2017

# Some studies on protein structure alignment algorithms

Shalini Bhattacharjee
*University of Windsor*

# Some studies on protein structure alignment algorithms

By

**Shalini Bhattacharjee**

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

2017

Some studies on protein structure alignment algorithms

by

Shalini Bhattacharjee

APPROVED BY:

---

M. Hlynka
Department of Mathematics and Statistics

---

D. Wu
School of Computer Science

---

A. Mukhopadhyay, Advisor
School of Computer Science

---

Y. P. Aneja, Co-Advisor
Odette School of Business

Nov 29, 2017

## 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

The alignment of two protein structures is a fundamental problem in structural bioinformatics. Their structural similarity carries with it the connotation of similar functional behavior that could be exploited in various applications. A plethora of algorithms, including one by us, is a testament to the importance of the problem. In this thesis, we propose a novel approach to measure the effectiveness of a sample of four such algorithms, $DALI$, $TM\text{-}align$, $CE$ and $EDAlign_{sse}$, for detecting structural similarities among proteins. The underlying premise is that structural proximity should translate into spatial proximity. To verify this, we carried out extensive experiments with five different datasets, each consisting of proteins from two to six different families.

In further addition to our work, we have focused on the area of computational methods for aligning multiple protein structures. This problem is known for its np-complete nature. Therefore, there are many ways to come up with a solution which can be better than the existing ones or at least as good as them. Such a solution is presented here in this thesis. We have used a heuristic algorithm which is the Progressive Multiple Alignment approach, to have the multiple sequence alignment. We used the root mean square deviation (RMSD) as a measure of alignment quality and reported this measure for a large and varied number of alignments. We also compared the execution times of our algorithm with the well-known algorithm MUSTANG for all the tested alignments.

# DEDICATION

*To my beloved grandfather Dulal Kumar Ghose, loving parents, Siddhartha and Tanuja Bhattacharjee, and my sister Seetom Dasgupta and brother-in-law Aritra Dasgupta and my loving niece Arisee and nephew Ian*

AKNOWLEDGEMENTS

This thesis owes its experience to the help, support and inspiration of several people. Firstly, I would like to express my sincere appreciation and gratitude to Dr. Asish Mukhopadhyay, without whom my masters degree would not have been possible. His constant support and guidance during my research, and his inspiring suggestions have been precious for the development of this thesis content.

I would also like to thank my thesis committee members Dr. Hlynka and Dr. Wu as well as my co-Advisor Dr. Yash P Aneja for their valuable comments and suggestions for writing this thesis. In addition, the entire computer science faculty members, graduate secretary and technical support staff deserves special thanks for all the support they provided throughout my graduation. Special thanks goes to my research partners, Zamilur and Udaya, they have been the fundamental support throughout my work. Finally, my deepest gratitude goes to my loving family for their unflagging love and unconditional support throughout my life and my studies, without whom my masters journey would not have been this incredible. You made me live the most unique, magic and carefree childhood that has made me who I am today!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

# CHAPTER 1

# *Introduction*

## 1.1 Overview

In recent years, a number of different pairwise alignment algorithm has been developed. Many focused on the areas of computational methods for aligning pairwise protein structures. Since the structural comparison problem is known to be np-complete, many different ways were proposed with good approximations. The newly proposed alignment algorithms try to find the better alignment than the previous or at least as good as an alignment achieved by the ones before.

Therefore, we have come up with an approach to evaluate some of the well- known and widely accepted pairwise alignment algorithms proposed previously. In that case, we would be able to examine which alignment algorithm works the best. In the process, we will also be able to reveal the most significant similarities that are possible within a protein family.

Additionally, we have presented a heuristic algorithm for aligning multiple protein structures. We have taken into account the progressive method in aligning the multiple proteins. To measure the similarity between the alignment, we have used the Root Mean Square Deviation (RMSD) as the metric.

## 1.2 Motivation

There are basically three main classes of motives behind these work:

1. Quality of Alignment - Once we have the pairwise evaluation techniques, we will be able to find the alignment quality of all the pairwise alignment algorithms, and as a result for a structural comparison application, or any alignment application we will be using the best algorithm, fetching the best results. It will be a measure to find the quality of new pairwise alignment algorithms. In short, we will be able to get the best alignment from the best pairwise alignment algorithm.

2. Infer Functional Properties - In the process of evaluating different pairwise alignment techniques, we will eventually be able to differentiate and classify different proteins with their functional properties. For example, proteins 1YZQ and 1T91 belongs to the same family resulting in same structures, so we can conclude that both the proteins have the high probability of performing same functions. Thus, the evaluation technique adopted will help us infer functional properties of an entire family or a group of unknown proteins.

3. Evolutionary Relationships - Biologists found that there are about 8 to 100 millions species of organisms living on Earth today. How amazing it is to think about the common ancestor of humans and mouse. A lot of studies based on the three-dimensional structures of proteins show that two proteins with insignificant sequence similarity have a common fold and therefore performing same or identical functions. Hence, it is useful to use the three- dimensional structures instead of the amino acid sequence similarities in finding evolution between distant proteins. In this work, we will be able to conclude evolutionary relationships depending on the functional similarity.

## 1.3 Problem Statement and Solution Outline

In this thesis, we proposed a novel approach to measure the effectiveness of a sample of four different pairwise alignment algorithms, DALI, TM-align, EDAlignSSE and Combinatorial Extension (CE) for detecting structural similarities among proteins and conclude their functional properties. As a result, we can classify them according to their families. We have come up with an approach and carried out extensive experiments. We have prepared five different datasets, each consisting of proteins from two to six different families. For each dataset, we computed a distance matrix, where each distance is the $cRMSD$ distance of a pair of protein structures. For each distance matrix, we used Principal Component Analysis to obtain an embedding of a set of points (each representing a protein) that realize these distances in a two-dimensional space. To compare the clustering of the families, we used the $k$-means clustering algorithm to cluster the points, sans family labels. Our conclusion is that of the four algorithms considered, $TM\text{-}align$ proved to be successful in translating structural proximity to spatial proximity followed by $CE$.

In further addition to our work, we propose a Multiple Structural Alignment method. MSA is a fundamental tool for correlating the structural similarity of proteins with their functional similarity. It is a heuristic algorithm for multiple sequence alignment, we have used the Progressive Multiple Alignment approach, in our algorithm. The steps involve building a guide tree representing the similarity between sequences; this tree will guide us through the alignment process. We will then build an alignment for each internal node of the tree, where the alignment at any internal node will have all the sequences previously aligned. The root mean square deviation (RMSD) as a metric measure of alignment quality is been used, and report this measure for a large and varied number of alignments. We will be comparing the execution times of our algorithm with the well-known algorithm MUSTANG for all the tested alignments.

## 1.4 Thesis Organization

The remainder of thesis is organized as follows. Chapter 2 explains our first problem on evaluating various widely available pairwise alignment algorithms, it gives a detailed description of the problem and the technique used for evaluating the various Pairwise Alignment Algorithm and outputs the best alignment algorithm among all. Chapter 3 presents our second problem on proposing a heuristic method in aligning more than two proteins using a progressive alignment approach. Chapter 4 contains the conclusions acquired from the two problems.

## 1.5 Fundamentals of Protein and Protein Structures

### 1.5.1 What are Proteins?

Small molecules resulting into gigantic sequential molecules are known as Proteins [11]. Made out of hundred different amino acids found in nature, they are connected by peptide bonds to form single or multiple polypeptide chains of polymers. In protein synthesis process, only twenty amino acids are created by ribosomes [59]. In the process of peptide bond formation between amino and carboxyl acid groups from adjacent amino acids, a water molecule is released and the residue formed is the remains of amino acid [59]. The amino acid sequence is called Primary Structure of a protein. Each Protein has a unique structure. The backbone of a protein comes into existence only when all the liberated amino acids have bonded together to form a residue. A protein comprises of Nitrogen (N) atom from one amino group, central $\alpha$-carbon atom, and a Carbon (C) atom from the carboxylic group repeated in a triplet, one for each residue which have very rigid peptide bonds between them resulting into only two rotatable bonds along the protein backbone; the bond between the Ca atom and its C atom neighbor, and between the Ca atom and its N atom neighbor

[59].

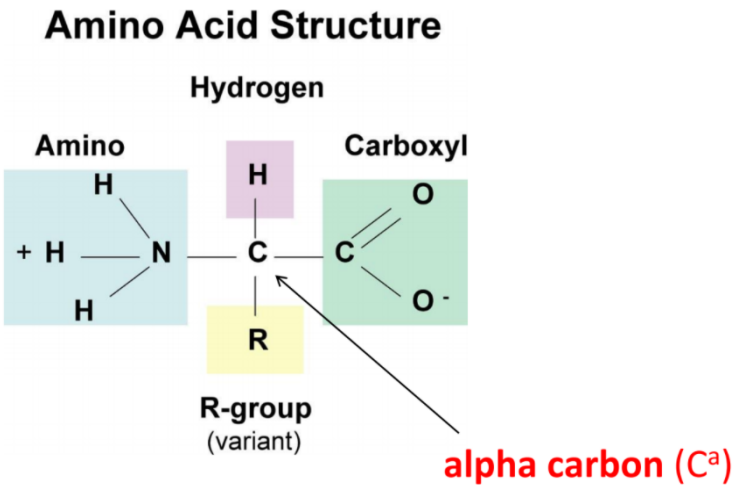



FIGURE 1: Amino acid Structure in Protein [33]

Thus, the complete 3D structure of proteins is determined by rotational states of the above-mentioned bonds in every residue. The angles of these two bonds are denoted by phi and psi [11]. There two main types of proteins, namely, Globular proteins and Fibrous proteins, of which, the former plays an active role in the expression of genes, categorizing metabolic process and replication; and later are more passive and often serve a structural purpose. Globular proteins are sphere-shaped, compact, nonrepetitive, between 100 to 300 residues in size and are considered to be the workhorses of the cell. Lastly, there are membranes, which regulates and controls various atoms and molecules traffic across the cells [59].

### 1.5.2 What are their functions?

There are several functions of a protein, the main are as follows:

1. Antibody - Protein helps in binding with foreign particles to help and protect the body from unfavorable conditions within the body. It helps in fighting with a virus inside the body. It

strengthens one's immune system.

2. Enzyme - Proteins involves in chemical reactions, resulting in the formation of new molecules by reading the information which is stored in DNA.

3. Messenger - Proteins plays a very important role in transmitting signals between cells, tissues, and muscles to coordinate biological processes. It helps in all the biological processes needed for the body to live and sustain.

4. Structural Component - Most importantly, proteins provides structure and support to cells which in return, helps in moving the body and performing the daily routine need in ones' life.

5. Transport - Proteins play a vital role in transporting atoms throughout the body by binding itself to a particle ad carrying atoms within cells from one part of the body to another.

### 1.5.3 How are Proteins structured?

Protein structure organization is spread across four levels, namely, primary, secondary, tertiary and quaternary structure [2], shown in Fig. 2.

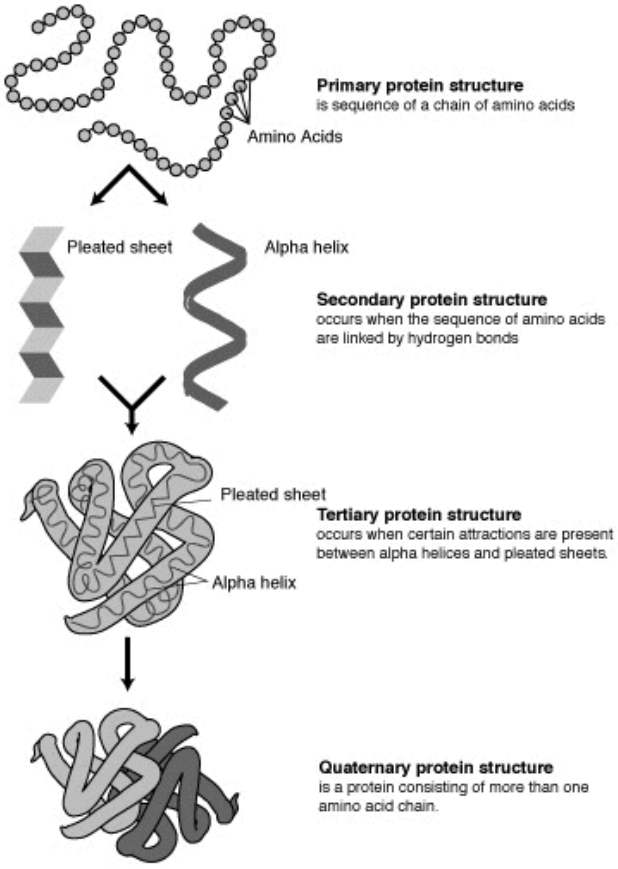



FIGURE 2: Different kinds of Protein structures [74]

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

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

FIGURE 3: Amino acids in Proteins

A structure comprising of elements that are stabilized by hydrogen bonds between the carboxyl group (C=O) and amide group (N-H) of two peptide bonds represents the secondary structure [59]. A large number of non-covalent interactions between amino acids results in three dimensional folded arrangement of protein, which is known as the tertiary structure. When non-covalent interactions bind multiple polypeptides into a one single larger protein, it is termed as quaternary structure. For example, Haemoglobin, which has a quaternary structure resulting from a bond between two alpha globin and two beta globin polyproteins [11].

### 1.5.4 Relatability between a pair of protein

The external conditions and chemical factors cause random genetic mutations resulting into the metamorphosis of proteins; this is a direct consequence of relatedness between proteins. The process of protein alignment is designed to find out the genetic information that is preserved amongst the proteins over the time and not reversing the evolution effects [59]. Using this technique common ancestor can be traced with the help of simulated agents which results in achieving categorization of evolutionary similarities and differences. This further helps in building an evolutionary tree based on which the families of different protein species can be grouped [59]. If any of the protein spices have a common ancestor, the proteins of that species can be closely related or not relatedly at all, depending on the evolution that might have occurred. There are 3 kinds of relatedness between the pair of protein(s):

1. Identity: If the formations of two proteins match with each other, they are termed as identical proteins.

2. Similarity: If the formations proteins are nearly related to each other without being identical, they are termed as similar proteins.

3. Homology: Homology is a special case in which proteins are expected to have common ancestors. Creation of super families of proteins is based on two kinds of homology that protein represents, namely, Sequence and Structural.

# CHAPTER 2

# *Low Dimensional Clustering to Evaluate Pairwise Protein Structure Alignment Algorithms*

## 2.1 Introduction

### 2.1.1 Life Origins

At the molecular level, protein molecules are known as the main drivers of all life processes. A protein molecule is defined as a linear polypeptide chain, which contains adjacent pairs of amino acids joined to each other by peptide bonds [54]. Therefore, it has the nomenclature "polypeptide". This linear polypeptide chain of the protein structure folds into a stable, low-energy 3-dimensional tertiary structure to perform its respective biological function [54]. There are other different structures in protein which are formed by loops joining together two types of secondary structures, known as $\alpha$-helices and $\beta$-sheets.

"The most important things we know about proteins have come therefore not from theory but observation and comparison of sequences and structures" is observed by Taylor et al. [68]. There is a wide number of heuristics approaches that have been proposed which shows the importance

of this problem, as a result, it led to a board spectrum of the literature of this problem and several large structural databases of proteins [49, 52, 30]. These databases help in classifying a large number of protein sequences into their structurally equivalent classes using an alignment or structure comparison algorithms.

There are namely two vital traits for pairwise protein structure alignment: (1) The way in which folding occurs; (2) How an accurate role is performed by assuming a particular structure. The protein folding is a well-known problem because of theorganic process drawback, predicting how a macromolecule can fold, given the amino acid compound sequence that makes up its peptide chain structure. A comprehensive solution is addressed for this disadvantage. The second drawback is that of predicting role of a known structure. Here a theoretical approach is a standard one: structural comparison with proteins of known functions [54], which in return give rise to the drawback of the structural alignment of a pairs of proteins.

In this work, we have come up with a solution to measure the effective of a sample of four different pairwise alignment algorithms. The four different pairwise alignment algorithms are: DALI, TM-Align, EDAlignSSE and Combinatorila Extension (CE).

### 2.1.2 Notations and Definitions of a Protein

A protein is modeled as a sequence of points, $P = \{p_i | p_i \in R^3, i = 1, 2, 3, ..., m\}$, in a 3-dimensional Euclidean space, where $m(= |P|)$ is the number residues and $p_i$ represents the coordinates of the central $\alpha$-carbon atom of the $i$-th residue. In what follows, we will use this sequence $P$ to refer to the protein it represents. [54]

Given two proteins $P$ and $Q$ of length $m$ and $n$ respectively, an alignment of $P$ and $Q$ is:

- a sequence of corresponding pairs of points of $P$ and $Q$, $S(P, Q) = \{(p_{i_1}, q_{j_1}), (p_{i_2}, q_{j_2}), ..., (p_{i_k}, q_{j_k})\}$

[54], where $1 \leq i_1 < i_2 < ... < i_k \leq m$ and $1 \leq j_1 \neq j_2 \neq ... \neq j_k \leq n$, together with

- a rigid transformation $t$, $t(Q) = \{t(q_j) = q_j^{'} | q_j^{'} \in R^3, j = 1, 2, 3, ..., n\}$, that optimizes some similarity measure for the above correspondence [54].

### 2.1.3 Pairwise Protein Structure Alignment

The outline of a Pairwise Protein Structure Alignment problem is: Given two protein structures, we find the transformation that produces the best superposition of one protein onto the other. From the computational point of view, the rotations and the translations required by one of the points set (protein $A$) which will produce a comparatively large superimpositions on the other set (protein $B$). The fundamental question to this problem is: How to find the best superposition of two protein structures? The problem of superimposing two structures of proteins is easy if we know the equivalent amino acids in both the structures. The hard part is to find this mapping of the corresponding atoms between two different protein structures.

Understanding a protein model clearly that will be used, is a vital step to design a pairwise alignment algorithm. Some of the previous alignment algorithms expressed the protein model as the central $\alpha$ carbon atom of every residue that are attached sequentially to form a polygonal chain in three dimensions [54]. In a more primitive model, protein is viewed as a collection of points (again the $\alpha$ carbon atoms) in 3-D space, where the alignment problem is allowed to view as that of matching two point sets [54]. However, it is crucial to supplant these different kinds of models with different features of the proteins like hydrophobicity, exposure to solvents, mutual affinities of amino acids, etc. to draw biologically meaningful conclusions from an alignment [54].

### 2.1.4 Similarity Measures

The root mean square deviation ($RMSD$) is widely used [37, 38] to measure the extent of structural similarity of two proteins. There are basically two different $RMSD$ [79, 62, 63, 81] measures that have been proposed in the background study:

1. coordinate root mean square deviation ($cRMSD$)

2. distance root mean square deviation ($dRMSD$)

For any two aligned structures of proteins $P$ and $Q$ of length $k$, these are defined as below [54].

$$dRMSD = \sqrt{\frac{2}{k^2 - k} \sum_{u=1}^{k-1} \sum_{v=u+1}^{k} (\|p_{i_u} - p_{i_v}\| - (\|q_{j_u} - q_{j_v}\|)^2} \tag{1}$$

$$cRMSD = \sqrt{\frac{1}{k} \sum_{u=1}^{k} \|p_{i_u} - t(q_{j_u})\|^2}. \tag{2}$$

[54] The similarity measures, $cRMSD$ and $dRMSD$, are both concerning absolute distances, therefore, the value of $RMSD$ gets poor even if there is any small presence of outliers irrespective of the fact that the two structures are globally similar to each other the $RMSD$ value will be poor. Many other researchers [81, 37, 70, 43] have observed such similar kind of behavior. Zhang and Skolnick [80] came up with a solution to overcome this problem, a sequence independent structural alignment measure (TM-score) which is a variation of a metric, originally defined by Levitt and Gerstein [41] [54]. Xu and Zhang have done a critical assessment of this TM-score. [76].

Given two proteins, a template protein $P$ and a target protein $Q$, $|P| \geq |Q|$, the structural similarity is obtained by a spatial superposition of $P$ and $Q$ that maximizes the following score [54]

$$\text{TM-score} = \frac{1}{|Q|} \sum_{i=1}^{k} \frac{1}{1 + (\frac{d_i}{d_0})^2}, \tag{3}$$

where $k$ is the number of aligned residues of $P$ and $Q$; $d_i$ is the distance between $i$-th pair of aligned residues and $d_0(= 1.24\sqrt[3]{|Q| - 15} - 1.8)$ is a normalization factor [54].

When the value of $d_0$ in equation (3) is set to $5A^o$, the resulting TM-score is known as a raw TM-score (rTM-score) [54].

Xu and Zhang [76] observed that two proteins are structurally similar and belong to the same fold when the TM-score $> 0.5$ [54].

## 2.2 Prior Work

The protein structure alignment problem is a vital issue in structural bioinformatics. All the protein alignment algorithms work mainly in three stages: finding correspondence between atoms, obtaining a rigid transformation in space for aligning them together and finally measure the similarity between the two aligned protein structure. The difference between all the protein structure alignment algorithm lies on the principle or the method chosen to find the correspondence between the atoms. Additionally, the similarity measure taken into account is different for different pairwise algorithm, which helps in refining the correspondence for a better similarity value.

### 2.2.1 DALI

DALI (Protein structure comparison by alignment of distance matrices [29]) is a distance matrix alignment method developed by Lisa Holm and Chris Sander. The DALI [29] method is based on the fact that similar 3-dimensional structures have similar intra-molecular distances. However, the main idea of this method revolves around the representation of each protein in a 2-dimensional matrix storing their intramolecular distances. It tries to overlap a matrix of one protein on top of another and gradually slide vertically and horizontally and stop when a common sub-matrix with

the best match is found between the two matrices [29].

The actual implementation can be broken into three steps:

1. The method decomposes each distance matrix into contact patterns as small sub-matrices which are of fixed sizes (hexapeptide-hexapeptide contact patterns)

2. Then it compares the contact patterns and tries to pair-up the sub-matrices, one from each protein, which is similar and storing the matched pairs in a pair list.

3. Finally, it assembles the similar sub-matrix pairs in the correct order to obtain the overall alignment.

### 2.2.2 TM-Align

TM-align: a protein structure alignment algorithm based on the TM-score [81] is a pairwise alignment method developed by Yang Zhang and Jeffrey Skolnick. The algorithm is nearly four times faster than Combinatorial Extension method and 20 times faster than DALI and SAL [81]. Regarding accuracy and coverage, the resulting structure is higher than any of the ones provided by other most often-used methods. This method uses the TM-score rotation matrix to increase the speed of the process by identifying the best structure alignments.

The method involves mainly of two steps:

1. Identifying initial structural alignments

2. Feed the obtained initial structural alignments into an iterative heuristic algorithm

The algorithm performs three different kinds of initial alignment :

1. The first alignment is obtained by aligning the secondary structures elements (SSEs) of two different proteins using dynamic programming (DP). The score matrix used for this alignment

is assigned to be 1 or 0 depending on whether or not the secondary structure elements of aligned residues are identical. The penalty values of 1 for gap-opening works the best [81].

2. The second alignment is based on the gapless matching of two protein structures. For the smaller of the two proteins, a gapless threading against the larger structure is performed, then the one with the best TM-score is selected [81].

3. The third initial alignment is obtained by Dynamic Programming using a gap-opening penalty of 1, but the score matrix used for the alignment is a half/half combination of the Secondary Structure score matrix [81].

Finally, the algorithm feeds the initial three structural alignments to a heuristic iterative algorithm, which is widely used in refining NP-hard structure-based alignments

### 2.2.3 EDAlign SSE

The EDAlign SSE (An eigendecomposition method for protein structure alignment [54]) is a pairwise alignment method developed by Satish Ch. Panigrahi and Asish Mukhopadhyay. This method is designed for both equal length and unequal length proteins. In this method, protein is considered as a polygonal chain of carbon residues in 3D. The solution of this method is depended on a matrix eigendecomposition method to the protein structure alignment problem [54]. Finally, this procedure reduces the protein structure alignment method to an approximate solution of a weighted graph matching problem. To measure the similarity between two aligned proteins it uses TM-score and cRMSD as the metrics [54].

For aligning equal length proteins, it refines the correspondence obtained from the matrix eigendecomposition approach and maximizes the similarity measure, TM-score, for the refined

correspondence. However, for the unequal length proteins, it works in three steps:

1. finds a correspondence between secondary structure elements (SSE-pairs);

2. finds a correspondence between residues within SSE-pairs;

3. applies a rigid transformation to obtain a structural alignment in space.

The final two steps are repeated until there is no further improvement found in the alignment.

### 2.2.4 Combinatorial Extension

Combinatorial Extension (Protein structure alignment by incremental combinatorial extension (CE) of the optimal path [63]) is an alignment algorithm developed by I N Shindyalov and P E Bourne.

Combinatorial Extension method work with aligned fragment pairs in every step of the method. At first, it breaks each structure in a series of fragments then tries to reassemble the fragments into a complete alignment. An alignment fragment pair is defined as a continuous sequence of protein A aligned against a continuous sequence of another protein B of same size [63]. If $n1$ and $n2$ are the lengths of protein $A$ and protein $B$, and AFP length is set to m, then there is a total of $(n1 - m)$ X $(n2 - m)$ AFPs. These series of pairwise combination of fragments which are called aligned fragment pairs (AFPs) defines a similarity matrix. This matrix helps in generating an optimal path to identify the final alignment. This optimal path increases linearly through the sequences, extending the alignment [63].

The algorithm steps are as follows:

1. Select some initial Aligned Fragment Pairs (AFP)

2. An alignment path is built gradually by incrementing AFPs in a way that satisfies a certain condition

3. Step (2) is repeated until each protein's length is traveled, or until no good AFPs remain

The Combinatorial Extension algorithm is a fast and accurate method that helps in finding an optimal structure alignment. It is suitable for analysis of large protein families

## 2.3 Proposed Approach and Details

### 2.3.1 Algorithm description

Our approach can be broken up into four major steps.

1. Creating protein datasets, each consisting of proteins from up to six different families of homologous proteins.

2. Constructing cRMSD distance matrices for each dataset, by running the selected pairwise alignment algorithms.

3. Using Principal Component Analysis [27] to embed the points that realize the distances in low-dimensional spaces

4. Applying a clustering algorithm to the embedded points, sans family labels, to test how well the alignment algorithms have succeeded in translating structural proximity to spatial proximity.

### 2.3.2 Creating Datasets

We have selected protein families which are completely different from one another structurally, implying divergence in functional behavior. The proteins that make up our datasets have been drawn from the following families.

- `Myosins`: a class of motor proteins that are crucial to muscle contraction and other motility processes in eukaryotes.
- `GTPases`: a large family of hydrolase enzymes that bind and hydrolase guanosine triphosphate (GTP). These help in regulating cell growth, cell differentiation, and cell migration [1].
- `Caspases` : a family of protease enzymes playing essential roles in programmed cell death and inflammation [12].
- `EF-Hand Proteins` : a large family of calcium-binding proteins, each with an EF-hand or alpha-loop-alpha motif [42].
- `Calmodulin` is a calcium transducer. It is a calcium-binding protein that can bind to and regulate the functions of different protein targets, thereby affecting many different cellular functions [46].
- `Phosphotransferase` : a class of enzymes that catalyze phosphorylation reactions [73]. Phosphorylation is crucial for protein function as it activates or deactivates nearly half of the enzymes. It is also a frequently-occurring post-translation modification in eukaryotic cells [73].
- `Cyclophilins`: a family of proteins found in vertebrates and other organisms that bind to cyclosporin A, an immunosuppressant commonly used to suppress rejection after an internal organ transplant [16].

The five datasets below were created from the above protein classes.

1. Two families, each consisting of 10 proteins

| Family No. | Family Name | Proteins (PDBs) |
|---|---|---|
| 1 | GTPases | 1YZQ, 1T91, 1YR9, 1XTS, 1MKY, 1PUJ, 2RCN, 2MSC, 2X2E, 1CEE |
| 2 | Myosins | 1W7J, 1W7I, 1B7T, 1OE9, 2AKA, 2MYS, 4P7H, 2EC6, 2OS8, 2OGT |

2. Three families, each consisting of 10 proteins

| Family No. | Family Name | Proteins (PDBs) |
|---|---|---|
| 1 | GTPases | 1YZQ, 1T91, 1YR9, 1XTS, 1MKY, 1PUJ, 2RCN, 2MSC, 2X2E, 1CEE |
| 2 | Myosins | 1W7J, 1W7I, 1B7T, 1OE9, 2AKA, 2MYS, 4P7H, 2EC6, 2OS8, 2OGT |
| 3 | Caspases | 1F1J, 1NW9, 1K86, 1QDU, 2C2O, 2H5I, 2NN3, 2CNK, 2CNL, 2CNN |

3. Four families, each consisting of 5 proteins

| Family No. | Family Name | Proteins (PDBs) |
|---|---|---|
| 1 | Caspases | 3DEI, 2C2O, 1K86, 2H5I, 1QDU |
| 2 | Myosins | 2AKA, 2EC6, 1OE9, 1B7T, 4P7H |
| 3 | ER Hand Proteins | 1XO5, 1IJ5, 2BE4, 2KQ6, 5BFX |
| 4 | Calmodulin | 1CLL, 1CFC, 3CLN, 1MXE, 1DMO |

4. Five families, each consisting of 6 proteins

| Family No. | Family Name | Proteins (PDBs) |
|---|---|---|
| 1 | Caspases | 3DEI, 2C2O, 1K86, 2H5I, 1QDU, 1RHK |
| 2 | Myosins | 2AKA, 2EC6, 1OE9, 1B7T, 4P7H, 1W7I |
| 3 | ER Hand Proteins | 1XO5, 1IJ5, 2BE4, 2KQ6, 5BFX, 1JUO |
| 4 | Calmodulin | 1CLL, 1CFC, 3CLN, 1MXE, 1DMO, 2F2P |
| 5 | Phosphotransferase | 2B0G, 1FYN, 2HWG, 1J7U, 1ND4, 4ORK |

5. Six families, each consisting of 7 proteins

| Family No. | Family Name | Proteins (PDBs) |
|---|---|---|
| 1 | Caspases | 3DEI, 2C2O, 1K86, 2H5I, 1QDU, 1RHK, 1V0D |
| 2 | Myosins | 2AKA, 2EC6, 1OE9, 1B7T, 4P7H, 1W7I, 1BR4 |
| 3 | ER Hand Proteins | 1XO5, 1IJ5, 2BE4, 2KQ6, 5BFX, 1JUO, 1JFK |
| 4 | Calmodulin | 1CLL, 1CFC, 3CLN, 1MXE, 1DMO, 2F2P, 2K0F |
| 5 | Phosphotransferase | 2B0G, 1FYN, 2HWG, 1J7U, 1ND4, 4ORK, 1PNJ |
| 6 | Cyclophilins | 1CWA, 1M9C, 1AK4, 2CPL, 2RMB, 2Z6W, 1NMK |

### 2.3.3 Running Pairwise Alignment Algorithms on the Datasets

For each dataset, we ran the four pairwise alignment algorithms, *DALI* [29], *TM-align* [81], *CE* [63] and *SSEAlign* [54], on each pair of proteins in the set to create a distance matrix. The distances are the *cRMSD* values of all the pairs. For example, from the first dataset, consisting of two families of ten proteins each, for a total of 20 proteins, we obtained a $20 \times 20$ symmetric distance matrix with 190 entries above the main diagonal.

### 2.3.4 Principal Component Analysis

Principal Component Analysis [27] is a dimensionality reduction technique that makes it possible to visualize data in low-dimensional spaces.

In this method, a new set of variables are obtained from the existing set of variables by a linear combination of the existing variables. The new set of variables are called Principal Components. The components are obtained in such a manner that the first one accumulates the maximum variation of the existing data. The succeeding component has the next highest variation and so on. In whatever reduced dimension the existing data is embedded, this process preserves the highest possible variance of the original set of data [57].

### 2.3.5 K-Means Clustering Algorithm

The $k$-means clustering algorithm partitions a set of $n$ data points in a $m$-dimensional Euclidean space into $k$-clusters. Each cluster consists of data points closest to the cluster center. The parameter $k$ is part of the input to the clustering algorithm.

The data points are obtained by applying Principal Component Analysis to the $cRMSD$ distance matrix. The returned set of points (representing proteins) lie in a low dimensional space (we have chosen two as the embedding dimension). The visualization of these points with their family labels shows a natural clustering that demonstrates how well an alignment algorithm translates structural proximity to spatial proximity.

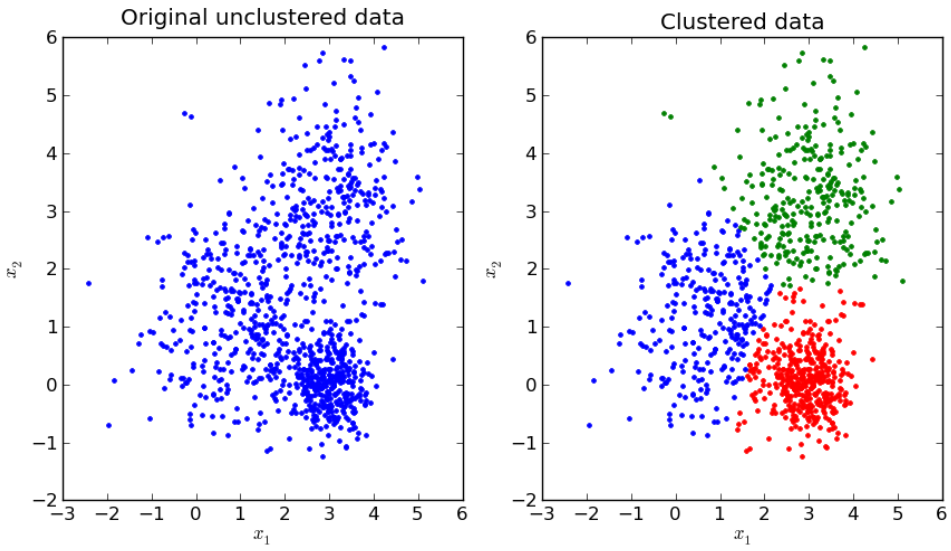



FIGURE 4: An example of how KMeans clustering look. [55]

When the same set of points, sans family labels, are subjected to the $k$-means clustering algorithm which uses only the spatial proximities of the points, we expect the clustering to remain largely unchanged with respect to the natural clustering. We discuss how well the four alignment algorithms have fared with respect to our expectations in the section on experimental results.

### 2.3.6 Proposed Algorithm for Evaluating Pairwise Alignment Algorithms

We call our algorithm $EPAA$, short for Evaluating Pairwise Alignment Algorithms. Below, we provide a formal description of $EPAA$ in pseudocode form.

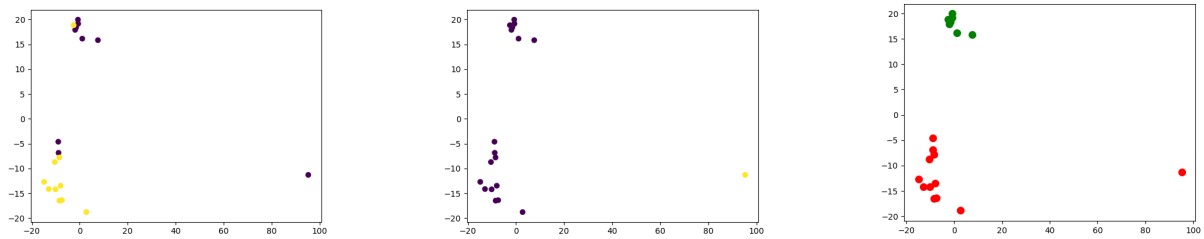
**Algorithm 1** *EPAA*

**for** Each alignment algorithm, $\mathcal{A}$ **do**
  **for** Each dataset, $\mathcal{DS}$ **do**
    Run $\mathcal{A}$ for every pair of proteins to build a $cRMSD$ distance matrix
    Input the distance matrix to the PCA algorithm to obtain a two dimensional embedding
    Run the $k$-means clustering/$k$-medians clustering method on the embedded point set
    Plot the points with original family labels
    Plot the clusters obtained from the $k$-means/$k$-medians algorithm, sans family labels
  **end for**
**end for**

## 2.4 Experimental Results and Discussion

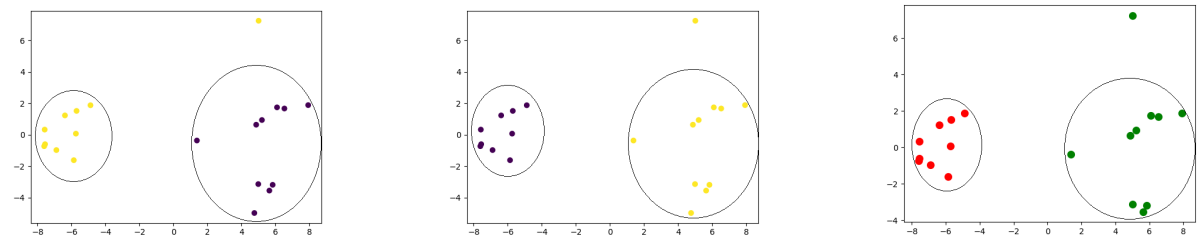
Experimental results obtained in 2-dimensional and 3-dimensional plots

1. Dataset 1- Two families, each consisting of 10 proteins



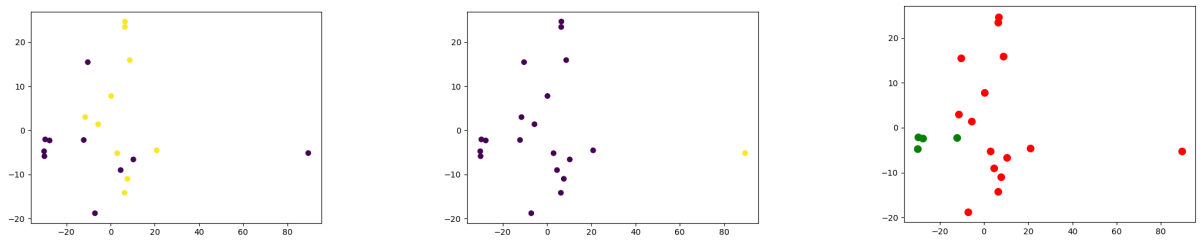
(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 5: Two family clusters formed by *DALI* in 2D.



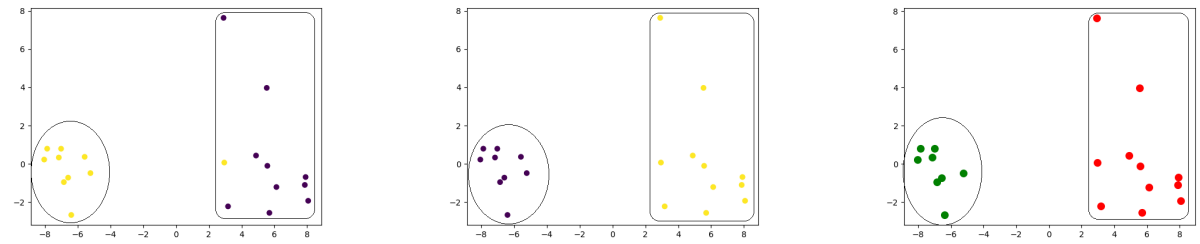
(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 6: Two family clusters formed by *TM-align* in 2D.



(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 7: Two family clusters formed by *EDAlignSSE* in 2D.



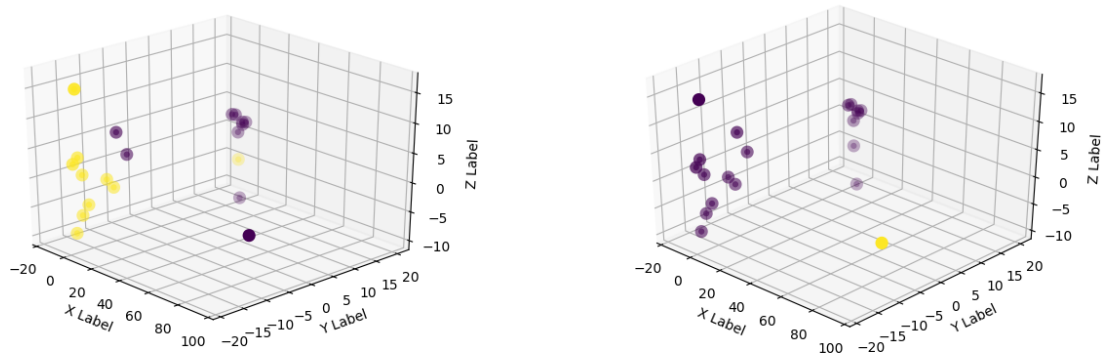
(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 8: Two family clusters formed by *CombinatorialExtension* in 2D.

Referring to Figs. 5-8 we find that with *TM-align* and *CE*, the structural clustering (Fig. 6a

and Fig. 8a) and the spatial clustering (Fig. 6b and Fig. 8b) are quite similar. The same cannot be said of *DALI* (Fig. 5) and *SSEAlign* (Fig. 7). In each of those cases, the proteins do not form two clearly distinguishable spatial clusters. For both *DALI* and *SSEAlign*, all the proteins except one got clustered in the same family. But, for *TM-align* and *CE* all proteins got clustered according to their family.

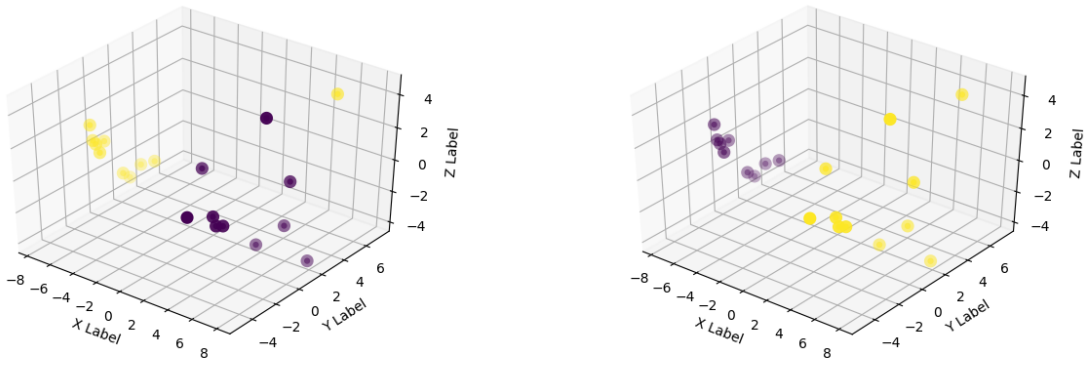
We have also used kmedians clustering to verify our result in terms of family mix. Referring to (Fig. 6c and Fig. 8c) of *TM-align* and *CE*, the spatially clustering using kmedians is similar to the the spatially clustering obtained by kmeans clustering whereas, for *DALI* and *SSE* (Fig. 5c and Fig. 7c), the spatial clustering are different than structural clustering.




(a) Known Family Labels

(b) Clusters with unknown family labels.

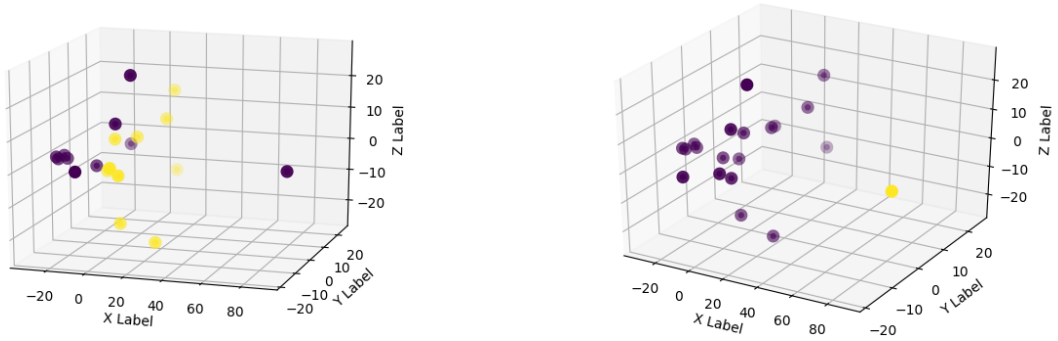
FIGURE 9: Two family clusters formed by *DALI* in 3D.




(a) Known Family Labels

(b) Clusters with unknown family labels.

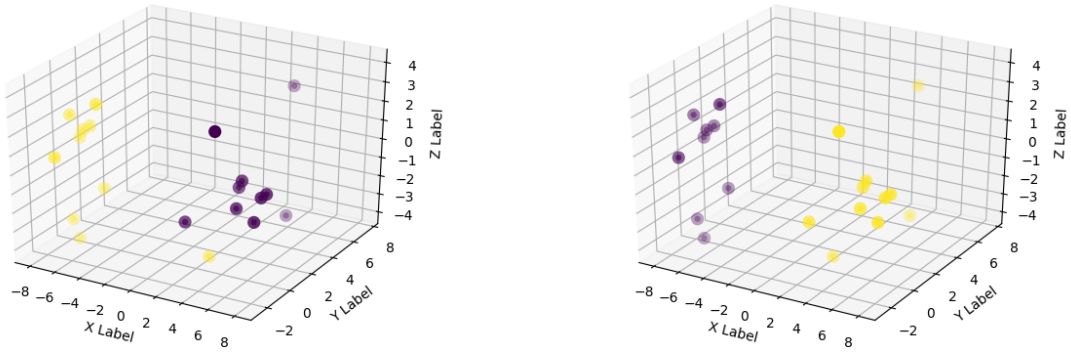
FIGURE 10: Two family clusters formed by *TM-align* in 3D.




(a) Known Family Labels

(b) Clusters with unknown family labels.

FIGURE 11: Two family clusters formed by EDAlign SSE in 3D.



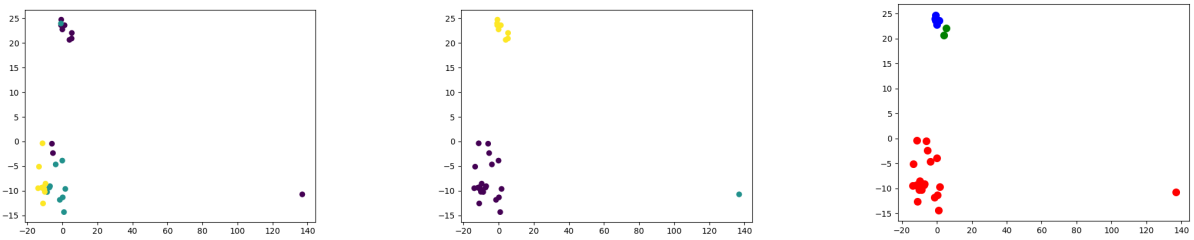

(a) Known Family Labels

(b) Clusters with unknown family labels.

FIGURE 12: Two family clusters formed by Combinatorial Extension in 3D.

Referring to Figs.9-12, we find that even in the higher dimensional plot, *DALI* (Fig. 9), *SSEAlign* (Fig. 11), *TM-align* (Fig. 10) and *CE* (Fig. 12) obtains the same clustering obtained in their 2-dimensional plot. This shows, there is no change in any family clustering when the dimension changes from 2D to 3D. The result is same for both the dimensional plot.

2. Dataset 2- Three families, each consisting of 10 proteins

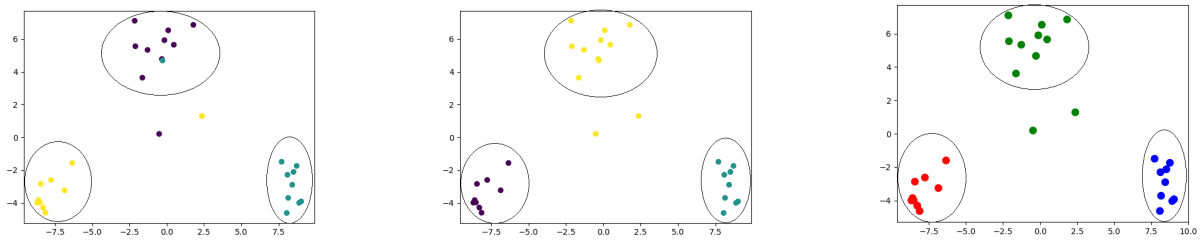


(a) Known Family Labels

(b) k-means clusters

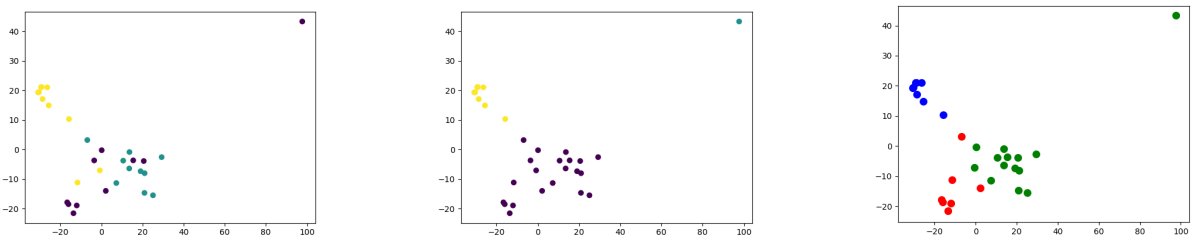
(c) k-medians clusters

FIGURE 13: Three family clusters formed by *DALI* in 2D.



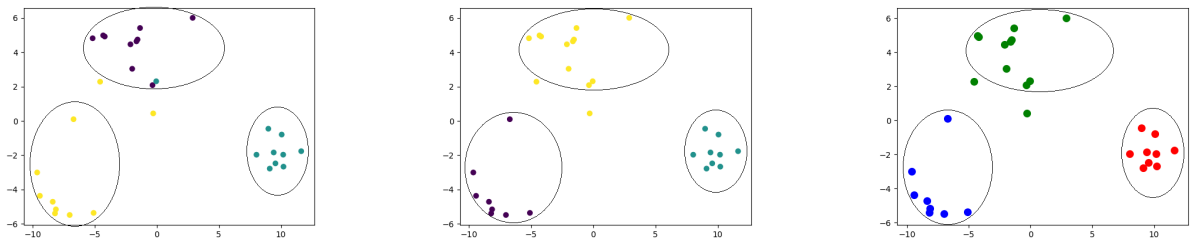
(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 14: Three family clusters formed by *TM-align* in 2D.



(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 15: Three family clusters formed by EDAlign SSE in 2D.



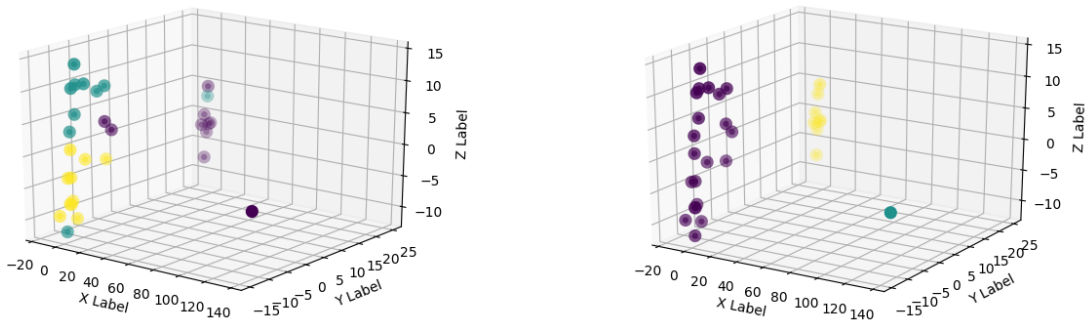
(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 16: Three family clusters formed by Combinatorial Extension in 2D.

Referring to Figs. 13-16, we find that with *DALI* (Fig. 13) and *SSEAlign* (Fig. 15), the

structural clustering (Fig. 13a and Fig. 15a, respectively) and the spatial clustering (Fig. 13b and Fig. 15b respectively) are very different. None of their clusterings resembles their actual family clustering. For both of them, two families out of three got mixed up in the spatial clustering. In the case of *TM-align* and *CE*, the structural clustering (Fig. 14a and Fig. 16a) and the spatial clustering (Fig. 14b and Fig. 16b) are the same. All the three families can be easily distinguished from the clusters formed. All the three families are well-clustered spatially according to their structural clustering.

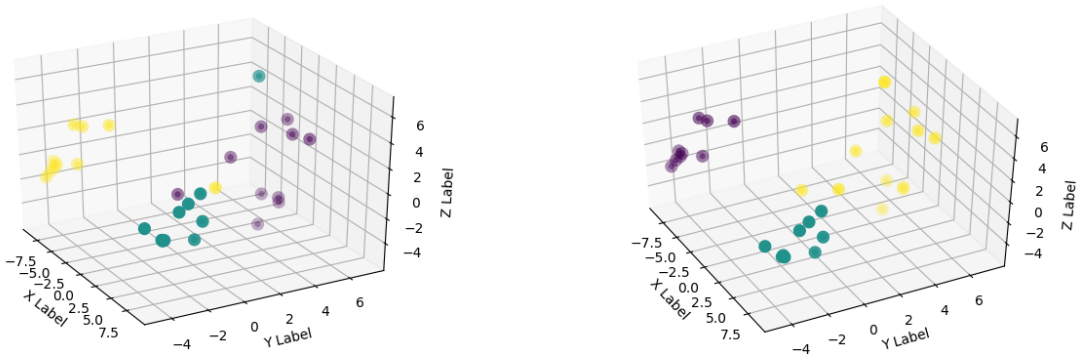
Referring to (Fig. 14c and Fig. 16c) of *TM-align* and *CE*, the spatially clustering is similar to the spatially clustering obtained by kmeans clustering. It is not the same for *DALI* and *SSE* (Fig. 15c and Fig. 13c), the spatial clustering obtained the kmedians clustering has also mixed all the family.




(a) Known Family Labels

(b) Clusters with unknown family labels.

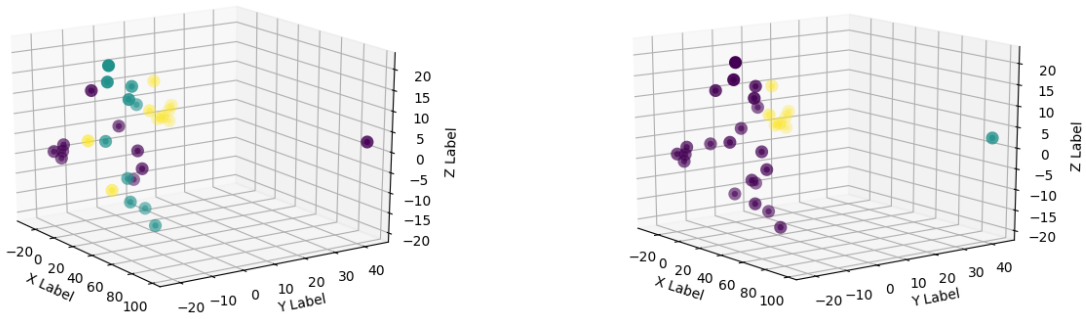
FIGURE 17: Three family clusters formed by *DALI* in 3D.




(a) Known Family Labels

(b) Clusters with unknown family labels.

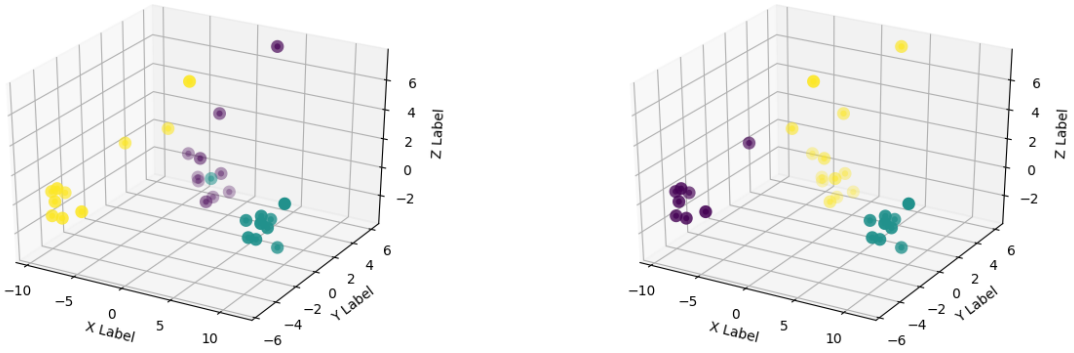
FIGURE 18: Three family clusters formed by *TM-align* in 3D.




(a) Known Family Labels

(b) Clusters with unknown family labels.

FIGURE 19: Three family clusters formed by EDAlign SSE in 3D.



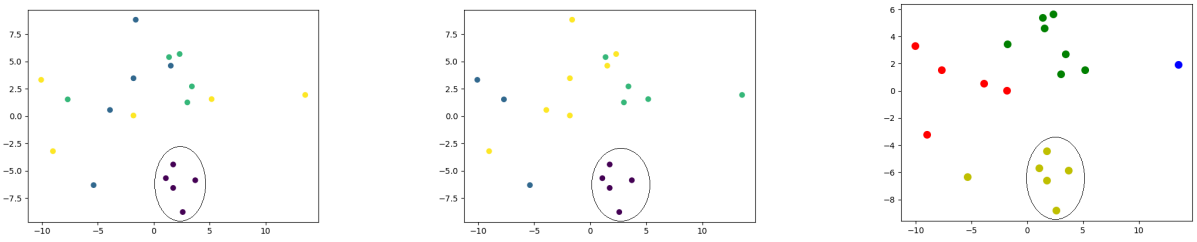

(a) Known Family Labels

(b) Clusters with unknown family labels.

FIGURE 20: Three family clusters formed by Combinatorial Extension in 3D.

Referring to Figs.17-20, we can say that even in this dataset the higher dimensional plot of, *DALI* (Fig. 17), *SSEAlign* (Fig. 19), *TM-align* (Fig. 18) and *CE* (Fig. 20) obtains the similar clustering achieved in 2D with this dataset, again proving the fact that the clustering is not affected with the dimension of plot.

3. Dataset 3- Four families, each consisting of 5 proteins

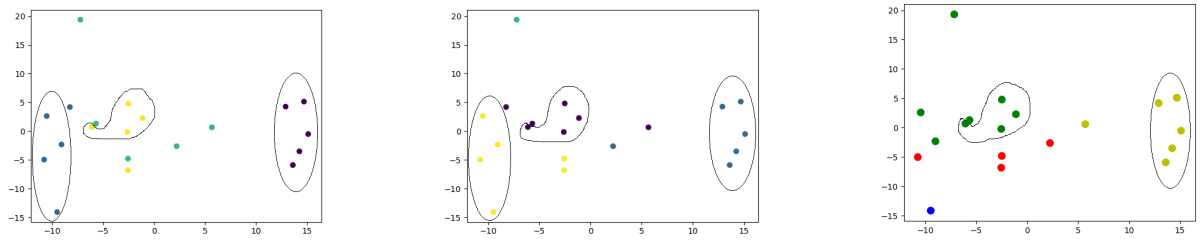


(a) Known Family Labels

(b) k-means clusters

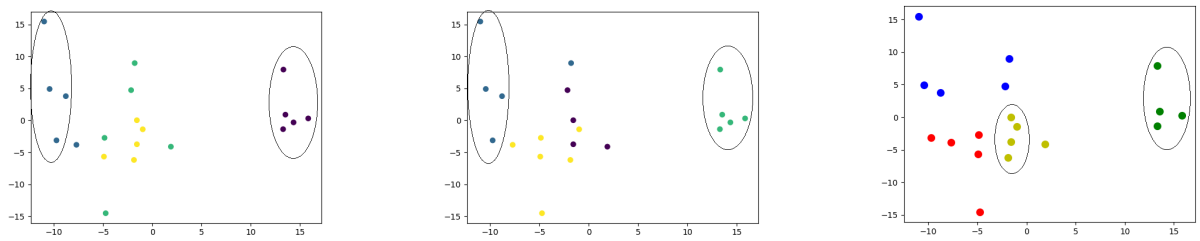
(c) k-medians clusters

FIGURE 21: Four family clusters formed by *DALI* in 2D.



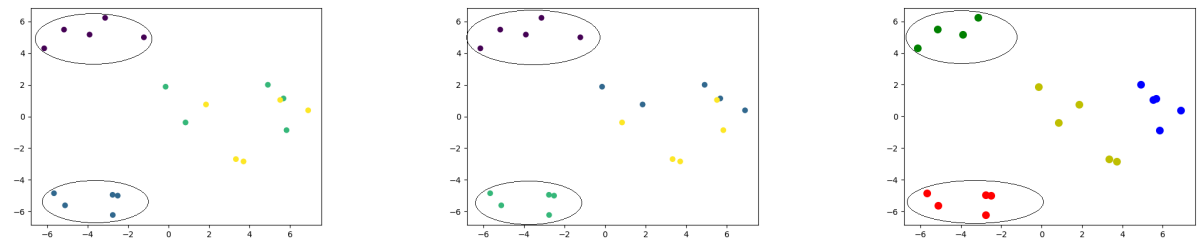
(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 22: Four family clusters formed by *TM-align* in 2D.



(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 23: Four family clusters formed by EDAlign SSE in 2D.



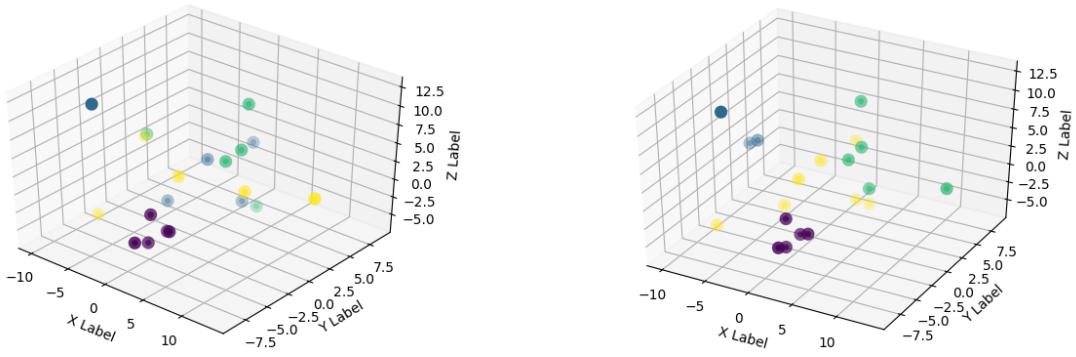
(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 24: Four family clusters formed by Combinatorial Extension in 2D.

Referring to Figs. 21-24, we find that with *TM-align*, the structural clustering (Fig. 22a) and

the spatial clustering (Fig. 22a) of the three families are very similar followed by $CE$ where two families out of 4 are well clustered. But in case of $DALI$ (Fig. 21), the spatial clustering of nearly all the families are dispersed. From Figs. 23a and 23b, we can say that the clustering in $SSEAlign$ is better than that of $DALI$ as it has formed proper spatial clustering in two families out of four. However, none is as good as $TM\text{-}align$.

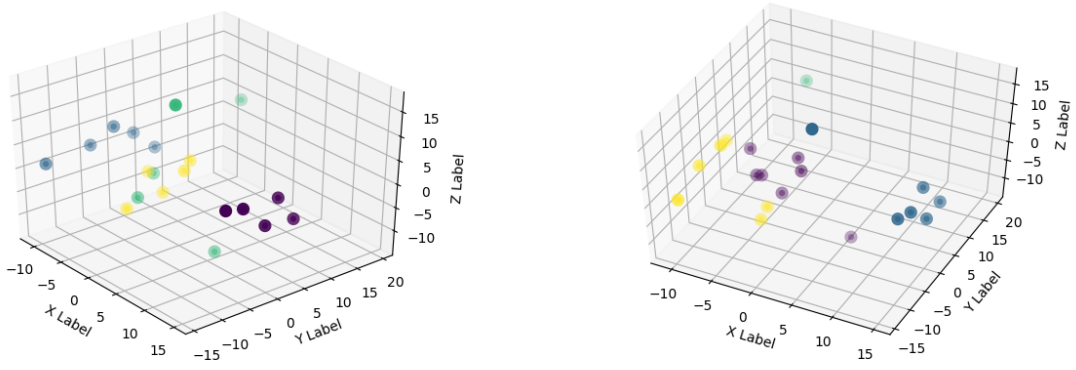
Referring to (Fig. 22c and Fig. 24c) of $TM\text{-}align$ and $CE$, the spatially clustering obtained by k-medians has successfully clustered two-three families correctly unlike $DALI$ and $SSE$ (Fig. 21c and Fig. 23c), they got wrongly clustered even using k-medians clustering.




(a) Known Family Labels

(b) Clusters with unknown family labels.

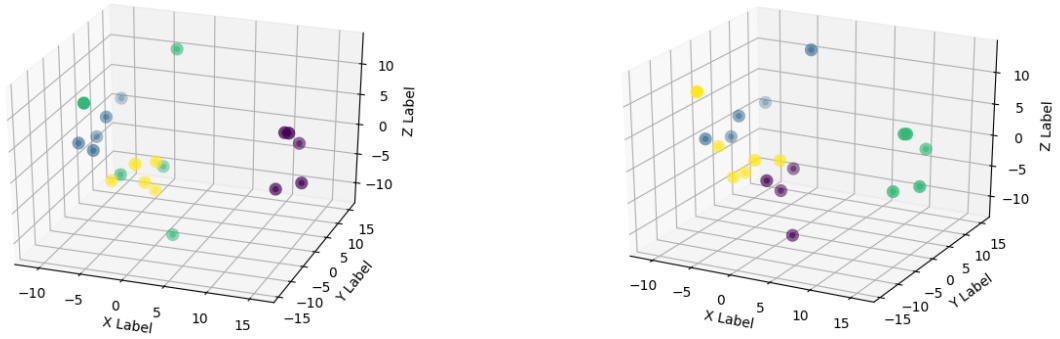
FIGURE 25: Four family clusters formed by $DALI$ in 3D.




(a) Known Family Labels

(b) Clusters with unknown family labels.

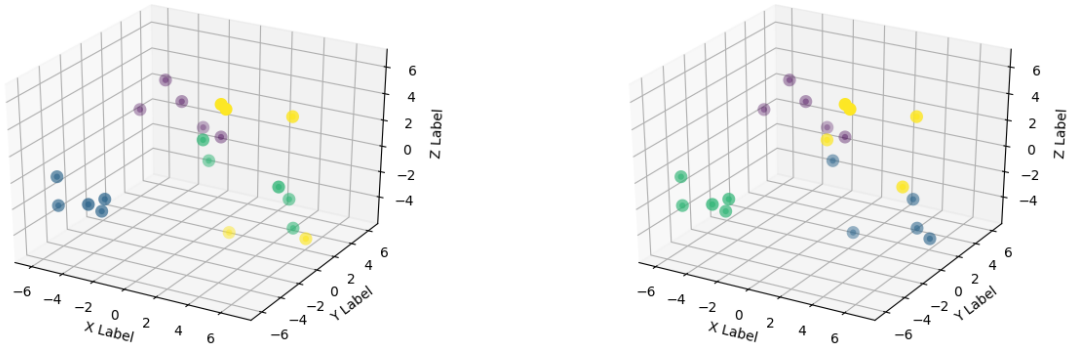
FIGURE 26: Four family clusters formed by *TM-align* in 3D.




(a) Known Family Labels

(b) Clusters with unknown family labels.

FIGURE 27: Four family clusters formed by EDAlign SSE in 3D.



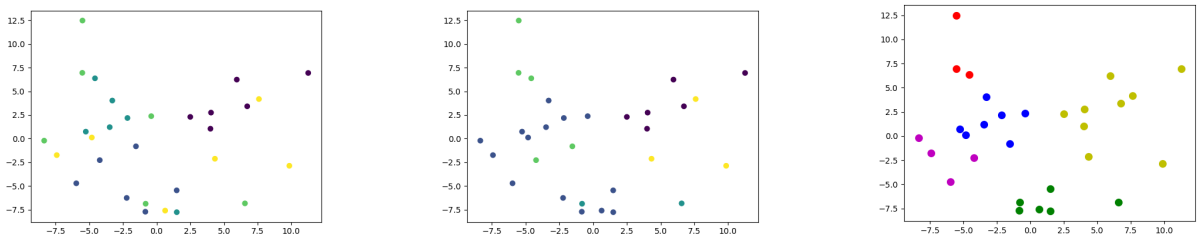

(a) Known Family Labels

(b) Clusters with unknown family labels.

FIGURE 28: Four family clusters formed by Combinatorial Extension in 3D.

Referring to Figs. 25-28, it again proves there is no change in any of the family clustering when the dimensions change from 2D to 3D. *DALI* (Fig. 25), *SSEAlign* (Fig. 27), *TM-align* (Fig. 26), and *CE* (Fig. 28), all of them, resemble the same clustering as in 2D.

4. Dataset 4- Five families, each consisting of 6 proteins

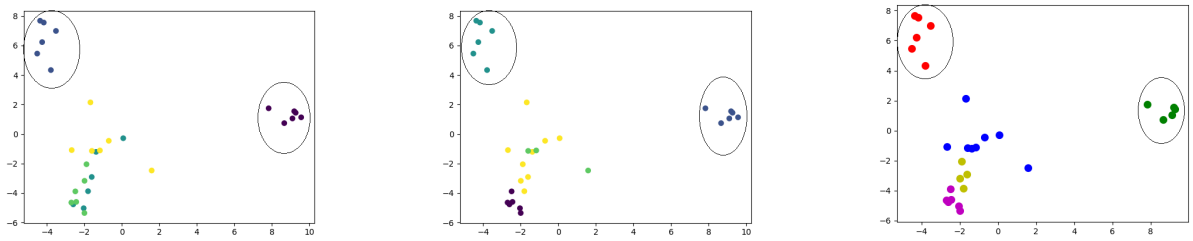


(a) Known Family Labels

(b) k-means clusters

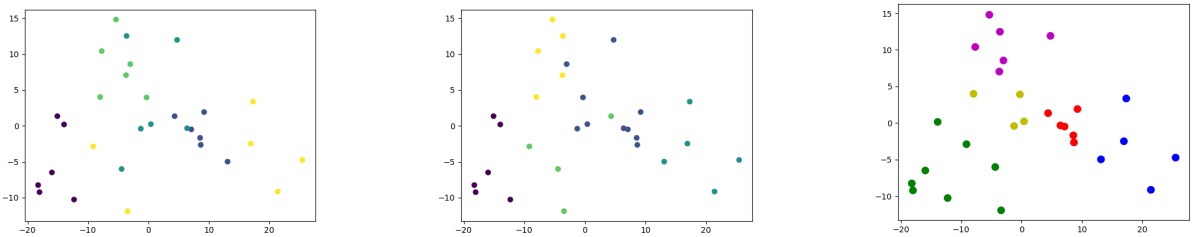
(c) k-medians clusters

FIGURE 29: Five family clusters formed by *DALI* in 2D.



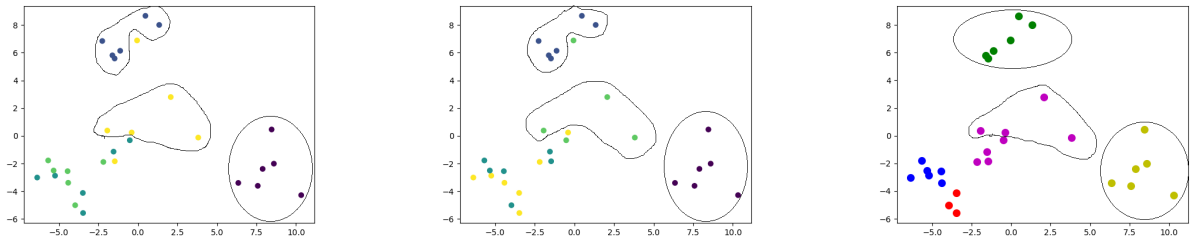
(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 30: Five family clusters formed by *TM-align* in 2D.



(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 31: Five family clusters formed by EDAlign SSE in 2D.



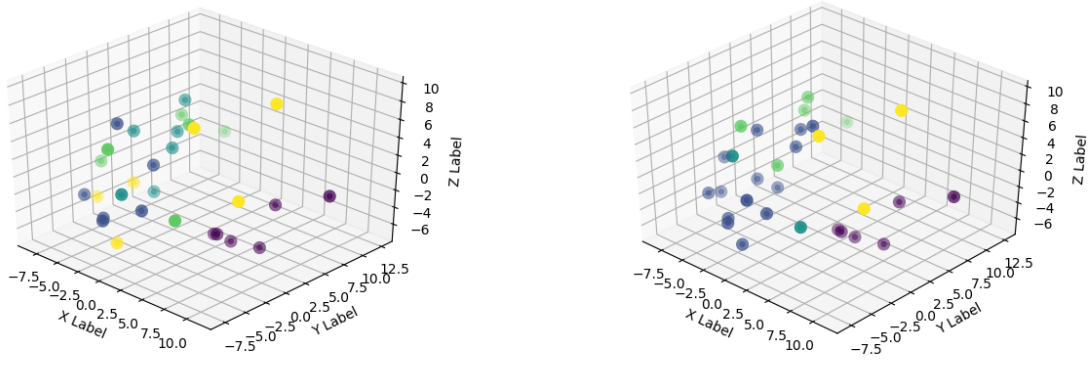
(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 32: Five family clusters formed by Combinatorial Extension in 2D.

Referring to Figs. 29-32, we again find that for *TM-align* and *CE*, the structural clustering

(Fig. 30a and Fig. 32a) and the spatial clustering (Fig. 30b and Fig. 32b) are similar for two to three families out of 5. Unlike *DALI* (Fig. 29) and *SSEAlign* (Fig. 31). We see that *SSEAlign* has got one family clustered properly, as compared to *DALI* which has failed for this large dataset. In this huge dataset, *TM-align* and *CE* behaved similarly.

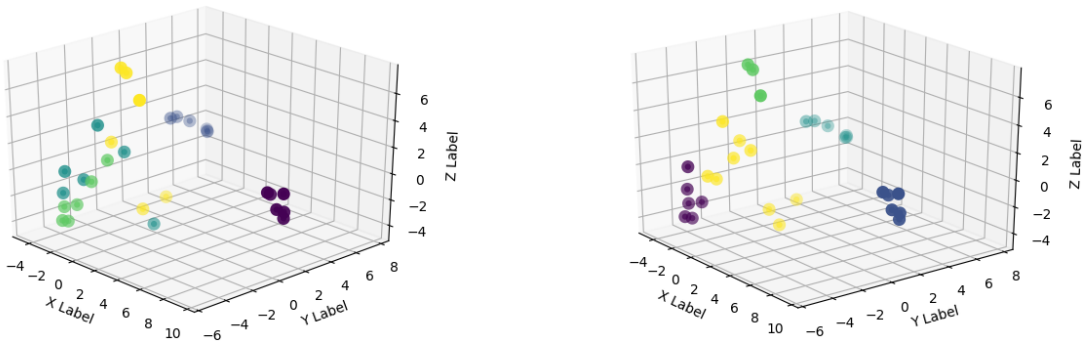
Similarly for this dataset, we again find that *TM-align* and *CE* (Fig. 30c and Fig. 32c), the spatially clustering obtained by k-medians is same as k-means clustering. But even using k-medians, *DALI* and *SSE* (Fig. 21c and Fig. 23c), have proved that they are unsuccessful in translating structural proximity to spatial proximity.




(a) Known Family Labels

(b) Clusters with unknown family labels.

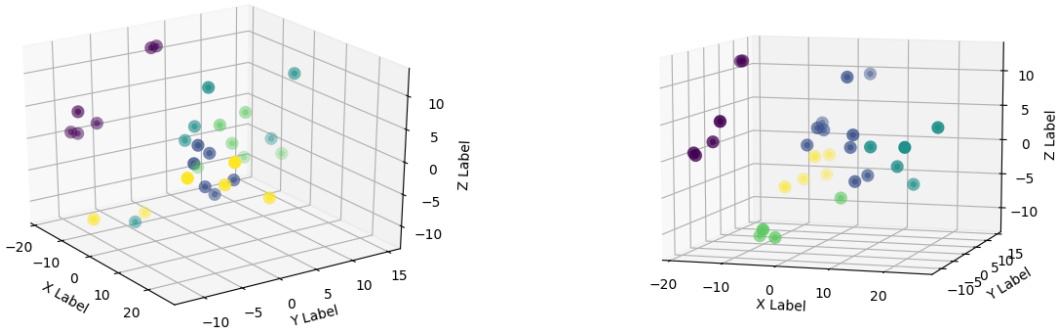
FIGURE 33: Five family clusters formed by *DALI* in 3D.




(a) Known Family Labels

(b) Clusters with unknown family labels.

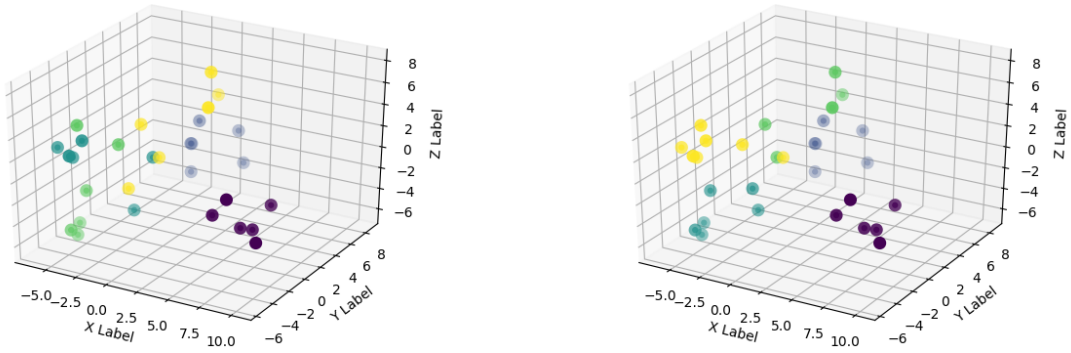
FIGURE 34: Five family clusters formed by *TM-align* in 3D.




(a) Known Family Labels

(b) Clusters with unknown family labels.

FIGURE 35: Five family clusters formed by EDAlign SSE in 3 D.



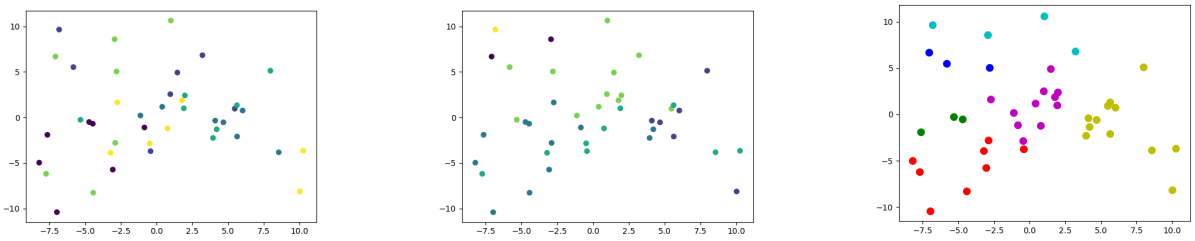

(a) Known Family Labels

(b) Clusters with unknown family labels.

FIGURE 36: Five family clusters formed by Combinatorial Extension in 3D.

Referring to Figs. 33-36, it shows even when the dataset gets higher along with the dimensions, the spatial clustering is unaffected.

5. Dataset 5- Six families, each consisting of 7 proteins

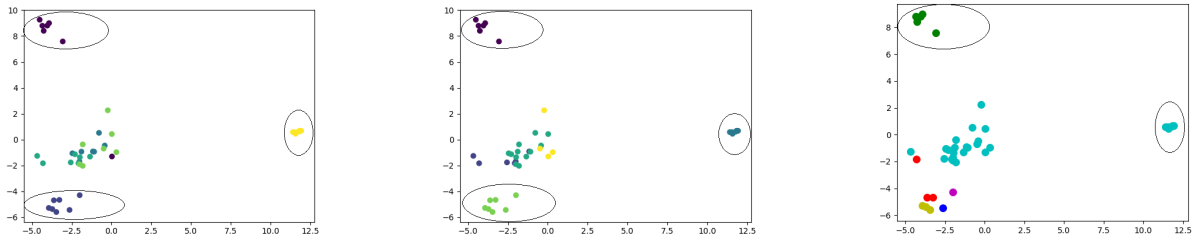


(a) Known Family Labels

(b) k-means clusters

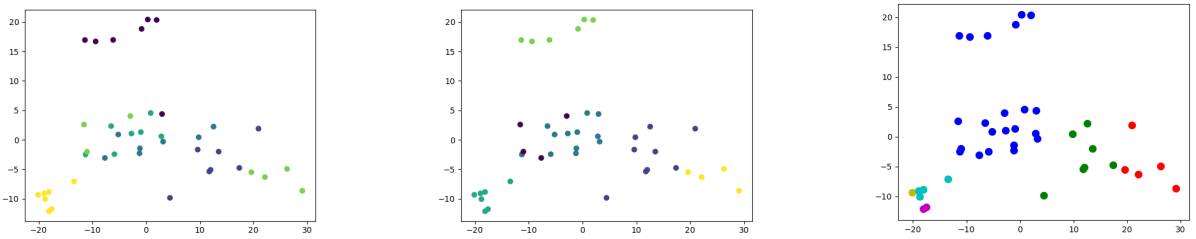
(c) k-medians clusters

FIGURE 37: Six family clusters formed by *DALI* in 2D.



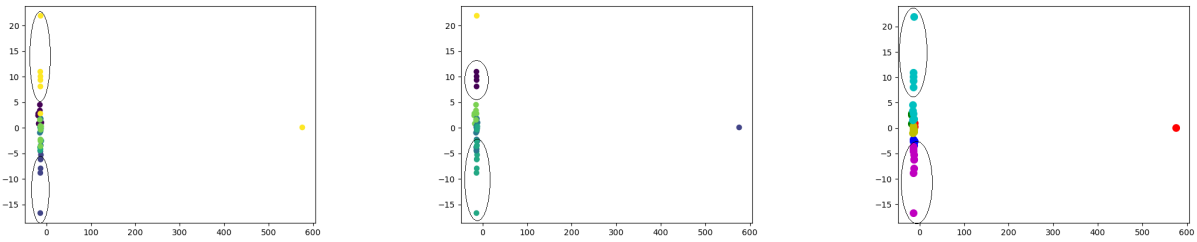
(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 38: Six family clusters formed by *TM-align* in 2D.



(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 39: Six family clusters formed by EDAlign SSE in 2D.



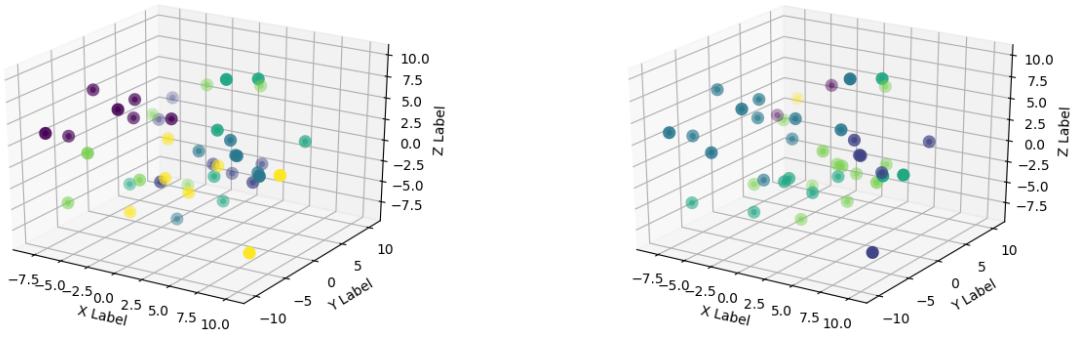
(a) Known Family Labels (b) k-means clusters (c) k-medians clusters

FIGURE 40: Six family clusters formed by Combinatorial Extension in 2D.

Referring to Figs. 37-40, we find that for this dataset *CE* and *SSEAlign* has performed better

than *TM-align*. The structural clustering (Fig. 39a) and the spatial clustering (Fig. 39b) of *SSEAlign* are similar for four families unlike *TM-align* which has similar clustering for three families. Most of the families are easily distinguishable. From Fig. 37a and Fig. 37b, we find that the structural clustering and the spatial clustering for *DALI* again proves to be the worst with the proteins from different families all dispersed among the different clusters.

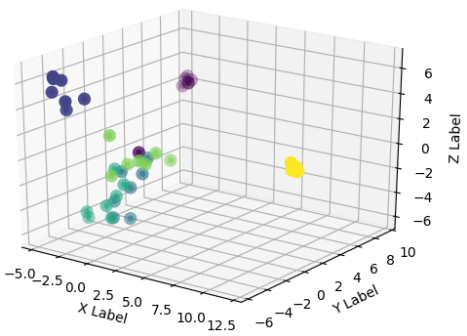
Finally, even in the last set using kmedians clustering, *TM-align* and *CE* (Fig. 38c and Fig. 40c), has successfully clustered according to the family. But, *DALI* and *SSE* (Fig. 37c and Fig. 39c), have proved mixed up the families again and proved that they are unsuccessful in translating structural proximity to spatial proximity.



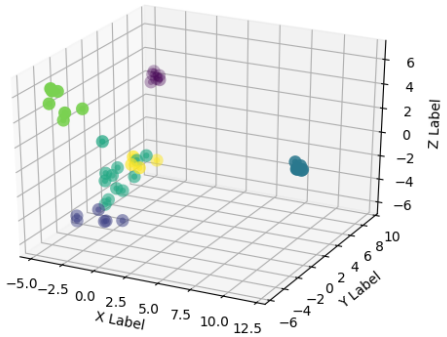

(a) Known Family Labels

(b) Clusters with unknown family labels.

FIGURE 41: Six family clusters formed by *DALI* in 3D.

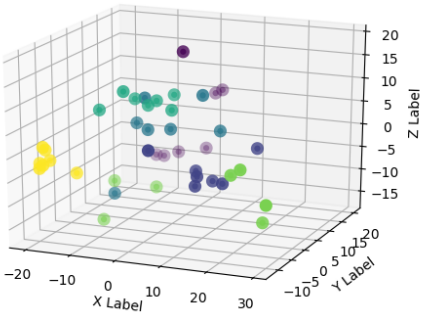



(a) Known Family Labels

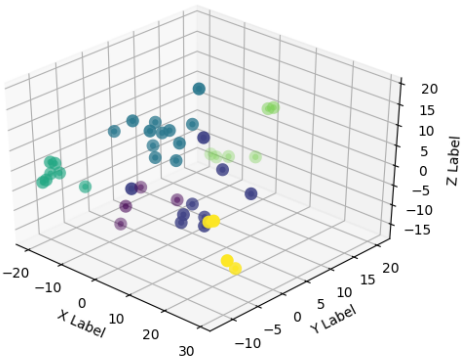



(b) Clusters with unknown family labels.

FIGURE 42: Six family clusters formed by *TM-align* in 3D.

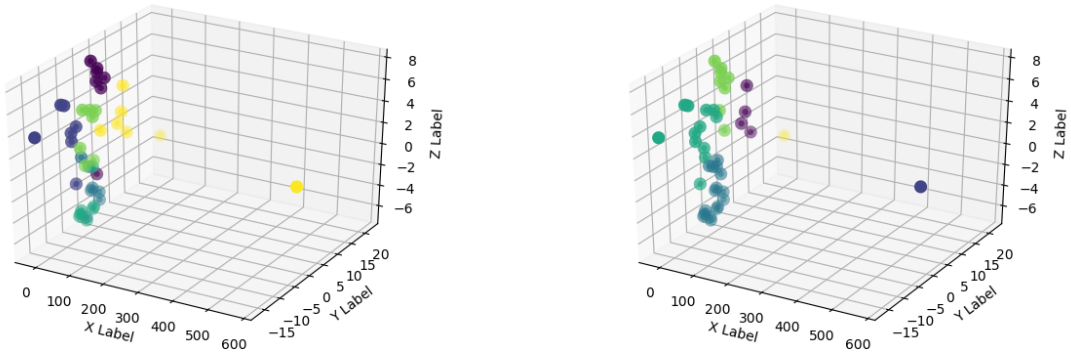



(a) Known Family Labels




(b) Clusters with unknown family labels.

FIGURE 43: Six family clusters formed by EDAlign SSE in 3 D.




(a) Known Family Labels

(b) Clusters with unknown family labels.

FIGURE 44: Six family clusters formed by Combinatorial Extension in 3D.

Referring to Figs.41-44, it resembles the same family cluster as it gets clusters in the 2D dimensional plot.

## 2.5 Conclusions

The above experiments showed that $TM-Align$ and $CE$ are able to cluster proteins from the same family spatially. Among all the four different algorithms taken into consideration, these two pairwise alignment algorithm has been successfully correlated structural proximity to spatial proximity. We have also confirmed our results by two ways:

1. Plotting in higher dimesnions to find if the cluster formation is same in both the dimension: We applyed the same method on a little higher dimensional data (3D), even on 3D dimensional plot, both $TM-Align$ and $CE$ has successfully clustered proteins from same family into the same group with minimum family mix. The clustering remained unchanged even when the

dimension had increased.

2. Used K-median clustering, which is known to be robust in nature: This clustering also obtained the similar family mix in $DALI$ and $SSE$ and proper clustering in $TM\text{-}align$ and $CE$, as obtained in k-means clustering. It has showed that the $TM\text{-}align$ and $CE$ have successfully translated structural proximity to spatial proximity but $DALI$ and $SSE$ has again got family mixed in the kmedians clustering

This work can be extended further by adding more pairwise alignment algorithms which are proposed in the past, to test and evaluate their ability to detect structural similarity among proteins. It can be used as a measure to find how good is the alignment algorithm. The paiwise alignment algorithm is considered to be more effective, if it has successfully been able to translate the structural proximity of the proteins to the spatial proximity.

# CHAPTER 3

# *Revised-MASCOT using Progressive Approach*

## 3.1 Introduction

The Multiple Protein Alignment problems have received a lot of attention from Structural Biologists because classification of protein structures and predicting their functions along with the other newly-sequenced proteins are done by the Multiple Structure Alignment Algorithm. It is also used for comparing the protein structures with the proteins of known functionalities. Additionally, the need for fast, robust and reliable MStA algorithms cannot be ignored because of the increased growth of the Protein Data Bank.

Based on the technique used, MStA algorithms can be classified roughly into one of four categories: MUSTANG [39], msTali [61], and CE-MC [22] imitates the progressive alignment approach used for multiple sequence alignment [59]. These algorithms using the progressive approach have succeeded in aligning multiple proteins way better than any other approaches. However, they fail in guaranteeing convergence to the global optimum [59]. There are other algorithms which are based on different approaches [47, 78] than progressive ones that have outperformed the progressive ones, both in speed and accuracy [59]. The second category of algorithms MATT [47], MultiProt [60],

Mass [14], Mapsci [32], and Smolign [64], which is taken into account when the goal is to find a structurally conserved subset of residues among the proteins to obtain knowledge about their evolution and origin. It refines the consensus structure, with repetitive iterations, and finds a core common to all the input proteins [59]. Unfortunately, such cores are pseudo-structures that, although geometrically interesting, are bereft of any biological significance. There is another method designed by Ye and Godzik which is graph-based POSA [78] which represents a protein as a directed acyclic graph (DAG) of residues, connected sequentially following the backbone of the protein structure. POSA creates a combined non-planar, multi-dimensional DAG, taking hinge rotation and coming up with residue equivalences among all the input proteins. Though POSA is known to incorporate the protein structures' flexibility, it completely misses the motifs on TIM-barrel and helix-bundle proteins [64], and incur a higher cost of alignments, for instance, MATT [47] or Smolign [64]. The pivot-based approach selects one of the input molecules, closest to all the other proteins as the pivot [59]. To obtain residue-residue correspondences, rest of proteins are aligned iteratively to the pivot either bottom-up manner [71] or top-down manner [77]. These correspondences are required to minimize some objective function and define a score as a similarity measure [59]. Mistral [48],[71] and [77] are some of the few published algorithms on this approach.

Our approach to this MStA problem is using the Progressive Alignment method used for multiple sequence alignment (MSA). We propose a new algorithm, **Revised MASCOT** (acronym for **Revised M**ultiple **A**lignment of **S**tructures using **C**enter **O**f pro**T**eins), which is very similar to the MASCOT [59] algorithm instead of using the center star method, we have taken the progressive method and tried aligning two or more protein structures together. Since we all know, SSEs are fundamental components of protein structures which serves as well-preserved scaffolds, we have taken this structural advantage of the protein polypeptide chain so that it contains all the

important information about the secondary structural elements (SSEs) [59]. Mutations do affect the loops which result in modifying the functionality, but the SSEs are evolutionarily remarkably conserved. For example, the substrate specificity of different serine proteases is governed by the confirmation of the binding loops [25]. Representing protein structures using their SSEs are also used on several pairwise alignment problems [36], [3], [4]; [21], [44].

In this paper, our goal is to design and develop a fast, elegant algorithm that uses the sequences of proteins using their SSEs to produce a structural alignment with high accuracy. Keeping this in mind, we have designed a version of MASCOT [59] called Revised MASCOT as a similar hybrid algorithm. It uses the most similar closely related proteins as a pair of sequences, and align those using global alignment. Then, we identify the next most closely related sequence pair; it can be a pair of sequences or a pair of alignments, and align them to each other. This concept of aligning two sequences or two alignments or a sequence with an alignment is obtained by finding the minimum sum of pairwise-distances between a pair of sequences.

Next, to align a pair of protein sequences, we find an optimal correspondence among the alpha carbon atom which is the backbone of the protein, using an inter-residue Euclidean distance threshold and compares the centerRMSD of the structures aligned in space as a measure of their similarity [59]. We have implemented Revised MASCOT and run experiments on the same set of proteins done by MASCOT. Again similar to [59], we too have included a comparison of the execution times of the Revised MASCOT with the well-known algorithm MUSTANG to show that it is really a fast algorithm [59].

### 3.1.1 Algorithm Aspects

#### What is an Alignment?

An alignment explains patterns of matching between some form of sequences, structures, and sequences with structures of proteins. An alignment gives a clear idea of the function and evolutionary distance of the proteins from a common ancestor.

Various methods like local geometry matching [75], comparison of distance matrices [28], maximal common subgraph detection [7], spectral matching [62], geometric hashing [51], contact map overlap [6, 13] and dynamic programming [81, 67] are used to obtain an initial equivalence set during alignment of two protein structures which is a 3-dimensional analogue of linear sequence alignment of peptide or nucleotide sequences [54]. Moreover, methods such as Monte Carlo algorithm or simulated annealing [28], dynamic programming [81, 67, 17, 18], incremental combinatorial extension of the optimal path [63] and genetic algorithm [66] can be used to further optimize these previously obtained equivalence set a goal to measure the amount of structural similarity for alignment of protein residue. This similarity measure is classified into four different categories (1) distance map similarity [28, 69, 5, 53] (2) root mean square deviation (RMSD) [62, 81, 63, 79] (3) contact map overlap [20] (4) universal similarity matrix [40, 56] are used to quantify the structural similarity; but even after all these classifications and research over the years, there is no universal definition of similarity score to measure structural similarity extent. A comprehensive list of different similarity measures is discussed by Hasegawa and Holm [24].

#### Global Alignment

In Global Alignment, the sequences are aligned along their entire lengths. This method of alignment is successful in finding the best alignment of a sequence with another sequence. The Global

Alignment technique is done using the Needleman-Wunsch algorithm, which is based on dynamic programming. The alignment received from this technique illustrates a lot about the protein and its sequences which as a result leads to the understanding of its functions and structures.

Let $A_m$, ... , $A_n$ and $B_m$, ... , $B_n$ be the two different protein sequences of length $m$ and $n$ respectively which we need to align optimally. It contains input alphabets consisting of symbols to represents different amino acids in the protein structure [59]. So, we define an alignment in a much formal way. To perform a global alignment of $A$ and $B$, we need to introduce gaps $(-)$ at the beginning or ending or between a pair of symbols, such that the alignment which is obtained must have the following properties.

1. It should be a 2 X L matrix where max(m,n) <= L <= m+n
2. First row has either a blank or a character from A and second row has either a blank or a character from B.
3. No column can have blanks in both the sequences.

Needleman-Wunsch Algorithm is a global alignment method used to find the optimal edit distance between two string [50] or protein sequences. The NeedlemanWunsch algorithm performs a global alignment on two sequences. It is an example of dynamic programming and was the first application of dynamic programming algorithm to biological sequence comparison. It is applied to optimization problems..

**Structure Alignment**

In protein structure alignment, there is a comparison taken place between the positions of atoms in two or more 3-D protein structures. The biological functions and properties of a protein are

determined by its 3D structure [59]. Holms and Sander stated "Comparing protein shapes rather than protein sequences are like using a bigger telescope that looks father into the universe, and thus farther back in time, opening the door to detect the most remote and most fascinating evolutionary relations" [31]. The statement means aligning the 3D structure of proteins gives more insight into their functions and evolutionary origins [31]. For a structural alignment to take place optimally, we must first find out which residues correspondences, it is only possible if we are unaware of the proteins 3D structure. There is a possibility when a residue cannot be mapped to any other residue, in that case, it will be aligned with a gap '-'. Therefore, in a set of $N$ proteins, one-to-one mapping of the protein residues gives rise to a protein structure alignment.

For two proteins of length $N_B$ and $N_A$, the number of possible alignment grows exponentially,

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

The number of possible alignment increases, when the number of residues in both the proteins is more. The problem is a pairwise alignment one, when $N = 2$, for which there are many widely accepted algorithms such as, Dali [28], SSAP [53], Eigenvalue Decomposition [54], Combinatorial Extension [63] etc. When $N > 2$, the problem is termed as Multiple Structure Alignment (MStA). Our problem belongs to this category of problem [59].

### 3.1.2 Problem Statement

Consider P = $P_1$, $P_2$ ... $P_N$ a set of $N$ protein structures and $L_1$, $L_2$ $L_N$ the number of residues in each proteins respectively. All the protein structures are represented by the coordinates of their alpha carbon ($C_\alpha$) atoms. $P_{ab}$ denotes the $b_{th}$ residue of the $a_{th}$ structure, for a = 1 ... N and b = 1 ... $L_N$.

The multiple structural alignment of P is Y = $(Y_{ab})$ , 1 ≤ a ≤ N, 1 ≤ b ≤ L , such that:

- Max ($L_1$, $L_2$ ... $L_N$) ≤ $L$ ≤ ($L_1$ + $L_2$ + ... + $L_N$).
- Each element in X either has to be one of the residues of $P_{ij}$ or a gap which is a special null residue, denoted by the symbol '-'.
- The $a^{th}$ row in Y contains the set of C positions of a protein structure $i$, it might contain gaps in between. This shows that the alignment makes sure that it preserves the order of residues.

Now that the matrix of equivalences Y are obtained, a set of rigid body transformations needs to be done, each having a proper rotation matrix $Rot_a$, where det($Rot_a$)= +1, which can be denoted as, TR( $Rot_a$ , $Trans_a$ ) 1 ≤ i ≤ N, where $Trans_a$ is a translation tuple which is acted upon each protein in Y, which will drive an optimal superposition of all the structures.

A superposition of minimum coordinate root mean square deviation (RMSD) is obtained with the help of an input set of reference 3D points of the equivalent residues.

$$RMSD = \sqrt{\frac{1}{n}\sum_{N}^{i=1}((v_{i_x} - w_{i_x})^2 + (v_{i_y} - w_{i_y})^2 + (v_{i_z} - w_{i_z})^2)}$$

where, ($v_{ix}$; $v_{iy}$; $v_{iz}$) represents 3D coordinates of residue $i$ which are obtained by superimposing structure $a$ on structure $b$. The distances between the corresponding residues $v_i$ and wi represents the Euclidean distance. $n$ denotes the number of aligned. Basically, our goal is to minimize root mean square deviation (RMSD) while maximizing the number of aligned residues.

## 3.2 Prior Work

We have studied a couple of papers, which are relevant to our work.

### 3.2.1 A center star approach - MASCOT

The center-star approach was first proposed by Gusfield [23]. This approach falls under the class of approximation algorithms as it aims to give a solution quality and a run-time bound which can be proved [59]. Gusfield stated the solution of Multiple Structure Alignment could be optimal up to a constant factor 2 with the sum-of-pairs metric [23].

The following properties must be satisfied by the cost function of sequences a, b, and c:

Cost[a, a] = 0 (reflexive)

Cost[a, b] = Cost[b, a] $\geq$ 0 (symmetric)

Cost[a, b] + Cost[b, c] $\geq$ Cost[x, z] (triangle inequality)

The center-star algorithm follows the following steps:

1. By minimizing the Sum-of-Pairs metric, we need to find the center protein sequence $Sc$ such that
$$\sum_{j=1}^{N} EditDistance(P_i, P_j)$$
is minimum.

2. Next, align all the rest $N-1$ protein sequences, $Si$, where i = 1 ... N, and i $\neq$ c to Sc, following once a gap always a gap policy [59].

For the first step, they perform a global alignment on every pair of protein sequences using Needleman-Wunsch algorithm [50] which requires an affine gap opening penalty, and a scoring matrix.

### 3.2.2 A progressive approach - MUSTANG

The progressive alignment algorithm falls under the class of heuristic algorithms as it sometimes comes out with an optimal solution and sometimes with not-so-good results. There are many algorithms that used this technique to solve the problem of multiple structure alignment. This method starts the process of multiple alignments by beginning with the most similar pair of protein structures and iteratively adding more and more distant structures to the alignment [59]. To do this process, it uses a guide tree. They use different algorithms such as neighbor-joining method [15] or Unweighted Pair Group Method with Arithmetic Mean (UPGMA) to build the guide tree.

Under this category of approach, Mustang [39] is one of the best paper worth reviewing. It is proposed by Konagurthu et al., in 2006. Mustang starts by first identifying the most similar fragment pairs, aligning them and then extending this pair to the find more distant pair and finally covering the entire protein length. It applies an all-pair-all-fragments scoring in the input proteins, finally obtaining a guide tree [59]. Using the guide tree, it progressively aligns multiple proteins and obtains the set of correspondence for each protein, required for the rigid body superposition [59].

This algorithm along with others also have got flaws, which cannot be ignored. For example, its dependencies on the initial alignments, once an error as occurred in the alignment process it gets propagated throughout the alignment and effects the final alignment result, it takes in large running time, and since the method is heuristic it gives relatively low accuracy result [59].

## 3.3 Proposed Approach and Details

### 3.3.1 Protein Data Bank

The crystallographic database for three-dimensional structure data of large biological molecules [72], first conceived at Brookhaven National Laboratories is known as Protein Data Bank (PDB) [59]. PDB [10], which is overseen by the organization called Worldwide Protein Data Bank (wwPDB), get its data by submission through biologists and biochemist using technologies such as X-ray crystallography and Nuclear Magnetic resonance. Due to the exponential growth of the data as well as the arrival of the internet, PDB was easily available through the website of its member organizations [59].

PDB repository contains all the known protein structures in PDB format since it is the key resource in areas of structural biology and mandatory for scientists to submit their structure data to PDB [59]. Every atom taking up a single line in the PDB file has a structure which shows its type, carbon atom position, residue number and type of residue. Below representation shows an entry in a PDB file [59].

ATOM 83 CA ALA A 8 24.850 0.000 17.421

The above line indicates that there is a carbon atom at the position (24.850, 0.000, 17.421). Moreover, the CA shows that it is the central C atom of a residue, namely residue 8 of type ALA from chain A [59]. The value 83 is a unique atom identifier within the file [59]. In short, a PDB file is a digitized version of the actual protein chemical.

### 3.3.2 Input data set

Protein structures are saved as PDB files in the Protein Data Bank [8]. Since the protein structures are growing rapidly in number, it contains more than 100000 structures. In this algorithm, the first step is to feed the algorithm with a proper and correct data, which can be retrieved from a standard protein database or from the local computer itself [59]. The vital step in the algorithm is to preprocess the data before proceeding with it in the algorithm since we have to align thousands of proteins [59]. This is an important step because, in the process of aligning thousands of proteins, the required proteins needs to be fetched and readied correctly by the next step in the algorithm. For example, an input set could be 1AOR: A 6ACM 2ACT 1TRQ: B, etc.

### 3.3.3 Representing the proteins

After taking the correct input set, we represent each protein in a manner that will make the subsequent processing easier and convenient as well as retain all the necessary information that is needed by the algorithm [59]. To obtain a robust and fast multiple alignment algorithms, the observations are dependent on this particular step.

We depict the proteins by motif sequences which are obtained by the DSSP-program [35]; this is done to keep a balance between complexity and functionality. The DSSP-program Assigns each residue of a protein to one of the eight possible structural motifs (see Table 1). As a result, a linear motif sequence is obtained as a backbone of the protein structure. For example, for SEA CUCUMBER CAUDINA (1HLM), it will show as follows,

...HHHHGGGZZIIIITTHHHHHHTTSSI...

This step converts the problem from structure alignment to a sequence alignment problem. So, we are now able to use an algorithm for sequence similarity in this problem to obtain an optimal

| *Symbol* | *Motif* |
|---|---|
| H | Alpha Helix |
| B | Beta bridge |
| G | Helix 3 |
| E | Beta strand |
| T | Turn |
| S | Bend |
| I | Helix 5 |
| Z | No motif |

TABLE 1: The DSSP code [59]

| – | H | B | G | E | T | S | I | Z |
|---|---|---|---|---|---|---|---|---|
| H | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| B | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| G | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| E | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| T | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| I | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| Z | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

TABLE 2: Example of a Scoring Matrix [59]

alignment of any two or more sequences. The next step in the algorithm is built on this observation.

### 3.3.4 Pairwise global alignment

This is the next step in the algorithm. We perform a pairwise global alignment on the different $N$ DSSP sequences obtained in the previous step, corresponding to the $N$ input proteins [59]. For the pairwise global alignment, we use the Needleman-Wunsch algorithm. The scoring matrix used was of Table 2 along with an appropriate gap opening penalty.

These pairwise alignments produce a rudimentary picture of SSE-SSE alignments. This is apparent from the pairwise alignments of the DSSP sequences of the globins 1DM1, 1MBC, 1MBA, shown below [59].

```
1DM1   ..ZZZHHHHHHHHHHHHHHHHHTHHHHHHHHHHHHHHHSGGG-...

1MBA   ..ZZZHHHHHHHHHHHHHHHHHT-HHHHHHHHHHHHHHHHZGGG...


1DM1   ..ZZZHHHHHHHHHHHHHHHHHTHHHHHHHHHHHHHHHH-SGGG...

1MBC   ..ZZZHHHHHHHHHHHHHHHHGGGHHHHHHHHHHHHHHHHZTHHH...


1MBC   ..ZZZHHHHHHHHHHHHHHHHGGGHHHHHHHHHHHHHHHHZTHH-H...

1MBA   ..ZZZHHHHHHHHHHHHHHHHHTHHHHHHHHHHHHHHHHZGGGGG...
```

We can see that the helices are properly aligned against one another. These alignments are saved in a list and referred to when needed.

### 3.3.5 Tree-based Progressive Alignment

The main idea is to align large groups of protein sequences by using a sequence of pairwise alignments; these alignments take place between closely related proteins. The alignment occurs from the tip and proceeds towards the root of the tree [19]. It follows the branching order of the guide tree. In progressive alignment method, we have developed an $N \times N$ symmetric distance matrix for multiple sequence alignment, an $N \times N$ (see Table 3) whose entries are calculated taking Kmer distance as the metric (values shown in the table have been arbitrarily assigned) between the various multiple sequences in the list mentioned above.

From this distance matrix, we find the least paiwise alignment scores between the sequences which are closely related to each other and form a cluster of those sequences and gradually the next

| $-$ | $P_1$ | $P_2$ | $\ldots$ | $P_N$ |
|---|---|---|---|---|
| $P_1$ | 0 | 10 | $\ldots$ | 20 |
| $P_2$ | 10 | 0 | $\ldots$ | 30 |
| $\vdots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\vdots$ |
| $P_N$ | 20 | 30 | $\cdots$ | 0 |

TABLE 3: A sample distance matrix [59]

alignment with the next related pair of sequences or a related pair of sequences and an alignment it will give us a guide tree. The leaves of the guide tree formed are considered to be the sequences and the node between two sequences will give us the alignment between the sequences. We call this tree as a guide tree because it will be guiding us through the process of multiple sequence alignment. The internal nodes of the tree are considered to be the alignment between a pair of sequences or a pair of sequence and an alignment or a pair of alignments. The alignment at a given internal node contains all of the sequences in the clade defined by that node. The alignment obtained at the root node is our multiple sequence alignment.

### 3.3.6 Guide Tree

In progressive alignment, the guide tree plays a vital role as it determines which sequences to be considered for the next step of alignment. A guide tree is a phylogenetic tree that is constructed dependent on the distance matrix obtained from the sequences. This phylogenetic tree shows the distance between the pairs of sequences when the edges of the tree are weighted. In this work, we will be using a rooted phylogenetic tree as our guide tree. Guide trees are built using clustering algorithms. In our work, the guide tree is developed using UPGMA (Unweighted Pair-Group

Method with Arithmetic mean) method [26]. We use to cluster the sequences with UPGMA and print out a dendrogram.

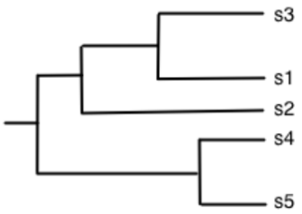



FIGURE 45: A guide tree

For example, the above guide tree will be shown as:

$$(((s1 : 0.23,\ s2 : 0.23) : 0.77,\ s2 : 0.87) : 1.76,(s4 : 0.99\ ,s5 : 1.00) : 1.52)$$

### 3.3.7 How did we build a guide Tree

We used UPGMA [26] clustering algorithm for developing our guide tree. Building a UPGMA tree requires only a distance matrix which is obtained using a pairwise distance between the protein sequences. Once we have the distance matrix, we will be able to cluster the sequences based on their similarities/dissimilarities.

For UPGMA method [26], we first find the pairwise distances between all the sequences. Then, we find the shorter distance of the sequences with the smallest value in the distance matrix. For example, the shorter distance between two sequences is $S_1$ and $S_2$. Now, $S_1$ and $S_2$ form a cluster named $C_1$ eliminating $S_1$ and $S_2$ from the distance matrix. Similarly, we find the next smallest distance and its corresponding sequences or clusters formed by previous sequences and form the new cluster. We repeat this process of finding new clusters from the existing sequences, merging and updating the distance matrix, until we have found a single cluster at the end. Once we have an alignment between two protein sequences, we find a sequence or any other alignment which has

a score similiar to that and merge that alignment to the previous alignment and this propagates through the tree from the leaves to the root.

### 3.3.8 Correspondence matrix

Formally, a correspondence matrix is an $N \times l$ matrix with respect to a new protein $P_o$ in the set of input proteins, $P = \{P_1, P_2, \ldots, P_N\}$ such that

$$max(|P_1|, |P_2|, \ldots, |P_N|) \leq l \leq |P_1| + |P_2| + \cdots + |P_N|$$

with the following properties:

1. The $i^{th}$ row contains the DSSP sequence of protein $P_i$, with added gaps.

2. No column consists entirely of gaps

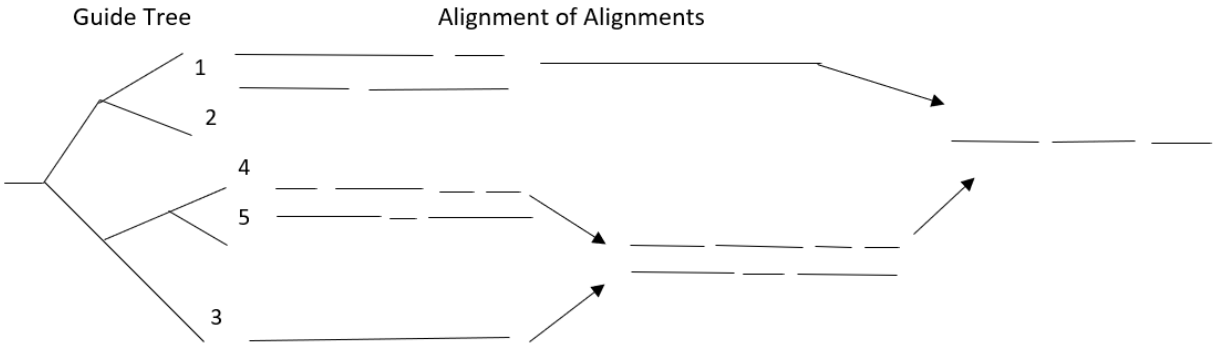



FIGURE 46: Alignment along the tree.

In this step, all alignment pairs between $P_o$ and every other protein are retrieved from the saved list, and merged sequentially, using Algorithm 3.2.2.

**Algorithm:** Correspondence Matrix

*Input*: List of proteins

*Output*: Multiple sequence alignment (correspondence matrix).

1: Initialize the correspondence matrix to null, calculate the pairwise alignment between all the pairs of protein sequences, starting with two closely related sequences obtained

2: **for** all combination of protein pair **do**

3: Compute the distance between the sequences

4: **end for**

5: Build a distance matrix

6: Build a dendogram for the guide tree using UPGMA method [26] in the distance matrix

7: **for** each closely related protein pairs or an alignment pair or a pair of protein and an alignment $p$ in the set of exsiting set of proteins **do**

8: apply a pairwsie global alignment with the alignment obtained in the previous step, it is done through the guide tree, until the root node is reached

9: **end for**

A sample correspondence matrix for the globin family is shown in Figure 47. The column-wise alignment of the SSEs of all the proteins is quite conspicuous.

At this point, we have identified conserved regions across all the proteins, but not aligned them in any way. Janardan's method [77] reaches a similar result, creating a correspondence matrix by carefully manipulating vectors.

The result is an MSA of DSSP sequences $S_i$ for proteins $P_i$, $1 \leq i \leq N$. The output from this step helps identify residue equivalences among the protein structures. To align them in 3D space,

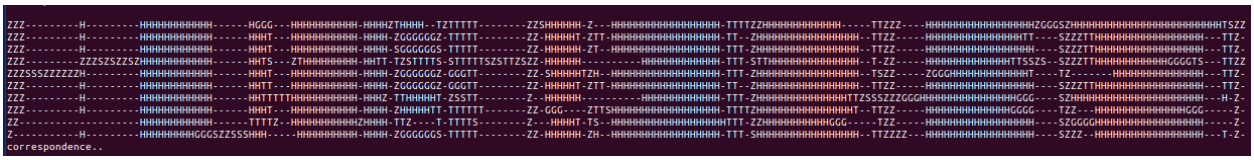



FIGURE 47: A correspondence matrix. [Notice there are no columns with gaps in all rows.]

we feed this output to the next step of our algorithm.

### 3.3.9 Rigid body superposition

We perform a rigid body superposition on the set of input proteins, to obtain a spatial alignment of the proteins. It is a transformation which involves a proper translation and rotation, preserving all the angles and distances between all the alpha carbon atoms of each protein. After this transformation, the distance between the alpha carbon atoms of equivalent residues is reduced below a threshold value [59].

For a rigid body superposition to take place, we keep the frame of reference to be fixed and the set of alpha carbon atoms equivalence. For this, we need a set of equivalences and a reference frame against which the rigid body superposition is to take place. We have both: the correspondence matrix from phase 2 gives us the residue-residue equivalences, and our chosen guide tree will guide us through the alignment.

We apply Kabsch's method [34] on each protein $P_i$ to superpose the structures from the leaf nodes of the guide for aligning them by walking through it to the root. The structure obtained in every step of the alignment is superposed with the previous alignment. For example, the spatial alignment of the globins 1DM1, 1MBC, 1MBA (see Figs. 48, 49, 50) generated by our algorithm is shown in Fig. 51.

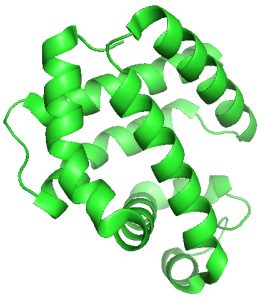


FIGURE 48: 1DM1

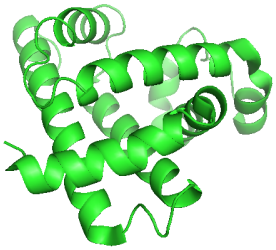


FIGURE 49: 1MBC

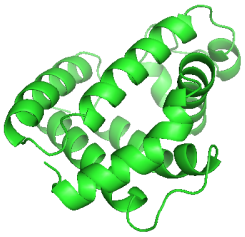


FIGURE 50: 1MBA

### 3.3.10 Dynamic programming and scoring

After applying rigid body transformation on the different proteins which were spatially aligned, the proteins are now very close to each other [59]. Since the distance between them are very close, we can compute the Euclidean distance between the pair of proteins so that we can increase the number of equivalence between the pairs [59].

The formula used to find the *enterRMSD* which determins the quality of the alignment is as follows:

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

Keeping the threshold value to (5Å), we can say that the pairs of alpha carbon atoms as equivalent pairs. An alignment which score is typically less than the threshold value is considered as a good alignment which means it has more equivalent pairs. Also, with similar biological properties, we find difficult alignments which result in exceeding this value by about 1.5Å [59]. The *centerRMSD* value of the above alignment, for example, is `0.36`Å.

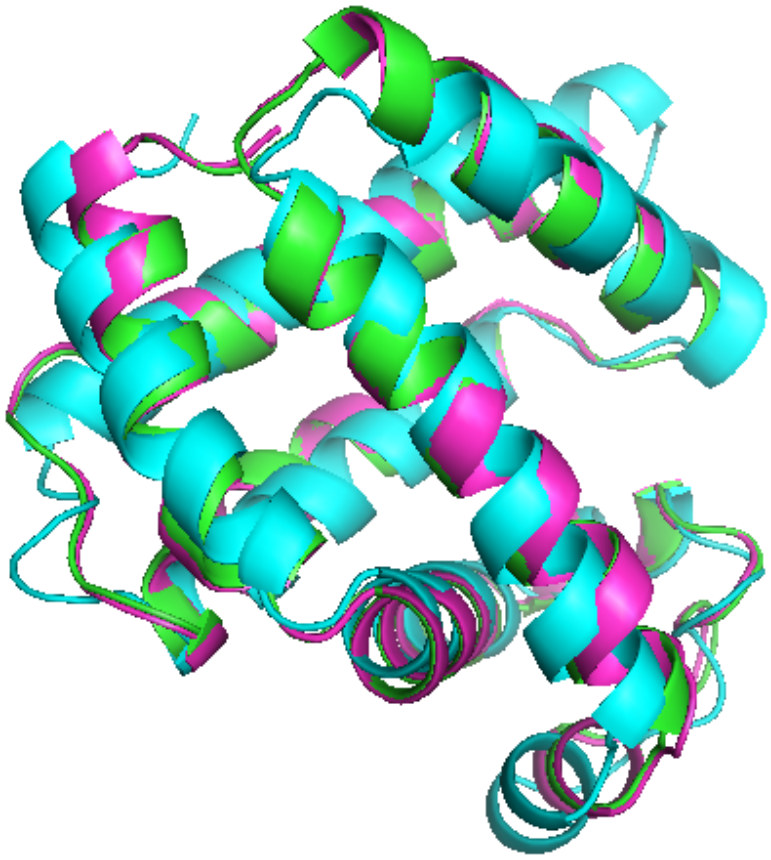


FIGURE 51: Alignment of 1DM1, 1MBC, and 1MBA

### 3.3.11 Algorithm description

A flowchart of Revised-MASCOT is given below:

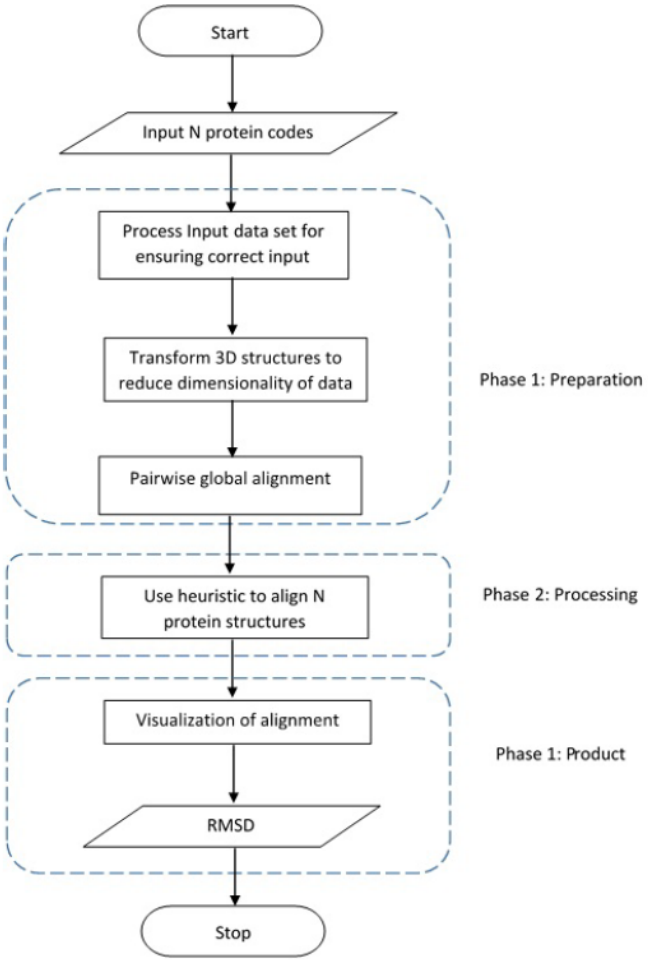



FIGURE 52: Flowchart of Revised-MASCOT [59]

A pseudo-code description of our algorithm is given below.

**Algorithm:** REVISED-MASCOT

*Input*: Protein $pdbids : (pdbid_1, pdbid_1, \cdots, pdbid_N)$

*Output*: Multiple alignment of proteins with files created for $pdbids_{1...N}$

▷ Phase 1

1: Extract protein structures into $P = \{P_1, P_2, \ldots, P_N\}$

2: Represent $P$ by corresponding DSSP-based sequences $S = \{S_1, S_2, \ldots, S_N\}$ consisting of DSSP-defined SSE motifs

3: compute the distances between all of the pairs of protein structures by measuring the similarity between every $(S_i, S_j)$ pair,

▷ Phase 2

4: Create an edit-distance matrix that stores the distances between every $(P_i, P_j)$ obtained using the distance computation method

5: Apply UPGMA method to the distance matrix and produce a guide tree

6: Create an MSA of $S$ w.r.t all the sequences alignments using the guide tree

▷ Phase 3

7: Treat all alignments of symbols with non-gaps as residue-residue equivalences of the pair $(P_i, P_j)$

8: Apply Kabsch's method to every $(P_i, P_o)$ pair to obtain $(trans_i, rot_i)$ for this pair

9: Use $(trans_i, rot_i)$ from Step 8 to transform and place $P_i$ in space, with $P_o$ being brought to origin first, to produce output pdb files

## 3.4 Experimental Results and Discussions

Revised MASCOT was implemented in Python 3.4, using packages from Bio-python 1.67 and Scikit-bio library, on a 64-bit HP desktop with a 3.4 GHz Intel CPU, running under Ubuntu 12.04.5. In the next few sections, we will be discussing the large variety of datasets and the experiments that were conducted. Given that $T_T$ and $T_G$ represent the time taken right from giving the input to producing the output files for Revised MASCOT and MUSTANG, respectively.

### 3.4.1 Globins

Studies show for the Multiple structural Alignment problems; Globin family plays an important role. This family is used for studying approximately 150-residue proteins [59]. Also, a common thing amongst the humans, for example, hemoglobin and myoglobin. Hence, the MStA algorithm is considered to be the best fit to find the similarity amongst the members of this family.

The following four sets of globins have been aligned using Revised MASCOT.

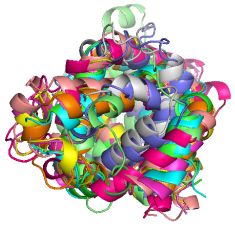


FIGURE 53: Set 1

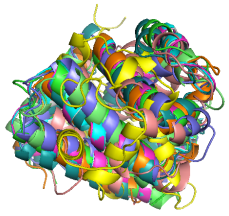


FIGURE 54: Set 2

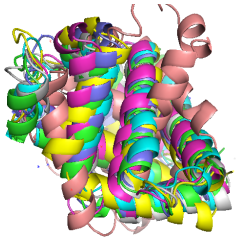


FIGURE 55: Set 3

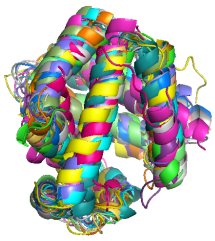


FIGURE 56: Set 4

We have taken Set 1 and experimented on this as it was used by [48], and [64] to show how their algorithms work on aligning globins. In our algorithm, the rmsd obtained for this superposition is `3.63`Å. Similarly, we have a set from [77], Set 2; it has obtained an rmsd of `3.00`Å to align the proteins. Set 3 is taken from [60], and obtaining an rmsd `3.30`Å. A random collection of 20 globins

TABLE 4: The table below shows the globins used in this section:

| Name | PDB ids | Count | $T_T$ | $T_G$ |
|---|---|---|---|---|
| Set 1 | 1HHO:A 2DHB:A 2DHB:B 1HHO:B 1MBD 1DLW 1DLY 1ECO 1IDR:A 2LH7 | 10 | 17s | 29s |
| Set 2 | 1MBC 1MBA 1DM1 1HLM 2LHB 2FAL 1HBG 1FLP 1ECA 1ASH | 10 | 18s | 26s |
| Set 3 | 5MBN 1ECO 2HBG 2LH3 2LHB 4HHB:B 4HHB:A | 7 | 13s | 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 | 27s | 1m 47s |

is Set 4 which is taken from [45] and [32] and gets an rmsd value of 3.10Å.

From all the alignments, we can see that the secondary structured elements of the pair of proteins are closely placed, it happens to be within a certain threshold distance.

### 3.4.2 Serpins

Serpins, which stands for Serine Protease Inhibitors, is considered to play a vital role in the biological world [9]. For example, they are considered to play important roles in human body like transporting hormones to various parts of the body which is done by a serpin Thyroxine-hiding globulin; and Maspin is another serpin which controls gene expression of certain tumors [9]

The following set of serpins have been aligned using Revised MASCOT.

TABLE 5: The table below shows the serpins used in this section:

| Name | PDB ids | Count | $T_T$ | $T_G$ |
|---|---|---|---|---|
| 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 | 2m 55s | 4m 14s |

Set 5 represents Serpins, and it is taken from [60]. It is one such kind of a set which is large in size and motif distribution, hence aligning this set of proteins is difficult and claimed to be quite difficult to align owing to their large size and motif distribution [59]. The set 5 figure, shows in spite of all these difficulties, it still manages to put all the secondary structures elements (beta sheets, hinges, and helices ) of the proteins in close proximity to each other. For the alignment purpose,[60] tried to find a common core. But in our algorithm, we applied a global alignment over

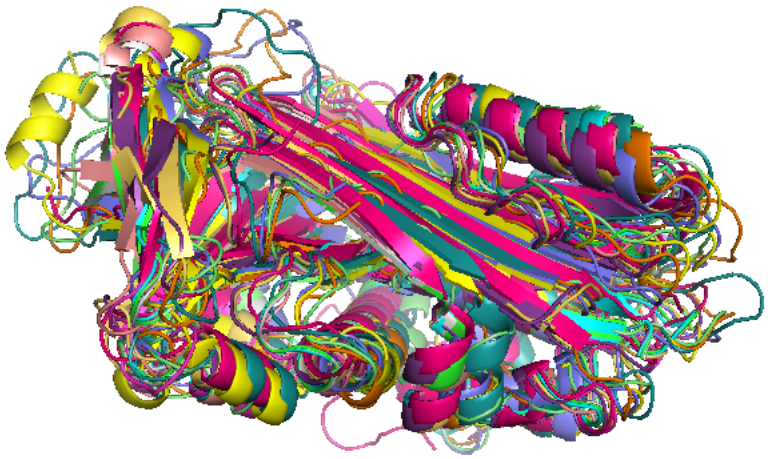


FIGURE 57: Set 5

the length of the proteins. The rmsd for this alignment is 3.82Å.

### 3.4.3 Barrels

The eight-stranded TIM-barrel, a family whose ancestry is still unknown and is still a mystery. There has been a strong debate on evolution history of this family, this is found in a lot of enzymes [59]. This ever-expanding family grows further after the adding the aligned TIM-barrel proteins [59]. The table below shows the aligned proteins in this set.

There is total of 66 molecules in set 6; this set is taken from MASS [14] which they used for an alignment algorithm to obtain some insight into how proteins with barrels align. Figure 58 shows how the alignment obtained by our new algorithm successfully superimposed the TIM barrel proteins [59]. The figure shows that this set of proteins share common structures and functions since all eight helices and eight beta sheets have been aligned. Revised MASCOT produced a rmsd

TABLE 6: The table below shows the barrels used in this section:

| Name | PDB ids | Count | $T_T$ | $T_G$ |
|---|---|---|---|---|
| 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 | 47m | 2h 32m |
| 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 12s | 1m 26s |

of 3.4Å for this set.

Set 7 has been extracted from the manually curated SCOP database Surprisingly, Figure 59 shows, our algorithm has been able to align the barrel motifs perfectly on each other and obtained an RMSD value of 3.55Å, in spite of the fact that all the proteins of this set belong from different superfamilies [59].

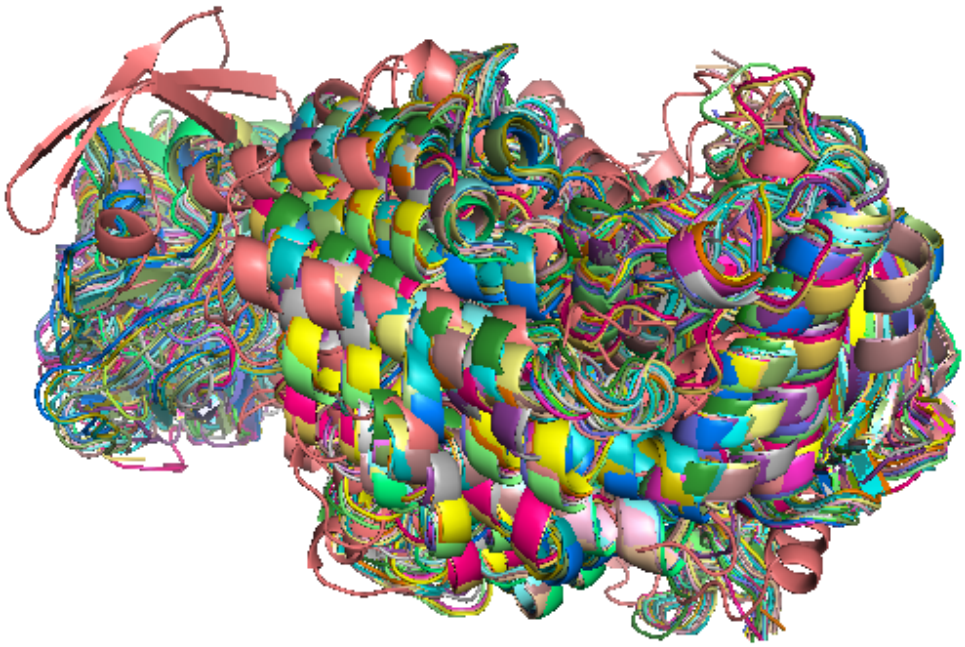


FIGURE 58: Set 6

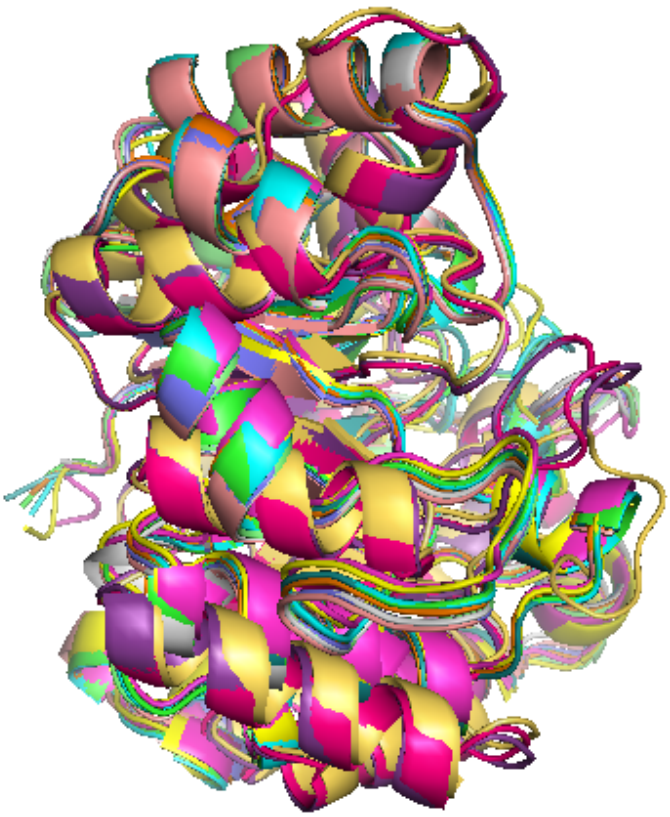


FIGURE 59: Set 7

### 3.4.4 Twilight-zone proteins

The structural comparison comes into play when the sequence similarity is less, but sequence alignment still stays an option until the proteins have 30% or more sequence identity [59]. The table 7 below shows datasets that belong to the twilight-zone.

TABLE 7: The table below shows the sets used in this section:

| Name | PDB ids | Seq. Identity | $T_T$ | $T_G$ |
|---|---|---|---|---|
| Set 8 | 1STF:I 1MOL:A 1CEW:I | <8% | 3s | 5s |
| Set 9 | 1BGE:A 1BGE:B 2GMF:A 2GMF:B | <12% | 6s | 15s |
| Set 10 | 1NSB 2SIM 1F8E 4DGR | <20% | 35s | 75s |

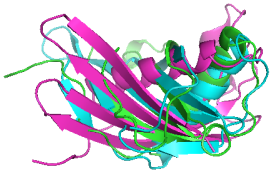


FIGURE 60: Set 8

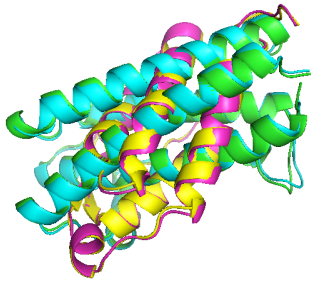


FIGURE 61: Set 9

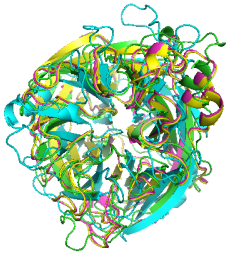


FIGURE 62: Set 10

For the significantly low sequence similarity proteins, we have created the above three sets [59]. Revised MASCOT proved itself better than any other versatile MStA algorithms since it has successfully aligned proteins that are very different from each other. It is only possible because Revised MASCOT uses Secondary structure elements as the sequential representation, unlike many other MStA algorithms which use primary residues[59]. Sets 8, 9, and 10 represent three bands of sequence identity within the twilight zone obtaining rmsd values of `3.71`Å, `4.00`Å, and `2.73` Å respectively [59].

### 3.4.5 Pig, Malaria, Human, and Dogfish - connected?

The Tree of life has sprung many branches over millennia. Could the branches for pigs, malaria parasites, humans, and dogfish have had a common root at some point in time? The below tables shows the structures used to seek more insights by alignment of the above species.

TABLE 8: The table shows the sets used in this section:

| Name | PDB ids | Count | $T_T$ | $T_G$ |
|---|---|---|---|---|
| Set 11 | 1MLD:A 1MLD:B 1MLD:C 1MLD:D 1T2D:A 1I0Z:A 1I0Z:B 1LDM:A | 8 | 1m 2s | 115s |

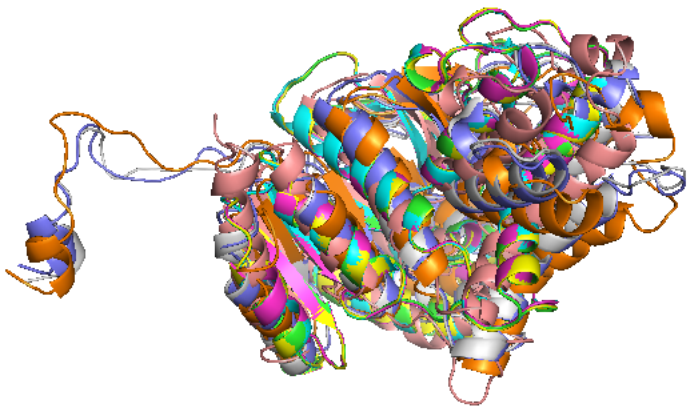


FIGURE 63: Set 11

The crystal structure of mitochondrial malate dehydrogenase from porcine heart (1MLD) con-

tains four identical subunits [9] [59]. The protein 1T2D uses, Plasmodium falciparum, the causative agent of malaria to enhance NAD+ regeneration [59]. New anti-malarial drugs [9] uses this protein. Protein from Homo sapiens is produced by the HRAS and HRAS1 genes [9] [59]. 1LDM represents the crystal structure of M4 apo-lactate dehydrogenase from the spiny dogfish (Squalus acanthius) [9] [59].

The alignment obtained from Set 11 is shown in Figure 63. A striking similarity is found by Revised MASCOT among these molecules with rmsd `2.885`Å, which proves that at some point in time these species might have had evolved from some common ancestor [59] .

### 3.4.6 Human, Chicken, Rabbit, Yeast, and Nematode

A collective group of protein from species like human, chicken, rabbit, yeast, and nematode were taken into consideration to check weather molecules taken from such diverse taxa be aligned to find structural similarity?

TABLE 9: The table below shows the sets used in this section:

| Name | PDB ids | Count | $T_T$ | $T_G$ |
|---|---|---|---|---|
| Set 12 | 1SSG:A 1SSG:B 1HTI:A 1HTI:B 1R2S:A 1R2T:A 1MO0:A 1MO0:B 7TIM 3YPI | 10 | 49s | 11m 15s |

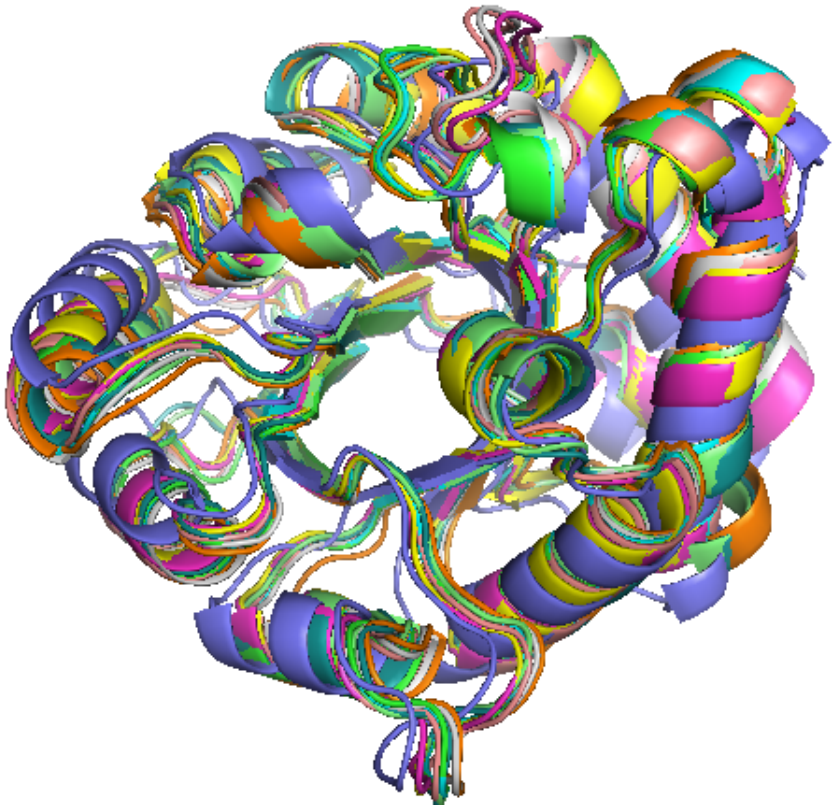


FIGURE 64: Set 12

Different family performs the same function in their way. For example, glycolysis is the 'metabolic pathway' [58] using which glucose is broken down to form free energy [59]. This process is done differently by a different group of species. Chicken does this using the protein 1SSG [9], Humans do the same thing using protein 1HTI[9]. Revised MASCOT is applied to structures taken from rabbit muscle (1R2S, 1R2T), baker's yeast (7TIM, 3YPI), and nematode (1MO0) obtaining a rmsd of `1.74`Å. We did this to confirm that these proteins are used for the same purposes [59].

### 3.4.7 Seafood allergy in Fish!

The presence of some proteins cause havoc in the immune system which is the general cause of seafood allergy amongst the humans and rats [59]. This further raises a question whether such behavior is exhibited amongst the fishes? This further raises a concern if all the fishes become allergic to seafood, how will the possibly survive? [59]

TABLE 10: The table below shows the sets used in this section:

| Name | PDB ids | Count | $T_T$ | $T_G$ |
|---|---|---|---|---|
| Set 13 | 1RWY:A 1RJV:A 4CPV 3PAL 1BU3 5PAL | 6 | 6s | 10s |

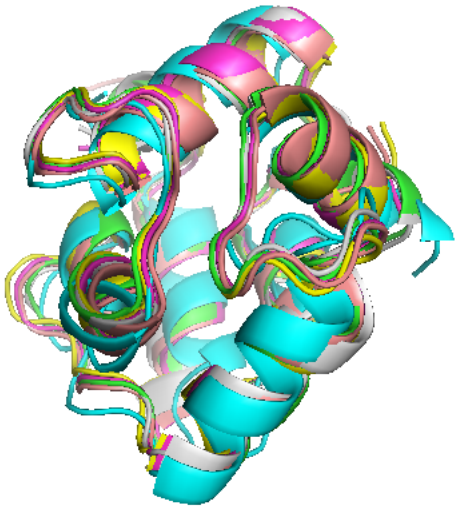


FIGURE 65: Set 13

Seafood allergy is caused by proteins 1RWY and 1RJV in common brown rats and humans.

We have seen, proteins 4CPV, 3PAL, 1BU3, and 5PAL taken from common carp, pike, silver hake and leopard shark have highly similar tertiary structures [59]. Therefore, after performing exhaustive experiments, we have come to this conclusion that there are some fishes that have a host of proteins with similar structure and function [59]. Is there a possibility that these proteins might cause seafood allergy in these fishes? It turns out that indeed they do. A recent study by Swoboda et al. [65] shows that parvalbumins, such as the ones taken above, are the major cross-reactive fish allergen. Figure 65 shows how Revised MASCOT correctly aligns the EF-hand motifs in these proteins, albeit with a rmsd of `3.82`Å [59] .

## 3.5 Conclusion

The above varied experiments shows Revised-MASCOT is very simple and fast in aligning multiple proteins efficiently. It works well with a wide range of proteins. But has struggled for proteins from different families and different ancestors. Keeping in mind, Revised-MASCOT is an heuristic algorithm it obtains a multiple structure alignment in lesser time as compared to the widely known, MUSTANG algorithm.

# CHAPTER 4

# *Conclusion and Future Work*

## 4.1 Evaluating Pairwise Alignment Algorithm

The evidence from the experiments suggests that of the four algorithms $TM\text{-}align$, and $CE$ are the successful ones in correlating structural proximity to spatial proximity. In $TM\text{-}align$ and $CE$, the majority of the proteins were clustered according to their family, and there was a minimal mix-up of the families as compared to the other two pairwise alignment algorithms. As for $EDAlignSSE$, it performed creditably as compared to $DALI$[29]; in fact, it managed to get some of the proteins clustered according to their families. This is particularly evident for dataset 3, in which case it formed four family clusters with similar proteins. However, only in this dataset, $CE$ performed better than $TM - Align$. In the final datasets, with 42 different proteins, $DALI$[29] showed very poor clustering. The huge data set accentuated its weakness more emphatically. Moreover, from the 3-dimensional plot that we have obtained from all the dataset, it verifies that the translation of the structural proximity to spatial proximity remains unchanged even when the dimensions change from 2D to 3D. The clustering of the families is the same for both the dimensions. Therefore, based on our clustering technique, we conclude that $TM\text{-}align$, and $CE$ are the more effective pairwise alignment algorithm than $EDAlignSSE$ or $DALI$ [29]. The alignment obtained from an algorithm is possibly, may depend on how the algorithm has abstracted a protein structure. All the

four algorithms taken for evaluation in my thesis has taken the backbone of a protein (the $C-\alpha$ atom) as an abstract of the protein structure. This abstraction of a protein structure plays a very crucial role in how the alignment algorithm performs.

This work can be extended further to test and evaluate many other pairwise alignment algorithms proposed in the past. It can be used as a measure to find the quality of new pairwise alignment algorithms. This work can also be used to infer evolutionary relationships from the clusters. Our solution also opens a problem on how to define an abstraction of a protein structure which is the first step in any alignment algorithm, this step will result in giving an efficient output. This can be considered as an important step on how an alignment algorithm works.

## 4.2 Revised MASCOT

Our goal was to design an algorithm, which aligns more than two proteins accurately, optimally and efficiently by comparing their 3-D structures, which will allow us to find biological similarities leading to functional similarities of protein. To achieve, we reduced the problem to a sequence similarity problem by converting the 3D structures to SSE elements of the proteins and performing a progressive heuristic approach.

We studied Revised MASCOT very carefully and compared the results with the famous MUSTANG algorithm. Similar to MASCOT [59], Revised MASCOT is also a simple, fast and elegant - features. It can be used to its potential use to solve more complex and more sophisticated multiple structure alignment problems [59]. Revised MASCOT has proved its ability and capability to align different kinds of proteins efficiently by obtaining the *centerRMSD* scores in a large variety of experiments. Despite being a heuristic method like MUSTANG, it took lesser time than MUSTANG for aligning multiple complex proteins. Datasets which contains proteins from different families

and are not closely related has shown not-so-good alignment but still managed to get the loops and bends of different proteins near each other. It successfully reveals the structural alignments with high accuracy and efficiency [59].

In future, we plan to extend our work on Revised MASCOT in the following directions:

1. Incorporate more protein flexibility into the algorithm [59].

2. Derive a common core structure from the aligned input proteins for use as a template for protein threading [59].

3. Try a different approach to the same problem and compare its result to the existing algorithm results.

# REFERENCES


[1] B. Alberts, D. Bray, K. Hopkin, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Essential cell biology*. Garland Science, 2013.

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

[3] V. Alesker, R. Nussinov, and H. J. Wolfson. Detection of non-topological motifs in protein structures. *Protein Engineering, Design and Selection*, 9(12):1103–1119, 1996.

[4] N. N. Alexandrov and D. Fischer. Analysis of topological and nontopological structural similarities in the pdb: new examples with old structures. *Proteins: Structure, Function, and Bioinformatics*, 25(3):354–365, 1996.

[5] N. N. Alexandrov, K. Takahashi, and N. Gō. Common spatial arrangements of backbone fragments in homologous and non-homologous proteins. *Journal of molecular biology*, 225(1):5–9, 1992.

[6] R. Andonov, N. Malod-Dognin, and N. Yanev. Maximum contact map overlap revisited. *Journal of Computational Biology*, 18(1):27–41, 2011.

[7] P. J. Artymiuk, R. V. Spriggs, and P. Willett. Graph theoretic methods for the analysis of structural relationships in biological macromolecules. *Journal of the Association for Information Science and Technology*, 56(5):518–528, 2005.

[8] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank, 1999–. In *International Tables for Crystallography Volume F: Crystallography of biological macromolecules*, pages 675–684. Springer, 2006.

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

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

[11] C. I. Branden et al. *Introduction to protein structure*. Garland Science, 1999.

[12] H. Y. Chang and X. Yang. Proteases for cell suicide: functions and regulation of caspases. *Microbiology and molecular biology reviews*, 64(4):821–846, 2000.

[13] P. Di Lena, P. Fariselli, L. Margara, M. Vassura, and R. Casadio. Fast overlapping of protein contact maps by alignment of eigenvectors. *Bioinformatics*, 26(18):2250–2258, 2010.

[14] O. Dror, H. Benyamini, R. Nussinov, and H. Wolfson. Mass: multiple structural alignment by secondary structures. *Bioinformatics*, 19(suppl_1):i95–i104, 2003.

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

[16] S. J. Flint, V. R. Racaniello, L. W. Enquist, A. M. Skalka, et al. *Principles of virology, Volume 2: pathogenesis and control.* Number Ed. 3. ASM press, 2009.

[17] M. Gerstein and M. Levitt. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. In *Ismb*, volume 96, pages 59–66, 1996.

[18] M. Gerstein and M. Levitt. Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *Protein Science*, 7(2):445–456, 1998.

[19] glenys. Ppt - multiple sequence alignment and phylogenetic trees ..., 2009.

[20] A. Godzik, A. Kolinski, and J. Skolnick. Topology fingerprint approach to the inverse protein folding problem. *Journal of molecular biology*, 227(1):227–238, 1992.

[21] H. M. Grindley, P. J. Artymiuk, D. W. Rice, and P. Willett. Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. *Journal of molecular biology*, 229(3):707–721, 1993.

[22] C. Guda, S. Lu, E. D. Scheeff, P. E. Bourne, and I. N. Shindyalov. Ce-mc: a multiple protein structure alignment server. *Nucleic acids research*, 32(suppl_2):W100–W103, 2004.

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

[24] H. Hasegawa and L. Holm. Advances and pitfalls of protein structural alignment. *Current opinion in structural biology*, 19(3):341–348, 2009.

[25] L. Hedstrom. Serine protease mechanism and specificity. *Chemical reviews*, 102(12):4501–4524, 2002.

[26] D. M. Hillis, J. J. Bull, et al. Experimental phylogenetics: Generation of a know phylogeny. *Science*, 255(5044):589, 1992.

[27] S. M. Holland. Principal components analysis (pca). *Department of Geology, University of Georgia, Athens, GA*, pages 30602–2501, 2008.

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

[29] L. Holm and C. Sander. Dali: a network tool for protein structure comparison. *Trends in biochemical sciences*, 20(11):478–480, 1995.

[30] L. Holm and C. Sander. Dali/fssp classification of three-dimensional protein folds. *Nucleic acids research*, 25(1):231–234, 1997.

[31] L. Holm, C. Sander, et al. Mapping the protein universe. *SCIENCE-NEW YORK THEN WASHINGTON-*, pages 595–602, 1996.

[32] I. Ilinkin, J. Ye, and R. Janardan. Multiple structure alignment and consensus identification for proteins. *BMC bioinformatics*, 11(1):71, 2010.

[33] JJ. Amino acid, 2017.

[34] W. Kabsch. 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):922–923, 1976.

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

[36] I. Koch, T. Lengauer, and E. Wanke. An algorithm for finding maximal common subtopologies in a set of protein structures. *Journal of computational biology*, 3(2):289–306, 1996.

[37] P. Koehl. Protein structure similarities. *Current opinion in structural biology*, 11(3):348–353, 2001.

[38] R. Kolodny and N. Linial. Approximate protein structural alignment in polynomial time. *Proceedings of the National Academy of Sciences of the United States of America*, 101(33):12201–12206, 2004.

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

[40] N. Krasnogor and D. A. Pelta. Measuring the similarity of protein structures by means of the universal similarity metric. *Bioinformatics*, 20(7):1015–1021, 2004.

[41] M. Levitt and M. Gerstein. A unified statistical framework for sequence comparison and structure comparison. *Proceedings of the National Academy of sciences*, 95(11):5913–5920, 1998.

[42] A. Lewit-Bentley and S. Réty. Ef-hand calcium-binding proteins. *Current opinion in structural biology*, 10(6):637–643, 2000.

[43] S. C. Li. The difficulty of protein structure alignment under the rmsd. *Algorithms for Molecular Biology*, 8(1):1, 2013.

[44] G. Lu. Top: a new method for protein structure comparisons and similarity searches. *Journal of Applied Crystallography*, 33(1):176–183, 2000.

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

[46] A. R. Means, M. F. VanBerkum, I. Bagchi, K. P. Lu, and C. D. Rasmussen. Regulatory functions of calmodulin. *Pharmacology & therapeutics*, 50(2):255–270, 1991.

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

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

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

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

[51] R. Nussinov and H. J. Wolfson. Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. *Proceedings of the National Academy of Sciences*, 88(23):10495–10499, 1991.

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

[53] C. A. Orengo and W. R. Taylor. [36] ssap: sequential structure alignment program for protein structure comparison. *Methods in enzymology*, 266:617–635, 1996.

[54] S. C. Panigrahi and A. Mukhopadhyay. An eigendecomposition method for protein structure alignment. In *International Symposium on Bioinformatics Research and Applications*, pages 24–37. Springer, 2014.

[55] pypr. kmeans clustering, 2010.

[56] S. Rahmati and J. I. Glasgow. Comparing protein contact maps via universal similarity metric: an improvement in the noise-tolerance. *International journal of computational biology and drug design*, 2(2):149–167, 2009.

[57] S. Ray. Beginners Guide To Learn Dimension Reduction Techniques. `https://www.analyticsvidhya.com/blog/2015/07/dimension-reduction-methods//`, 2015.

[58] A. Romano and T. Conway. Evolution of carbohydrate metabolic pathways. *Research in microbiology*, 147(6-7):448–455, 1996.

[59] K. Roy, S. C. Panigrahi, and A. Mukhopadhyay. Multiple alignment of structures using center of proteins. In *International Symposium on Bioinformatics Research and Applications*, pages 284–296. Springer, 2015.

[60] M. Shatsky, R. Nussinov, and H. J. Wolfson. Multiprot-a multiple protein structural alignment algorithm. In *WABI*, volume 2, pages 235–250. Springer, 2002.

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

[62] Y. Shibberu and A. Holder. A spectral approach to protein structure alignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(4):867–875, 2011.

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

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

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

[66] J. D. Szustakowski and Z. Weng. Protein structure alignment using a genetic algorithm. *Proteins: Structure, Function, and Bioinformatics*, 38(4):428–440, 2000.

[67] W. R. Taylor. Protein structure comparison using iterated double dynamic programming. *Protein Science*, 8(3):654–665, 1999.

[68] W. R. Taylor, A. C. May, N. P. Brown, and A. Aszódi. Protein structure: geometry, topology and classification. *Reports on Progress in Physics*, 64(4):517, 2001.

[69] W. R. Taylor and C. A. Orengo. Protein structure alignment. *Journal of molecular biology*, 208(1):1–22, 1989.

[70] S. Wang, J. Ma, J. Peng, and J. Xu. Protein structure alignment beyond spatial proximity. *Scientific reports*, 3, 2013.

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

[72] Wikipedia. Protein data bank, 2009.

[73] Wikipedia. Phosphotransferase, 2017.

[74] Wikipedia. Protein strcutures, 2017.

[75] T. D. Wu, S. C. Schmidler, T. Hastie, and D. L. Brutlag. Regression analysis of multiple protein structures. *Journal of Computational Biology*, 5(3):585–595, 1998.

[76] J. Xu and Y. Zhang. How significant is a protein structure similarity with tm-score= 0.5? *Bioinformatics*, 26(7):889–895, 2010.

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

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

[79] A. Zemla. Lga: a method for finding 3d similarities in protein structures. *Nucleic acids research*, 31(13):3370–3374, 2003.

[80] Y. Zhang and J. Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

[81] Y. Zhang and J. Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

# VITA AUCTORIS

NAME: Shalini Bhattacharjee

PLACE OF BIRTH: Darjeeling, India.

EDUCATION: Bachelor of Technology in Computer Science, West Bengal Univeristy of Technology, Kolkata, India, 2015.

Master of Science in Computer Science, University of Windsor, Windsor, Ontario, Canada, 2017.