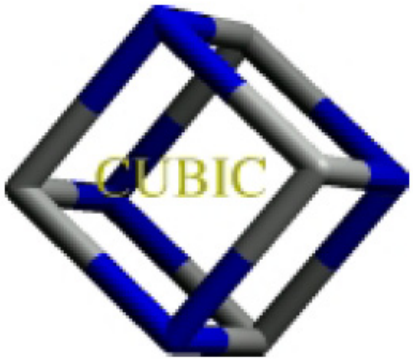
Benchmarking and Extension of Protein Structure Alignment tool (Protein3DFit)

Project Report

by

Nasir Mahmood

Course: Applied Bioinformatics



Cologne University Bioinformatics Center (CUBIC)
Institute of Biochemistry

University of Cologne

Supervisor 1: Prof. Dr. Dietmar Schomburg
Supervisor 2: Pascal Benkert, M.Sc

Acknowledgments

I would like to thank my supervisors, Prof. Dr. Dietmar Schomburg and Mr. Pascal Benkert for their expert guidance and encouraging attitude during this work.

I feel utmost pleasure to express my gratitude to my parents and wife who stood by me in solving all my troubles with love.

Further thanks go to the secretarial staff of the CUBIC and the computing support staff for their support on technical and all around computing support.

Nasir Mahmood

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Proteins are essential for the structure and function of all living cells and viruses. Proteins are composed of a polypeptide chain of amino acids joined by peptide bonds. The sequence of amino acids of a protein is called its primary structure. The shape protein naturally folds into is called "native state" and is determined by the sequence of amino acids. There are 20 amino acids that are commonly found in proteins, each with similar but yet unique structure. Amino acid chain is arranged locally into a secondary structure by hydrogen bonding within the peptide backbone. The most common secondary structure elements (SSEs) in proteins are alpha (α) helix and beta (β) sheet. The global folding of a single polypeptide chain is called tertiary structure of protein. Proteins achieve their functions by binding to other molecules, and tertiary structure controls the existence and placement of binding sites.

With rapid increase in number of protein structures stored in Protein Data Bank (PDB), there is an immense need for an efficient structural alignment tool to perform analysis and comparison of three-dimensional structures [14]. Currently, PDB contains more than 32000 structures and 10-20 structures are being submitted daily. When this number was small, structures were aligned by visual inspection but with increase in available proteins and recent advances in structural genomics, need for efficient structural alignment is evident [21].

It is believed that protein function is determined from its three-dimensional structure because the binding of proteins to proteins and to ligands depends purely on the stability and mechanical aspects of three-dimensional structure. Similar structures may perform similar functions and similarity of structures can be used for determination of their functions. During evolution process, structure of protein remains more conserved than sequence and on the basis of this fact, high similarity of sequences of two proteins almost implies structure similarity but the opposite is not always true. Therefore, alignment of proteins structures provides significant clues by identifying the structural similarities that purely sequence-based methods cannot [12] [26].

Structural information of proteins provides invaluable information about evolutionary and functional characteristics of protein and it has very important application in drug design efforts. In theory, structures of protein are determined by three methods; by use of experimental information from X-ray crystallography or NMR spectroscopy, by purely theoretical methods, or by homology modelling. In foreseeable future, experimental methods are incapable of determining the structures of a fraction of billions of proteins in the world. As far as theoretical approaches to solve protein structures are concerned, they are lacking in providing high-resolution information about most of the protein structures [25]. Therefore, techniques of threading and homology modelling are getting more attention for protein structure prediction/determination. These methods make use of information from proteins whose structures have already been determined experimentally called templates to predict structure of a sequence. Structural alignment is an integral part of threading and homology modelling based protein structure prediction methods and is used as a "gold standard" for testing of structure prediction algorithms [23]. Homology modelling consists of four steps: (1) identification of homologs/template(s); (2) alignment of target with template(s); (3) building of model based on alignment of target with template(s); (4) evaluation of the model. Template protein structures are aligned to determine common sub-

structure to provide a base to the modelled protein structure. To make sure that resolved structure of new protein is in agreement with those of templates used, it is aligned with predicted model [23].

There are thousands of proteins with sequence identity less 25% but have been evolved naturally from the same ancestor into similar structures. In such cases, threading techniques are used to align protein sequences with known structures in order to find structural models for the unknown fold of a given sequence. Development, testing and evaluation of these methods depend on a library of similar structures and structural equivalence among them. Relevant structure-to-structure alignments are used to rate the predicted sequence to structure alignments [27].

1.1 Structural Alignment Problem

Given two proteins A and B, find two subchains P and Q of equal length such that

1. $A(P)$ and $B(Q)$ are similar, and
2. Correspondence length $|P| = |Q|$

is maximal under condition 1. As we are interested in relative position and orientation of structures of two proteins, the structure of one protein is kept fixed while that of second one is rigidly transformed by rotating and translating it without disturbing its internal structure.

Structural alignment problem is structural analogy of well-known sequence alignment problem but former is performed between known structures of two proteins and is based on the Euclidean distance between corresponding pairs of residues whereas later is based on distance between amino acid types.

Given an optimized separable scoring function, optimal sequence alignment can always be found using dynamic programming. But it is very difficult to find out the appropriate set of parameters of the scoring function that results in similarity between amino acid residues. Substitution matrices are used to find meaningful equivalences between amino acids and to help to figure out biologically meaningful sequence alignments.

For the alignment of two structures, it is very hard to find the optimal alignment because rotation and translation of one structure must be found to superimpose it onto the other. All solutions to structural alignment are based on heuristics and therefore only provide an optimal approximate solution. Similarity in structural alignment is measured by coordinate root mean square (cRMS) of aligned C_α atoms of protein structures. In addition to cRMS, number of aligned residues and number and length of alignment gaps are also used as similarity measures [10].

1.2 Structural Alignment Issues

1.2.1 Is there a unique answer?

The assumption of unique structural alignment cannot be warranted, as it exists in the field of homology searches and sequence comparison. The fundamental difference between these two fields is that when we compare two sequences of protein we are certain that there is a unique, correct solution but very rarely we are able to find it. As result of a series of mutations, deletions and insertions, both proteins have evolved from the same common ancestor and there is a molecular process to link one sequence into other one. Therefore, there is a unique, one to one correspondence between positions in each protein and positions in the common ancestor. If this correspondence could be known, it could be used to create unique sequence alignment.

As we are unable to find out this true correspondence, therefore all the sequence alignments are just approximate solutions depending on our approximate knowledge about the process of evolution of sequences of protein. As we have discussed earlier, all structural alignment approaches are based on some heuristic and use simplifications of a scoring function or search procedure. Different methods see unquestionable similarity between structures of protein in different ways due to their different optimization methods, such as the dynamic programming, two level dynamic programming and Monte Carlo minimization. These differences are very few on the level of secondary structure elements (SSEs), such as helices and strands but become clearly visible by differing in 2-4 positions if an algorithm tries to align structures at residue level.

Different similarity measures can also lead to different alignments and structural alignment optimizing one similarity measure can be close, but by no means identical, to the alignment optimizing a different similarity measure. Therefore, for the same pair of proteins, we can have two completely different alignments generated by two different structural alignment algorithms. That is why, there is no such alignment that could be used as a standard of truth to judge and validate other alignment methods, such as threading or sequence alignment [1].

1.3 Similarity Measures / Scoring Functions

Though most of the algorithms for protein structure alignments use similarity measures, which differ from each other, there are two main methods to quantify similarity. In first method, internal distances between corresponding atoms in the two proteins are calculated. This distances measure is called distance root mean square *(dRMS)*.

$$dRMS = \sqrt{\frac{1}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (d_{ij}^{A} - d_{ij}^{B})^2}$$

where d_{ij}^A and d_{ij}^B are the distances between atoms i and j in molecules A and B, respectively.

The second method which is called coordinate root mean square *(cRMS)* uses the actual Euclidean distance between corresponding atoms in the two proteins under consideration. For that, method must also have to find out the rigid transformation that optimally superimposes one structure onto the second one.

$$cRMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(||x(i) - y(i)||^2)}$$

Where N is the number of atoms in the list of equivalences, and x and y are the coordinates of atom indexed i in protein A and protein B, respectively. Both *cRMS* and *dRMS* are based on L2-norm (i.e. the Euclidian norm) and, as such, they suffer from the same draw back as the residual, X^2 , in least-squares minimization: the presence of outliners introduces a bias in the search of the fit may be artificially poor because of the sole presence of these outliers. As a result, RMS is a useful measure of structural similarity only for closely related proteins. Several other measures have therefore been proposed to circumvent these problems. We will discuss later a scoring function used in our algorithm to convert distances between superimposed corresponding C_α atoms to similarity score to be used by dynamic programming to create optimal alignment. On the basis of above two methods for quantifying similarity, there are two approaches to address the problem of structural alignment of two proteins. In case of first approach, heuristic algorithms have been developed to compare internal distance matrices of proteins consideration. One advantage of such type of algorithms is that they do not need to find an optimal rigid body transformation to superimpose one structure onto other. DALI, most commonly used structural alignment server (http://www.ebi.ac.uk/dali/), belongs to this group. In second approach, heuristic algorithms have been designed to find optimal correspondence and transformation both simultaneously. Neither first

nor second approach based heuristics algorithms are able to find an optimal alignment with respect to any scoring function [11].

1.4 Search Algorithms

Structural alignment of a pair of proteins is an NP hard problem and it is not possible to find the unique solution in realistic period of time. Therefore, development of heuristic algorithms is a good choice. Different algorithms based on different heuristics may not produce exactly the same solution to a structural alignment problem. An algorithm can be designed to find either local or global similarities between two structures but recent approaches are trying to find a middle path to detect both local and global similarities (equivalences) and for that several approaches are being used. Among them most important ones are; comparison of distance matrices, fragment matching, geometric hashing, maximal common sub-graph detection and local geometry matching. Residue equivalences found by these methods are then optimized by dynamic programming. Some of these methods will be discussed later in little bit more detail.

1.4.1 Iteration of alternating Superposition

Initial seed alignments/correspondences are found by making pairs of short fragments (4-6 residues) called aligned fragment pairs (AFPs). Given a set of seed alignments in the form of AFPs, a superposition algorithm can be used to find a transformation minimizing an RMSD measure. When one structure is superimposed onto the other by using the transformation, the distances between all pairs of atoms (residues) are then calculated to use them in a scoring matrix or to make new correspondences between residues of structures. The scoring matrix is used to make an alignment and correspondences to find a transformation to further minimize RMSD measure. This iteration could be continued until RMSD is minimized. Rao & Rossmann in [17], Rossmann &

Argos in [18] [19], Gerstein & Levitt in [4], and Lessel & Schomburg in [14] have used this approach to align protein structures.

1.4.2 Dynamic programming

Dynamic programming approaches try to find exact solution to alignment problem but are dependent on the target function, which might not reflect information about the alignment of other parts of protein molecule. If the structures are superimposed and we have a scoring scheme to construct similarity matrix, then dynamic programming can optimally align two structures of protein. Depending on superposition of AFPs, one might ideally like wish to align the superposed structures to optimize a score but alignment of any two substructures affects the scoring of alignment of the complete structures and independency requirement of dynamic programming is violated. To solve this problem, several algorithms have been proposed to extend dynamic programming [3]. In a heuristic method by Sali & Blundell in [20], several alignments are made, one for each relation. In residue-by-residue matrix, a residue pair is assigned a high score if it is found in many relationships based on alignments. Finally, this matrix is combined with property information to construct a new matrix to be used by dynamic programming for final alignment. In SSAP [24], Taylor & Orengo used dynamic programming at two levels called double dynamic programming. At lower level, many dynamic programming matrices are calculated and the highest scoring path from each lower level matrix is propagated to higher level dynamic programming matrix to find over all best alignment.

1.4.3 Geometric Hashing

Geometric hashing is a method for efficiently finding geometric objects of the same or similar shape, even though they may be rotated or otherwise transformed. In structural alignment of proteins, the aim of geometric hashing is to find common substructures. Coordinates of all or a subset of the elements

(atoms/residues/SSEs) are transformed into local coordinate systems. Set of elements from the two structures with the same mutual spatial relations are used to make common substructures. A highly redundant representation of the structures is used, which is independent of rotation, translation, and sequence order. Hash tables are used for storing and comparing local geometrical information, hence the name is called geometric hashing. Three points (x, y, z) are used to make three-dimensional frame for local reference systems. In [8], Holm and Sander have proposed a method where geometric hashing is used with vector representation of SSEs in right-handed coordinate systems. Ordered pairs of SSEs are used in coordinate system with origin at midpoint of first vector directing y-axis along that vector. In [16], Nussinov and Wolfson also used geometric hashing to align structures in sequence independent way.

1.4.4 Graph Theory

A graph representation of protein molecule enables to apply graph-theoretical approaches to solve problem of structure alignment. The approach involves three typical steps: (a) graph representation of protein structure, (b) matching of representation graphs and (c) finding common sub-graph to establish similarity between structures. Common sub-graph can be created by another correspondence graph in each node is formed from a corresponding pair of nodes from representation graphs of proteins under consideration. An edge is added between two nodes in correspondence graph if the edges in the original representation graphs are equivalent to a common sub-graph [12] [13]. Three-dimensional graphs of chemical structures connecting all atoms with distance labelled edges and with special labels for chiral centres are not applicable to compare protein structures because of high cost of graph matching [12]. Size limitation of graph theory can be overcome by using few less elementary objects as graph vertices. Protein secondary structure elements play important role in determining functions of a protein and remain conserved during process of evolution. For identification of folds, SSEs have been used as elementary

objects (vertices) in many studies [15] [7] [6] [22] [9].

Chapter 2

Method

2.1 Detection of Fragment Pairs

Structural alignment of protein consists of two basic operations: (a) finding initial seed alignments/correspondences to be used as anchor points for calculation of transformation, (b) Superposition of two structures by transforming target structure onto the reference with the help of an optimal transformation.

Seed alignments consist of protein fragment pairs that are combined in the following step into quartets to provide anchor points for the transformation of structures. For a give minimum length m, each fragment of first molecule is compared with each possible fragment of molecule B. This comparison is based on the intra-molecular C_α distances of the molecules under consideration. A fragment pair is made if the root mean square (r.m.s) deviation of corresponding C_α distances is below a threshold value of r_1 as given in equation 2.1 [14].

To generate fragment pairs (sometimes are called aligned fragment pairs (AFPs)), C_α distance matrices of molecules A and B are constructed. Only those parts of $N_A N_B$ distance matrices are compared which are representatives of the fragments under consideration.

$$\sqrt{\frac{\sum_{i=1}^{m_1-1}\sum_{j=0}^{i-1}(a_{(p1+i,p1+j)} - b_{(f1+i,f1+j)})^2}{\frac{1}{2}.m_1(m_1-1)}} \leq r_1 \tag{2.1}$$

According to equation 2.1, a fragment A_1 from a_{p1} to a_{q1} in molecule A will be matched with another fragment B_1 from b_{f1} to b_{g1} in molecule B if r.m.s deviation between corresponding C_α atoms of the fragments is less than or equal to r_1. As C_α distance matrices are symmetrical, elements of main diagonal plus $m_1(m_1-1)/2$ of the C_α difference in each distance matrix can be omitted.

Distances to the adjacent C_α atoms are also not so informative because they will also be around 3.8 A^0 with a small degree of deviation [14].

$$\sqrt{\frac{\sum_{i=1}^{m_1-1}(a_{(q1+1,p1+i)} - b_{(g1+1,f1+i)})^2}{m_1}} \leq r_1 \tag{2.2}$$

According to equation 2.2, a fragment pair can be extend to any number of C_α atoms as far as relationship of new C_α atom to other C_α atoms in parts A_1 and B_1 holds. It means fragment pair A_1 and B_1 is elongated by C_α atoms a_{q1+1} and b_{g1+1} if equation 2.2 is valid. On each addition of an C_α atom to fragment pairs, m_1 is increased by one and this process continues until 2.2 stands true.

2.2 Combination of Fragment Pairs - Quartets

After having identified and elongated all the fragment pairs $A_i|B_j$, it is necessary to check if several of these fragment pairs can be superimposed by the same translation and rotation. For that purpose, two fitting fragments are

combined into a quartet. A quartet composed of two fragment pairs $A_1|B_1$ and $A_2|B_2$ is accepted if the following conditions are fulfilled:

i) the participating fragment pairs A_1 and A_2 or B_1 and B_2 of molecule A or B respectively should not overlap. To better differentiate local and global similarities and to save processing time, a minimum number of residues can be defined between the fragments A_1 & A_2 and respectively B_1 & B_2.

ii) corresponding parts of C_α distance matrices, representing intra-molecular C_α distances between atoms within the fragments, should be similar.

$$\sqrt{\frac{\sum_{i=1}^{m_1-1}\sum_{j=0}^{m_2-1}\left(a_{(p1+i,p2+j)}-b_{(f1+i,f2+j)}\right)^2}{m_1.m_2}} \leq r_2 \qquad (2.3)$$

By equation 2.3, a quartet is made by fragments A_1, B_1, A_2 and B_2 where fragment A_1 ranges from a_{p1} to a_{q1}, B_1 from b_{f1} to b_{g1} , A_2 from a_{p2} to a_{q2}, and B_2 from b_{f2} to b_{g2} with fragment lengths of m_1 and m_2:

$m_1 = q1 - p1_1 = g1 - f1 + 1$, and
$m_2 = q2 - p2 + 1 = g2 - f2 + 1.$

The r.m.s deviation of two spatially distant parts of proteins is usually expected to be larger than that of a local one. Therefore, the different values of r_1 and r_2 can be chosen for detection of fragment pairs and quartets respectively.

iii) When proteins are superimposed based on quartets, the r.m.s deviation between fitted pairs should not exceed r_2 to avoid enantiomeric fragments.

2.3 Optimization

2.3.1 Superposition optimization

By this point, we have found several superimposable quartets of fragments. Each of these quartets provides a superposition by corresponding transformation (rotation/translations) and these superpositions can further be optimized based on newly found C_α matches after superposition.

Matching fragments below initially defined minimum length m are not yet recognized. On the other hand, these may represent a large part of the common substructure of the two proteins to be compared if, for example, very similar parts of the proteins are interrupted by insertions and/or deletions. Thus, beginning with a quartet as an initial seed alignment, a new superposition is generated including all C_α atoms that have equivalents below a defined maximum distance r_3 in the superimposed second protein structure. After this additional superposition cycle, it is possible that new C_α atoms get partners within the maximum distance. So this procedure is repeated until the number of C_α matches does not increase further. This process is carried out for all superimposable quartets of fragments and best fragment pairs (with maximum number of C_α matches greater than m). To save computer-processing time the optimization procedure is stopped if less than 25 C_α atoms have equivalents with the reasonable assumption that these quartets/seeds will not give the best superposition. The flow chart of above described algorithm is shown in figure 2.1.

While optimization of a superposition, single lonely found C_α matches are not considered on superposition of two structures because these matches may result by superposition of two secondary structures in completely different orientation. And if such matches are used for further optimization, they might stop the two structures from reaching at most favourable high scoring superposition by jamming it at some intermediate state.

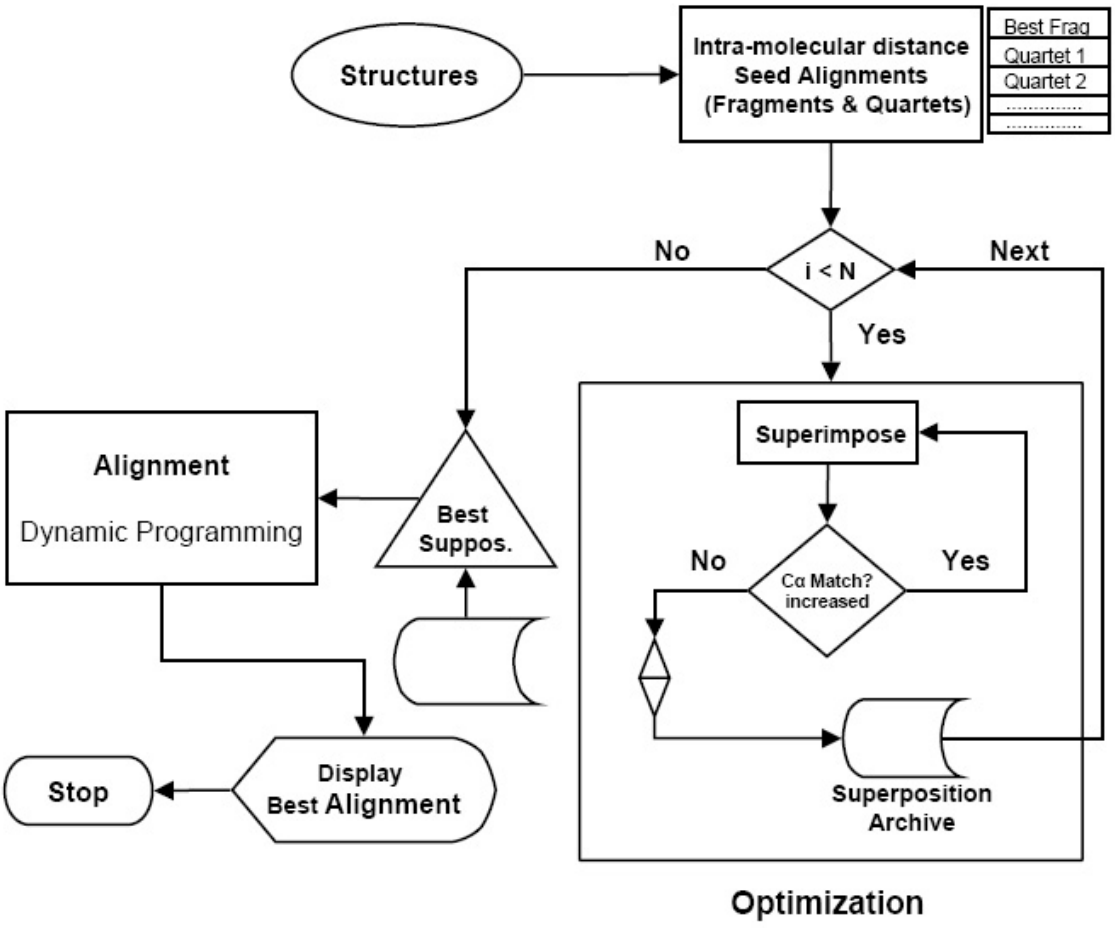


Figure 2.1: Protein3DFit flowchart - old version

2.3.2 Optimization of best superpositions

Now we have a sorted list of optimized superpositions that were initially superimposed based on quartets of fragments. From this list, N best superpositions are selected for second round of optimization. Best superpositions are further improved by alternative iteration between two steps;

i) alignment of equivalent C_α atoms by dynamic programming and

ii) optimization (same step that was performed in first round of optimization) of superposition based on newly aligned C_α atoms.

The iteration continues until there is no further improvement in the quality of alignment. The flow chart of extend algorithm with second round of optimization is shown in figure 2.2.

The alignment of protein structure is desired to meet the contradictory requirements of achieving a lower r.m.s.d. and a higher number of mapped (aligned) C_α atoms. A quality filter Q given in equation 2.4, originally proposed by Krissinel & Henrick in [12], was used as a measure of quality alignment after dynamic programming step to decide whether it should be carried on further or next best superposition should be tried.

$$Q = \frac{N_{align}^2}{((1 + (RMSD/R_0)^2)N_1N_2)} \tag{2.4}$$

Alignment results in mapping of $min(N_1, N_2)$ C_α atoms, where N_1 and N_2 are the number of C_α in the aligned structures. Q score reaches 1.0 only for identical structures as it can be seen from equation 2.4 ($N_{align} = N_1 = N_2$ and RMSD = 0) and decrease with decreasing similarity i.e. by increasing RMSD and/or by decreasing align N . Therefore, higher Q score implies better alignment of structures.

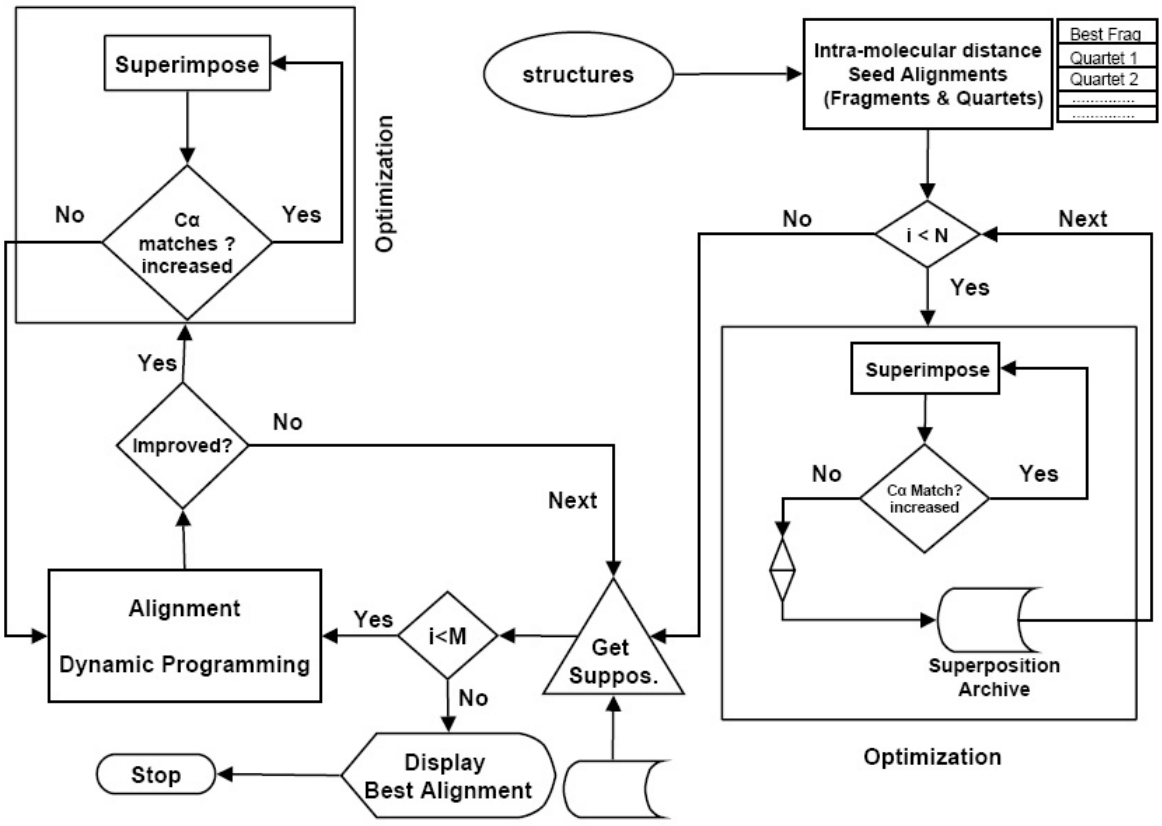


Figure 2.2: Protein3DFit flowchart - extended version

As shown in the flow chart in figure 2.2, during optimization one structure is superimposed onto the other and then all pairwise distances between each C_α atom in the first structure and every atom in the second structure are computed. This results into an inter-molecular distance matrix where each entry d_{ij} corresponds to distance between C_α atom i in the first structure and C_α atom j in the second one. During alignment, this matrix needs to be converted into a similarity matrix S_{ij} , similar to the one used in sequence alignment, by application of the formula in equation 2.5 proposed by Gerstein & Levitt in [5].

$$S_{ij} = \frac{M}{1 + (\frac{d_{ij}}{d_0})^2} \tag{2.5}$$

M is the arbitrarily chosen maximum score of a match and it is 20 in our case. d_0 is the distance at which similarity falls to half of its maximum value. It is taken to be 1.8 A_0 in our case. One applies dynamic programming to this similarity matrix to get optimal equivalences. If this were normal sequence alignment it would be finished at this point but in structural alignment these equivalences are used to superimpose two structures by Diamond method [2].

Chapter 3

Results and Discussions

As it can be seen in figure 2.2 in comparison to figure 2.1 that our basic approach to align two protein structures is still same but an extension to this approach has been made to further improve the alignment between structures. Originally, our algorithm started with a list of seed alignments, called quartets, found on the basis of intra-molecular distances. Each seed alignment was then used one by one to superimpose two structures. After superposition, all the matching C_α atoms were counted and used as seed/anchor points for the succeeding superposition and this iteration initially started by a quartet seed alignment continues until C_α matches were increasing. For best superposition, a binary similarity matrix was built i.e. $S_{ij} = 1$ if an C_α atom in first structure matches to an C_α atom in the second one otherwise $S_{ij} = 0$. And in the end dynamic programming was applied to the binary similarity matrix to find optimal alignment.

As shown in table 3.1, though algorithm was able to produce good alignments of smaller structures with high sequence similarity, it could not find comparable optimal alignments in case of big structures with low sequence similarity.

Table 3.1: Performance of old version of Protein3DFit against 10 difficult alignment test

chain 1	**chain 2**	**Protein3Dfit** $N^A/r.m.s.d.$	**VAST** $N^A/r.m.s.d.$	**Dali** $N^A/r.m.s.d.$	**CE** $N^A/r.m.s.d.$
1FXI:A	1UBQ:_	39/1.0	48/2.1	-	-
1TEN:_	3HHR:B	71/1.0	78/1.6	86/1.9	87/1.9
3HLA:B	2RHE:_	42/1.1	-	63/2.5	85/3.5
2AZA:A	1PAZ:_	52/1.2	74/2.2	-	85/2.9
1CEW:I	1MOL:A	54/1.2	71/1.9	81/2.3	69/1.9
1CID:_	2RHE:_	59/1.1	85/2.2	95/3.3	94/2.7
1CRL:_	1EDE:_	100/1.2	-	211/3.4	187/3.2
2SIM:_	1NSB:A	129/1.2	284/3.8	286/3.8	264/3.0
1BGE:B	2GMF:A	52/1.1	74/2.5	98/3.5	94/4.1
1TIE:_	4FGF:_	70/1.1	82/1.7	108/2.0	116/2.9

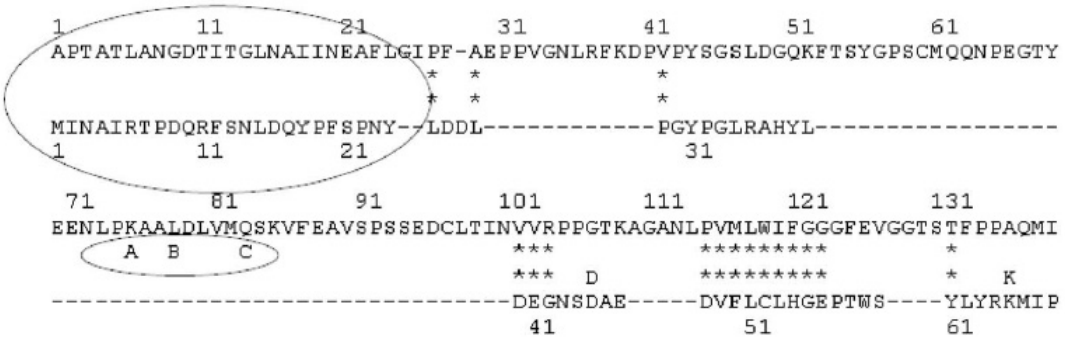


Figure 3.1: An alignment generated by old version of Protein3DFit

The quality of alignment in most of alignments was very poor as an example alignment has been shown in figure 3.1. This example alignment represents a lot of problems which were so common in most of the alignments we investigated. Firstly, the part of structure in the start of alignment, which is not aligned, should have not been made part of alignment. Secondly, randomly occurring alphabetically labelled matches do not make any sense to be aligned when are alone and without any sequence. Actually these matches are by chance matches, which usually result from alignment of two secondary structures in completely different orientation and ultimately result in binding two structures in a position, which prevents convergence to an optimal alignment of two structures. Thirdly, the biggest hindrance in getting quality alignments with comparable number of aligned C_α atoms and root mean square deviation (rmsd) was absence of further optimization of alignment of structures by dynamic programming. Lastly, it is also important to mention that the number, which has been shown in the table 3.1 as rmsd for alignments, is actually not a true rmsd value. It is just average of the distances between only aligned C_α atoms. To get rid of all of these problems to enhance quality of alignment and number of aligned C_α atoms with comparable rmsd, we made few extensions and modifications to the original algorithm, which resulted in very promising results. The results and all the amendments made have been discussed below in detail.

As shown in figure 3.2, the optimization step initiated by a seed alignment (quartet) superimposes (by Diamond method [2]) two structures and tries to reach at a converged optimal superposition of two structures by iteratively superimposing two structures on the basis of newly found C_α matches under a threshold of 2.24 A^0 . At the end of optimization of superposition of two structures, an inter-molecular distance matrix is constructed which contains distance of an C_α atom in first structure from each of C_α atoms in the second one.

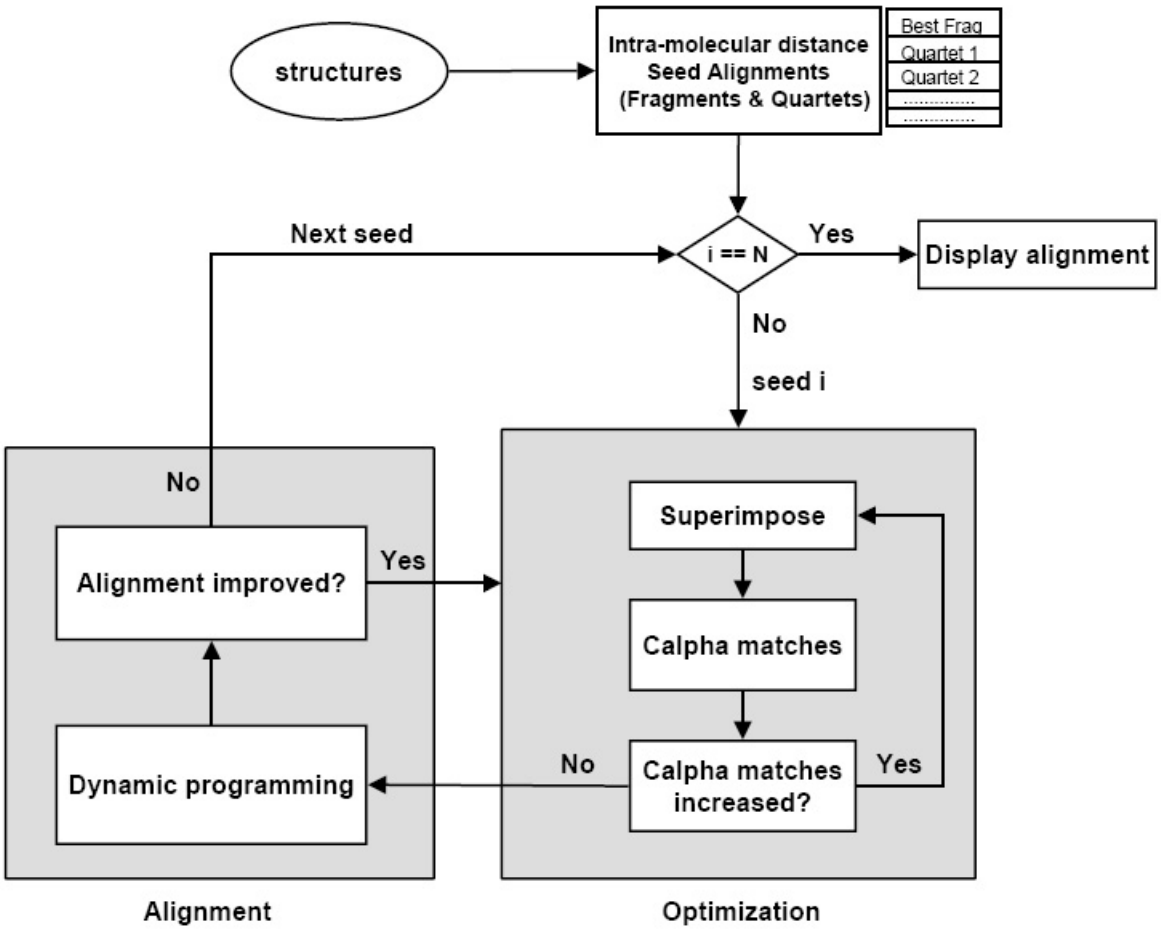


Figure 3.2: Optimization and then refinement by dynamic programming

The inter-molecular distance matrix is converted into a similarity matrix by using formula in equation 2.5, which assigns a score from 0 to 20 depending upon the strength of a match. Then dynamic programming is performed on the similarity matrix to align two optimally superimposed structures. After alignment, quality function given in equation 2.4 evaluates quality of alignment on the basis of rmsd and number of aligned C_α atoms under threshold of 2.90 A^0 . Threshold of alignment step is a little higher than that of optimization due to less chances of getting wrong matches after dynamic programming. Quality function takes into account both contradictory factors of number of aligned residues and rmsd to decide whether alignment is being improved or not. Q function provides good quality measure because most of the time when number of aligned residues is raised rmsd value goes high as well. If quality function says alignment is getting better, two structures are superimposed and optimized on the basis of aligned C_α atoms below 2.9 A^0.

Above described (figure 3.2) interplay between optimization and alignment dynamic programming can not be applied to all the seed alignments because high cost of computation particularly for dynamic programming. The number of seed alignments is usually in hundreds and if this criterion is applied to each seed then computation time to generate an alignment of two protein structures could be in minutes rather than seconds. Therefore, the same scheme to reach at best optimal alignment by jumping between optimization and alignment was applied to the M number of best-optimized superpositions as shown in figure 2.2. The value of M in our case was kept 15 and it resulted in getting best optimal alignment within less than 10 seconds.

Though a huge large-scale benchmark could not be performed due to lack of time, extended version of Protein3DFit was tested against 10 difficult test cases of structures alignment. Results of difficult test case alignments have been shown in table 3.2. The quality of alignment has also been improved a lot without lonely standing single matches labelled with alphabet characters as

Table 3.2: Performance of new version of Protein3DFit against 10 difficult alignment test

chain 1	**chain 2**	**Protein3Dfit** $N^A/r.m.s.d.$	**VAST** $N^A/r.m.s.d.$	**Dali** $N^A/r.m.s.d.$	**CE** $N^A/r.m.s.d.$
1FXI:A	1UBQ:_	59/2.9	48/2.1	-	-
1TEN:_	3HHR:B	82/1.7	78/1.6	86/1.9	87/1.9
3HLA:B	2RHE:_	69/3.4	-	63/2.5	85/3.5
2AZA:A	1PAZ:_	80/2.3	74/2.2	-	85/2.9
1CEW:I	1MOL:A	76/2.1	71/1.9	81/2.3	69/1.9
1CID:_	2RHE:_	91/2.6	85/2.2	95/3.3	94/2.7
1CRL:_	1EDE:_	181/3.2	-	211/3.4	187/3.2
2SIM:_	1NSB:A	265/3.2	284/3.8	286/3.8	264/3.0
1BGE:B	2GMF:A	81/3.0	74/2.5	98/3.5	94/4.1
1TIE:_	4FGF:_	105/2.6	82/1.7	108/2.0	116/2.9

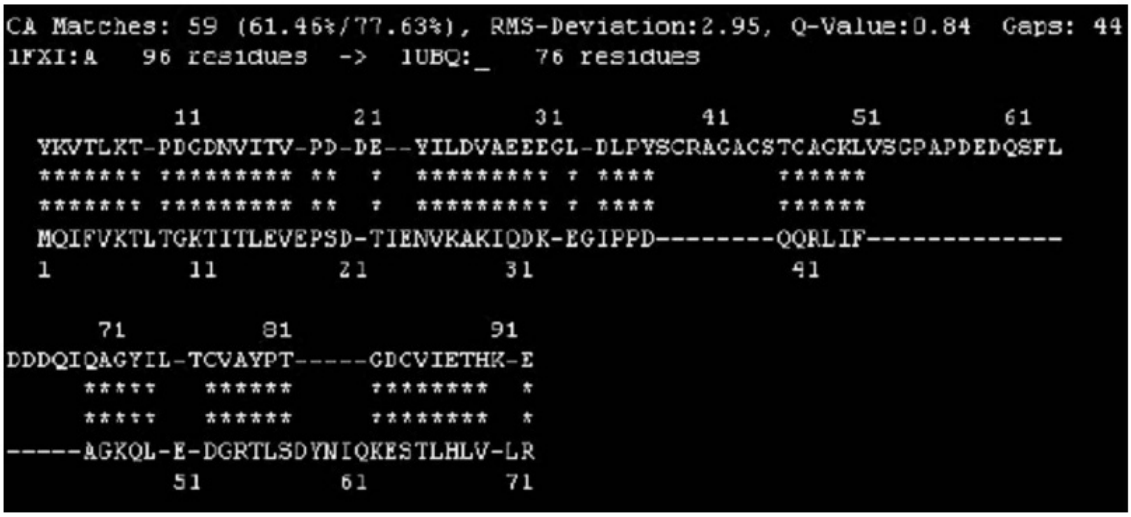


Figure 3.3: An alignment by extend Protein3DFit

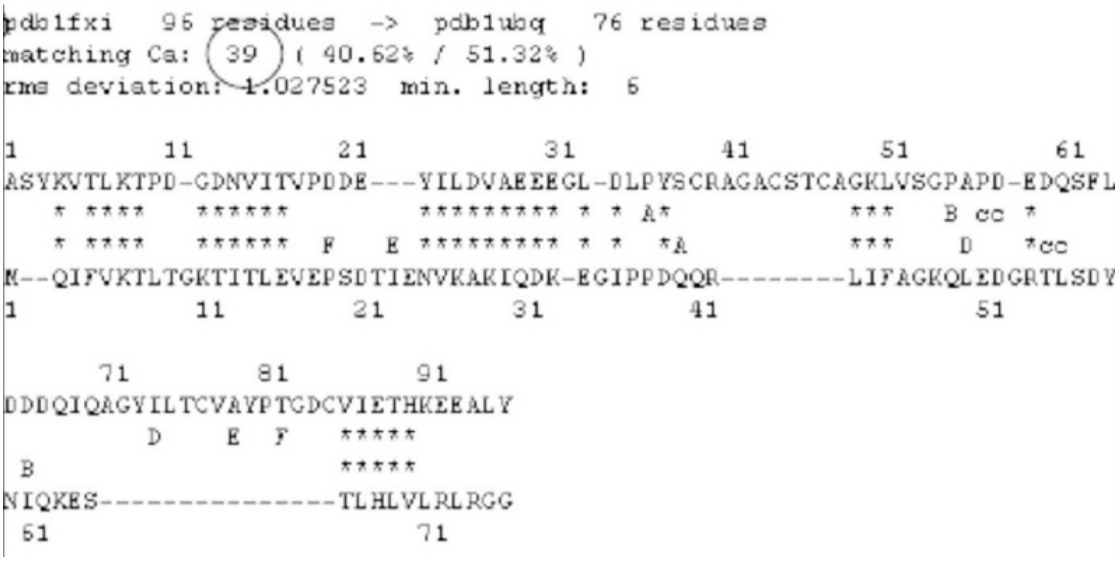


Figure 3.4: An alignment by old version Protein3DFit

an example of same alignment by old and new extend version of Protein3DFit is shown in figure 3.3 and 3.4. Not only the number of aligned residues has been improved but our alignments are in strong agreement with those of Dali and CE but still we have a performance gap of 10-15% to fill. Now rmsd value for alignments is also real rmsd rather than average of distances between aligned residues of two structures.

Bibliography

[1] Godzik A. The structural alignment between two proteins: Is there a unique answer? *Protein Science*, 5:1325–1338, 1996.

[2] R. Diamond. A note on the rotational superposition problem. *Acta. Cryst.*, A44:211–216, 1988.

[3] I. Eidhammer, I. Jonassen, and W.R. Taylor. Structure comparison and structure patterns. *J. Comp. Biol.*, 7:685–716, 2000.

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

[5] M. Gerstein and M. Levitt. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. *M. Proc Fourth Int. Conf on Intell. Sys Mol Biol.*, pages 59–67, 1996.

[6] J.F. Gibrat, T. Madej, and S.H. Bryant. Surprising similarities in structure comparison. current opinion in structural biology. *Current Opinion in Structural Biolog*, 6:377–385, 1996.

[7] H.M. Grindley, P.J. Artymiuk, D.W. Rice, and P.J. Willet. Identification of tertiary structure resemblance in proteins using a maximal common sub-graph isomorphism algorithm. *Mol. Biol.*, 229:707–721, 1993.

[8] L. Holm and C. Sander. The fssp database: fold classification based on structure-structure alignment of proteins. *Nucleic Acids Res.*, 24:206–209, 1996.

[9] G.J. Kleywegt and T.A. Jones. Detecting folding motifs and similarities in protein structures. *Methods Enzymol*, 277:525–545, 1997.

[10] R. Kolodny, P. Koehl, and M. Levitt. Comprehensive evaluation of protein structure alignment methods: Scoring by geometric measures. *J. Mol. Biol.*, 346:1173–1188, 2005.

[11] R. Kolodny and N. Linial. Approximate protein structural alignment in polynomial time. *Proc. Natl. Acad. Sci.*, 101 (33):12201–12206, 2003.

[12] E. Krissinel and K. Henrick. Secondary-structure matching (ssm), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica Section D: Biological Crystallography*, 6(1):2256–2268, 2004.

[13] N. Leibowitz, R. Nussinov, , and H.J Wolfson. Automated method for multiple structure alignment and detection of common motifs: Application to proteins. *J. Comp. Biol.*, 8:93–121, 2001.

[14] U. Lessel and D. Schomburg. Similarities between protein 3-d structures. *Protein Engineering*, 7:1175–1187, 1994.

[15] E. M. Mitchell, P.J. Artymiuk, D.W. Rice, and P.J. Willet. Use of techniques derived from graph theory to compare secondary structure motifs in proteins. *Mol. Biol.*, 212:151–166, 1990.

[16] R. Nussinov and H.J. Wolfson. Efficient detection of three-dimensional motifs in biological macromolecules by computer vision techniques. *Proc. Natl. Acad. Sci. USA*, 88:10495–10499, 1991.

[17] S.T. Rao and M.G. Rossmann. Comparison of super-secondary structures in proteins. *J. Mol. Biol.*, 76:241–256, 1973.

[18] M.G. Rossmann and P. Argos. A comparison of the heme binding pocket in globins and cytochrombe b5. *J. Biol. Chem.*, 250:7523–7523, 1975.

[19] M.G. Rossmann and P. Argos. Exploring structural homology of proteins. *J. Mol. Biol.*, 105:75–96, 1976.

[20] A. Sali and T.L. Blundell. Definition of general topological equivalence in protein structures: A procedure involving comparison of properties and relationships through simulated annealing and dynamic programming. *J. Mol. Biol.*, 212:403–428, 1990.

[21] M. Shatsky, R. Nussinov, and H. Wolfson. Flexprot: an alignment of flexible protein structures. *Proteins: Structure, Function, and Genetics*, 48:242–256, 2002.

[22] A.P. Singh and D.L. Brutlag. Hierarchical protein structure superposition using both secondary structure and atomic representation. *Proc. Intelligent Systems for Molecular Biology ISMB*, pages 284–293, 1997.

[23] J.D. Szustakowski and Z. Weng. Protein structure alignment using a genetic algorithm. *Proteins: Structure, Function, and Genetics*, 38:428–440, 2000.

[24] W. R. Taylor and C. A. Orengo. Protein structure alignment. *J. Mol. Biol.*, 208:1–22, 1989.

[25] B. Wallner and A. Elofsson. All are not equal: A benchmark of different homology modelling programs. *Protein Science*, 14:1315–1327, 2005.

[26] J. Zhu and Z. Weng. A novel protein structure alignment algorithm. *PROTEINS: Structure, Function, and Bioinformatics*, 58:618–627, 2005.

[27] F. Zu-Kang and M.J. Sippl. Optimum superimposition of protein structures: ambiguities and implications. *Folding and Design*, 1:123–132, 1996.