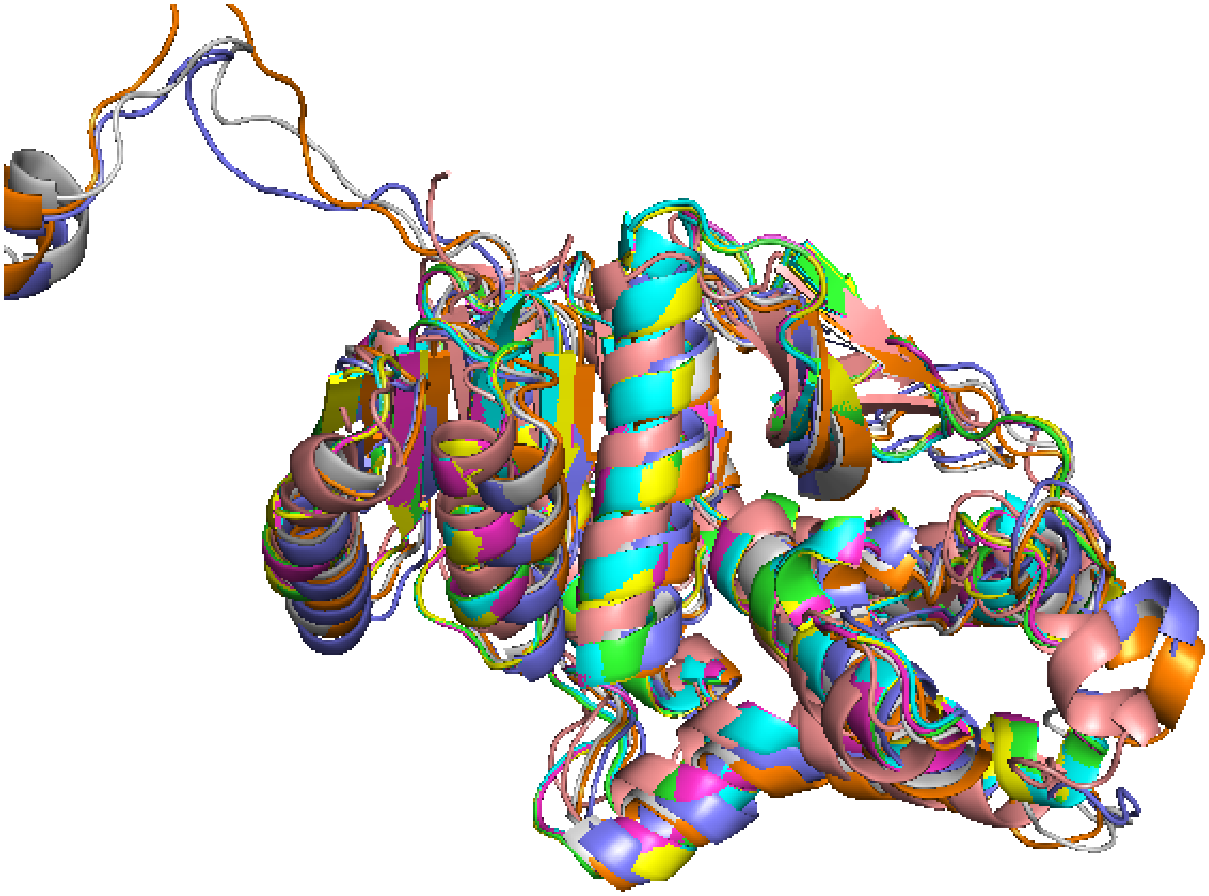


Fig. 4.1.13: Set 11

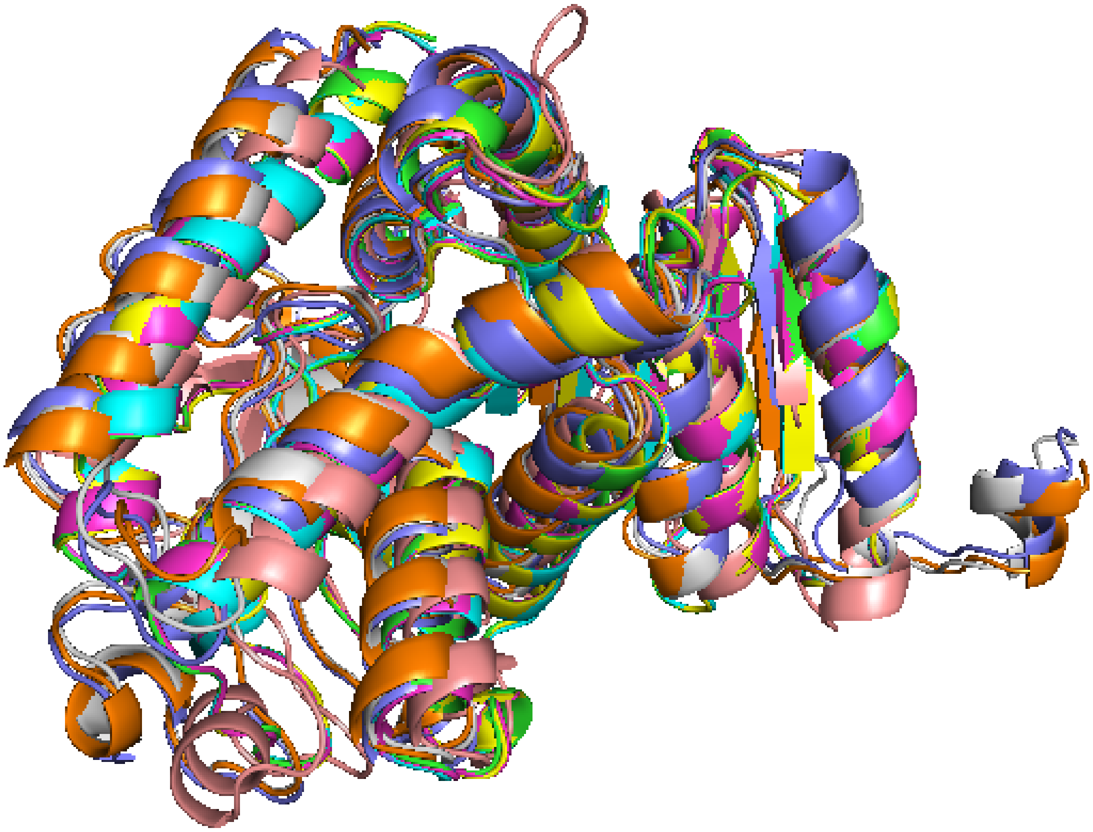


Fig. 4.1.14: Set 11

The crystal structure of mitochondrial malate dehydrogenase from porcine heart (1MLD) contains four identical subunits. Plasmodium falciparum, the causative agent of malaria, uses the protein 1T2D to enhance NAD+ regeneration. Incidentally this protein is being used for new anti-malarial drugs [1]. 1IOZ, a protein from Homo sapiens, is produced by the HRAS and HRAS1 genes [1]. 1LDM represents the crystal structure of M4 apo-lactate dehydrogenase from the spiny dogfish (Squalus acanthius) [1].

The figures above show how MASCOT finds striking similarities among these molecules with rmsd 2.885, indicating that at some point of time the branches for these species might indeed have had some common ancestor.

4.1.6 Human, Chicken, Rabbit, Yeast, and Nematode

An ensemble group of proteins have been taken from the species mentioned above. Could molecules taken from such diverse taxa be aligned to find structural similarity?

Table 4.1.6: The table below shows the sets used in this section:

Name	PDB ids	Count	T
Set 12	1SSG:A 1SSG:B 1HTI:A 1HTI:B 1R2S:A 1R2T:A 1MO0:A 1MO0:B 7TIM 3YPI	10	47s

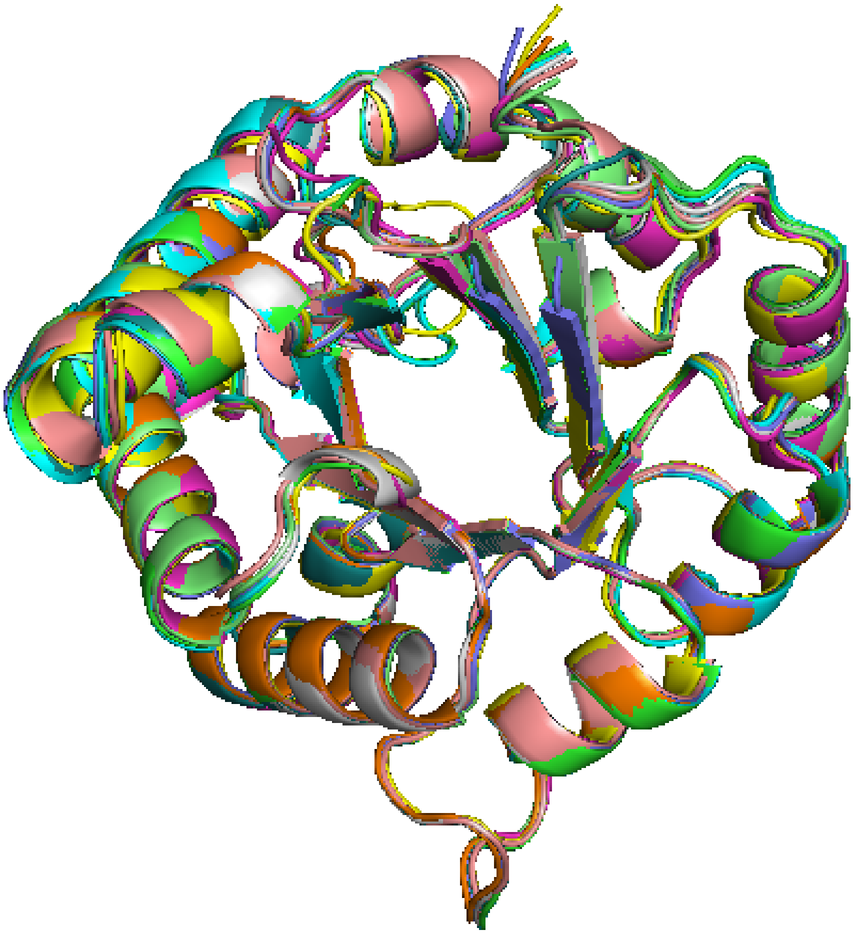


Fig. 4.1.15: Set 12

The notion that different taxa perform the same function in their own way is well known. However, the process of identifying the proteins responsible, in each organism, is daunting and expensive. Using MStA methods we can easily get around that problem by identifying one of the proteins in a wetlab, and finding similar proteins in other species to extrapolate such functionalities onto the molecules for which the function is not known. For example, glycolysis is the 'metabolic pathway' [68] using which glucose is broken down to form free energy. Now, we know that chicken does this using the protein 1SSG [1]. We could apply pairwise structural alignment to find out that humans do the same thing using protein 1HTI. But again this is time consuming when sought for many species at the same time. Instead, we can apply MASCOT to align multiple structures respectively taken from rabbit muscle (1R2S, 1R2T), baker's yeast (7TIM, 3YPI), and nematode (1MO0) and come to the conclusion that the given species perform glycolysis using these proteins. Further, an rmsd as low as 1.74 really helps us confirm that.

4.1.7 Seafood allergy in Fish!

Rats and humans are known to have allergy towards seafood. This is generally caused due to the presence of some proteins causing havoc in the immune system. Can such propensity be exhibited among fishes? This further begs the question - if at all fishes become allergic to seafood, how can they possibly survive?

Table 4.1.7: The table below shows the sets used in this section:

Name	PDB ids	Count	T
Set 13	1RWY:A 1RJV:A 4CPV 3PAL 1BU3 5PAL	6	5s

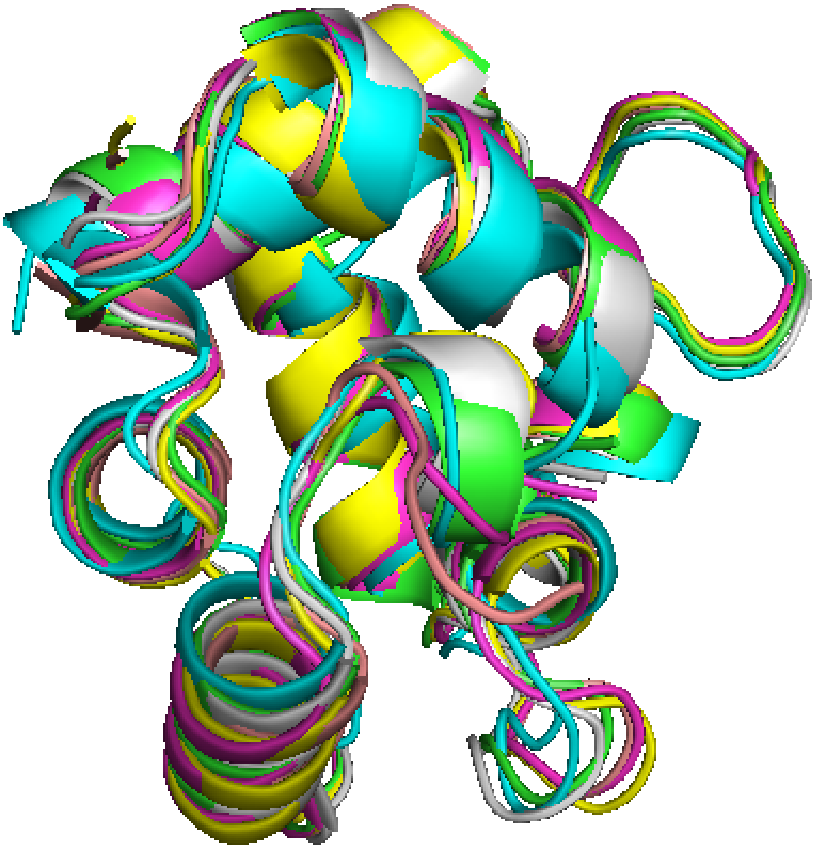


Fig. 4.1.16: Set 13

1RWY and 1RJV are known to cause seafood allergy in common brown rats and humans [1]. After experimenting on a host of proteins we found out some fishes too have proteins with similar structure. For example, proteins 4CPV, 3PAL, 1BU3, and 5PAL subsequently taken from common carp, pike, silver hake, and leopard shark have highly similar tertiary structures. Could this be an indication that these proteins might cause seafood allergy in these fishes? It turns out that indeed they do. A recent study by Swoboda et al [69] suggests that parvalbumins, such as the ones taken above, are major cross-reactive fish allergen. The picture above shows how MASCOT correctly aligns the EF hand motifs in these proteins, albeit with an rmsd of 3.82.

As to the question of how fishes allergic to their only source of food survive, we come back full circle to Darwin's theory of natural selection. There are only two possibilities at this point - either the fish will evolve through time to adapt and provide for themselves, or they will simply perish.

4.2 Conclusions

This thesis contributes towards the goal of comparing more than two protein structures, and finding biologically relevant similarities within them. To this end we focused on using a novel approach by reducing the complexity of the three dimensional structures into meaningful SSE elements, and adopting a center-star approach to arrive at equivalences.

The research work started in chapter 2 with a detailed review of MSA and MStA techniques. Important concepts and basic building blocks were defined in order to construct a solid knowledge base centered on molecular structures, mathematical methods, and current alignment strategies. A number of MSA and MStA algorithms were explored in depth, and a summary presented.

The next chapter introduced MASCOT and its inner workings. MASCOT has been designed to overcome the major hurdles of a multiple alignment by using a sum-of-pairs heuristic that associates all proteins with the one that is 'closest' to the others among the input set.

The core of this work took the form of experiments. A representative set of results from these experiments have been presented in chapter 4. Sets 1 to 6 are standard data sets used by other published algorithms. MASCOT can efficiently align the proteins belonging to the globin, serpin, and tim barrel superfamilies. The quest for uncovering hidden knowledge, in structures, continues with sets 7 through 13. Set

7 represents data taken from a gold standard database (SCOP), which is a sort of litmus test for MStA methods. Sets 8, 9, and 10 show how MASCOT totally ignores the primary sequence and finds common motifs in spite of low sequence identity. The most interesting conclusions can be drawn from sets 11, 12, and 13. For example, set 11 shows that protein structures across these species have been conserved. So, during creation of phylogenetic trees based on structure, MASCOT can be used to process subsets of proteins as sub-problems, which later combine leading up to a tree. Set 12 is a classic case of structure-function association. Looking at how proteins, from species like humans, yeast, nematode, etc. having the same structure, can also have the same function, shows results from MASCOT operate as close to biological relevance as possible. The final set, 13, was chosen not just to reassert structure-function relationship, but also to raise some important questions. Such as, is our degree of adaptability engrained down to the molecular level? Also, say somehow we manage to find a protein that nullifies the allergy-causing mutation in one species, can MStA techniques be again applied to find similar proteins to the new protein, such that a cure presents for every other species? Raising questions is a vital part of all research since it gives us something to look forward to in future, which is taken up in the next section.

4.3 Future work

With structural bioinformatics foraying into the world of bio-molecular engineering [70], and rational protein design taking over conventional drug design for personalized medicine [71], more and more analysis is being done on theoretical proteins. MASCOT currently has limited accuracy when applied on such proteins. This is where upcoming enthusiasts can find a way to improve and expand the horizon of this new algorithm. Another avenue of enrichment would be the capability to genuinely account for flexibility in proteins. As of now, the turns and bends (hinges) are being aligned intrinsically, but MASCOT does not take advantage of the flexible nature of the proteins. Further, the algorithm does not produce any 'core' structure as one of its outputs, and cases where a core structure could potentially give more insight will have to wait till MASCOT gets an upgrade, which I'm sure it will in the near future.

Chapter 5

Web Server

A user-friendly graphical user interface based web server has been made for trying out a new pairwise alignment algorithm [38]. The web server (and not the algorithm) has been part of my work and this section attempts to provide some design and documentation for the project.

To begin with, the application is available at http://uwindsor.ca/cslocal/edalign and it's hosted on our university server. Readers of this thesis are encouraged to input a pair of protein structures into this algorithm and watch what happens.

It often so happens that good algorithms exist but elude the common researcher due to lack of access and testability. The project, named EDAlignW, is an extension of EDAlign [38], in terms of usability and accessibility. It should be noted that in the making of the software, care has been taken to follow professional software development methodologies, since stability, longevity, and extensibility of any application depend heavily on them. For example, there was a feasibility study conducted to assess the amount of projected resource requirements in terms of time, man-power and hardware availability. Design was started early on and comprised of the following activities:

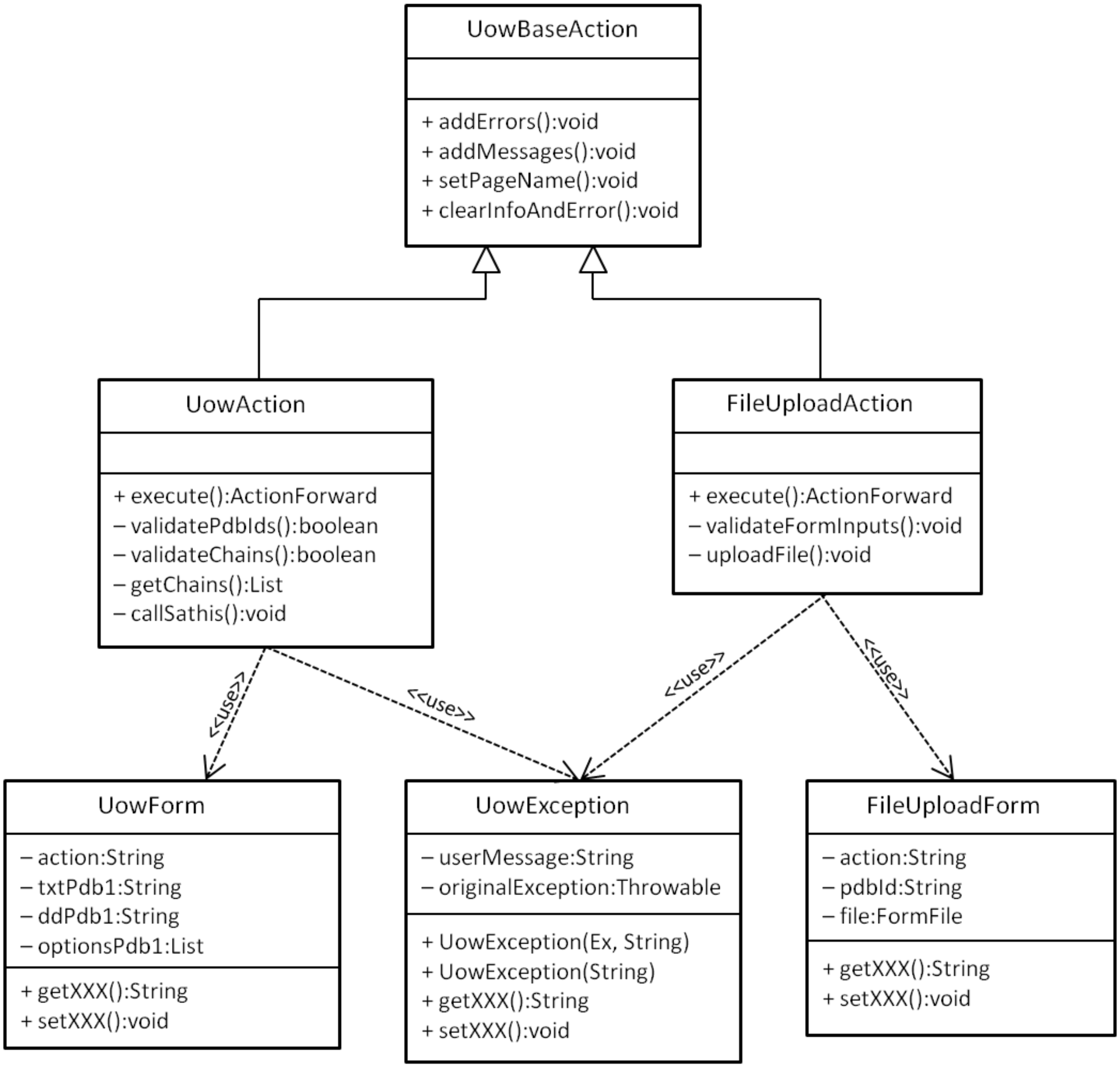
a) A pen-and-paper model for judging the scope of the application and its relevant features, taking into account the ease of use as well as satisfactory output production.

b) Playing with different technologies such as Java, Python, Django, Tomcat, Applets etc. to see which combination would provide the smoothest interconnection between modules.

c) Setting up an implementation plan, along with some tentative dates for checking of progress.

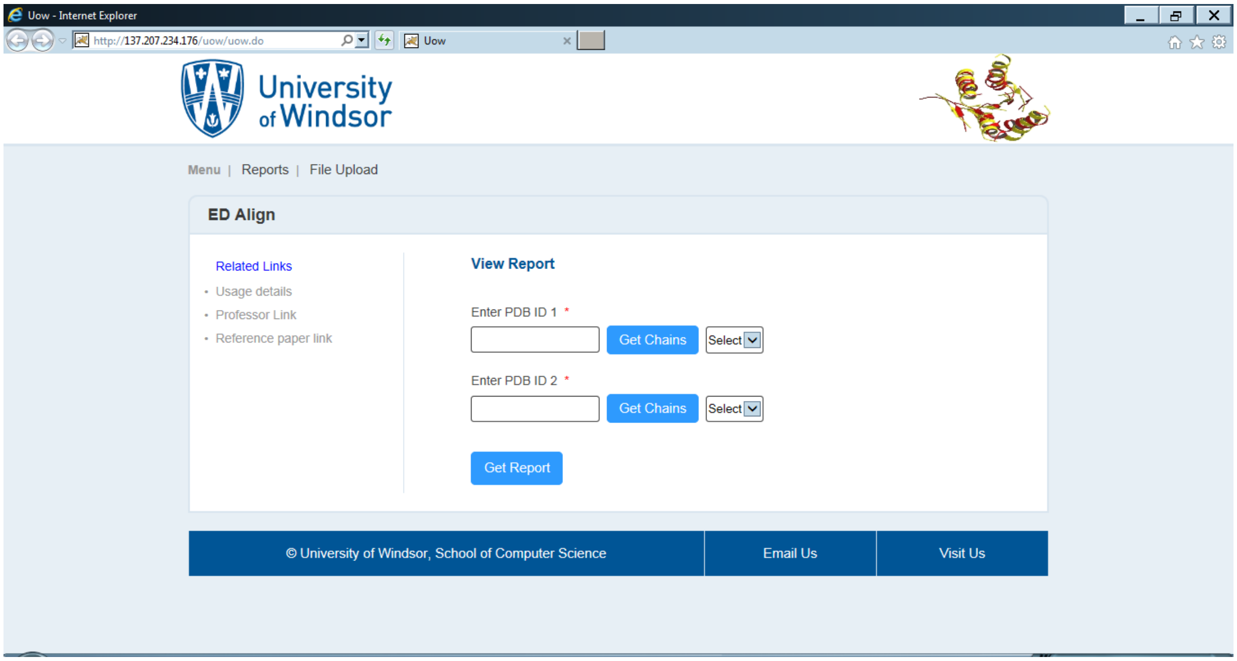
The implementation started by setting up a dedicated server, for the application, in our lab. Server is a powered by Intel Core i7-4770 and 12 GB DDR3 RAM. The web container is Apache Tomcat 7.0, and the underlying model is model view architecture (MVC) powered by Struts framework. An early prototype was created with Python and Django but on facing issues with applets not running on Django we decided to create a java wrapper around the original python application. JMol applets provide the visualization for the output.

The following self-explanatory UML diagram shows the design put into EDAlign-Web:



Note: getXXX() (set) means all getter (setter) methods used for the properties.

Its quite easy to operate the application because of the minimalist GUI. One just has to enter the two pdb ids and a choice of chain for each before clicking the submit button. JMol comes up with beautiful cartoon representations of the aligned pair of proteins. A sample execution is show with screenshots below:



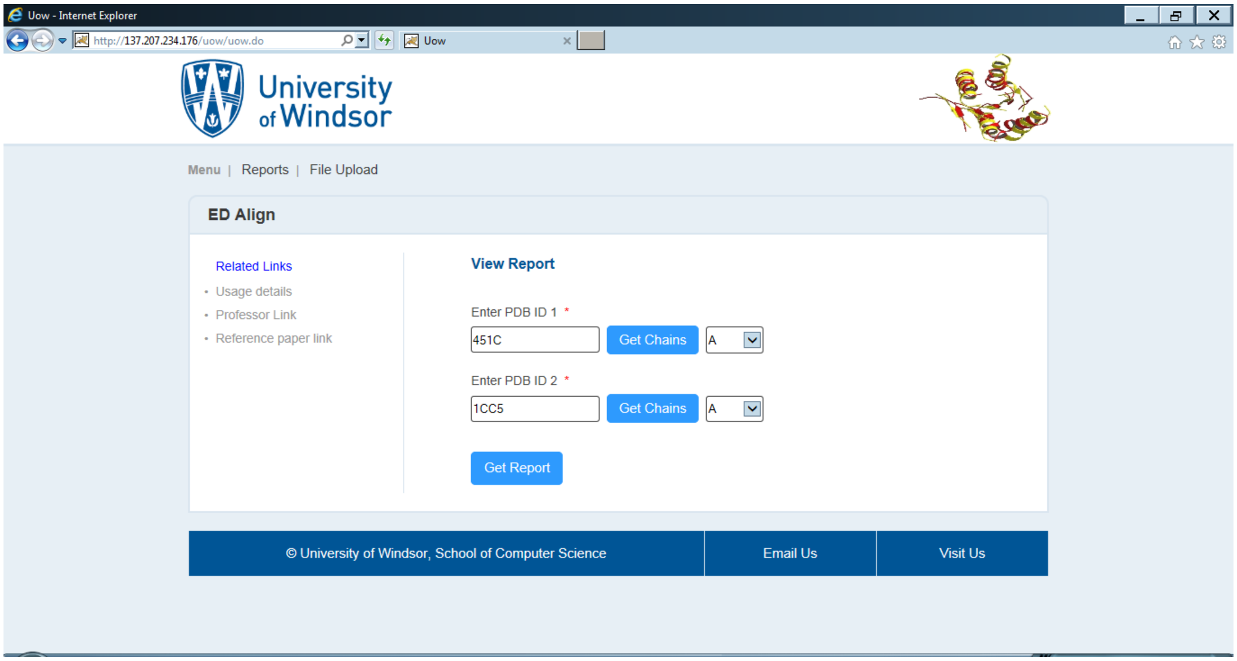
ED Align

View Report

Enter PDB ID 1 *

Enter PDB ID 2 *

The application opens with empty fields for pdb and chain ids. If the input pair is 1CC5:A and 451C:A, the screen becomes as follows:



ED Align

View Report

Enter PDB ID 1 *

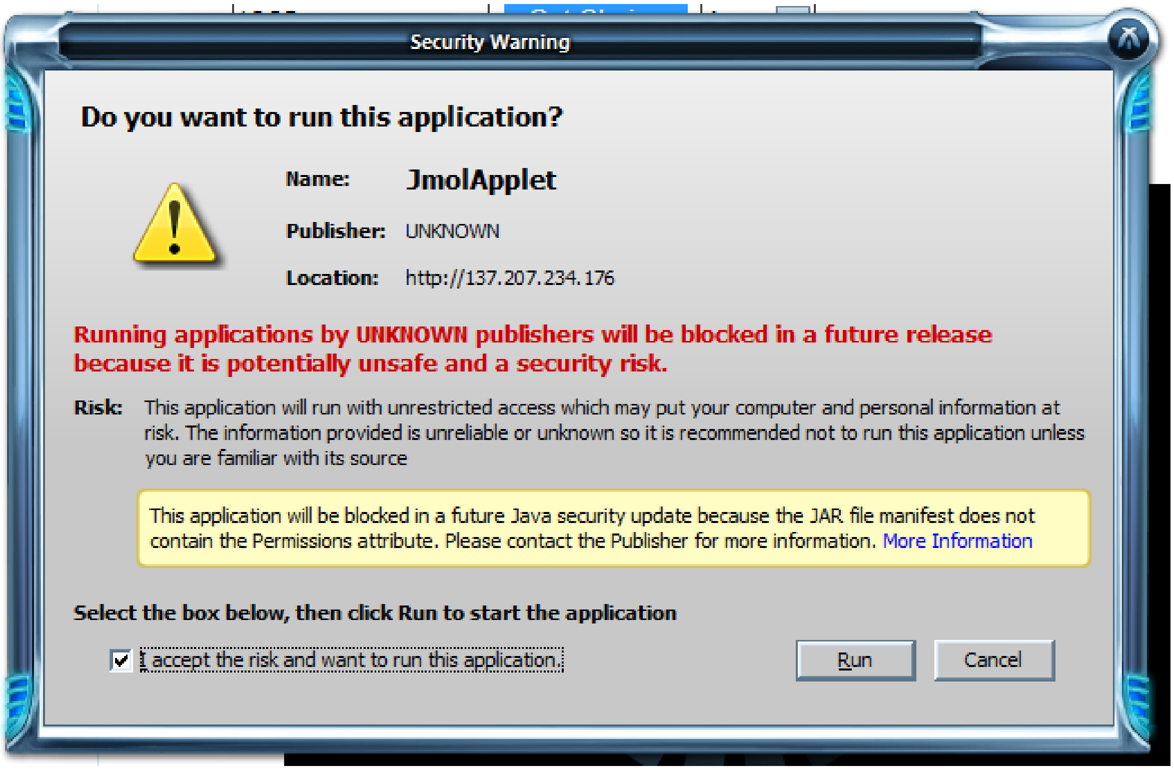
451C A

Enter PDB ID 2 *

1CC5 A

Note that one has to click on the 'get chains' button for each protein to load the dropdown menu with chains for that particular protein. The 'get chains' button internally calls an asynchronized Ajax operation that runs a bio-java script to come up with the chains which are then loaded into the dropdown menu.

On clicking the 'submit' button, an intermediary applet access confirmation window appears, since JMol provides signed archives which need to be explicitly allowed to run.



Security Warning

Do you want to run this application?

Name: JmolApplet

Publisher: UNKNOWN

Location: http://137.207.234.176

Running applications by UNKNOWN publishers will be blocked in a future release because it is potentially unsafe and a security risk.

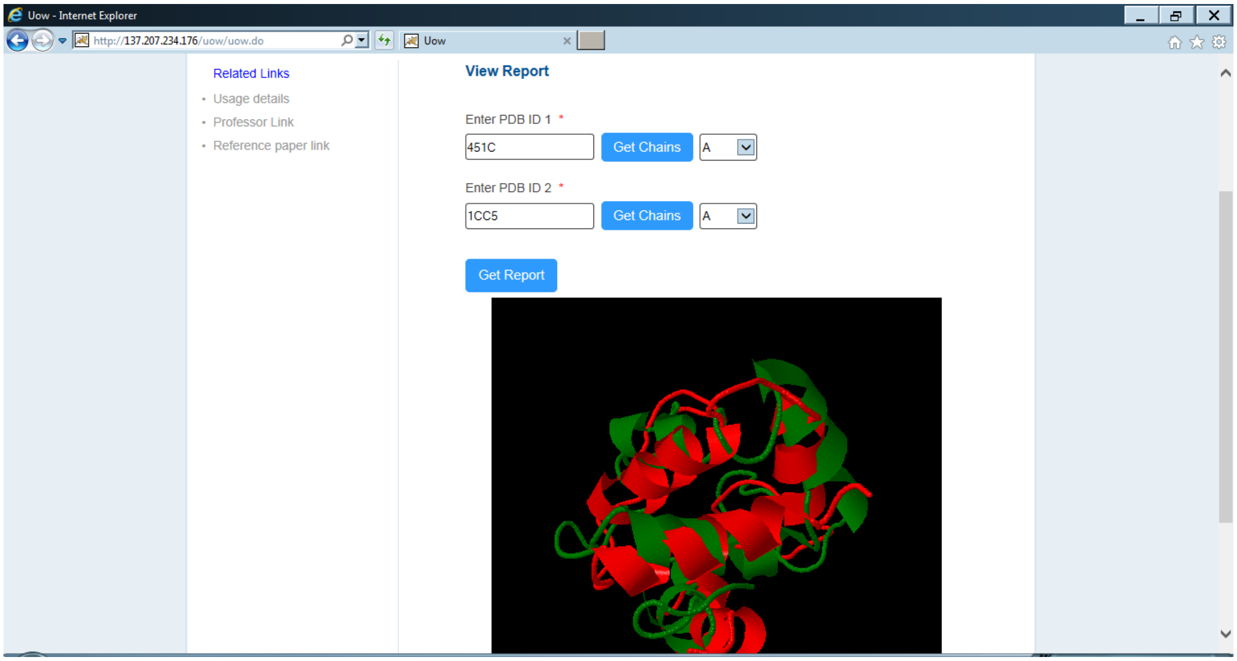
Risk: This application will run with unrestricted access which may put your computer and personal information at risk. The information provided is unreliable or unknown so it is recommended not to run this application unless you are familiar with its source

This application will be blocked in a future Java security update because the JAR file manifest does not contain the Permissions attribute. Please contact the Publisher for more information. More Information

Select the box below, then click Run to start the application

☑ I accept the risk and want to run this application.

If one choses to follow through, then the output presents as follows:



Enter PDB ID 1 * 451C A

Enter PDB ID 2 * 1CC5 A

EDAlignW will receive an update in a weeks' time when all the links will be operational, but as of now the main objective of alignment is fully functional.

Bibliography

[1] Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucleic Acids Research **28**(1) (2000) 235–242

[2] Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: Scop: a structural classification of proteins database for the investigation of sequences and structures. Journal of molecular biology **247**(4) (1995) 536–540

[3] Orengo, C.A., Michie, A., Jones, S., Jones, D.T., Swindells, M., Thornton, J.M.: Cath–a hierarchic classification of protein domain structures. Structure **5**(8) (1997) 1093–1109

[4] Piateski, G., Frawley, W.: Knowledge discovery in databases. MIT press (1991)

[5] Holm, L., Sander, C.: The fssp database: fold classification based on structure-structure alignment of proteins. Nucleic Acids Research **24**(1) (1996) 206–209

[6] Haeckel, E.H.: Generelle Morphologie der Organismen allgemeine Grundzuge der organischen Formen-Wissenschaft, mechanisch begrundet durch die von Charles Darwin reformirte Descendenz-Theorie von Ernst Haeckel: Allgemeine Entwickelungsgeschichte der Organismen kritische Grundzuge der mechanischen Wissenschaft von den entstehenden Formen der Organismen, begrundet durch die Descendenz-Theorie. Volume 2. Verlag von Georg Reimer (1866)

[7] Balaji, S., Srinivasan, N.: Comparison of sequence-based and structure-based phylogenetic trees of homologous proteins: Inferences on protein evolution. Journal of biosciences **32**(1) (2007) 83–96

[8] Balaji, S., Sujatha, S., Kumar, S.S.C., Srinivasan, N.: Pali - a database of phylogeny and alignment of homologous protein structures. Nucleic Acids Research **29**(1) (2001) 61–65

[9] Ofosu, F.A.: The United States food and drugs administration approves a generic enoxaparin. Clinical and Applied Thrombosis/Hemostasis **17**(1) (2011) 5–8

[10] Hopkins, A.L., Groom, C.R.: The druggable genome. Nature reviews Drug discovery **1**(9) (2002) 727–730

[11] Stolovitzky, G., Monroe, D., Califano, A.: Dialogue on reverse-engineering assessment and methods. Annals of the New York Academy of Sciences **1115**(1) (2007) 1–22

[12] Darwin, C.: The origin of species. Oxford University Press Oxford (1951)

[13] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., Walter, P.: The shape and structure of proteins. (2002)

[14] Branden, C., Tooze, J., et al.: Introduction to protein structure. Volume 2. Garland New York (1991)

[15] Kabsch, W., Sander, C.: Dssp: definition of secondary structure of proteins given a set of 3d coordinates. Biopolymers **22** (1983) 2577–2637

[16] Taylor, W.R.: A 'periodic table' for protein structures. Nature **416**(6881) (2002) 657–660

[17] Wang, S., Zheng, W.M.: Fast multiple alignment of protein structures using conformational letter blocks. Open Bioinformatics Journal **3** (2009) 69–83

[18] Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology **48**(3) (1970) 443–453

[19] Sankoff, D.: Matching sequences under deletion/insertion constraints. Proceedings of the National Academy of Sciences **69**(1) (1972) 4–6

[20] Dayhoff, M.O., Schwartz, R.M.: A model of evolutionary change in proteins. In: Atlas of protein sequence and structure, Citeseer (1978)

[21] Henikoff, S., Henikoff, J.G.: Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences **89**(22) (1992) 10915–10919

[22] Cosner, M.E., Jansen, R.K., Moret, B.M., Raubeson, L.A., Wang, L.S., Warnow, T., Wyman, S.K.: A new fast heuristic for computing the breakpoint phylogeny and experimental phylogenetic analyses of real and synthetic data. In: ISMB. (2000) 104–115

[23] Sinha, S., Blanchette, M., Tompa, M.: Phyme: a probabilistic algorithm for finding motifs in sets of orthologous sequences. BMC bioinformatics **5**(1) (2004) 170

[24] Andrusier, N., Mashiach, E., Nussinov, R., Wolfson, H.J.: Principles of flexible protein–protein docking. Proteins: Structure, Function, and Bioinformatics **73**(2) (2008) 271–289

[25] Gusfield, D.: Efficient methods for multiple sequence alignment with guaranteed error bounds. Bulletin of Mathematical Biology **55**(1) (1993) 141–154

[26] Notredame, C.: Recent progress in multiple sequence alignment: a survey. Pharmacogenomics **3**(1) (2002) 131–144

[27] Higgins, D.G., Sharp, P.M.: Clustal: a package for performing multiple sequence alignment on a microcomputer. Gene **73**(1) (1988) 237–244

[28] Notredame, C., Higgins, D.G., Heringa, J.: T-coffee: A novel method for fast and accurate multiple sequence alignment. Journal of molecular biology **302**(1) (2000) 205–217

[29] Edgar, R.C.: Muscle: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research **32**(5) (2004) 1792–1797

[30] Do, C.B., Mahabhashyam, M.S., Brudno, M., Batzoglou, S.: Probcons: Probabilistic consistency-based multiple sequence alignment. Genome research **15**(2) (2005) 330–340

[31] Feng, D.F., Doolittle, R.F.: Progressive sequence alignment as a prerequisite to correct phylogenetic trees. Journal of molecular evolution **25**(4) (1987) 351–360

[32] Thompson, J.D., Higgins, D.G., Gibson, T.J.: Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic acids research **22**(22) (1994) 4673–4680

[33] Thompson, J.D., Gibson, T.J., Plewniak, F., Jeanmougin, F., Higgins, D.G.: The clustal_x windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. Nucleic acids research **25**(24) (1997) 4876–4882

[34] Holm, L., Sander, C.: Mapping the protein universe. Science **273**(5275) (1996) 595–602

[35] Goldman, D., Istrail, S., Papadimitriou, C.H.: Algorithmic aspects of protein structure similarity. In: Foundations of Computer Science, 1999. 40th Annual Symposium on, IEEE (1999) 512–521

[36] Holm, L., Sander, C.: Protein structure comparison by alignment of distance matrices. Journal of molecular biology **233**(1) (1993) 123–138

[37] Orengo, C.A., Taylor, W.R.: Ssap: sequential structure alignment program for protein structure comparison. Computer methods for macromolecular sequence analysis (1996)

[38] Panigrahi, S.C., Mukhopadhyay, A.: An eigendecomposition method for protein structure alignment. In: Bioinformatics Research and Applications. Springer (2014) 24–37

[39] Shindyalov, I.N., Bourne, P.E.: Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. Protein engineering **11**(9) (1998) 739–747

[40] Shatsky, M., Nussinov, R., Wolfson, H.J.: Multiprot - a multiple protein structural alignment algorithm. In: Algorithms in Bioinformatics. Springer (2002) 235–250

[41] Eargle, J., Wright, D., Luthey-Schulten, Z.: Multiple alignment of protein structures and sequences for vmd. Bioinformatics **22**(4) (2006) 504–506

[42] Lupyan, D., Leo-Macias, A., Ortiz, A.R.: A new progressive-iterative algorithm for multiple structure alignment. Bioinformatics **21**(15) (2005) 3255–3263

[43] Krissinel, E., Henrick, K.: Multiple alignment of protein structures in three dimensions. In: Computational Life Sciences. Springer (2005) 67–78

[44] Konagurthu, A.S., Whisstock, J.C., Stuckey, P.J., Lesk, A.M.: Mustang: a multiple structural alignment algorithm. Proteins: Structure, Function, and Bioinformatics **64**(3) (2006) 559–574

[45] Shealy, P., Valafar, H.: Multiple structure alignment with mstali. BMC bioinformatics **13**(1) (2012) 105

[46] Léonard, S., Joseph, A.P., Srinivasan, N., Gelly, J.C., De Brevern, A.G.: mulpba: an efficient multiple protein structure alignment method based on a structural alphabet. Journal of Biomolecular Structure and Dynamics **32**(4) (2014) 661–668

[47] Guda, C., Lu, S., Scheeff, E.D., Bourne, P.E., Shindyalov, I.N.: Ce-mc: a multiple protein structure alignment server. Nucleic acids research **32**(suppl 2) (2004) W100–W103

[48] Zhang, D., Iyer, L.M., He, F., Aravind, L.: Discovery of novel denn proteins: implications for the evolution of eukaryotic intracellular membrane structures and human disease. Frontiers in genetics **3** (2012)

[49] Agarwal, G., Rajavel, M., Gopal, B., Srinivasan, N.: Structure-based phylogeny as a diagnostic for functional characterization of proteins with a cupin fold. PLoS One **4**(5) (2009) e5736

[50] Vanhee, P., Reumers, J., Stricher, F., Baeten, L., Serrano, L., Schymkowitz, J., Rousseau, F.: Pepx: a structural database of non-redundant protein–peptide complexes. Nucleic acids research **38**(suppl 1) (2010) D545–D551

[51] Menke, M., Berger, B., Cowen, L.: Matt: local flexibility aids protein multiple structure alignment. PLoS computational biology **4**(1) (2008) e10

[52] Ye, Y., Godzik, A.: Multiple flexible structure alignment using partial order graphs. Bioinformatics **21**(10) (2005) 2362–2369

[53] Zhou, T., Chen, L., Tang, Y., Zhang, X.: Aligning multiple protein structures by deterministic annealing. Journal of bioinformatics and computational biology **3**(04) (2005) 837–860

[54] Dror, O., Benyamini, H., Nussinov, R., Wolfson, H.: Mass: multiple structural alignment by secondary structures. Bioinformatics **19**(suppl 1) (2003) i95–i104

[55] Ye, J., Ilinkin, I., Janardan, R., Isom, A.: Multiple structure alignment and consensus identification for proteins. In: Algorithms in Bioinformatics. Springer (2006) 115–125

[56] Sun, H., Sacan, A., Ferhatosmanoglu, H., Wang, Y.: Smolign: a spatial motifs-based protein multiple structural alignment method. Computational Biology and Bioinformatics, IEEE/ACM Transactions on **9**(1) (2012) 249–261

[57] Lee, C., Grasso, C., Sharlow, M.F.: Multiple sequence alignment using partial order graphs. Bioinformatics **18**(3) (2002) 452–464

[58] Madhusudhan, M., Webb, B.M., Marti-Renom, M.A., Eswar, N., Sali, A.: Alignment of multiple protein structures based on sequence and structure features. Protein Engineering Design and Selection **22**(9) (2009) 569–574

[59] Laborde, J.M.: Elastic shape analysis of rnas and proteins. (2013)

[60] Ye, Y., Godzik, A.: Flexible structure alignment by chaining aligned fragment pairs allowing twists. Bioinformatics **19**(suppl 2) (2003) ii246–ii255

[61] Fober, T., Mernberger, M., Klebe, G., Hüllermeier, E.: Graph-based methods for protein structure comparison. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **3**(5) (2013) 307–320

[62] Micheletti, C., Orland, H.: Mistral: a tool for energy-based multiple structural alignment of proteins. Bioinformatics **25**(20) (2009) 2663–2669

[63] Ye, J., Janardan, R.: Approximate multiple protein structure alignment using the sum-of-pairs distance. Journal of Computational Biology **11**(5) (2004) 986–1000

[64] Wang, S., Zheng, W.M.: Clepaps: fast pair alignment of protein structures based on conformational letters. Journal of bioinformatics and computational biology **6**(02) (2008) 347–366

[65] Bernstein, F.C., Koetzle, T.F., Williams, G.J., Meyer, E.F., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T., Tasumi, M.: The protein data bank. European Journal of Biochemistry **80**(2) (1977) 319–324

[66] Kabsch, W.: A solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography **32**(5) (1976) 922–923

[67] Bernardo, M.M., Meng, Y., Lockett, J., Dyson, G., Dombkowski, A., Kaplun, A., Li, X., Yin, S., Dzinic, S., Olive, M., et al.: Maspin reprograms the gene expression profile of prostate carcinoma cells for differentiation. Genes & cancer **2**(11) (2011) 1009–1022

[68] Romano, A., Conway, T.: Evolution of carbohydrate metabolic pathways. Research in microbiology **147**(6) (1996) 448–455

[69] Swoboda, I., Bugajska-Schretter, A., Verdino, P., Keller, W., Sperr, W.R., Valent, P., Valenta, R., Spitzauer, S.: Recombinant carp parvalbumin, the major cross-reactive fish allergen: a tool for diagnosis and therapy of fish allergy. The Journal of Immunology **168**(9) (2002) 4576–4584

[70] Chou, K.C.: Structural bioinformatics and its impact to biomedical science. Current medicinal chemistry **11**(16) (2004) 2105–2134

[71] Lahti, J.L., Tang, G.W., Capriotti, E., Liu, T., Altman, R.B.: Bioinformatics and variability in drug response: a protein structural perspective. Journal of The Royal Society Interface **9**(72) (2012) 1409–1437

VITA AUCTORIS

Kaushik Roy was born in Calcutta, India in the year 1984. He passed bachelor of science in computer science from Calcutta University in 2005 in Calcutta, India. Later, he attended St. Xaviers' College under University of Calcutta, where he was awarded the Master of Science degree in Computer Science in the year 2007. He is currently a candidate for the Masters degree in Computer Science at the University of Windsor, Ontario and to graduate in Summer 2014.