Carnegie Mellon University

CARNEGIE INSTITUTE OF TECHNOLOGY

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF Doctor of Philosophy

TITLE Optimization Models and Algorithms for Protein Structure Alignment

PRESENTED BY Shweta Shah

ACCEPTED BY THE DEPARTMENT OF

Chemical Engineering

N. Sahinidis — NIKOLAOS SAHINIDIS, ADVISOR — 9/23/11 — DATE

A. Lee Gellman — ANDREW J. GELLMAN, DEPARTMENT HEAD — 10/7/11 — DATE

APPROVED BY THE COLLEGE COUNCIL

Vijayakumar Bhagavatula — DEAN — October 14, 2011 — DATE

Optimization Models and Algorithms for Protein Structure Alignment

submitted in partial fulfillment of the requirements

for the degree of

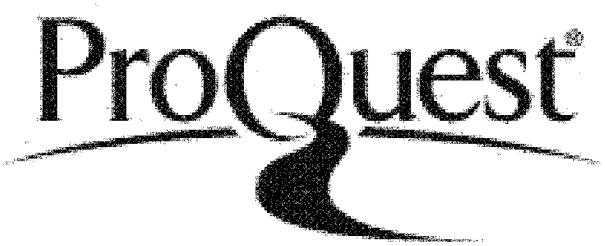
Doctor of Philosophy
in
Chemical Engineering

Shweta Shah

M.Tech., Process Systems Design and Engineering, Indian Institute of Technology, Bombay
B.Tech., Chemical Engineering, Indian Institute of Technology, Bombay

Carnegie Mellon University
Pittsburgh, PA

September, 2011



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Acknowledgements

I would like to express my sincere thanks to Dr. Sahinidis, who has been a great advisor and a wonderful person to work with. His vast knowledge, extreme patience and invaluable feedback have been extremely useful to define and guide my research. I am very much impressed with his ability to maintain a perfect balance between work and family, and have learnt a great deal through him. I am grateful that he supported my family related responsibilities and allowed me to work remotely from home for the past one year.

I would also like to thank my committee members, Dr. Biegler, Dr. Schneider, Dr. Van Hoeve, and Dr. Langmead for providing useful feedback towards my research. They helped me understand my work better and improved the quality of my research. I would also like to thank the Department of Chemical Engineering for supporting me through the course of my research work.

My parents, Dr. Bhupendra Shah and Mrs. Sandhya Shah, have been the greatest inspiration to me all through my educational endeavors. My father always wanted me to be a doctor (of medicine rather than philosophy). I am proud today to fulfill his dream, partially if not completely. My mother has

been a friend, a teacher, a role model to me through all my road-blocks and achievements. I am extremely thankful to both to always support my dreams and make all this possible for me.

My brother, Sameep Shah, has been a great source of encouragement and a person who has always kept me firmly routed to the path towards achieving my goals. I am thankful to him to always keep a check on my progress and not letting me drift at any point.

I am thankful to my husband, Dr. Parag Jain, who has been a great support and a source of inspiration throughout the course of my Ph.D. research. He is responsible for making every depressed evening an inspirational one and every failed effort a new chapter learnt in life.

Finally, I would like to thank all my friends in Pittsburgh, India, and other cities of United States. They have been a source of help, fun and support to me through the course of my Ph.D.

Abstract

Proteins are complex 3D organic compounds formed from amino acid residues. As protein functionality is strongly dependent on structural conformation, understanding structural similarities between proteins helps obtain useful insights in their functional relationships. Specifically, structural similarities often help identify functional relationships that cannot be predicted from sequence similarity alone.

In the past three decades, a variety of computational tools have been developed to address the problem of identifying structural similarities between proteins. These tools identify functionally similar parts of two given proteins by aligning their amino acid residues, i.e., by identifying a correspondence between similar parts of the two protein structures. However, most algorithms for structural alignment provide only approximate rigid sequential alignments between the protein structures under comparison. The incapability of structure alignment tools to provide nonsequential and nonrigid alignments limit their applicability to accurately identify conformation changes within similar structures.

In this thesis, we present two very different approaches to protein structure alignment. First, we revisit the state-of-the-art exact structure alignment algorithm CMOS [XS07] and introduce improved reduction schemes in the CMOS algorithm, resulting in an an over five-fold increase in the computational efficiency of the algorithm. This improvement has increased the applicability of the CMOS algorithm to comparisons between large proteins within reasonable computing times. However, the CMOS algorithm is still insufficient to perform an all-to-all comparison of proteins in the protein structure database and is also limited to sequential structure alignments.

In the second part of the thesis, we introduce a novel model for protein structure alignment that lends itself to an efficient computational procedure for protein structure alignment. The model is based on a reformulation of the traditional approach to structure alignment where alignments are evaluated by rigid structure superposition of the proteins. We develop a new algorithm, SAS-Pro, based on this approach. The new formulation does not require the sequentiality constraints, thus making it possible to discover non-sequential protein alignments and similarities. Alignments obtained with SAS-Pro have better RMSD values and larger lengths than those obtained from other alignment tools. Moreover, for non-sequential alignment problems, SAS-Pro leads to alignments with high degree of similarity with known reference alignments.

In the final part of the thesis, we extend the SAS-Pro model to allow for nonrigid superposition of the proteins structures under comparison. We utilize derivative-free optimization (DFO) methodologies for searching for the

global optimum of the proposed model. We perform an extensive analysis of the performance of 22 different DFO solvers to determine a suitable solution approach for flexible protein structure alignment. Our results indicate that the proposed methodology provides excellent quality alignments for problems where conformational changes are observed.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Proteins are complex 3D organic polymers found in living organisms and are responsible for all bodily functions. They are formed from 20 different amino acids joined together in numerous possible combinations by peptide bonds. The resulting complex 3D structure of the proteins are responsible for their functional properties. Thus, a better understanding of protein structures and structural similarity relationships among them provides information that is critical to function elucidation, fold family classification, and developing homology based inferences about proteins. As a result, knowledge of these similarity relationships has a vast array of applications in a variety of industries, including the drug design and bio-catalysis industries.

Extensive information about known proteins is documented in protein data banks, which keep a detailed record of structural and physical features of proteins. As of July 2011, the UniProtKB/Swiss-Prot database [uni] holds sequences of 531,473 proteins and this number is increasing in size every day

with the addition of newly discovered proteins to the database. Figure 1.1 shows the number of sequences in the UniprotKB/Swissprot database since 1985. The 3D structural information of proteins in stored in the Protein Data Bank (PDB) [pdb]. While there are only over 75,000 protein structures currently in the PDB database, this number is also increasing at a rapid rate, as observed from Figure 1.2. This multitude of data necessitated the development of large number of mathematical modeling and optimization tools for the fast and accurate analysis of protein sequences and structures, especially as they relate to potentially new enzymes and drugs.

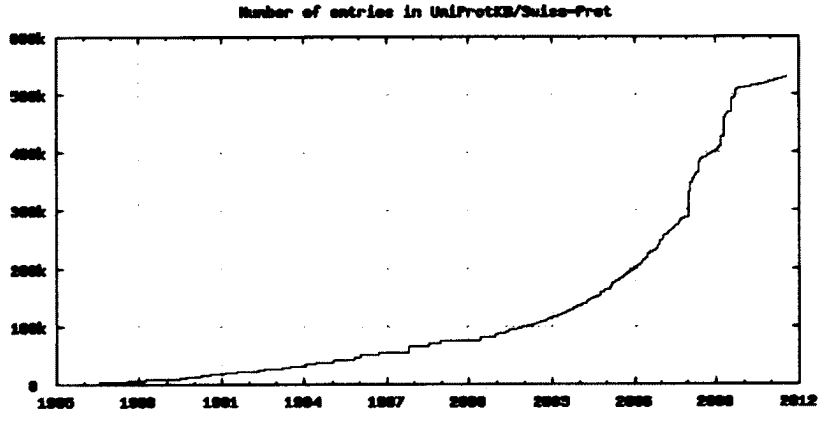


Figure 1.1: Data on the growth in the number of proteins in the Uniprot since 1985. Statistics for the figure are taken from the Uniprot website (http://www.uniprot.org).

Amongst the several unresolved problems in the field of bioinformatics and proteomics, the protein alignment problem has gained tremendous research importance due to its applicability in protein clustering, identifying homology

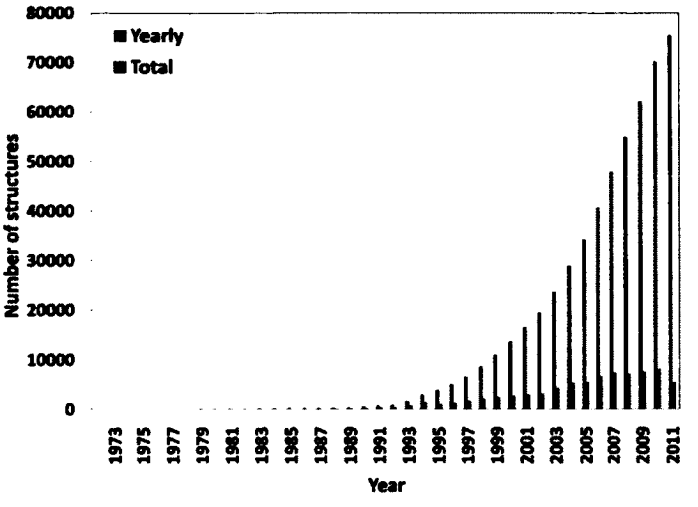


Figure 1.2: Data on the growth in the number of proteins in the PDB since 1976. The total number of proteins is shown in red bars, and the number of proteins per year in blue bars. Statistics for the figure are taken from the PDB website (http://www.pdb.org).

relationships, and inferring unknown information about new and existing proteins. Proteins may be compared with each other through sequence alignment, where the similarities between the proteins are identified through similarities within their amino acid residue sequences. Research on protein sequence alignment has led to the development of numerous dynamic programming algorithms [NW70, SW81] that are central to the BLAST code [AGM90, AS97], an alignment tool that radically transformed the bioinformatics field and found extensive applications in the biotechnology industry. However, structural information of proteins is difficult to infer from sequence information alone. While sequence similarity guarantees structural similarity between proteins,

there exist a large number of protein pairs, *e.g.* haemoglobin and myoglobin found in the human body, that are structurally similar but possess low sequence similarities (a.k.a. twilight zone proteins). Physical comparisons of protein structures [FKR+70, HESF71] further demonstrate the need for direct comparison of 3D protein structures, also known as the protein structure alignment problem, which is the focus of this dissertation.

1.1 Protein structure alignment

The aim of protein structure alignment problem is to determine structural similarities between a given pair of proteins so that further functional relationships between them may be identified. The problem involves determining an assignment of corresponding amino acid residues of the proteins, as well as a suitable measure of the degree of similarity between the two proteins. The protein structure alignment problem is thus formulated as an optimization problem that matches amino acid residues of two proteins in a way that maximizes the degree of structural similarity, as measured by a similarity function, while satisfying certain biological constraints. However, the complex geometry of the 3D structures and the exponential number of potential alignments between the two proteins makes the protein structure alignment problem computationally challenging.

The basic approach to the development of protein structure alignment tools involves a three step process, described in Figure 1.3. First, a suitable math-

ematical representation of the protein structures is determined. Proteins may be represented as graphs, where the residues represent the nodes of the graphs and interactions between them are represented as edges; distance matrices, where the inter-residue distances are represented in a matrix form; secondary structure vectors, where secondary structures are represented by vectors in 3D space; or simply by their 3D co-ordinates. Next, a suitable similarity measure that may be optimized is decided. These similarity criteria, except for contact map overlap (as explained in Chapter 2), are evaluated based on the superimposed structures of the proteins. Finally, based on the mathematical structure representation and the similarity function to be optimized, the protein structure alignment problem is formulated as an optimization problem and addressed by various optimization algorithms, dynamic programming techniques, and heuristic search methods.

The computational complexity of the resulting protein structure alignment problem prohibits the development of fast alignment tools that may provide globally optimal structural alignments. Hence, most structure alignment tools utilize approximate and heuristic methods for fast evaluation of structure alignments. As a result of these approximations, alignment tools may provide inaccurate structural and functional classification of proteins. A few exact structure alignment tools have also been developed to address this structure alignment problem [LCWI01, CCI+04, XS07]. However, they are often computationally expensive and currently not applicable to large scale comparisons of proteins.

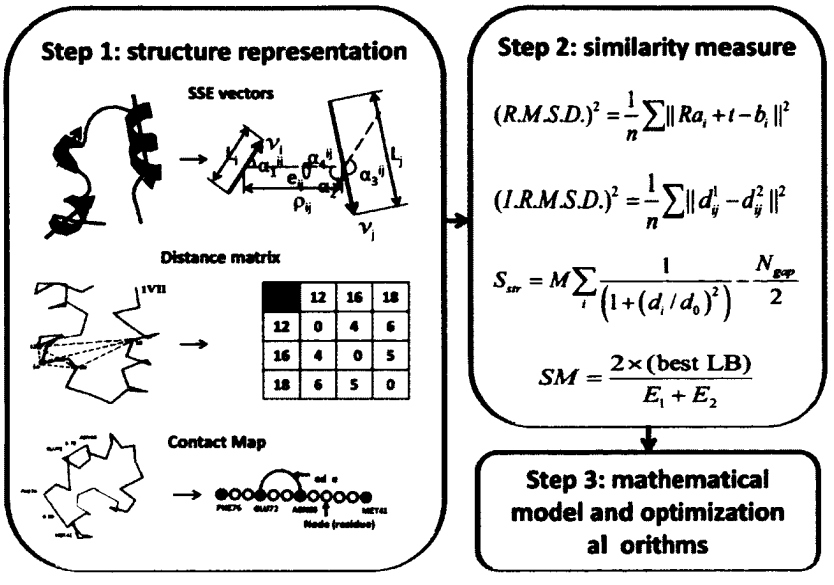


Figure 1.3: Steps in designing a protein structure alignment tool. 1) Protein structure representation, 2) Development of similarity measures, and 3) Problem formulation and development of optimization solution techniques.

In addition to involving computationally complex optimization problems, the structure alignment problem poses additional interesting challenges. Most alignment tools provide good quality alignments for 'similar sized' and structurally similar proteins. However, their performance deteriorates considerably when aligning different sized proteins or those that possess low levels of similarity. Moreover, existing structure alignment algorithms are limited in providing only rigid and sequential alignments between two protein structures under comparison. In many protein pairs, the different parts of protein structures that come together to form a functional unit, may not always occur in the same sequential order in both proteins. Thus, while the proteins present similar structures, they align in a nonsequential manner and the order of the

corresponding amino acid residue sequences is not preserved. Moreover, a large number of proteins present multiple conformational changes, which can be accounted for in structural comparisons only through flexible protein structure alignment. Nonsequential and flexible alignment problems are more complex and present a great opportunity for continued research in this field.

1.2 Research objectives

In this dissertation, we aim at developing fast and accurate structure alignment tools which can provide solutions to some of the challenges of the protein structure alignment problem. Specifically, we have addressed the following issues:

1. Improving the **computational efficiency** of an existing structure alignment tool

2. Developing approaches to obtain optimal/near-optimal **nonsequential structure alignments**

3. Developing approaches to obtain optimal/near-optimal **flexible structure alignments**

In the remainder of the thesis, we begin by presenting improvements to the state-of-the-art exact structure alignment tool CMOS, by improving the computational efficiency of this algorithm. We then introduce a new structure alignment approach that allows for flexibility and nonsequential alignments,

resulting in the development of a new, fast, and accurate structure alignment tool.

1.3 Thesis outline

In Chapter 2 we present a comprehensive literature survey of the structure alignment tools developed in the past three decades. The literature survey provides useful insights in the approaches to protein structure alignment problems. It also demonstrates the difficulties and challenges presented by the structure alignment problem.

In Chapter 3, we revisit the state-of-the-art exact structure alignment algorithm, CMOS [XS07]. The CMOS algorithm is the fastest performing known exact structure alignment tool and is based on a graph representation of protein structures known as contact maps. The protein structure alignment problem is modeled as an integer program, discussed in more detail in Chapter 3. The CMOS algorithm is based on a branch-and-bound approach to this problem, and utilizes many reduction schemes which are mainly responsible for the computational efficiency of the algorithm. In this chapter, we present improvements to the CMOS algorithm through introduction of new reduction schemes, based on physical information from the 3D structure of the proteins. We systematically investigate four different physical properties–hydrogen bonding, hydropathy, torsion angles, and solvent accessibility, and the impact of implementing these biological/physical constraints on the CMOS algorithm.

In Chapter 4, we propose a new alignment tool, SAS-Pro, which is capable of providing flexible nonsequential structure alignments between the protein structures under consideration. The SAS-Pro alignment tool is based on a novel bilevel optimization model for protein structure alignment. We further discuss the implementation of the SAS-Pro algorithm and present computational results for both sequential and nonsequential structure alignment data sets. We also demonstrate the performance of the SAS-Pro alignment tool in comparison with other state-of-the-art structure alignment tools.

In Chapter 5, we enhance the SAS-Pro model and SAS-Pro alignment tool with flexibility variables that allow up to two bends within one of the protein structures before superposition. We perform an extensive analysis of the performance of 22 different derivative-free optimization (DFO) solvers in the context of flexible protein structure alignment. The performance of these solvers is analyzed to determine the most effective techniques suitable for our model, in terms of computational requirements and solution quality. We also discuss the applicability of the SAS-Pro alignment tool with flexibility to similar protein pairs with conformational changes.

Finally, in Chapter 6, we conclude the dissertation, highlight its key contributions to the field of protein structure alignment, and suggest directions for future work.

Chapter 2

Literature survey

The protein structure alignment problem can be formulated as an optimization problem that matches amino acid residues of two proteins in a way that maximizes the degree of structural similarity, as measured by a similarity function, while satisfying certain biological constraints. Several measures of similarity and optimization formulations have been proposed for this purpose. In general, the problem of identifying an optimal structural alignment is known to be NP-hard [GPI99], which refutes the possibility for the existence of an exact algorithm that runs in polynomial time. As a result, many algorithms have been proposed in the quest of developing tools that perform well in practice without necessarily sacrificing solution accuracy. In addition to the similarity function they utilize, these approaches differ primarily in the way they represent protein structure mathematically. Several protein representations emphasize secondary structure elements, bond lengths and angles, relative distances and placements of amino-acid residues in 3D space. Others rely on graphical

representations, such as contact maps. We review and contrast these representations and the algorithmic and software tools that have been developed for protein alignment over the past three decades.

The early protein structure comparisons were based on computing the root mean square deviation (RMSD) amongst two protein structures of known residue correspondence. In order to make such comparisons on a large-scale, McLachan [McL82] and Sippl [Sip82] developed algorithms for fast RMSD computations. These algorithms were then used to construct the first protein structure alignment tools [AF96, ATG92, LKSD00] that were based on determining the optimal correspondence amongst individual residues of two proteins. In the 1990s, several protein structure representations were explored for making fast and accurate alignments leading to the development of tools such as DALI [HS93], CE [SB98], and STRUCTAL [SLL93]. These tools have been instrumental in the development of various protein structure databases like FSSP [HS96], SCOP [MBHC95], CATH [OMJ+97] and HOMSTRAD [MDBO98], which provide extensive information on classification of protein folds and domains.

Protein structure alignment has been the subject of several review papers that present a comprehensive comparison of various structure alignment tools. Gibrat et al. [GMB96] and Lancia et al. [LCWI01] reviewed the alignment tools that focused on specific protein structure representations such as secondary structure based representation and contact map representations respectively. Kolodny [KKL05], Singh and Brutlag [SB01], and Novotny et al.

[NMK04] performed large computational experiments to compare the various alignment tools. Their results indicate that different tools perform with different performance levels for different cases, and thus no single tool is the best. These reviews are systematic and comprehensive, each concentrating on a subset of the algorithms developed for structure alignment. We complement these works by providing coverage of a much larger number of algorithms and techniques, and put the relative strengths and weaknesses of all these approaches into perspective.

We classify protein structural alignment tools based on the protein structure representations they use. Representations based on coordinates, secondary structures, contact maps, and various other elements, are discussed in Sections 2.1, 2.2, 2.3, and 2.4, respectively. In Sections 2.5 and 2.6, we review commonly used similarity measures and protein fold databases. Finally, we conclude in Section 2.7 with a discussion of the current state-of-the-art in the protein structure alignment field and presentation of standing challenges.

2.1 Co-ordinate based protein representation

In co-ordinate based protein structure representation every atom in the protein is denoted via its 3D co-ordinates. The alignment problem is then formulated as a semi-continuous optimization problem where the degrees of freedom are the rotational and translational parameters (continuous) for superposition of the protein structures (figure 2.1), as well as the decision variables (discrete)

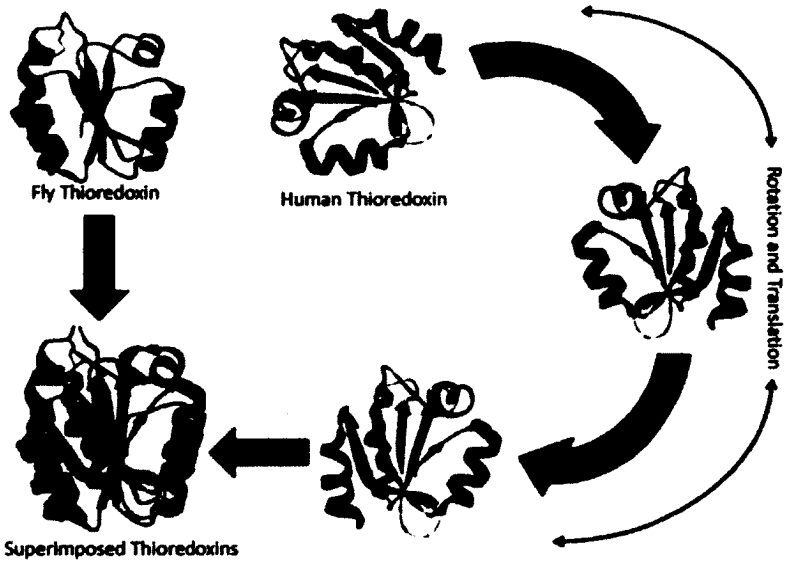


Figure 2.1: Protein structure alignment through structure superposition

representing the correspondence between the amino acid residues. In this section we review the various protein structure alignment tools employing this structural representation.

The co-ordinate based representation has been extensively used in development of early alignment tools which treated proteins as rigid structures without allowing gaps in the alignment. In order to allow for gaps, tools such as SARF [ATG92, AF96] and ProSup [LKSD00] were developed which utilize the rigid body superposition techniques [McL82, Sip82] to align smaller fragments of proteins, which were further joined together to provide a complete alignment.

Using the co-ordinate based representation, the structure alignment problem is often approached in a two stage process where a suitable alignment between the residues of the proteins is first determined and then the pro-

teins are superimposed to employ a similarity function to obtain the merit of alignment. This process is performed iteratively until a desired alignment is obtained. This two stage approach usually provides approximate alignments with no guarantees of optimality, although the alignments obtained for highly similar protein pairs are observed to be close to the optimal solution. Levitt and coworkers [SLL93, GL96] developed the STRUCTAL tool based on dynamic programming methods to obtain an alignment between two proteins and accessed the similarity using the newly developed STRUCTAL score. The alignments obtained through STRUCTAL have since been used as a benchmark for other alignment tools and have been instrumental in the development of the SCOP protein fold database [MBHC95]. Andreani et al. [AMMY08] developed a heuristic method for finding suitable alignments, and used a Gauss-Newton approach to minimize the RMSD. They further improved this alignment tool [AM08] by replacing the heuristic method by a dynamic programming method, and by using the STRUCTAL score as the similarity measure. Their method provides fast and accurate alignments for proteins possessing a high degree of similarity ($> 85\%$), however has limited applicability for proteins with lower similarity. Bhattacharya et al. [BBC06] introduced dynamic programming methods based on novel 'neighborhood preserving projection vectors' obtained from inter-residue distances where the optimal rotation translation parameters are obtained by solving a continuous optimization problem. Ortiz et al. [OSO02] combined the continuous superposition method [McL82] with dynamic programming methods and developed

the MAMMOTH algorithm. This algorithm was the first to be generalized to obtain multiple structure alignments simultaneously [LLMA05].

A few recent co-ordinate representation based approximate algorithms guarantee near-optimal alignments within polynomial time for some special cases of structure alignment problems. Kolodny et al. [KLL04] employed a heuristic procedure where a polynomial number of sample structure superposition transformations are chosen and optimal sequential alignment for each transformation is obtained by dynamic programming. The alignment with the best similarity score provides a near-optimal alignment. Shibuya and coworkers [Shi07, Shi10b, SJS10, Shi10a] constructed a protein structure database search tool providing a set of similar protein structures within a given RMSD value in linear time complexity. Starting with alignments based on rigid protein structures permitting no gaps, they generalized their methodology in a step-wise manner to incorporate small number of gaps, as well as flexible alignments with few or no gaps.

One of the key challenges in structure alignment problem is obtaining solutions that allow for flexibility within protein structures. Some of the recent tools like FlexProt [SNW02], FATCAT [YG03], ProtDeform [RSWD09], and FlexSnap [SZB10] address this issue by aligning smaller rigid fragments of proteins and joining them together, allowing for twists and turns in the overall alignment. The FlexProt [SNW02] tool joins the aligned fragments in a rigid fashion, and introduces a bend whenever RMSD exceeds the desired value. In the FATCAT [YG03] algorithm an upper bound on total allowable bends

is chosen *apriori* heuristically and updated for improving the RMSD value. FlexSnap [SZB10] furthermore generalized the FlexProt and FATCAT tools by introducing non-sequential flexible alignments. In ProtDeform [RSWD09] an alignment based on secondary structure representation of proteins is found and structural flexibility is introduced for obtained the best match.

2.2 Secondary structure based algorithms

Hydrogen bonding within amino acids gives rise to the formation of some commonly occurring structural motifs called secondary structures. These secondary structures also form the building blocks for the functional units of proteins. As a result, there is a growing interest in the development of various structure alignment algorithms based on secondary structures.

The vector alignment search tool (VAST) [MGB95] is the earliest alignment tool based on secondary structures. In VAST an initial alignment of secondary structures is found and then refined through a Monte Carlo technique on the aligned backbone. VAST has been an instrumental structure alignment tool utilized in protein database search and has been recognized as one of the major structure similarity search tools by NCBI.

Singh and Brutlag [SB97] developed a unique method of representing proteins as a set of vectors, one for each secondary structure, as shown in figure 2.2. Each vector extends from the beginning to the end of the secondary structure, and the angles represents the relative placement of the secondary

structures in 3D space. They developed a dynamic programming based algorithm, LOCK, which aligned the secondary structure vectors and further refined the alignment by a heuristic method which minimized the RMSD. This protein structure representation was further employed in the development of the current state-of-the-art alignment tool, Secondary Structure Matching (SSM) [KH04]. In SSM, the initial alignment of secondary structure vectors is determined heuristically, and then refined through a fast 3D superposition technique. SSM is currently the most popular, fast and accurate alignment tool available and is employed in the SCOP database [MBHC95] to evaluate and classify protein similarities.

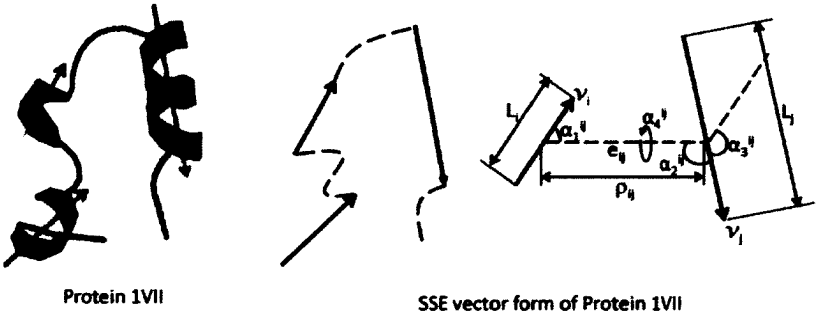


Figure 2.2: Secondary structure vector representation of 1VII protein introduced by Singh and Brutlag [SB97]

Protein structures have also been represented as sequences of secondary structures and aligned by string matching. TM-align alignment tool [ZS05] utilizes protein structures represented as strings of α-helices and β-strands while TOPSCAN [Mar00] utilizes strings of seven different types of secondary structures determined by DSSP [KS83]. TOPS [VG01] and TOPS+ [VGV10]

suggest improvements over TOPSCAN by incorporating information about chirality and hydrogen bonding, and loops and directed connectivity among secondary structures, respectively.

VAST and TM-align have been instrumental in providing new similarity measures for evaluating protein alignment. VAST, like its namesake BLAST, utilizes a unique similarity function which provides a probability value representing the biological relevance of the alignment and provides an appropriate ranking of similarity amongst a database of proteins. TM-score is inspired by the STRUCTAL score [GL96], and provides a weighted RMSD measure incorporating suitable weight for aligned residues and number of gaps. Currently the TM-score is amongst the most commonly used similarity measures and results in obtaining more biologically meaningful alignments.

2.3 Contact maps

First introduced by [GKS93] for visualizing patterns in protein structures, a contact map is a graphical representation of a protein structure, where the nodes of the graph correspond to the amino acid residues of the protein and an edge between two nodes of the graph encodes the interactions between residues. Interactions are modeled by a distance cut-off between the residues in the 3D structure. An example of a contact map is shown in Figure 2.3. Figure 2.3(a) illustrates the construction of a contact map from the 3D structure of part of the backbone of protein 1VII. A cutoff a 7Å was used to create this contact

map. Thus, nodes in the contact map were joined by edges only for those amino acid residues that are within 7Å from each other. This, for instance, was the case for residues ASN68 and GLU72. On the other hand, there is no edge between two nodes of the contact map if the distance between the corresponding residues is larger than 7Å. This, for instance, is the case for residues MET41 and PHE76. The complete contact map of protein 1VII in presented in Figure 2.3(b).

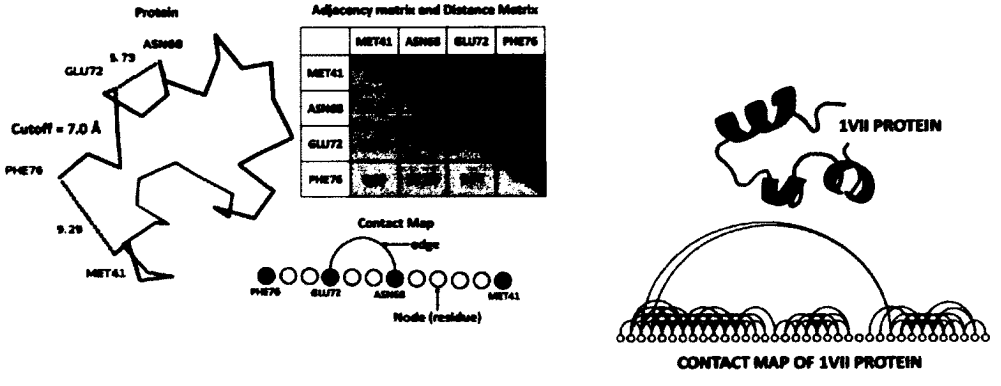


Figure 2.3: (a) Contact map generation, and (b) Contact map for 1VII protein

The contact map encodes information about the entire structure of the protein. In comparison to coordinates-based representation of proteins, contact maps provide a more robust representation since the map does not change considerably with small perturbations within the structure. In addition, the corresponding alignment problem formulation is independent of the number of gaps between aligned residues, and provides the flexibility of introducing non-rigidity in the protein structure.

The contact maps of two proteins may be compared to identify structural

similarity between the corresponding 3D protein structures. Similar subgraphs of the contact maps imply similar sub-structures of their corresponding protein structures. Thus, viewed through contact maps, the protein alignment problem becomes equivalent to finding a maximum-sized common subgraph among the two contact maps. In addition, it is customary to require the alignment to maintain the sequential order of amino acid residues. The problem of finding the maximum contact map overlap is referred to as MAX-CMO. The MAX-CMO formulation of protein structure alignment was shown to be NP-hard by Goldman [GPI99]. Hence, exact algorithms designed to solve the protein structure alignment problem are often computationally expensive.

Figure 2.4 illustrates an alignment between two contact maps A and B. Here, a dotted line between a node of contact map A and a node of contact map B denotes a pair of aligned residues. Lines l and m in the figure are examples of such an alignment. The binary variables x_l and x_m represent the decision variables for lines l and m. Here, the binary variable x_l corresponding to line l takes the value of 1 if the residues at the ends of line l are aligned.

The MAX-CMO formulation requires finding a set of non-intersecting such lines which maximizes the number of overlapping edges. These edges are shown in thick lines in Figure 2.4. The parameter e_{lm} equals 1 if there exists an overlapping edge in both contact maps for nodes aligned by lines l and m, and 0 otherwise. The condition of non-intersecting lines is imposed to maintain the sequential order of amino acid residues in the alignment. The mathematical model that stems from this formulation is a nonlinear integer

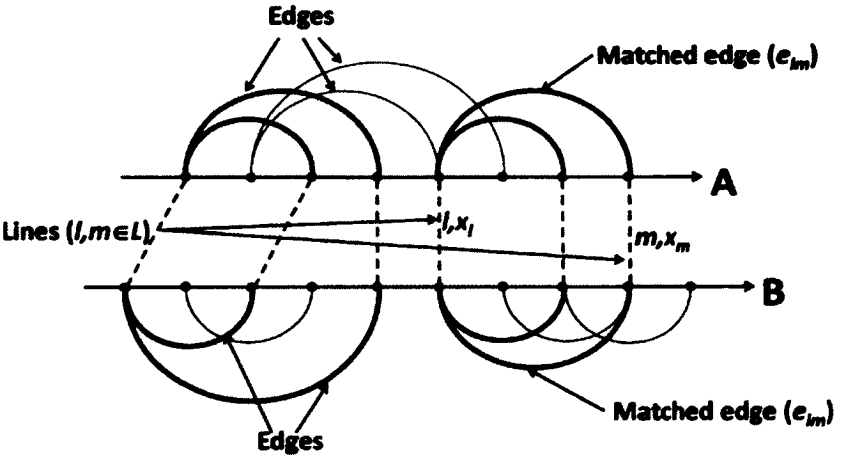


Figure 2.4: A schematic of contact map overlap problem

program, first suggested by Lancia and co-workers [CLI00], and is given as:

$$
\begin{aligned}
(\mathrm{IP}-\mathrm{I}) \qquad \max \quad & \sum_{l \in L} \sum_{m \in L} e_{lm} x_l x_m & & \\
\text{s.t.} \quad & \sum_{l \in I} x_l \leqslant 1, & \forall I \in \Gamma \qquad & \text{Clique Inequalities} \\
& x_l \in \{0,1\}, & \forall l \in L, &
\end{aligned}
$$

where L is the set of lines, I is the set of incompatible, i.e. , intersecting lines, and Γ is the set of all sets of incompatible lines. The constraints for maintaining the order of the amino acid residues in the alignment are known as clique inequalities. These inequalities arise from a graph representation of MAX-CMO. Every node in this graph represents a line in the MAX-CMO problem, and every edge in the graph represents a pair of interesting lines. The condition of no intersecting lines in MAX-CMO translates to having no

cliques in the graph.

The nonlinear integer program (IP – I) can be converted to a linear integer program through the addition of variables $y_{lm} = x_l x_m$ to obtain:

$$
\begin{aligned}
\text{(IP – II)} \qquad \max \quad & \sum_{l \in L} \sum_{m \in L} e_{lm} y_{lm} \\
\text{s.t.} \quad & \sum_{l \in I} x_l \leqslant 1, \qquad \forall I \in \Gamma \\
& y_{lm} \leqslant x_l, \qquad \forall l, m \in L \\
& y_{lm} = y_{ml}, \qquad \forall l, m \in L, l < m \\
& x_l \in \{0, 1\}, \qquad \forall l \in L
\end{aligned}
$$

The continuous relaxation of the integer program does not provide any useful information. The relaxation is very weak and the solution suggests that every possible alignment between the proteins is equally likely. Thus enumeration methods based on branch-and-bound were developed for obtaining exact solutions to the problem. The order preserving property involves the addition of exponentially many constraints, known as the clique inequalities [CLI00] to the integer programming formulation. Carr et al. [CLI00, LCWI01] suggested a solution to deal with this exponential number of inequalities through an iterative method of adding only the most violated inequality as a cut at every step of a branch-and-cut algorithm. In a more recent work [CL04], the same group also suggested the use of a more compact formulation, which provides faster solutions than the branch-and-cut algorithm. The same group developed different bounding heuristics for the branch-and-cut algorithm. The ini-

tial heuristics were based on a genetic algorithm. The group further improved these bounding heuristics by suggesting a lower bound obtained by solving a Lagrangian relaxation of the integer program [CL02, CCI+04]. In 2006, a branch-and-reduce algorithm was developed by Xie and Sahinidis [XS06] for MAX-CMO. The reduction and bounding parts of this algorithm are based on dynamic programming techniques and are largely responsible for the speed of the overall approach. This algorithm was further improved in 2007, via the addition of stronger reduction schemes based on optimality arguments. This algorithm currently represents the state-of-the-art exact algorithm for MAX-CMO in the sense that it is at least an order of magnitude faster than other exact algorithms for this problem [16, 12, 65] on a large collection of test. In addition, this algorithm was able to obtain the global optimum for some previously unsolved large protein structure alignments.

Strickland et al. [SBS05] expressed the MAX-CMO problem as a graph theory problem of finding a maximum cardinality clique (MAX-CLIQUE) in a graph of size $|E_1 \times E_2|$. The MAX-CLIQUE problem was solved exactly using coloring techniques that exploit the special structure of the graph. Also exploiting MAX-CLIQUE, Pullan [Pul07] came up with a local search technique for finding an approximate solution. This method was shown to be faster than the method proposed by Strickland et al. [SBS05] by an order of magnitude while providing the true global optimum in most cases.

The MAX-clique formulation has the drawback of creating very large graphs, even for small proteins. As a result, it requires a large amount of memory to

store the graph structure and cannot be used for large problems. This drawback has been addressed by Melvin et al. [MST09] where a new data-structure has been developed to store the graph. This development has resulted in the ability of solving much larger problems using the MAX-CLIQUE formulation.

The MAX-CLIQUE algorithms, however, are not as fast as other exact algorithms like the branch-and-reduce algorithm developed by Xie and Sahinidis [XS06, XS07], since the reduction and bounding schemes developed by Xie and Sahinidis are much faster and efficient at finding the solution.

In addition to exact algorithms for contact map overlap maximization, there have been efforts to develop faster approximate solution procedures. Godzik et al. [GSK93] made the very first attempt to use contact maps for protein structure alignment using a Monte-Carlo based simulated annealing approach. This was further generalized to multiple structure/sequence alignment by the same group [GS94]. Gramm [Gra04] proposed a poly-time algorithm for a special case of contact maps, namely 2-interval sets. The suggested algorithm may be used as a building block to more general MAX-CMO algorithms. Another approximate poly-time algorithm for MAX-CMO was developed by Xu et al. [XJB07], where a suitable alignment of protein contact maps is found using a tree-decomposition algorithm derived from discretizing the rotation angles. Jain and Lappe [JL07] used a continuous optimization algorithm for solving the maximum common subgraph problem and then converted the solution into a MAX-CMO solution using a dynamic programming approach. The resulting solution is approximate but the algorithm is fast and

provides good solutions. Pelta et al. [PGV08] developed a multi-start variable neighbor search metaheuristic, which provided good approximate alignments of the contact maps, although at the expense of more CPU time than other alignment heuristics and algorithms.

In order to provide a quick estimate of the level of similarity between two proteins, the universal similarity metric (USM) [KP04] was developed in 2004. USM arises from the application of Kolmogorov complexity to protein structures, described in terms of contact maps. The Kolmogorov complexity gives a measure of the information stored in an object, and can be used to derive a measure of the information distance between two objects. The smaller the information distance, the more similar the two objects are. USM can be used to estimate the similarity between two proteins without actually aligning them. This measure can be used efficiently as a pre-processing technique before exact algorithms are invoked.

The USM measure was successfully applied to a more general class of contact maps, known as the fuzzy contact maps. Fuzzy contact maps [PGK05, PKBC$^+$05] were introduced to capture short as well as long distance relationships, through the use of multiple thresholds to define contact maps. Pelta et al. [PGK05] used USM to evaluate the alignment obtained for fuzzy CMOs and observed very good agreement with the clustering observed in nature. They also demonstrated in [PKBC$^+$05] that a simple neighbor search heuristic provided very good clustering for the Chew-Kedem and the Skolnick data sets.

There have been several efforts to obtain approximate solutions to the maximum contact map overlap using memetic evolution and genetic algorithms (GAs). The idea is to use traditional GA operators like crossover and mutation rules, accompanied by a local search. Krasnogor [Kra04], Carr et al. [CHK+02] and Lancia et al. [LCWI01] use traditional GA moves along with some modifications. Lancia et al. [LCWI01] use GAs to find an approximate solution to be used as feasible solution in their branch-and-cut algorithm. Carr et al. [CHK+02] associate a local search with every instance of the problem, and pass on the local search to the new generation during cross-over. Krasnogor [Kra04] suggested an improvement with some added processes of imitation, innovation, and mental simulation. The order in which meme processes are applied is also different. However, the approximate solutions obtained through memetic evolution are inferior to other approximate solution algorithms including LGA and Lagrangian relaxation algorithms. Kolbeck et al. [KMSG+06] used a hybrid alignment technique combining secondary structures and contact maps. At the first level the secondary structures of proteins were aligned using GA and at the second level the corresponding contact map alignment was refined. The algorithm was further improved [GK08] by using a combinatorial technique instead of the GA, at the first level.

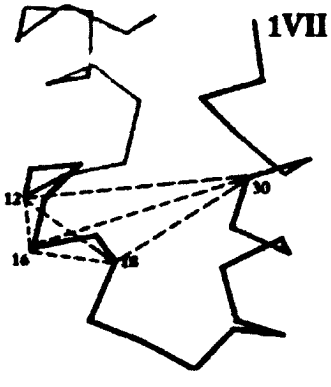
2.4 Miscellaneous structure representation models

Dynamic programming has been the key to the development of fast and accurate sequence alignment algorithms. In the case of structure alignment problems, any good alignment provides useful information for obtaining the most meaningful structure alignment. Since dynamic programming techniques are fast and useful for problems of sequential nature, they can be used efficiently to obtain good alignments.

Proteins structures have been represented as sequences of objects other than amino-acid residues and then aligned using dynamic programming techniques. Sali and Blundell [SB90] developed the comparer alignment tool based on a sequence representation of the structure, as a weighted sum of alignments based on primary and secondary structure sequence representations and physical properties such as hydrophobicity, hydrogen bond orientation, dihedral angles etc. In the tool VeaR developed by Leluk et al. [LKR03] the protein structure is represented as sequences of dihedral angles and radius of curvature of five residue long fragments. Ye et al. [YJL05] use a geometric representation which captures orientation of the corresponding amino acid residues. Wu et al. [WSHB98] use dynamic programming on proteins represented as sequences of radius of curvature of smaller fragments.

Taylor et al. [TO89a] developed the SSAP algorithm which involves a bi-level dynamic program. For this algorithm, every residue is represented as

sequence of vectors pointing at other residues. Dynamic programming is used first to determine the correspondence and then the alignment of the amino acid residues. The algorithm was further improved [TO89b] to include physical properties of protein structures like hydrogen bonding, hydrophobicity etc. Further to expand the scope of the problem to multiple structure alignment, Taylor et al. [TFO94a] suggested the integration of MULTAL, a multiple sequence alignment tool [TFO94b], and the SSAP method of structure alignment [TO89a, TO89b], to obtain better multiple sequence alignments of proteins.



Distance Matrix for 1VII protein

Residue Number

Residue Number	12	16	18	30
12	0	4.51	6.45	12.56
16	4.51	0	5.37	12.36
18	6.45	5.37	0	9.21
30	12.56	12.36	9.21	0

All distances are in Å

Figure 2.5: Distance matrix representation of protein 1VII

DALI and CE are amongst the oldest and most popular structure alignment tools. Holm and Sanders [HS93] developed DALI which uses the distance matrix representation of proteins, which is a matrix of inter-residue distances of the protein, as depicted in figure 2.5. The algorithm finds smaller submatrices of the distance matrix that are similar to each other, and then the alignment obtained is further improved using the Monte Carlo method. The results obtained by DALI have since become the benchmark for structure

alignment. An improved version of DALI, known as DaliLite [HP00], is now available for structure alignment. Combinatorial Extension (CE) developed by Shindyalov and Bourne [SB98] also uses distance matrix representation of proteins. Various distance comparison heuristics have been employed to choose among the combinatorial number of possible alignments of the proteins. CE is also used as a benchmark for protein structure alignments. Both these tools are currently employed as structure comparison tools for the FSSP protein fold database [HS96]. More recently, Tai et al. [TVKL09] developed a tool termed as seed extension (SE) based on distance matrix representation where they identify potential small alignments which are extended further using dynamic programming to provide the final alignment.

A variety of novel protein representations are used for developing tools focusing on different geometric features of protein structures. Falicov etc *al.* [FC96] create a triangulation between two proteins, where the residues are the vertices of the triangles. For two proteins to be similar, they must superimpose perfectly on each other, thus making the area of the triangulation zero. The algorithm developed by Falicov et al. focuses on obtaining a minimum area triangulation between the two proteins. TOPFIT [IAL04] is based on vornoi diagram representation of protein structures. Initially the tessellation of the protein structure is found and the tetrahedra are classified as per volume and size. Then the tetrahedra are matched to obtain initial alignment, which is then expanded through addition of neighboring tetrahedral matches. Zhao et al. [ZFAS08] suggested an algorithm, SLIPSA, based on feedback for structure

alignment. They use the idea of stitching local alignments together to get a global alignment, which is then provided as a starting point to the algorithm, as a feedback. The algorithm uses a graphical representation of proteins. Yang and Tung [YT06] developed a structural alphabet for proteins using the angles between the C-alpha atoms and then used dynamic programming to align the structural alphabet sequence.

2.5 Similarity Metrics

The quality of alignment, or means to measure the quality of alignment, is very important to evaluate the true significance of the similarity found between proteins. Every alignment tool has a different metric to measure similarity, giving different degree of importance to four main factors, namely, (a) RMSD, (b) length of alignment, (c) size of the proteins compared, and (d) number of gaps. Every similarity measure is a function of these four quantities. Most similarity measures try to minimize the RMSD and number of gaps in the alignment, while maximizing the length of alignment as fraction of size of proteins being compared. It is not clear which factor is important to what degree and every measure gives different weights to these factors. This has resulted in multiple similarity metrics with no clear consensus on the best similarity measure.

Amongst the wide variety of similarity metrics the more commonly used ones are geometric measures such as the RMSD, wRMSD [WSHB98], SI score

[KJ94], SAS score [SLL93], the LG score [LG98], and TM-score [ZS05]. For special protein representations, special similarity metrics such as contact map overlap, and the VAST p-score are also used, though they are not as common. We use the RMSD, SI, SAS and contact map overlap similarity measures through the course of this thesis to evaluate the quality of the alignments and compare different alignment tools. The equations below provide the definitions for some of the similarity measures utilized in protein structure alignment.

SIMILARITY MEASURES

$$\begin{aligned}
\textbf{RMSD} &= \sqrt{\sum_i \sum_j S_{ij} ||\theta(r(a_i)) - r(b_j)||^2 / \sum_i \sum_j S_{ij}} \\
\textbf{wRMSD} &= \sqrt{\sum_i \sum_j w_{ij} * S_{ij} ||\theta(r(a_i)) - r(b_j)||^2 / \sum_i \sum_j S_{ij}} \\
\textbf{SI} &= RMSD * \frac{min(L_1, L_2)}{N_{align}} \\
\textbf{SAS} &= RMSD * \frac{100}{N_{align}} \\
\textbf{LG-score} &= M \sum_i \sum_j \frac{S_{ij}}{(1 + (d_{ij}/d_0)^2)} - \frac{N_{gap}}{2} \\
\textbf{CM overlap} &= \frac{2 * (LB)}{E_1 + E_2} \\
\textbf{TM-score} &= MAX \frac{1}{P_1} \sum_i \sum_j \frac{S_{ij}}{(1 + (d_{ij}/d_0)^2)}
\end{aligned}$$

Here, a_i represents the i^{th} residue of protein A, and b_j represent the j^{th} residue of protein B. $r(a_i)$ and $r(b_j)$ represent the 3D coordinates of the corresponding amino-acid residues. S_{ij} is a binary variable that equals 1 when a_i is aligned to b_j and 0 otherwise. θ represents the rotation-translation transfor-

mation applied to protein A. d_{ij} represents the distance between the i^{th} residue of protein A and j^{th} residue of protein B. P_1 represents the size of protein A. E_1 and E_2 represent the sizes of contact maps A and B. N_{align} represents the number of aligned residues. d_0 and M are parameters.

2.6 Databases

There have been some efforts to classify the existing proteins into fold families. However, due to the limitation of fast structure alignment tools, not many folds are identified and only a small percentage of proteins are actually classified. Some of the popular databases storing the classification of these proteins are the CATH database [OMJ+97], the SCOP database [MBHC95], the FSSP database [HS96], and the HOMSTRAD database [MDBO98]. As of September 1, 2011, the CATH database has classified proteins into 1282 fold families (also known as topologies), with over hundred thousand domains.

Barthel et al. [BHB+07] developed a web-based server called ProCKSI, which makes an intelligent choice of a suitable alignment tool to align two proteins from a set of different alignment tools based on a variety of alignment measures. ProCKSI has provided a common platform for structure alignment tools by integrating various tools and similarity measures on a single server.

2.7 Conclusions

A large number of structure alignment tools have been developed in the past three decades, based on very different and innovative approaches to the alignment problem. Most algorithms provide good quality alignments for protein pairs with high amount of structural similarity, that are in agreement with the clustering for proteins in the SCOP and CATH databases. The increasing size of the protein database further emphasizes the importance to obtain fast alignment tools to do an all-to-all comparison of proteins in the PDB. Amongst inexact tools the SSM algorithm is the state-of-the-art since it is quite fast and provides biologically relevant approximate alignments. Amongst exact alignment tools based on contact map formulations, the CMOS tool provides quick alignments with guarantees of global optimality.

Even though structures with similar sizes and with high amount of similarity are easier to compare, the challenge for most approaches remains to identify similarity between similar structures of different sizes and between structures with medium level of similarity. Also most algorithms are able to deal with rigid sequential alignments with few or no gaps. The problem of fast nonrigid and nonsequential alignment of proteins is still a challenge. For sequential comparison, dynamic programming techniques have proven to be quite effective, while for non-sequential alignment continuous alignment methods seem to be promising.

In the remaining dissertation, we provide computational solutions to ad-

dress some of the limitations of current alignment tools. As a result, we improve existing and provide new freely usable structure alignment software.

Chapter 3

Exploiting physical information in the CMOS algorithm

Among the many factors affecting protein conformation are physical factors, such as hydrogen bonding, torsion angles, hydropathy, and ionic interactions. Some of these factors, such as hydrogen bonding, are easily identified from the 3D protein structure, while others, such as long-range ionic interactions, are difficult to infer from the structure. Moreover, the contribution of each of these factors towards shaping and imparting function to a protein has not been not quantified. Due to these reasons, these factors have not been used effectively in protein structure alignment tools. The only exception is hydrogen bonding information, which is utilized through secondary structures that are commonly observed in 3D protein structures.

As discussed in Chapter 2, secondary structure information is utilized by many alignment tools, including VAST [MGB95], TM-align [ZS05], and SSM

[KH04]. These tools are based on representations of secondary structures, such as a sequence of secondary structure types [MGB95, ZS05], or vector representations [SB97, KH04]. Amongst these alignment tools, SSM is the most computationally efficient. SSM provides better quality alignments than other alignment tools, such as CE, DALI, and Structal, that do not utilize any physical property information (cf. [KKL05]). Thus, exploitation of physical properties may be very significant in determining structural features of proteins and improving the performance of structure alignment tools.

The CMOS algorithm [XS07] was shown to be up to an order of magnitude faster than other exact algorithms. In addition, this algorithm provided exact alignments for some previously unsolved structure alignment problems. The CMOS algorithm has been used extensively as a benchmark for structure alignment algorithms [JO09, LFM+10, SHL10, WDK10]. In addition, the similarity measure utilized by CMOS has been used in other bioinformatics applications, such as protein model evaluation [MVR+10]. The computational efficiency of the CMOS algorithm results from a variety of reduction schemes that are employed by the algorithm in order to reduce the size of the search space. These reduction schemes utilize only geometric constraints imposed by the structure of the proteins, and are not dependent directly on any physical properties of the proteins under comparison.

We propose improvements to the CMOS algorithm through exploitation of physical property information of the proteins under comparison. While the mere incorporation of physical properties in search space reduction schemes

in any structural alignment tool is an obvious way to induce computational benefits, it gives rise to several interesting questions. In particular, one must then determine how to go about quantifying physical properties and how to use quantitative measures of properties to perform alignment space reductions in a way that does not eliminate biologically meaningful solutions. We have performed systematic computations to answer these questions in the context of CMOS. Our computational results demonstrate that exploiting physical properties in CMOS results in a five-fold reduction in the computational requirements of the CMOS algorithm. Furthermore, this increased efficiency increases the applicability of CMOS to larger structure alignment problems, which were previously unsolvable by this algorithm. We first present a brief description of the CMOS algorithm in Section 3.1. In Section 3.2, we present the biological significance of the various physical properties that we have considered and discuss models that relate these physical properties to structural information. Finally, in Sections 3.3 through 3.5, we present a detailed analysis of the role of these physical properties in improving the performance of CMOS, making it more viable for large structure alignments.

3.1 Protein structure alignment and the CMOS algorithm

Protein structure alignment tools are based on a variety of mathematical formulations derived from different representations of the 3D structures of proteins. Currently, the contact map protein representation provides an excellent framework for formulating sequential non-rigid structure alignment problems. Most exact structure alignment tools developed in the past are based on the contact map representation of protein structures.

The mathematical formulation of the protein structure alignment problem using the contact maps is also known as the MAX-CMO formulation. Figure 2.4 illustrates a schematics of the MAX-CMO formulation through an alignment between two contact maps A and B. Here, a dotted line between a node of contact map A and a node of contact map B denotes a pair of aligned residues. Lines l and m in the figure are examples of such an alignment. The binary variables x_l and x_m represent the decision variables for lines l and m. Here, the binary variable x_l corresponding to line l takes the value of 1 if the residues at the ends of line l are aligned.

The MAX-CMO formulation requires finding a set of non-intersecting such lines which maximizes the number of overlapping edges. These edges are shown in thick lines in Figure 2.4. The parameter e_{lm} equals 1 if there exists an overlapping edge in both contact maps for nodes aligned by lines l and m, and 0 otherwise. The condition of non-intersecting lines is imposed to

maintain the sequential order of amino acid residues in the alignment.

Xie and Sahinidis [XS07] developed the branch-and-reduce algorithm, CMOS, which provides an exact solution for the MAX-CMO formulation. These authors demonstrated the CMOS algorithm to be an order of magnitude faster than prior exact algorithms for MAX-CMO. In addition, this algorithm obtained the global optimum for some previously unsolved large protein structure alignment problems.

The CMOS algorithm is based on a branch-and-bound approach. The reduction and bounding schemes of this algorithm are based on dynamic programming and are largely responsible for the speed of the overall approach. Hence, the algorithm is also referred to as a branch-and-reduce algorithm.

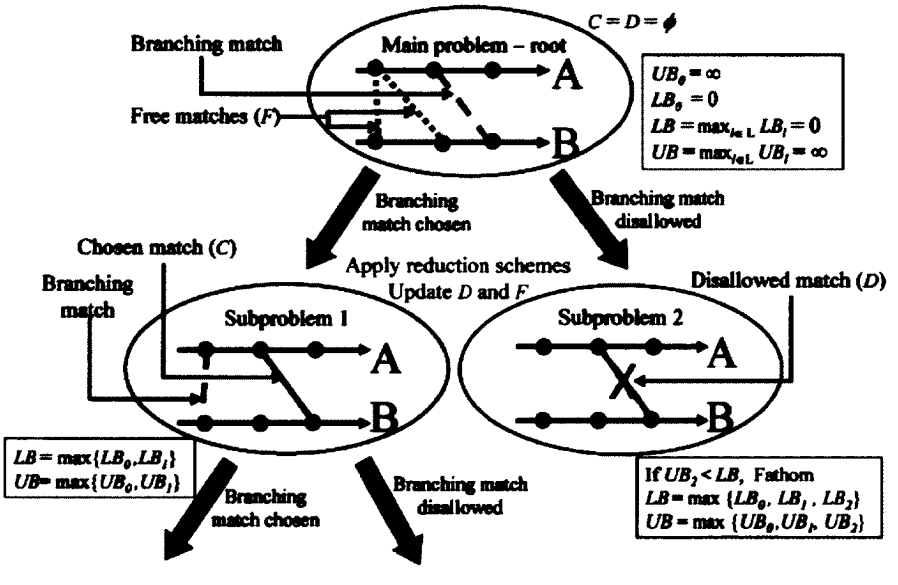


Figure 3.1: Schematic of the branch-and-reduce tree generated by CMOS

The search tree of CMOS algorithm is illustrated in Figure 3.1. Each

node of the CMOS branch-and-reduce tree corresponds to a subproblem of the MAX-CMO problem. The algorithm dynamically generates this tree and processes one node of the tree in every iteration. For each node/subproblem, three sets of lines are evaluated: the set C includes lines corresponding to fixed amino acid residue alignments for the subproblem, the set D includes lines corresponding to disallowed amino acid residue alignments for that subdomain of the problem, and the set F includes lines representing potential amino acid alignments, about which no conclusion is made yet. Alignments from the set F act as variables of the subproblem. Reduction schemes which enforce geometrical and mathematical constraints of the problem are employed for reducing the size of F in every node of the tree. Upper and lower bounds for the objective, i.e. , the number of aligned edges, are calculated for each subproblem. The maximum (best) lower bound of the branch-and-reduce tree is updated in every iteration. Dynamic programming methods are used for lower and upper bound calculations at each node. Inferior nodes, i.e. , subproblems with an upper bound that does not exceed the maximum lower bound of the tree, are pruned. At every node, a potential alignment, referred to as the branching alignment, is chosen from set F, and the node is branched (partitioned) into two descendant nodes, one where the branching alignment is enforced (adding branching alignment to set C), and the other where the branching alignment is disallowed (adding branching alignment to set D). The algorithm terminates when the maximum lower bound equals the maximum upper bound of the tree. The alignment which provides the maximum lower bound is then declared as

an optimal alignment. The algorithm is initialized with $C = D = \emptyset$, $F = \{$all possible alignments$\}$, upper bound $UB = \infty$ and lower bound $LB = 0$. A step-by-step implementation of the CMOS algorithm is as follows:

Stepwise implementation of the CMOS algorithm

1. Initialize the root node such that $C = D = \emptyset$, $F = \{$all possible alignments$\}$, $UB = \infty$, and $LB = 0$. Let list of all nodes $\mathcal{L} = \{$root node$\}$.
2. While $\mathcal{L} \neq \emptyset$
   (a) Choose a working node k from $\mathcal{L}$.
   (b) Apply reduction schemes. Update sets D and F for the working node.
   (c) Calculate UB_k and LB_k for the current node. Update $UB = \max_{l \in \mathcal{L}} UB_l$ and $LB = \max_{l \in \mathcal{L}} LB_l$.
   (d) Prune/delete inferior nodes l with $UB_l \leqslant LB$.
   (e) Check for termination, i.e. , stop if $UB = LB$.
   (f) Choose branching alignment $\in F$. Create descendant nodes, which inherit D and F from the parent node, and add them to $\mathcal{L}$. Delete current working node k from $\mathcal{L}$.
3. end

The bounding and reduction schemes are the key computational ingredients of CMOS and provide several opportunities for enhancements. Introduction of new reduction schemes will result in further reducing the variable search space, and hence decreasing the computational time for dynamic programming based bounding schemes at every subproblem. Thus, inclusion of physical property information for determining and excluding from the search potential residue alignments that are biologically unimportant may prove useful towards improving the performance of CMOS algorithm.

3.2 Alignment space reduction by physical property exploitation

Physical property information about the protein structures under comparison can be obtained *before* aligning the structures. This information can then be used to determine physically and biologically incompatible parts of the proteins under comparison. With this strategy, we have examined four physical properties–hydrogen bonding, hydropathy, torsion angles, and solvent accessibility. For each of these four physical properties, we first briefly discuss their role in shaping the 3D structure of proteins. Then, we describe numerical scales to quantify these physical properties using existing computational tools. We further use these numerical scales in order to devise reduction schemes based on physical property values. These reduction schemes eliminate some of the potential residue alignments from the search space before the actual application of the CMOS algorithm.

While all these physical properties are responsible for shaping the protein structures in some way, not all of them may be useful in accelerating a protein structure alignment algorithm. Their usefulness will be assessed by analyzing the fraction of the number of aligned residues of known optimal alignments for which a given physical property of corresponding residues agrees within a certain threshold. We refer to such residues as *matching residues* and to the corresponding thresholds as *matching thresholds.* For any given matching threshold, a higher fraction of matching residues indicates a higher correlation between the physical property and protein structure. Matching thresholds further become parameters for designing reduction schemes that disallow all potential alignments whose properties fall outside their corresponding thresholds. Tighter thresholds lead to elimination of a larger number of potential residue

alignments, thus expediting the search algorithm. However, large reductions may also eliminate biologically meaningful alignments. A key question here is how to identify matching thresholds that provide the best trade-off between computational time reduction and quality of alignments obtained after reduction.

We performed a wide range of computational experiments to assess the usefulness of physical properties in protein structure alignment. These experiments involved pairwise comparisons of the proteins in the Sokol data set [CLI00] and the Skolnick data set [LCWI01]. The former contains nine small proteins, while the latter contains forty large proteins from five different fold families from the SCOP database. The entire testing set comprises a total of 850 protein pairs, including 222 similar protein pairs and 628 dissimilar protein pairs. The choice of matching thresholds is made through a detailed computational study of the proteins in the Sokol data set. These computational experiments were performed on an INTEL dual core 2.1 GHz machine. A limit of 10000 iterations was imposed on CMOS in all runs. Within this iteration limit, the CMOS algorithm was able to provide optimal structure alignments for 205 of the 850 problems from the Sokol and Skolnick data sets. These 205 pairs are analyzed and used to draw assess different reduction strategies.

In the following, we examine each physical property in detail, using the analysis techniques discussed above.

3.2.1 Secondary structures

Secondary structures, formed by intra- and inter-molecular hydrogen bonding within amino acid residues, impart important functional properties to proteins. Different types of secondary structures exhibit very different physical

and structural properties. For this reason, we consider search space reduction techniques that eliminate alignments between different types of secondary structures. Specifically, we introduce search space reduction schemes that ensure that α-helices and β-strands are not aligned with each other.

We use the DSSP model developed by Kabsch and Sander [KS83] to determine the secondary structure type of each amino acid residue in a protein based on its 3D structure. This model identifies three types of α-helices and two types of β-strands with 100% accuracy. For reduction purposes, we do not discriminate between different variants of α-helices and β-strands. In other words, we lump the three different types of α-helices into a single α-helix type and the two different types of β-strands into a single β-strand type.

To investigate the applicability of secondary structures to protein structure alignment, the known optimal structure alignments for 205 protein pairs in the Sokol and Skolnick data sets were analyzed. Figure 3.2 shows the number of these problems as a function of the fraction of matching residues. As seen in this figure, secondary structure types agree completely for nearly 50% of the aligned residues in all optimal alignments. In addition, the distribution in this figure is skewed heavily towards the 100% value. These observations suggest that secondary structures correlate well with structural similarity.

The secondary structure based reduction scheme was implemented in CMOS by updating the set of disallowed alignments D at the root node based on secondary structure type (SS-type) of amino acid residues as follows:

$$D = D \cup \{x_l \mid \text{SS-type of } a_l \neq \text{SS-type of } b_l\} \tag{3.1}$$

The impact of this reduction mechanism on the solution quality and computational requirements of the CMOS algorithm is assessed in Figure 3.3,

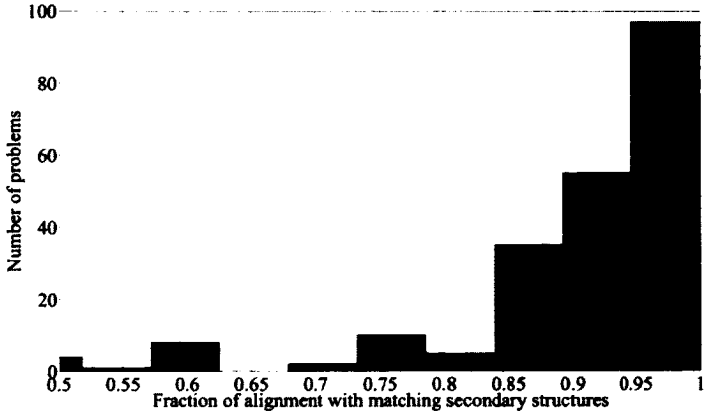


Figure 3.2: Distribution of alignment problems with respect to the fraction of aligned residues with an identical secondary structure type

which compares optimal alignments obtained with and without the inclusion of Equation (3.1). The figure presents the % deviation of the objective function values of these alignments versus the % reduction in search space gained by using Equation (3.1). The secondary structure based reduction scheme resulted in an average (maximum) deviation of 1.6% (10%) from the optimal, while producing an average (maximum) 16% (22%) reduction in the variable search space.

3.2.2 Hydropathy

The hydrophobic or hydrophilic nature of amino acid residues provides information about residue affinity towards water and other solvents. Hydrophobic residues tend to form hydrophobic cores in proteins and prefer the interior of the protein structure, as compared to hydrophilic residues, which tend to form outer protein surfaces and help in stabilizing the protein through solvent interaction.

The hydropathy of amino acid residues is measured by the hydropathy index (HPI) that is determined by theoretical and experimental analysis over

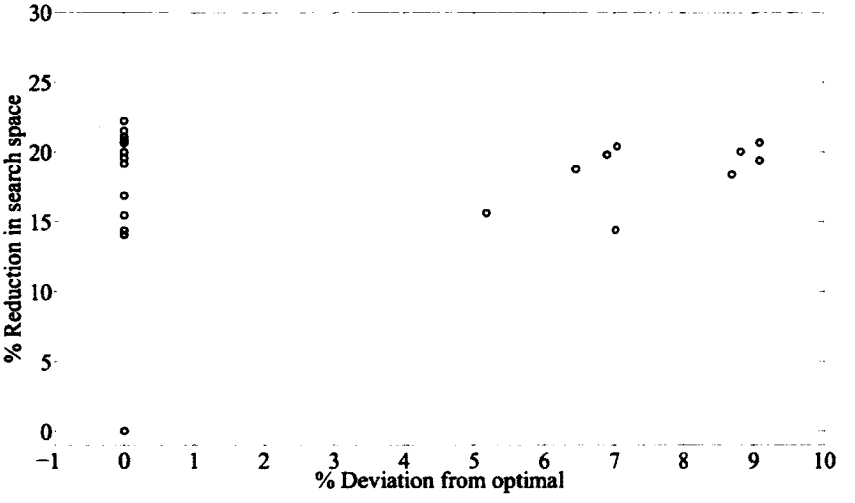


Figure 3.3: Impact of secondary structure based reduction on CMOS algorithm. The graph shows % deviation of objective function from optimal vs. % reduction in search space when reduction mechanism is applied to the Sokol data set.

a large data set of proteins. The HPI scale developed by Kyte and Dolittle [KD82] is based on characteristics of amino acid residues placed individually in a solvent environment. In contrast, the HPI scale developed by Rose et al. [RGL+85] is based on the observed behavior of amino acid residues as a part of the whole protein structure placed in a solvent environment. Henceforth, we will refer to the scales of Kyte and Dolittle [KD82] and Rose et al. [RGL+85] by HPI-KD and HPI-R, respectively. Since the two scales differ considerably in the ranking of the amino acid residues, we have compared the utility of both of these in protein structure alignment.

To investigate the applicability of a hydropathy-based reduction test, the known optimal structure alignments for the 205 protein pairs in the Sokol and Skolnick data sets were analyzed in terms of the fraction of matching amino acid residues for a given matching threshold. Figures 3.4 and 3.5 are based on the HPI-KD and HPI-R values, respectively, and show the number of aligned

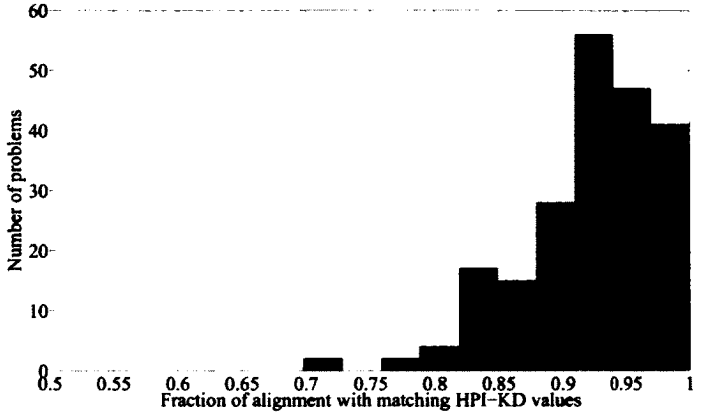


Figure 3.4: Distribution of optimally aligned protein pairs with respect to the fraction of aligned residues with matching hydropathies. Data shown are for the Sokol and Skolnick data sets using the HPI-KD hydropathy scale

protein pairs that had a certain fraction of aligned residues with matching HPI values. The hydropathy values of the amino acids range between -5 and 5 on the HPI-KD scale, and between -1.8 and 0.9 on the HPI-R scale. Thus, the range of the thresholds of the difference between the HPI values is 0 to 10 HPI-KD units and 0 to 2.7 HPI-R units respectively. The Figures 3.4 and 3.5 were constructed using a mid-threshold value of 6 HPI-KD units and 1.5 HPI-R units, respectively. For these 205 protein pairs, it was observed that HPI values matched for more than 70% of the aligned residues. For a large number of pairs, the fraction of matching residues was over 90% of aligned residues. Clearly, hydropathy information characterized by both the HPI-KD and HPI-R scales is matched for large lengths of optimal alignments.

The reduction schemes we designed are based on the HPI values and eliminate alignments where the HPI values of the aligned amino acid residues differ by more than a threshold value. This was implemented in the CMOS algorithm

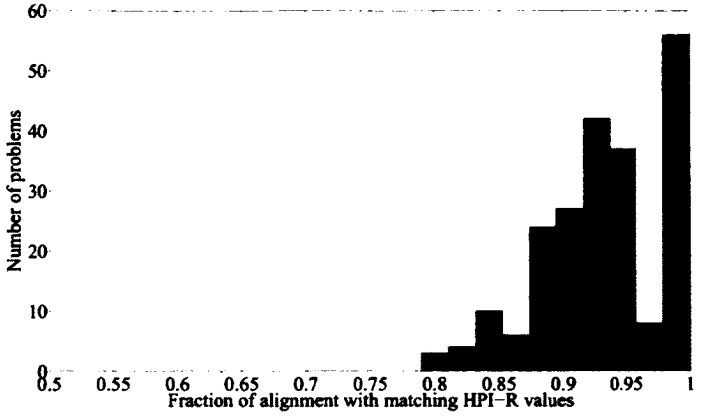


Figure 3.5: Distribution of optimally aligned protein pairs with respect to the fraction of aligned residues with matching hydropathies. Data shown are for the Sokol and Skolnick data sets using the HPI-R hydropathy scale

by updating the set of disallowed alignments D at the root node by

$$D = D \cup \{x_l \mid |\text{HPI index of } a_l - \text{HPI index of } b_l| > \text{HPI}_{\text{threshold}}\} \quad (3.2)$$

The reduction scheme described in Equation (3.2) was also investigated in the context of the CMOS algorithm by computing the effect of the chosen threshold on objective function deviations from optimum as well as search space reduction. For different threshold values, the average deviation of the objective obtained after the inclusion of Equation (3.2) from the objective value before the inclusion of Equation (3.2), and the reduction in search space size caused by Equation (3.2) were analyzed for the Sokol data set. The results are shown in Figures 3.6 and 3.7.

Figure 3.6 shows the average % deviation from the optimal and the average % reduction in search space obtained for different $\text{HPI}_{\text{threshold}}$ values in the HPI-KD scale. These $\text{HPI}_{\text{threshold}}$ values were varied between 2 and 10 HPI-KD units. With thresholds of 8 or more HPI-KD units, only alignments between highly hydrophobic and highly hydrophilic residues are eliminated.

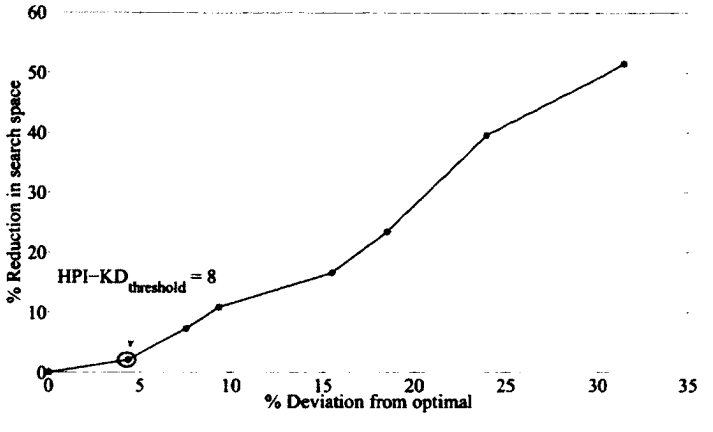


Figure 3.6: Average % deviation from optimal vs. % reduction in search space. Data shown are for the Sokol data set using the HPI-KD hydropathy scale

Figure 3.6 shows that only with a large threshold value of 8 HPI-KD units, the average deviation of the solution from the optimum falls below 5% of the geometric optimum, thus maintaining the quality of the optimum alignment. However, with this threshold value of 8 HPI-KD units, the average (maximum) reduction in the search space is only 2% (5%), which is not enough to produce a significant improvement in the performance of the CMOS algorithm.

Similarly, the Figure 3.7 shows the average % deviation from the optimal and the average % reduction in search space obtained for different $HPI_{threshold}$ values in the HPI-R scale. These $HPI_{threshold}$ values were varied between 0.4 and 2.7 HPI-R units. Figure 3.7 shows that only with a large threshold value of 2.2 HPI-KD units, the average deviation of the solution from the optimal falls below 5% of the geometric optimal value, while producing an average (maximum) search space reduction of only 2% (8%). This reduction is also insufficient to produce a significant improvement in the performance of the CMOS algorithm.

Overall, the hydropathy based reduction schemes were unable to produce considerable search space reductions for the CMOS algorithm. While lower

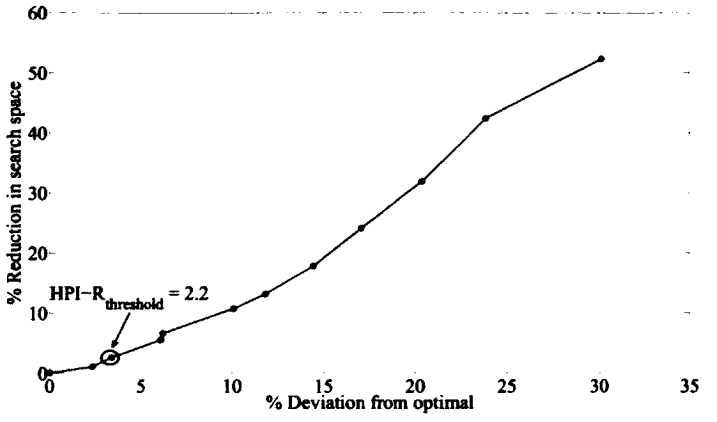


Figure 3.7: Average % deviation from optimal vs. % reduction in search space. Data shown are for the Sokol data set using the HPI-R hydropathy scale

threshold values did produce large reductions in the search space, they also led to the elimination of large number of optimal aligned residue pairs leading to loss of quality in the optimal solution. Higher threshold values produced reductions while maintaining the quality the quality of the solutions. However, the reduction was insufficient to produce a significant improvement in the performance of the CMOS algorithm. Thus, hydropathy information is not used in the updated version of the CMOS algorithm.

3.2.3 Torsional angles

The local structure of proteins is characterized by the torsion angles ϕ and ψ about the N-C_α and C_α-C bonds, respectively. These angles determine the turns and twists in the C_α-backbone of the proteins that we compare while performing structure alignments. Torsion angle information is readily available in the PDB file of each protein. These torsion angles vary between 0^o and 360^o and are utilized to match the local structure of the proteins within thresholds of 0^o to 180^o.

To determine the role of torsion angles in protein structure alignment,

we analyzed the fraction of optimal alignments where the torsion angles of the aligned residues were within a pre-specified threshold. Figures 3.8 and 3.9 show the distribution of these alignment problems versus the fraction of the alignment for which the ϕ and ψ values, respectively, were within a mid-threshold value of 90^o of the aligned residues. These figures show that, for most problems, the torsion angles matched for more than 50% of the aligned residues. However, for some problems, the match was as low as 30% of the optimal alignment length. Thus, while there were many alignments for which torsion angles matched for a large number of aligned residues, there were also many problems for which the length of match was not significant. Moreover, the distribution for the ϕ torsion angles was more uniform, indicating a weak correlation between the torsion angles and optimal structure alignments.

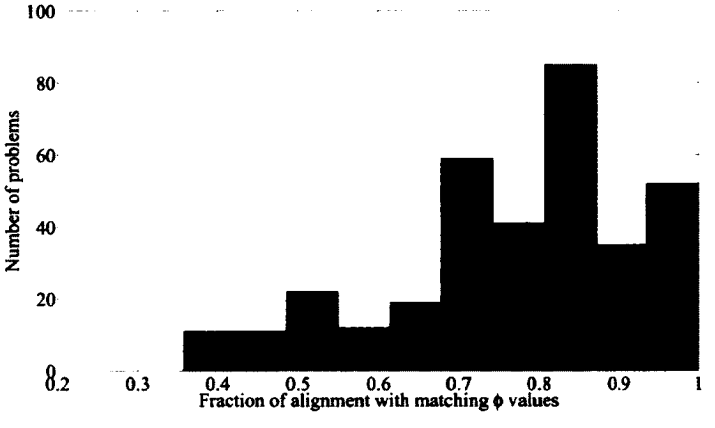


Figure 3.8: Distribution of alignment problems with respect to the fraction of optimal alignments where torsion angles match for the Sokol and Skolnick data sets for torsion angle ϕ

We designed two separate reduction schemes, one for each of the two torsion angle types ϕ and ψ. These reduction schemes eliminate residue alignments when the difference between the corresponding torsion angles is larger than a specified threshold value. The corresponding reduction schemes are imple-

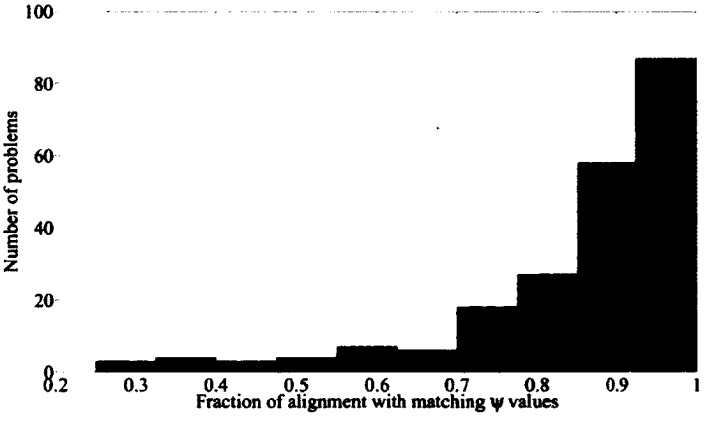


Figure 3.9: Distribution of alignment problems with respect to the fraction of optimal alignments where torsion angles match for the Sokol and Skolnick data sets for torsion angle ψ

mented in CMOS by updating the set of disallowed alignments D to

$$D = D \cup \{x_l \mid |\phi \text{ angle of } a_l - \phi \text{ angle of } b_l| > \phi_{\text{threshold}}\} \quad (3.3)$$

$$D = D \cup \{x_l \mid |\psi \text{ angle of } a_l - \psi \text{ angle of } b_l| > \psi_{\text{threshold}}\} \quad (3.4)$$

We further investigated the trade-off between the deviation from the optimal value before and after inclusion of Equations (3.3) and (3.4), and the reduction in the search space caused by these equations for different $\phi_{\text{threshold}}$ and $\psi_{\text{threshold}}$ values, respectively. The corresponding results are presented in Figures 3.10 and 3.11, respectively. The $\phi_{\text{threshold}}$ and $\psi_{\text{threshold}}$ values were varied between 40^o and 180^o in intervals of 20^o. A maximum difference of 180^o represents aligned bonds in exactly opposite directions. Figures 3.10 and 3.11 show that, for even for a large value of 160^o for the $\phi_{\text{threshold}}$ and $\psi_{\text{threshold}}$, the average deviation of the solution from the optimal was larger than 5% of the geometric optimal value. The choice of a threshold greater than 160^o

would represent allowing matches up to bonds in nearly opposite directions, providing no biologically meaningful reduction in the CMOS search space. We therefore decided not to incorporate these reduction schemes in the CMOS algorithm.

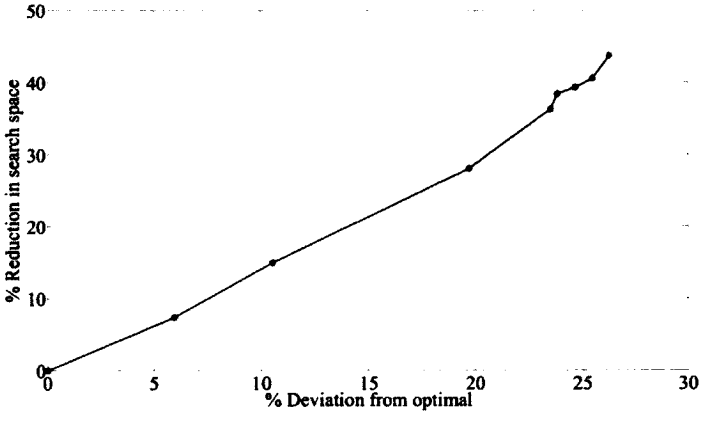


Figure 3.10: % deviation from optimal vs. % reduction in search space. Data shown are for the Sokol data set using the torsion angle ϕ

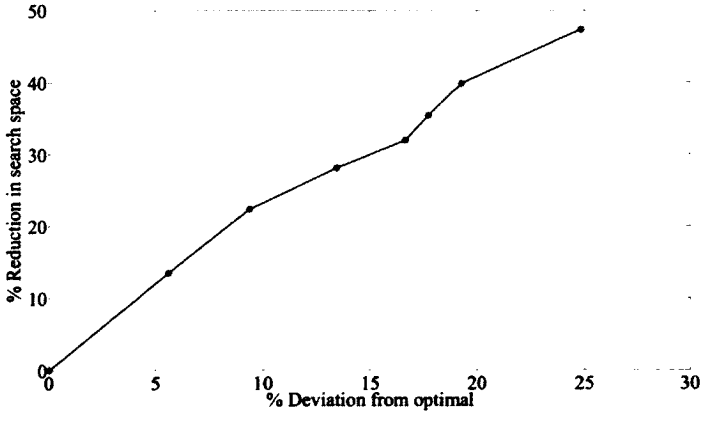


Figure 3.11: % deviation from optimal vs. % reduction in search space. Data shown are for the Sokol data set using the torsion angle ψ

3.2.4 Solvent accessibility

Solvent accessibility (SA) for protein surfaces plays an important role in determining the function and binding sites of a protein. Solvent accessibility is measured by the number of accessible solvent molecules or the solvent accessible area around an amino acid residue. The shape of the protein surface as well as the interior and exterior protein parts can be determined accurately from solvent accessible area at each of the amino acid residues. Solvent accessibility can be quantified using the DSSP tool [KS83], which provides a measure of the solvent accessible area around every amino acid residue of the protein.

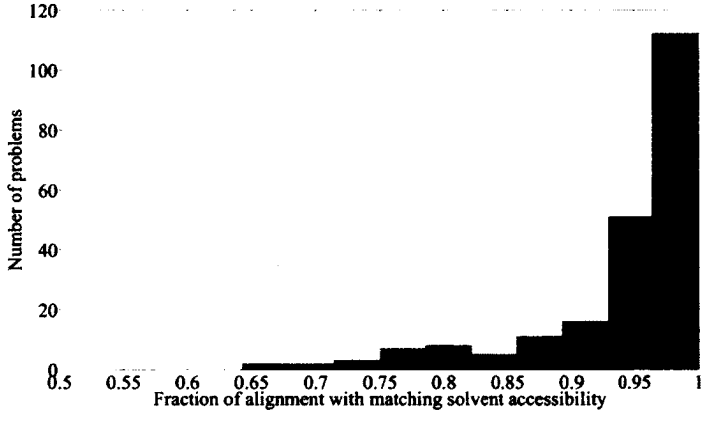


Figure 3.12: Distribution of alignment problems with respect to the fraction of optimal alignments where solvent accessibilities match for the Sokol and Skolnick data sets

The utility of solvent accessibility in protein structure alignment was analyzed using the 205 exactly solved structure alignment problems from the Sokol and Skolnick data sets. Figure 3.12 shows the distribution of these alignment problems versus the fraction of the alignment for which the solvent accessibility values were within an $SA_{threshold}$ of 60Å^2, or 6 water molecules, of their aligned residues. As seen in this figure, solvent accessibility values matched for more than 50% of the optimal alignment length. The peak of the

distribution curve is near 90% of the optimal alignment length, suggesting a very good correlation between the solvent accessibility and optimal structure alignments.

Reduction schemes based on solvent accessibility eliminate residue alignments which differ in their solvent accessible areas by more than a $SA_{threshold}$ value. The reduction scheme based on solvent accessibility is implemented in the CMOS algorithm by updating the set of disallowed alignments D to

$$D = D \cup \{x_l \mid |\text{SA of } a_l - \text{SA of } b_l| > SA_{threshold}\} \tag{3.5}$$

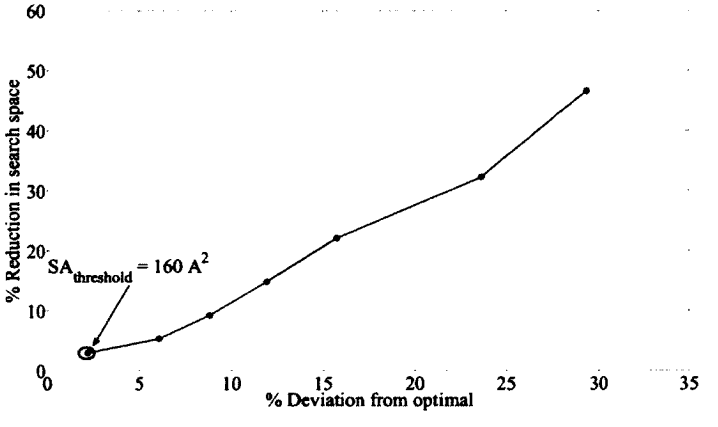


Figure 3.13: % deviation from optimal vs. % reduction in search space. Data shown are for the Sokol data set using solvent accessibilities

Figure 3.13 depicts the trade-off between the deviation from the optimal, and the reduction in the search space, after the application of the reduction scheme presented in equation 3.5, for different $SA_{threshold}$ values. The $SA_{threshold}$ values were varied between 4 and 16 accessible water molecules, or 40Å^2 to 160Å^2 of solvent accessible area. From amongst these, an average deviation of less than 5% from the optimal could be obtained only for a tolerance of

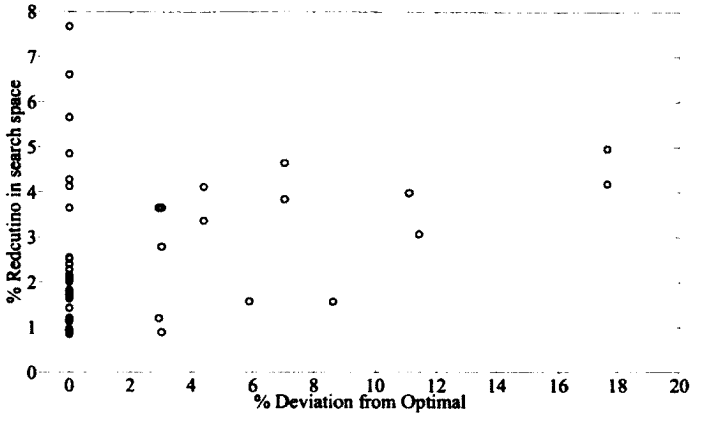


Figure 3.14: Distribution of the % deviation from optimal and % reduction in search space for solvent accessibilities

16 water molecules or an area of 160Å^2. This also resulted in providing a reduction in the search space of an average of only 3%. This reduction value is quite low and cannot produce any considerable computational advantage, while compromising on the quality of solution by 2%. The impact of the inclusion of Equation 3.5 in CMOS is shown in Figure 3.14, which presents the deviation in the optimal value and the reduction in search space for all problems in the Sokol data set. The figure shows that solvent accessibility leads to a maximum search space reduction of 8%. Also, while the deviation from optimal is low for most problems, there are a few problems for which the deviation from the optimal is as large as 18%, even for a large threshold value of 160Å^2. For this reason, solvent accessibility information was not utilized as a reduction scheme in the algorithm.

3.3 Accelerating CMOS with physical properties

The computational results of the previous section suggest that, amongst the four physical properties considered, only secondary structure information may be utilized for considerable improvement in the CMOS algorithm without compromising the quality of the solution obtained. The reduction scheme based on secondary structure types produced a considerable reduction in search space with very low deviation from optimal alignments for the Sokol data set. The effect of incorporating this reduction scheme into the CMOS algorithm is studied in more detail in this section through a computational study that relies on the entire Skolnick data set. In both cases, we compare alignments obtained by the algorithm with and without the incorporation of secondary structure information represented by Equation 3.1. From now on, we use CMOS to refer to the original CMOS algorithm and we use CMOS-SS to refer to the version of the algorithm that utilizes secondary structure based reduction. The comparison between the two approaches was made on the basis of: (a) the average gap between the upper and lower bounds at the root node, (b) the number of branch-and-reduce iterations required by the algorithm to terminate within a 10% gap between the upper and lower bounds, and (c) the number of problems solvable to near optimality. Table 3.1 summarizes the results from these comparisons.

	CMOS	CMOS-SS
Average root node gap	90%	60%
Average # iterations for 10% gap	> 10000	2000
Solvable problems in Skolnick set	20%	80%

Table 3.1: Improvements in the CMOS algorithm by introduction of secondary structure information. Results shown are for the Skolnick data set.

As seen in Table 3.1, the introduction of secondary structure based reduction resulted in an average of 30% decrease in the root-node gap. The root-node gap reduction was especially large for hard alignment problems, comprising large dissimilarity in sizes and low to medium levels of structural similarities. Proteins with large differences in sizes produce a combinatorially explosive number of possible alignments. Thus, even a small number of eliminated alignments by the secondary structure information results in a large reduction in the number of possible alignments to search, thus improving algorithm performance considerably.

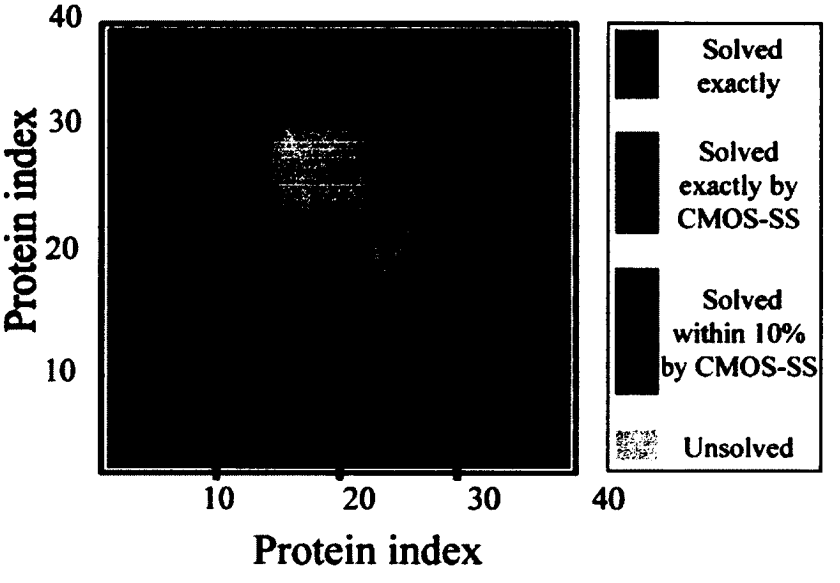


Figure 3.15: Problems of the Skolnick data set solved exactly by CMOS [red], solved exactly by CMOS-SS [blue], solved within 10% by CMOS-SS [green], and unsolved [yellow]

Table 3.1 further reports the number of iterations required to obtain a solution within a tolerance of 10% of the optimal solution. It was observed that CMOS-SS terminated in an average of 2000 iterations, while CMOS was unable to attain a 10% tolerance within the maximum limit of 10000 iterations. Thus,

the secondary structure information resulted in an over five-fold improvement in the computational time of the algorithm. Figure 3.15 presents the problems of the Skolnick data set that were solved exactly (with $LB = UB$) or to near-optimality (nonzero gap between LB and UB) by CMOS and CMOS-SS. CMOS-SS provided exact solutions for 25% problems (red+blue in Figure 3.15) and solutions with 10% gap for 75% of the remaining problems (green in Figure 3.15). The original version of CMOS was able to obtain exact solutions for only 20% of the problems (red in Figure 3.15), and was unable to terminate with near-optimality proof for any of the remaining problems. Overall, within a maximum limit of 10000 iterations, the CMOS-SS algorithm terminated with no more than a 10% gap between LB and UB for 80% of the problems. The original CMOS algorithm succeeding in doing so for only 20% of the problems.

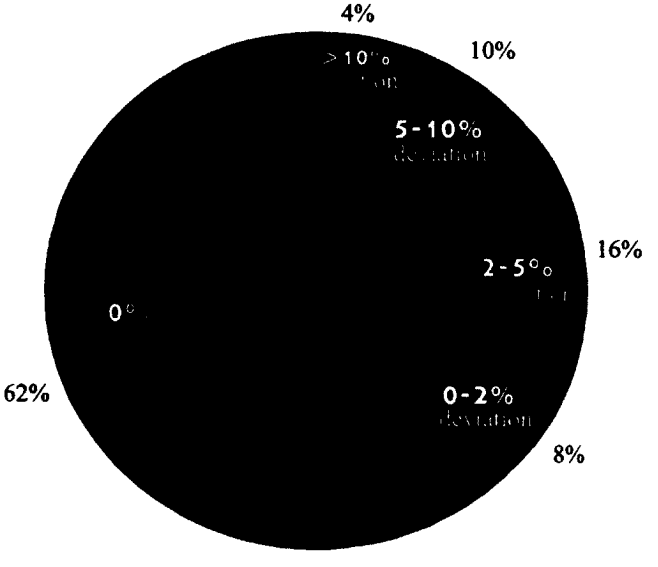


Figure 3.16: Pie chart depicting deviation of CMOS-SS optimal solution from the geometric optimum obtained by CMOS for the 155 optimally solved problems in the Skolnick data set

CMOS-SS imposes reduction constraints that may result in obtaining a suboptimal alignment as compared to the true geometric optimum. Figure 3.16

shows the distribution of the deviation from the optimum for solutions obtained from CMOS-SS as compared to true optimal solutions obtained by CMOS. We observed that in over 60% of the cases, the optimal solution remained the same as the geometrical optimal, and differed less than 10% from the geometrical optimal value in 90% of the remaining cases. Violations greater than 10% were observed in only 4% of the cases. However, most of differences between the CMOS and CMOS-SS solutions were observed to be isolated alignments with a low number of edges in the contact maps. These alignments are usually biologically irrelevant and do not affect the quality of the alignment.

For problems which were solved to near-optimality by CMOS-SS, no exact solution for the alignment problem in available. However, CMOS-SS terminated with a better objective value than CMOS for about 40% of these problems. The improvements in the objective value ranged between 3% to 12% of the objective values obtained by CMOS. The objectives function values of the remaining 60% of the problems are all within 5% of the solutions obtained by CMOS, suggesting essentially no loss of quality due to the inclusion of secondary structure information.

3.4 Special cases

There are certain cases of proteins for which secondary structure based reduction appears to be very effective. One of these cases is when the proteins under comparison have both α-helical and β-sheet characteristics. Figure 3.17 represents the 2PTF and 3B5M proteins, both of which have both α-helical and β-sheet characteristics. These proteins have a sequence identity of 22% and show homologous characteristics. A comparison of these proteins with CMOS requires 5000 iterations, while CMOS-SS provided an optimal solution

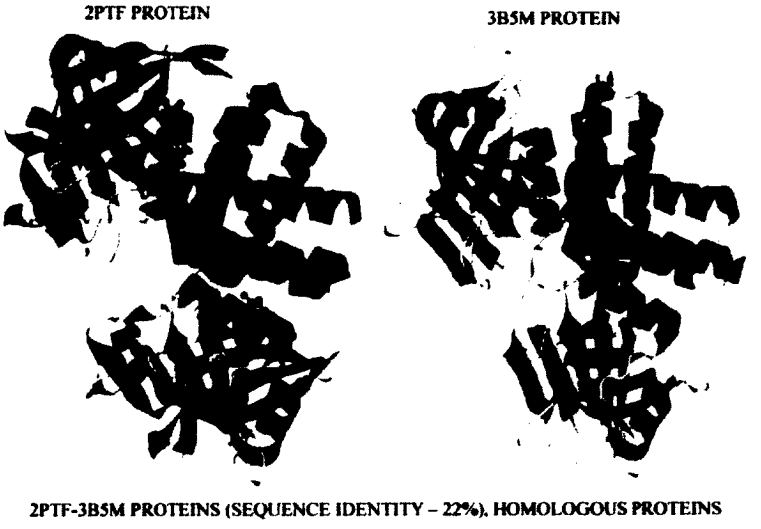


Figure 3.17: Proteins 2PTF and 3B5M are homologous proteins. The α-helices are marked in pink, β-sheets are marked in yellow

in only 1000 iterations. Thus, with the use of secondary structure information, the CMOS algorithm provided the solution five times faster than the original algorithm.

Considerable benefits from reduction based on secondary structures are also observed when the proteins under consideration are of very different sizes. As an example, in Figure 3.18, we compare the 3CHY protein with the 3LFT protein. These proteins are of very different sizes and are homologous in nature through a shared domain. The original CMOS algorithm resulted in a root node gap of 189, while the root node gap was reduced to only 86 by CMOS-SS. Also, the original CMOS algorithm could not provide a solution within 10% solution gap in 10000 iterations, while CMOS-SS terminated with a 10% tolerance solution within only 3000 iterations.

While hydropathy information may not be used in devising a general alignment tool, hydropathy information was observed to be useful in comparing proteins with a hydrophobic core. As an example, in Figure 3.19, we have the

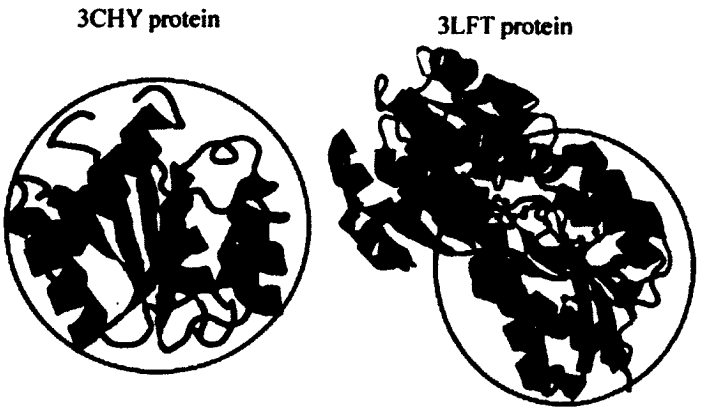


Figure 3.18: Proteins 3CHY (length 128) and 3LFT (length 296) are homologous proteins. The aligned domains are marked in red

1B00 and 3CHY proteins with hydrophobic cores where the use of hydropathy-based reduction is very useful. Here, the reduction scheme, as described in Equation 3.2, eliminated alignments where the difference in the corresponding HPI-KD values [KD82] was in excess of 6 HPI-KD units. The CMOS algorithm produced an optimal alignment in 529 iterations. When the hydropathy-based reduction scheme was applied, the optimal solution was obtained in only 101 iterations, providing a five-fold improvement in the performance of the algorithm.

3.5 Conclusions

We studied the effects of four physical properties of proteins, namely, hydrogen bonding through secondary structures, hydropathy, torsion angles and solvent accessibility, on the CMOS algorithm for structural alignment. While all of these properties may be useful in special cases of protein alignments, it was

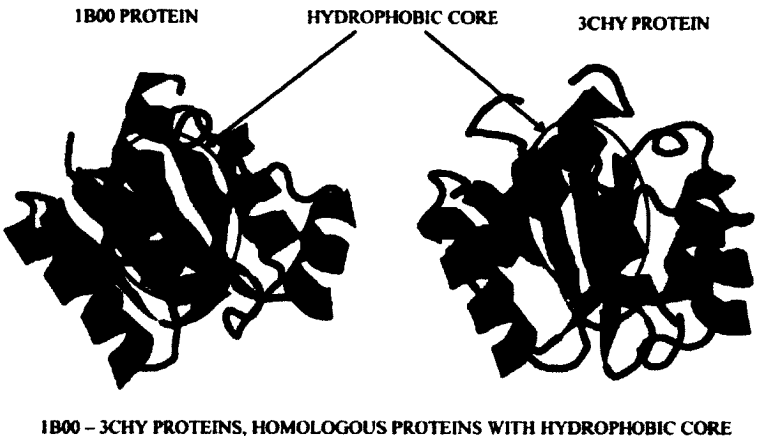


Figure 3.19: Proteins 1B00 and 3CHY homologous proteins. The hydrophobic residues are marked in red, hydrophilic residues are marked in blue

observed that only secondary structures prove to be useful in general structure alignment problems. Other physical property constraints provide inferior alignments as compared to the geometric optimum for a large number of cases.

Computational studies with secondary structure information indicated an over five-fold improvement in the computational efficiency of the CMOS alignment tool. At the same time, the quality of the optimal solution also remained within 10% of the original geometric optimal value for over 95% of the test problems. Thus, secondary structures provided enhanced performance of the CMOS algorithm, without compromising the quality of the solution.

CMOS-SS also provided near-optimal solutions for 80% of the problems that were previously unsolved by CMOS. Amongst these, CMOS-SS provided lower bounds to the solutions than those obtained by CMOS for 40% of the problems. Thus, the incorporation of secondary structure information has increased the utility of the CMOS algorithm to solve more difficult problems.

Chapter 4

SAS-Pro: Simultaneous residue assignment and structure superposition for protein structure alignment

The structure alignment problem is traditionally formulated as a continuous optimization problem, where similar protein substructures are superimposed onto each other to evaluate structural similarity. Here, the proteins are represented using the 3D coordinates of all the C_α atoms representing the protein backbone. To obtain an alignment, one of the proteins is rotated and translated to superimpose it onto the other protein structure, while optimizing a measure of similarity between them. Current structure alignment tools address the alignment optimization problem through a two-step process. In the first step, 'assignment' between amino-acid residues of two proteins is established using dynamic programming or heuristic methods. The objective here is to obtain the largest possible sequential alignment between the two pro-

teins. In the second step, 'superposition' is achieved via computing optimal values for rotation-translation variables by various convex optimization techniques. In the superposition step, the RMSD value or a variant of the RMSD value is minimized. An iterative application of this process results in obtaining the final alignment. Structal [GL96], MAMMOTH [OSO02], and alignment tools developed by Wu et al. [WSHB98], Andreani and Martinez [AM08], and Andreani et al. [AMMY08] are all based on this two-step approach. These approaches differ in the algorithms they use for assignment evaluation and structure superposition, as well as the choice of the objective functions in the two stages of alignment. Nearly all these methods determine the assignment by basic dynamic programming, and utilize different ways of building the similarity matrices based on different structural characteristics of the proteins. The exception is Andreani et al. [AMMY08], who determine the assignment of amino-acid residues by a heuristic method.

The two-step approach to structural alignment has clear computational advantages and results in very fast implementations. However, by decoupling the inter-dependence between the assignment and superposition problems, alignment tools based on this approach may produce suboptimal alignments. In this work, we present a novel approach, Simultaneous Alignment and Superposition of PROteins (SAS-Pro), that combines the evaluation of the assignment and the rotation-translation problems into a single bilevel optimization formulation. We further propose a combination of optimization algorithms, which we demonstrate leads to a practical computationally efficient approach for the solution of the proposed formulation. In addition, by eliminating the residue-sequentiality constraints, the SAS-Pro approach is capable of providing both sequential and non-sequential structure alignments.

4.1 The problem and a natural decomposition

Consider proteins A and B to be structurally aligned. Let a_i represent the i^{th} residue of protein A, and b_j represent the j^{th} residue of protein B. In addition, let $r(a_i)$ and $r(b_j)$ represent the 3D coordinates of the corresponding amino-acid residues. We seek to align amino-acid residues of A to amino-acid residues of B so that, when A is rotated-translated onto B, a similarity measure between the two proteins is minimized. The RMSD function will be used to determine the similarity between the protein structures and is defined as

$$\text{RMSD}(\text{S},\theta) = \sqrt{\frac{\sum_i \sum_j S_{ij} ||\theta(r(a_i)) - r(b_j)||^2}{\sum_i \sum_j S_{ij}}}. \tag{4.1}$$

Here, S_{ij} is a binary variable that equals 1 when a_i is aligned to b_j and 0 otherwise, and θ represents the rotation-translation transformation applied to protein A.

The problem of minimizing the RMSD may be represented as the following mixed-integer nonlinear optimization program:

$$\begin{aligned}
\textbf{(MINLP)} \qquad & \min_{S,\theta} \ \text{RMSD}(\text{S},\theta) \\
& \text{s.t.} \quad \sum_i S_{ij} \leqslant 1 \quad \forall j && (4.2)\\
& \qquad \sum_j S_{ij} \leqslant 1 \quad \forall i && (4.3)\\
& \qquad \sum_i \sum_j S_{ij} \geqslant r_m && (4.4)\\
& \qquad S_{ij} \in \{0,1\} \quad \forall i,j && (4.5)
\end{aligned}$$

Here, the parameter r_m is the minimum number of residues that must be aligned to ensure that the global optimum attains a non-trivial value and is enforced through Constraint (4.4). Constraints (4.2) and (4.3) ensure that no

more than one amino-acid residue of protein A is aligned with an amino-acid residue of protein B and vice versa. Constraint (4.5) enforces the binary nature of the assignment variables S.

4.1.1 Two-stage approach

A two-stage solution approach employed by existing alignment tools decouples the effects of S and θ variables and evaluates the effect of the assignment variables S and rotation-translation variables θ separately. The two-stage optimization problem may be viewed as follows:

Stage 1

$$\begin{aligned} \min_S \quad & f(S, \theta_0) \\ \text{s.t.} \quad & \sum_i S_{ij} \leqslant 1 \quad \forall i \\ & \sum_j S_{ij} \leqslant 1 \quad \forall j \\ & \sum_i \sum_j S_{ij} \geqslant r_m \\ & S_{ij} \in \{0,1\} \quad \forall i,j \end{aligned} \tag{4.6}$$

Stage 2

$$\min_\theta \quad \text{RMSD}(S_0, \theta)$$

where S_0 and θ_0 are optimal values of S and θ, respectively, obtained in Stage 1 and Stage 2 of an iteration of the two-stage optimization problem. Constraint (4.6) in Stage 1 is imposed implicitly in the model by solution procedures utilized to solve for S_0.

In typical approaches, values for the assignment variables S are determined by heuristic methods and dynamic programming techniques. The function f is thus selected as the dynamic programming objective function based on

different similarity matrices designed for the alignment tool. The similarity matrices currently in use are based on structural features of the proteins, including inter-residue distances [GL96, AM08], bond angles [OSO02], and radii of fragment curvature [WSHB98]. These heuristic methods and dynamic programming techniques do not guarantee optimality of the alignment obtained with respect to the objective of Stage 2, the RMSD value. Thus, the final alignment obtained from the iterative procedure is not guaranteed to be globally optimal, and is known to be dependent on the initialization of the process [GL96, AMMY08, AM08]. Hence, the two-stage formulation may provide only a feasible solution of the MINLP and not necessarily a global optimum. Global optimality cannot be guaranteed unless the MINLP is somehow solved directly.

4.2 SAS-Pro model

The SAS-Pro model reformulates the MINLP model into a single bilevel optimization problem. For any given θ, the function SRMSD(θ) may be defined as

$$\text{SRMSD}(\theta) = \min_{S} \text{RMSD}$$

The master problem of the SAS-Pro model optimizes over the solution of

the subproblem SRMSD(θ). The bilevel SAS-Pro model is as follows:

(SAS-Pro master problem)

$$\begin{aligned} \tau &= \min_{\theta} \{\min_{S} \text{RMSD}(S, \theta)\} \\ &= \min_{\theta} \text{SRMSD}(\theta) \end{aligned}$$

(SAS-Pro subproblem)

$$\begin{aligned} \text{SRMSD}(\theta) = \min_{S} \quad & \text{RMSD}(S, \theta) \\ \text{s.t.} \quad & \sum_{i} S_{ij} \leqslant 1 \quad \forall j \\ & \sum_{j} S_{ij} \leqslant 1 \quad \forall i \\ & \sum_{i} \sum_{j} S_{ij} \geqslant r_m \\ & S_{ij} \in \{0, 1\} \quad \forall i, j \end{aligned}$$

The master problem objective function SRMSD(θ) is in the space of the θ variables alone. Yet, it is trivial to see that any assignment/superposition feasible to the MINLP is also feasible to the SAS-Pro master problem. Hence, our reformulation maintains optimality.

Evaluation of the function SRMSD(θ) involves solving the subproblem and determining the optimal assignment variables S, for given values of θ and parameter r_m. Our key observation is that, for a given value of θ, the subproblem can be reformulated as the following k-cardinality linear assignment problem

(k-LAP):

$$\textbf{(k-LAP)} \quad \kappa_\theta = \min_S \sum_i \sum_j a_{ij} S_{ij} \tag{4.7}$$

$$\text{s.t.} \quad \sum_i S_{ij} \leqslant 1 \quad \forall j$$

$$\sum_j S_{ij} \leqslant 1 \quad \forall i$$

$$\sum_i \sum_j S_{ij} \geqslant r_m \tag{4.8}$$

$$S_{ij} \in \{0,1\} \quad \forall i,j$$

where $a_{ij} = ||\theta(r(a_i)) - r(b_j)||^2$, $\forall i,\ j$. A highly efficient polynomial-time algorithm, SKAP [DLM01], has been developed to solve the k-LAP problem and can be readily utilized in this context. The solution to the k-LAP problem will provide an assignment of exactly r_m amino-acid residues, as constrained in equation (5.4). The numerical value of SRMSD(θ) can be obtained from the objective value in equation (5.3) of the k-LAP problem as SRMSD(θ) $= \sqrt{\kappa_\theta / r_m}$. The k-LAP model does not include any sequence preserving constraints. Thus, the SAS-Pro model is designed to provide an optimal assignment and structure superposition of protein structures for specified values of the parameter r_m, with no sequence-preserving constraints. We later show how to recover a sequential alignment, if desired, from the SAS-Pro alignment.

Kolodny and Linial [KLL04] also present a bilevel approach to structure alignment by utilizing the SAS [SLL93] similarity measure as the objective function in the master problem, as opposed to the RMSD value. They obtain values for the assignment variables S through a dynamic programming methodology and determine the rotation-translation variables by enumeration over a grid in the θ space. Our approach differs from their approach in

three major aspects. First, the objective function used by Kolodny and Linial in the subproblem to determine the assignment variables S (dynamic programming based objective) differs from their master problem objective (SAS score). We use the same objective in both the subproblem as well as the master problem of the SAS-Pro model, which guarantees that a SAS-Pro optimal solution is optimal also for the original MINLP problem. Second, we utilize efficient search techniques to solve the master problem and obtain near-optimal rotation-translation variables, as opposed to the expensive enumeration approach used by Kolodny and Linial. Finally, our approach has the added capability of providing both sequential and non-sequential structure alignments for protein pairs.

As mentioned above, an optimal solution of the MINLP is feasible to our reformulation. In order for an optimal solution to be identified, suitable algorithms must be used to solve the master problem to global optimality. Indeed, there exist derivative-free optimization (DFO) algorithms that can achieve this goal based on dense sampling of the domain [RS11]. However, in the search of the most computationally efficient approach, in the next section we will also evaluate local search techniques for solving the master problem. With the same goal in mind, we will introduce a heuristic approach for determining the optimal parameter r_m as well as for curtailing the number of degrees of freedom for the alignment problem.

4.3 Algorithm

4.3.1 Derivative-free optimization

The landscape of the RMSD function with varying values of the rotation angles β and γ is presented in the contour plot of Figure 4.1 for proteins 1B00 and

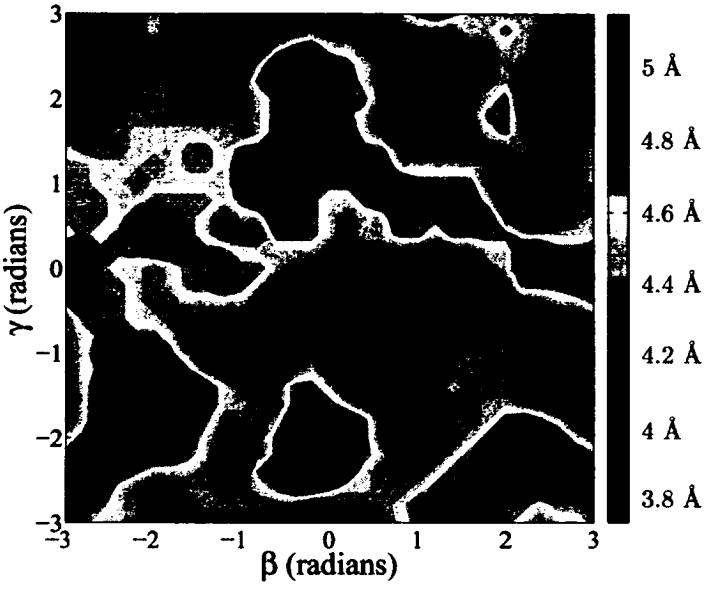


Figure 4.1: Contour plot of the landscape of the RMSD function for 1B00 and 1DBW proteins in the $\beta - \gamma$ rotation angles plane

1DBW. As seen in this figure, the objective function in the SAS-Pro model is highly multi-modal and nonlinear. This multi-modality can be addressed by optimization techniques that span the entire search space of the problem in the search for global optima. Furthermore, an explicit algebraic form for the SRMSD objective function for the master problem is not available, thus making it difficult to utilize derivative-based optimization methods. Thus, we opted to employ DFO techniques in order to solve the SAS-Pro model.

We performed extensive computational analysis with 28 different DFO solvers, based on a variety of techniques that included direct search, pattern search, surrogate management frameworks, domain partitioning methods, local search, global search, deterministic and stochastic algorithms [RS11]. Our experiments indicated that the derivative-free solver SNOBFIT [HN08] provides the best performance for a small number of function evaluations. This observation is consistent with the results reported in [RS11]. Keeping the

number of function evaluations low was dictated by our desire to design an algorithm that would take no more than a few CPU minutes on a standard computer workstation for the alignment of protein pairs that are routinely analyzed nowadays.

Our interface to SNOBFIT is based on the 'mydfo' interface developed by Rios [Rio09]. We have limited SNOBFIT to 500 function evaluations for each value of the parameter r_m. Every RMSD function evaluation for a given value of θ involves solving the k-LAP problem using the SKAP code developed by Dell'Amico and Martello [DLM01].

4.3.2 Choice of parameter r_m

The solution to the SAS-Pro model is dependent on the parameter r_m. Different values of r_m may lead to very different optimal alignments. The best alignment is found when the value of r_m is close to the number of biologically relevant residue matches. It is therefore important to determine the right value of the parameter r_m.

Proteins with high level of similarity have a large length of alignment, usually corresponding to 85% or more of size of the smaller protein. Hence the number of biologically relevant residues matches are expected to be between to 85% to 100% of the size of the smaller protein. To identify the best value for r_m, we systematically vary the value of r_m from 100% to 85% of the size of the smaller protein, until an alignment with a good similarity measure cutoff is obtained. The similarity measure used here is SAS_{nseq}, a modified version of the SAS score, that is further discussed in Section 4.3.5. In our implementation, we select the value of r_m for which an SAS_{nseq} score of less than 4Å is obtained.

4.3.3 Reducing the number of degrees of freedom

The solution to the SAS-Pro model involves determining the optimal values of both the assignment variables S as well as the rotation-translation variables θ. The assignment variables S are obtained as an exact solution to the SAS-Pro subproblem. Thus, the only degrees of freedom available in the SAS-Pro master problem are the three translation vector components t_x, t_y, and t_z along the X, Y and Z axes, respectively, and the three rotation angles α, β, and γ about the X, Y and Z axes, respectively.

In the course of our computational experimentations, we observed that, for proteins with similar sizes, a good approximation of the translation vectors is very often obtained if the centroids of the two protein structures are required to coincide. Thus, while comparing proteins of similar sizes, the number of degrees of freedom for optimization may be reduced to only the three rotation angles. As demonstrated in [RS11], for a collection of over 500 test problems, problems with up to three or four variables were almost always solved to global optimality by a variety of DFO algorithms. Thus, while solving the SAS-Pro optimization problem, the small number of degrees of freedom provides a computational advantage in terms of obtaining globally optimal structure alignments.

For structural comparison of proteins with different sizes, the SAS-Pro algorithm offers an option to utilize all six degrees of freedom. In this case, in order to maintain solution quality of the DFO solvers, we found it necessary to increase the number of function evaluations to 1000 for each value of r_m considered.

4.3.4 Extracting sequential alignments

The solution to the SAS-Pro model is usually a non-sequential structure alignment between the two proteins. However, a sequential alignment is easy to extract from the non-sequential alignment obtained from the SAS-Pro algorithm in a post-processing step. A dynamic programming algorithm was designed to identify the largest sequential alignment amongst the aligned residues provided by SAS-Pro. This algorithm sequentially evaluates the largest length of sequential alignment terminating at residue $a(i)$ of protein A and stores it in the vector $LenSeq(i)$. The algorithm maintains a pointer to the residue before $a(i)$ in the sequential alignment in the vector $PREV(i)$. $M(a(i))$ denotes the residue $b(j)$ of protein B which is aligned to $a(i)$. The largest value of $LenSeq(i)$ provides the length of the largest sequential alignment terminating at residue i. Backtracking the residues from this value of i using the vector $PREV(i)$ provides the corresponding alignment. A pseudo-code of the algorithm is presented below:

INITIALIZE
for $(i = 1 \rightarrow M)$ **do**
    $LenSeq(i) \leftarrow 1$
    $PREV(i) \leftarrow i$
end for
MAIN ALGORITHM
for $(i = 1 \rightarrow M)$ **do**
    for $(j = 1 \rightarrow i - 1)$ **do**
        if $(M(a(i)) < M(a(j))$ **and** $LenSeq(j) \geqslant LenSeq(i))$
        then
            $LenSeq(i) \leftarrow LenSeq(j) + 1$
            $PREV(i) \leftarrow j$
        end if
    end for
end for
SOLUTION
$MaxLength \leftarrow \max_i LenSeq(i)$
$MaxI \leftarrow arg(\max_i LenSeq(i))$

$j \leftarrow MaxI$
for $(i = 1 \rightarrow MaxLength)$ **do**
  $Alignment \leftarrow (j, M(a(j)))$
  $j \leftarrow PREV(j)$
end for

4.3.5 Similarity measure

For sequential protein alignments, where the sequence of the amino acid residues is preserved in the alignment, many suitable similarity measures, such as the Structure Alignment Score SAS [SLL93] and the Similarity Index SI [KJ94], have been defined. These measures are based on weighted ratios of the RMSD value and the length of alignment produced by the algorithm. However, for non-sequential structure alignments, the length of alignment is not properly defined and hence cannot be used to calculate the SAS and SI measures. We introduce a new measure of length of alignment, the total fragment length (N_{frag}), to extend the definition of the SAS similarity measure to non-sequential structure alignments. The total fragment length is defined as the sum of lengths of aligned continuous fragments of five or more residues. Sequentiality of the amino-acid residues in the fragment is not required, thus providing for a measure of the length of alignment that is applicable to both sequential and non-sequential structure alignment.

The similarity between proteins is then determined using the proposed SAS_{nseq} measure, which is defined as

$$\text{SAS}_{\text{nseq}} = \frac{\text{RMSD}}{N_{\text{frag}}/100}$$

This measure reduces to the SAS measure for the case of sequential structure alignments.

The best non-sequential structure alignment obtained from the SAS-Pro

algorithm may include multiple local small-length matches as opposed to a single large global alignment. This disorder of the alignment can be measured by the value of the fragment length. A disordered alignment is expected to have a small fragment length, while a biologically relevant ordered alignment is expected to have a large fragment length, thus providing lower SAS_{nseq} values for biologically relevant alignments. Hence, the best alignment for a given pair of proteins is expected to be one with the lowest SAS_{nseq} score.

4.4 Implementation and computational results

We performed computational experiments based on three data sets:

- the Sokol data set [CLI00], which is a set of 9 small size proteins with proteins from three different fold families,
- the Skolnick data set [LCWI01], which is a set of 40 large size proteins from five different fold families, and
- the RIPC data set [MDL07], which is a set of 23 complex structure alignment problems.

An all-to-all pairwise alignment for all the proteins in the Sokol and Skolnick data sets was obtained, resulting in 850 pairwise alignment problems with 222 similar protein pairs and 628 dissimilar protein pairs. The similar pairs in these data sets exhibit sequential similarity. The RIPC data set consists of 23 protein alignment problems for which a biologically relevant reference alignment is available. These 23 alignment problems are complex and exhibit non-sequential structure similarities. The complexity of these alignments arises from repetitions, insertions/deletions, permutations, and conformational

changes between the protein pairs that are not easily handled by alignment algorithms.

In all tests, the typical computing time requirements for SAS-Pro were around 1 CPU minute per protein pair on an Intel Quad Core 2.83 GHz processor with 6 GB RAM, while providing sequential and non-sequential alignments with exceptional classification ability.

4.4.1 Sequential structure alignments

The Sokol and Skolnick data sets were analyzed to evaluate the performance of SAS-Pro in obtaining sequential alignment problems. To obtain sequential alignments from the non-sequential alignments provided by SAS-Pro, the procedure described in Section 4.3.4 was used. Alignments were compared using the RMSD values as well as the geometric similarity measures SI and SAS.

A comparison of the RMSD, SI, and SAS values obtained by SAS-Pro for similar and dissimilar proteins is presented in Table 4.1. For protein pairs within the same fold family, alignments with low RMSD, SI, and SAS values were obtained. For pairs from different fold families, the values of RMSD, SI, and SAS were comparatively higher than the corresponding values for similar proteins. In addition, the alignments obtained from the SAS-Pro alignment tool were near-sequential for similar protein pairs and were 96% in agreement with known optimal alignments between the proteins that were obtained from the exact structure alignment tool CMOS [XS07]. These optimal alignments contain both large fragments of aligned residues as well as a few isolated aligned residues. SAS-Pro matches the large fragments of aligned residues with these optimal alignments exactly. However, the alignments may differ in isolated residue matches, that are not of biological consequence, resulting in an average of 96% agreement between the alignments between SAS-Pro and CMOS.

	Sokol set		Skolnick set	
	Similar	Dissimilar	Similar	Dissimilar
RMSD	0.60	2.9	1.72	3.94
SI	1.17	7.04	3.15	9.77
SAS	1.61	7.37	2.19	8.51
% match with optimal alignment	96	N.A.	96	N.A.

Table 4.1: Average RMSD value, SI score, SAS score, and match with reference alignments for the Sokol and Skolnick data sets for similar and dissimilar protein pairs.

	% Problems where					
	SAS-Pro is better			SAS-Pro is at par		
Solver	RMSD	SI	SAS	RMSD	SI	SAS
CE	57	51	51	12	12	12
SSM	47	36	36	12	12	12
STSA	44	40	40	21	21	21

Table 4.2: Comparison of SAS-Pro with CE, SSM, and STSA for the similar protein pairs of the Sokol and Skolnick data sets using RMSD, SI, and SAS measures.

The alignments obtained from SAS-Pro were also compared with those obtained from the CE [SB98], SSM [KH04], and STSA [SZB09] alignment tools. The results are summarized in Table 4.2. The SAS-Pro approach provided alignments with better or equal RMSD for over 59 to 69% of the similar alignment problems. Moreover, the RMSD values of more than three quarters of the remaining problems were observed to exceed those in CE and SSM by only 1Å on average, while preserving a 96% similarity with the corresponding sequential structure alignments. Consequently, the corresponding SI and SAS scores for these problems were also within 1Å of those from CE and SSM.

The Sokol and Skolnick data sets together include 222 similar protein pairs and 628 dissimilar protein pairs. A classification of these 850 problems into similar and dissimilar pairs was sought based on the SAS scores of the alignments obtained. The CE, SSM, and SAS-Pro alignment tools provided exact classification of these protein pairs. The STSA algorithm, however, produced

very short alignments for 5 of the similar pairs, leading to an imperfect classification.

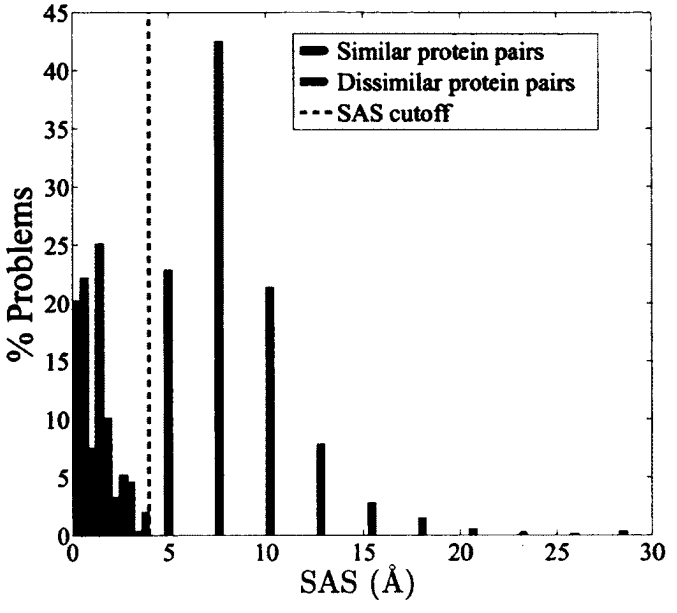


Figure 4.2: Distribution of SAS values obtained by SAS-Pro for similar and dissimilar proteins in the Skolnick data set

Figure 4.2 shows the distributions of the SAS values obtained for similar and dissimilar protein pairs for the Skolnick data set by SAS-Pro. The distributions for the similar and dissimilar proteins were observed to be completely disjoint, with lower SAS scores for similar proteins and higher SAS scores for dissimilar proteins. A SAS score cutoff of 4Å produced a perfect classification of the alignment problems into similar and dissimilar protein pairs. Based on this observation, a termination criterion for the SAS-Pro code was implemented. For computations reported in the sequel, SAS-Pro was designed to terminate if (a) an alignment with a SAS score of 4Å or less is obtained, or (b) all values of r_m between 85% and 100% of the size of the smaller protein are explored. In either case, the best alignment and the corresponding RMSD, SAS score, and fragment length of the alignment are returned by the software.

4.4.2 Non-sequential structure alignments

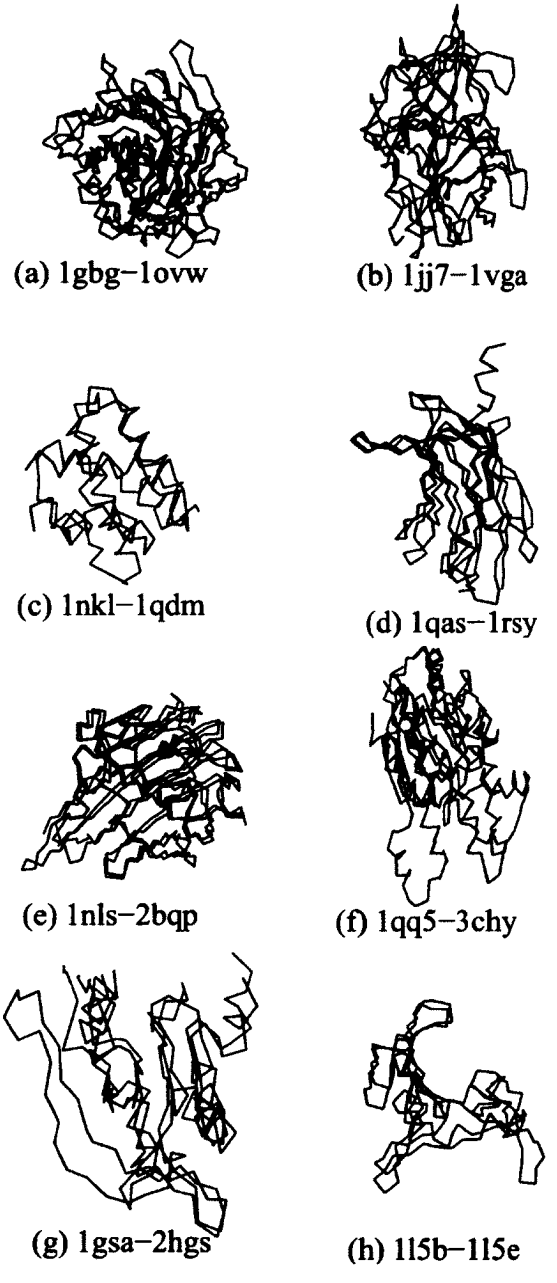


Figure 4.3: Alignments obtained by SAS-Pro for the RIPC data set. These alignments are in 100% agreement with the reference alignments [MDL07]

We performed a computational study to determine the quality of SAS-Pro's non-sequential structure alignments utilizing the RIPC data set and the non-

sequential alignment problems presented by Salem and Zaki [SZB09]. Salem and Zaki [SZB09] provided two examples of the non-sequential structure alignments for which their alignment tool, STSA, performs better than other structure alignment tools. We performed an alignment of the corresponding two protein pairs, 2LH3:A with 2HPD:A, and 1FSF:A with 1IG0:A, and obtained comparable alignments for both cases. For the 2LH3:A and 2HPD:A proteins, we obtained an alignment with length 126 and RMSD 3.17Å, as compared to STSA's alignment of length 117 and RMSD 3.27Å. For the 1FSF:A and 1IG0:A proteins we obtained an alignment with length 117 and RMSD 2.68Å, as compared to STSA's alignment of length 104 and RMSD 5.4Å. We present a quantitative comparison of the SAS-Pro alignment between the 2LH3:A and 2HPD:A proteins and other solvers in Table 4.3. As the results in this table demonstrate, SAS-Pro provides an RMSD in the same ball-park range as most other tools but with larger alignment length, thus providing a superior structure alignment as the SAS_{nseq} values indicate.

We next present results from a computational study with the 23 protein pairs in the RIPC data set. For this test set, SAS-Pro provided alignments which are 30% to 100% in agreement with the reference alignments. The median agreement of SAS-Pro is 70% and the mean is 62%. SAS-Pro provides alignments with greater mean and median agreements than CE, DALI, FATCAT, MATRAS, CA, SHEBA, SARF, and LGA. The corresponding box and whisker plot of percentage agreement with reference alignments is shown in Figure 4.4. STSA provides alignments with better mean and median agreements with reference alignments than SAS-Pro. However, SAS-Pro provides excellent quality alignments with 100% agreement with the reference alignments for eight problems, while STSA provides alignments in 100% agreement with reference alignments for only four problems.

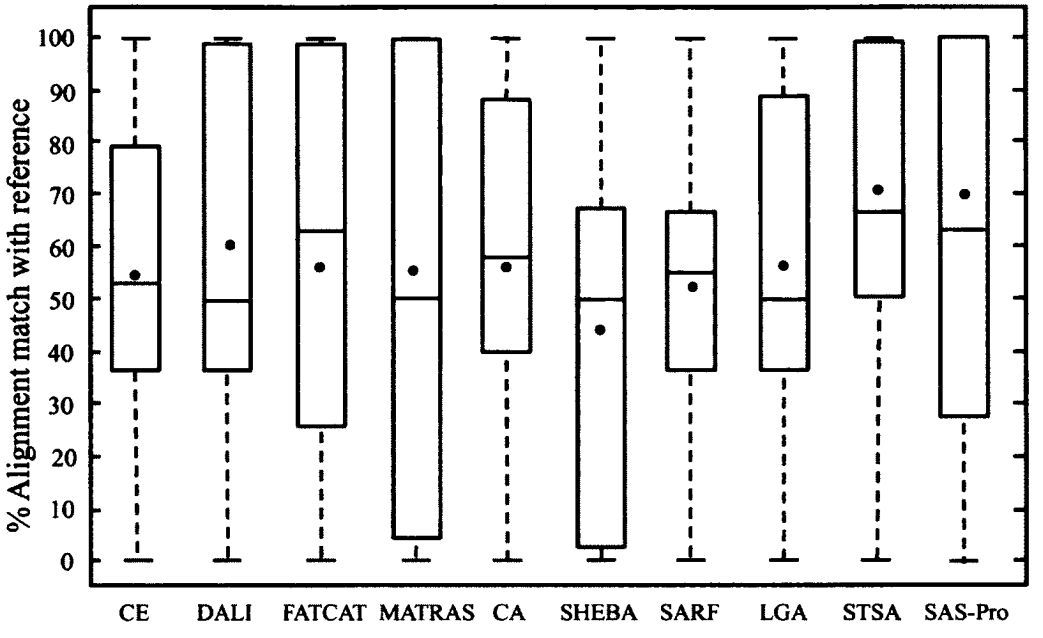


Figure 4.4: Box and whisker plot for the performance of different alignment tools for the RIPC data set. The red line represents the mean and the dot represents the median of the box. (All results, except for SAS-Pro and CE, were taken from [SZB09]).

The eight alignments for which SAS-Pro is in complete agreement with reference structures are shown in Figure 4.3. These eight protein pairs represent alignment problems spanning all four types of alignment challenges encountered in the RIPC data set, namely, repetitions, insertions/deletions, permutations, and conformational changes. The protein pairs 1gbg-1ovw (Figure 4.3(a)) and 1jj7-1vga (Figure 4.3(b)) present alignments with large requirements of insertions/deletions, not handled by all alignment tools. Specifically, protein 1jj7-1vga consists of a loop containing different numbers of β-strands, which require insertion/deletion of the right β-strands to obtain the correct alignment. SAS-Pro places no limit on number of insertions/deletions, result-

ing in a very good alignment for this protein pair. Protein pairs 1nkl-1qdm (Figure 4.3(c)), 1qas-1rsy (Figure 4.3(d)), 1nls-2bqp (Figure 4.3(e)), and 1qq5-3chy (Figure 4.3(f)) are examples of proteins with permutations. In the 1nls-2bqp protein pair, the N-terminus of one protein aligns with the C-terminus of the other protein, and *vice versa*. Most alignment codes match only the N-terminus half of 1nls with the C-terminus half of 1bqp, while SAS-Pro aligned the entire protein accurately. The 1qq5-3chy, 1nkl-1qdm, and 1qas-1rsy proteins consist of multiple α-helices, which do not align sequentially. SAS-Pro correctly aligns the right α-helices with each other, producing biologically relevant alignments. Finally, protein pairs 1gsa-2hgs (Figure 4.3(g)) and 1l5b-1l5e (Figure 4.3(h)) present conformational changes which cause slight bends in the structures. SAS-Pro was able to provide the correct structural alignment with a 100% match with the reference.

Alignment tool	RMSD (Å)	N_{align}	SAS_{nseq}
SAS-Pro	3.17	126	2.5
STSA	3.37	117	2.9
SARF2	3.05	108	2.8
STRUCTAL	2.27	56	4
DALI	4.8	87	5.5
CE	4.05	91	4.4

Table 4.3: Comparison of performance of alignment tools for aligning 2LH3:A and 2HPD:A proteins. (All results, except SAS-Pro, taken from [SZB09])

There are three problems in the RIPC data set for which the agreement of the SAS-Pro alignment with the reference in less than 50%. These three problems are from the permutation class of alignments for which, as Mayr et al. [MDL07] suggest, biologically relevant alternative alignments may exist. Hence, it is likely that SAS-Pro's performance may be even better than what the results of this section suggest.

Mayr et al. [MDL07] and Salem and Zaki [SZB09] have discussed eight

protein pairs from the RIPC data set that are difficult to align. Amongst these, Salem and Zaki [SZB09] reported the 1nkl-1qdm protein pair and the 1qq5-3chy protein pair, for which most alignment tools provided a 0% match with the reference alignment. For both of these pairs, SAS-Pro and STSA provided a 100% match with the reference alignment. Amongst the remaining six protein pairs, SAS-Pro provided high quality alignments with 100% agreement with the reference for three pairs and over 50% agreement with the reference for the remaining three pairs.

4.5 Discussion

We presented a novel formulation of the protein structure alignment problem as a single bilevel optimization problem that addresses the assignment of amino acid residues and the structural superposition of proteins simultaneously. We employed derivative-free optimization techniques to deal with the multi-modality and non-differentiability of the RMSD function in the proposed formulation. The proposed structure alignment methodology is capable of providing both sequential and non-sequential alignments.

Our computational experiments demonstrate that the SAS-Pro model captures similarities within proteins accurately and provides alignments with lower RMSD values and larger lengths of alignments as compared to CE, SSM, and STSA for a majority of problems in the Sokol and Skolnick data sets. Moreover, SAS-Pro exhibits very good performance for the RIPC data set, for which it provided alignments with 100% agreement with the reference for a large number of protein pairs.

While the present methodology addresses both sequential and non-sequential alignments, we will investigate the introduction of flexibility within proteins

through including additional degrees of freedom for bond rotations in Chapter 5. The introduction of flexibility is a step towards the development of a more comprehensive structure alignment tool.

Chapter 5

Structural flexibility in SAS-Pro

In the previous chapter, we presented a novel approach that combines the evaluation of the assignment and the rotation-translation variables through a single optimization problem. We further generalized the alignment procedure by eliminating the residue-sequentiality constraint, thus allowing for nonsequential alignments within the proteins under comparison.

Most alignment tools evaluate similarity within the proteins through rigid structure superposition of their structures. Thus, the alignments obtained do not account for flexibility within the proteins. Some alignment tools such as FlexProt [SNW02], FATCAT [YG03], ProtDeform [RSWD09], and FlexSnap [SZB10] address this issue by aligning smaller rigid fragments of proteins and joining them together, allowing for twists and turns in the overall alignment. However, these tools (except FlexSnap) are limited to providing only sequential structure alignments. SAS-Pro provides nonsequential alignments but is limited to rigid structural alignments. In this chapter, we extend the scope of SAS-Pros by introducing flexibility within the superimposing protein structures. This is achieved by allowing up to two bends within one of the protein structures under comparison. This increases the complexity of the structure

alignment problem by introducing additional variables within the structure alignment problem and making the RMSD and SAS similarity functions more nonlinear and nonsmooth.

In order to address the multi-modularity and nonlinearity of the RMSD and SAS function utilized in this approach, we employ and investigate several derivative-free optimization (DFO) techniques to determine suitable solutions. These DFO techniques provide near optimal solutions to the corresponding optimization problems. While these techniques perform quite well with small number of degrees of freedom (as in SAS-Pro), we investigate their performance under increasing number of degrees of freedom with a highly nonlinear and nonsmooth objective function obtained due to the inclusion of flexibility within protein structures. We determine the best DFO tool that may be utilized in the context of nonsequential and flexible protein structure alignment to provide good quality structural alignments.

In the remainder of this chapter, we discuss the mathematical model and the additional modifications to SAS-Pro in Section 5.1. Further, in Section 5.2 we present a brief description of the derivative-free solvers analyzed in this study. Finally, we conclude in Sections 5.3 and 5.4 with computational results and present an analysis of the computational experiments.

5.1 Mathematical model

Consider proteins A and B to be structurally aligned. Let a_i represent the i^{th} residue of protein A, and b_j represent the j^{th} residue of protein B. Also let $r(a_i)$ and $r(b_j)$ represent the 3D coordinates of the corresponding amino acid residues. We seek to align amino acid residues of A to amino acid residues of B so that when A is rotated-translated onto B, a similarity measure between

the two proteins is minimized. The main similarity function utilized in the present approach is the RMSD function. The RMSD function is defined as

$$\text{RMSD} = \sqrt{\left(\sum_i \sum_j S_{ij} ||\theta(r(a_i)) - r(b_j)||^2\right) / \left(\sum_i \sum_j S_{ij}\right)} \qquad (5.1)$$

Here, S_{ij} is a binary variable that equals 1 when a_i is aligned to b_j and 0 otherwise, and θ represents the rotation-translation and flexibility transformations applied to protein A. Thus, θ includes the three angles of rigid rotation, three rigid translation vector components, and the bend positions and three angles of bend for each bend introduced in protein A, as represented in Figure 5.1.

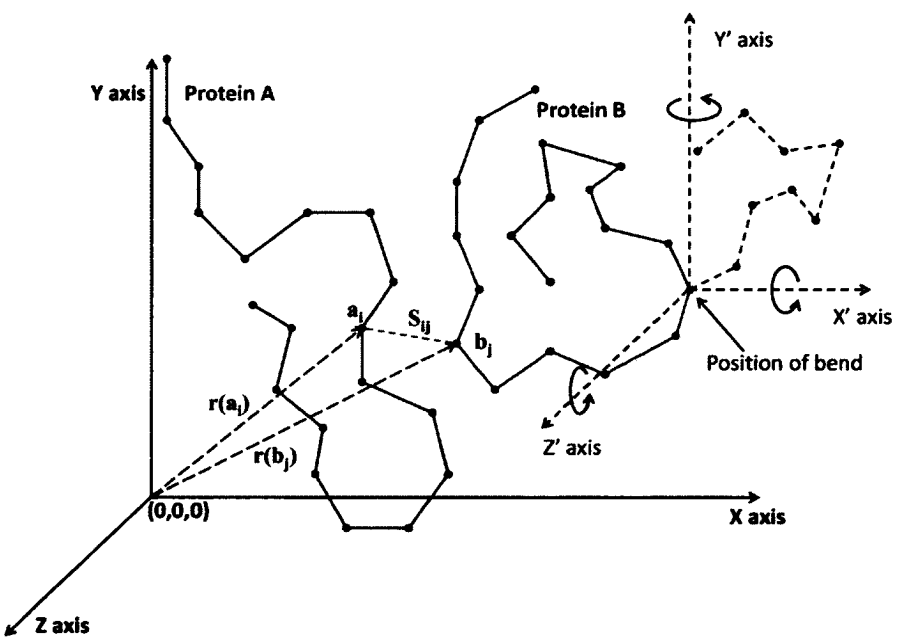


Figure 5.1: Schematic of variables and degrees of freedom in protein structure alignment

The SAS-Pro model that minimizes the RMSD function is presented as a single bilevel optimization problem as follows:

(SAS-Pro master problem)

$$\begin{aligned} \tau &= \min_{\theta} \{\min_{S} \text{RMSD}(S, \theta)\} \\ &= \min_{\theta} \text{SRMSD}(\theta) \end{aligned}$$

(SAS-Pro subproblem)

$$\begin{aligned} \text{SRMSD}(\theta) = \min_{S} \quad & \text{RMSD}(S, \theta) \\ \text{s.t.} \quad & \sum_{i} S_{ij} \leqslant 1 \quad \forall j \\ & \sum_{j} S_{ij} \leqslant 1 \quad \forall i \\ & \sum_{i} \sum_{j} S_{ij} \geqslant r_m \\ & S_{ij} \in \{0, 1\} \quad \forall i, j \end{aligned}$$

Here, r_m is the minimum number of residues that must be aligned to ensure that the global optimum attains a non-trivial value. For any given θ, the function SRMSD(θ) is defined as

$$\text{SRMSD}(\theta) = \min_{S} \text{RMSD} \tag{5.2}$$

The master problem objective function SRMSD(θ) is in the space of the θ variables alone. In this approach, the θ variables include the rotation and translation vectors for rigid body superposition as well as the variables for introducing flexibility within the protein structures.

Evaluation of the function SRMSD(θ) involves solving the subproblem and determining the optimal assignment variables S, for given values of θ and parameter r_m. We obtain the solution to the subproblem by solving the following k-cardinality linear assignment problem (k-LAP):

$$\textbf{(k-LAP)} \quad \kappa_\theta = \min_S \sum_i \sum_j a_{ij} S_{ij} \tag{5.3}$$

$$\text{s.t.} \quad \sum_i S_{ij} \leqslant 1 \quad \forall j$$

$$\sum_j S_{ij} \leqslant 1 \quad \forall i$$

$$\sum_i \sum_j S_{ij} \geqslant r_m \tag{5.4}$$

$$S_{ij} \in \{0,1\} \quad \forall i,j$$

where $a_{ij} = ||\theta(r(a_i)) - r(b_j)||^2$, $\forall i, j$. The θ variables in the original SAS-Pro model include only the rotation and translation variables for the superposition of rigid protein structures. We further introduce flexibility within the protein structures being compared by allowing up to 2 bends in the protein structures. This introduces four additional degrees of freedom per bend that includes a bend position (integer variable) and 3 angles of rotation (continuous variables) for the bend within protein A in the θ variable vector. Thus, the inclusion of bends in protein A allows a non-rigid superposition of the protein structures.

A highly efficient polynomial-time algorithm, SKAP [DLM01], is utilized to solve the k-LAP problem. The solution to the k-LAP problem will provide an assignment of exactly r_m amino-acid residues, as constrained in equation (5.4). The numerical value of SRMSD(θ) can be obtained from the objective value in equation (5.3) of the k-LAP problem as SRMSD(θ) = $\sqrt{\kappa_\theta / r_m}$. The k-LAP model does not include any sequence preserving constraints. Thus, the SAS-Pro model is designed to provide an optimal assignment and structure superposition of protein structures for specified values of the parameter r_m, with no sequence-preserving constraints.

The objective function utilized in this approach is highly nonlinear and

nonsmooth. This necessitates the utilization of optimization techniques which span the entire search space of the problem in the search for global optima. Furthermore, an explicit algebraic form for the objective function is not available, making it difficult to utilize derivative-based optimization methods. Hence, derivative-free optimization (DFO) techniques have been utilized to solve the master problem of the SAS-Pro model. DFO methods are often useful in obtaining near-global solutions for highly multi-modal and nonlinear functions.

In this study, we have utilized 22 different DFO solvers, based on a variety of techniques that included direct search, pattern search, surrogate management frameworks, domain partitioning methods, and stochastic algorithms [RS11]. These algorithms are designed to address the global optimization of nonlinear and nonsmooth functions with nonlinear and integer constraints and optimized for best performance with smaller number of variables.

While comparing proteins with similar sizes, we observed that the centroids of the superimposed proteins almost overlap with each other. Our data set includes a large number of protein pairs with comparable sizes. Thus, in order to analyze the performance of the DFO solvers with lower number of degrees of freedom, we introduced the assumption of overlapping centroids. This eliminated the translation degrees of freedom and reduced the number of degrees of freedom by 3. We explore the performance of the DFO solvers with and without the inclusion of this assumption. Further, inclusion of every bend introduced 4 more degrees of freedom per bend. Thus, we compare the performance of the DFO solvers with 3, 6, 7, 10, and 11 degrees of freedom. The value of the parameter r_m is set to the size of the smaller protein.

The protein structure alignment problem can also be addressed by optimizing the Structure Alignment Score (SAS score), instead of the RMSD function

in the MINLP formulation. The SAS function is defined as

$$\mathrm{SAS} = \frac{\mathrm{RMSD}}{N_{\mathrm{frag}}/100}$$

Here, the N_{frag} represents the total fragment length, defined as the sum of lengths of aligned continuous fragments of five or more aligned residues. The fragment length is defined for nonsequential alignments and is equivalent to the length of alignment for sequential alignments.

The bilevel optimization problem formulation of the SAS-Pro model can be easily reused with the SAS function as the objective. The master and subproblems of this reformulation are summarized below:

(SAS objective master problem)

$$\begin{aligned} \tau &= \min_{\theta} \{\min_{S} \mathrm{SAS}(S, \theta, r_m)\} \\ &= \min_{\theta} \quad \mathrm{SSAS}(\theta, r_m) \end{aligned}$$

(SAS objective subproblem)

$$\begin{aligned} \mathrm{SSAS}(\theta) = \min_{S} \quad & \mathrm{SAS}(S, \theta, r_m) \\ \text{s.t.} \quad & \sum_i S_{ij} \leqslant 1 \quad \forall j \\ & \sum_j S_{ij} \leqslant 1 \quad \forall i \\ & \sum_i \sum_j S_{ij} \geqslant r_m \\ & S_{ij} \in \{0, 1\} \quad \forall i, j \end{aligned}$$

where, for any given θ and r_m, the function $\mathrm{SSAS}(\theta, \mathrm{r_m})$ may be defined as

$$\mathrm{SSAS}(\theta, \mathrm{r_m}) = \min_{S} \mathrm{SAS} \tag{5.5}$$

We observe that a feasible solution to the subproblem for this reformation with the SAS objective function can also be obtained by solving the K-LAP problem described in Equations 5.3. The value of $\mathrm{SSAS}(\theta, \mathrm{r_m})$ for the feasible solution can be obtained from the optimal solution of the K-LAP problem as $\mathrm{SSAS}(\theta, \mathrm{r_m}) = \sqrt{\kappa_\theta / r_m} \times \frac{100}{N_{\mathrm{frag}}}$. The value of N_{frag} is dependent on the value of the parameter r_m. Thus, while using the SAS objective value, we include r_m as an additional degree of freedom along with the S and θ variables. The optimal value of r_m is determined from the solution of the master problem, along with the θ variables. Thus, while using the SAS objective function, the additional degree of freedom allows us to compare the performance of the DFO algorithms up to 12 degrees of freedom.

Note that the solution obtained from the k-LAP problem is only a feasible solution to the SAS-based SAS-Pro subproblem since the value of N_{frag} is not necessarily optimal for the corresponding k-LAP solution. Hence, while this approach produces very good quality solutions, the optimality of the solution obtained for the SAS objective function is not guaranteed. However, in practice, solutions from the k-LAP problem are observed to provide a very good estimate to the optimal sub-problem solution.

5.2 Derivative-free optimization solvers

We have compared the performance of 22 DFO solvers in the context flexible protein structure alignment. These encompass a variety of approaches to derivative-free optimization and their application to this highly nonlinear optimization problem. A list of all the solvers employed in this study in presented on Table 5.1.

Global model-based search methods utilize a surrogate model to guide the

Solver	Version	Language
ASA	26.30	C
BOBYQA	N/A	Fortran
CMA-ES	3.26.beta	Matlab
DAKOTA/DIRECT	4.2	C++
DAKOTA/EA	4.2	C++
DAKOTA/PATTERN	4.2	C++
DAKOTA/SOLIS-WETS	4.2	C++
DFO	2.0	Fortran
FMINSEARCH	N/A	Matlab
GLOBAL	1.0	Matlab
HOPSPACK	2.0	C++
IMFIL	0.86	Matlab
MCS	2.0	Matlab
NEWUOA	N/A	Fortran
NOMAD	3.3	C++
PSWARM	1.3	C, Matlab
SID-PSM	1.1	Matlab
SNOBFIT	2.1	Matlab
TOMLAB/GLCCLUSTER	7.3	Matlab
TOMLAB/LGO	7.3	Matlab
TOMLAB/MULTIMIN	7.3	Matlab
TOMLAB/OQNLP	7.3	Matlab

Table 5.1: Derivative-free solvers considered

optimization of the real model. In this study, we have included model-based search methods such as NEWUOA, Bound Approximation BY Quadratic Approximation (BOBYQA), and Radial Basis Function based optimization (TOMLAB/RBF).

Lipschitzian-based partitioning techniques construct and optimize an underestimator of the original objective function. By constructing this underestimator in a piecewise fashion, these methods provide possibilities for the global, as opposed to only local, optimization of the original problem. We have explored the branch-and-fit algorithm (SNOBFIT), the DIRECT algorithm based solver (TOMLAB/GLCCLUSTER), and a branch-and-bound based solver (TOMLAB-LGO), which are all based on this technique.

The list of solvers includes stochastic solvers such as Adaptive Simulated Annealing (ASA), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), genetic algorithms (DAKOTA/EA), and particle swarm algorithms (PSWARM). We also included other global optimization solvers such as GLOBAL, Hybrid Optimization Parallel Search PACKage (HOPSPACK), Multilevel coordinate search (MCS), TOMLAB/MULTIMIN and TOMLAB/OQNLP solvers.

Local search methods such as trust region algorithms (DFO), Mesh adaptive direct search methods (NOMAD), Nelder-Mead Simplex based methods (FMINSEARCH), pattern search algorithms (DAKOTA/PATTERN, SID-PSM), IMplicit FILtering (IMFIL), and greedy search algorithm (DAKOTA/SOLIS-WETS) were also explored in this study.

Thus, these solvers span all the different approaches to derivative-free optimization techniques. The SAS-Pro model with flexibility presents an opportunity to test these solvers to optimize a black-box non-smooth multimodal objective function.

5.3 Implementation and Results

We conducted a detailed computational study of the SAS-Pro model with the DFO solvers with the aim of analyzing the performance of the DFO solvers in obtaining near-optimal structure alignments, with different iteration limits and number of degrees of freedom. The experiments involved the Skolnick data set [LCWI01], which is a set of 40 large size proteins from five different fold families. An all-to-all pairwise alignment for all the proteins in the data sets is obtained, resulting in 850 pairwise alignment problems.

The performance of the solvers and the flexibility methodology is also tested with 10 problems from the RIPC data set. These problems present similar-

ities with conformational changes which are expected to align well with the inclusion of flexibility within protein structures.

In this study we compare the performance of the DFO solvers for both the RMSD and the SAS similarity measures as objective functions. Table 5.2 presents the variable types and bounds for all the variables utilized in the flexibility model. The formulation of the SAS-Pro model with the SAS objective also includes the parameter r_m as a variable which takes integer values. Thus, the additional integer variable makes the corresponding objective more non-smooth and difficult to optimize as compared to RMSD objective function with identical problem specifications.

Variable	Type	Lower bound	Upper bound
RMSD	objective	N.A.	N.A.
SAS	objective	N.A.	N.A.
Rotation angles	continuous	$-\pi$	π
Translation vectors	continuous	-100	100
Bend angles	continuous	$-\pi$	π
Bend position	integer	1	$min(P_1, P_2)$
r_m	integer	$0.85 \times min(P_1, P_2)$	$min(P_1, P_2)$

Table 5.2: Solver settings for the DFO solvers. P_1, P_2 represent the sizes of the two proteins.

The 22 DFO solvers were analyzed for varying number of degrees of freedom and iteration limits. The flexibility of the protein structures is included a variable number of degrees of freedom. Structure alignment problems with no flexibility involve smaller number of degrees of freedom (less than 6), while those where flexibility within the protein structures is included have larger number of degrees of freedom (more than 7). Each degree of freedom is analyzed for a limit of 500, 1000, 5000 and 10000 iterations.

5.3.1 Skolnick data set

We first compare the performance of the DFO solvers with respect to their computational requirements and the ability to provide near-optimal solutions with varying number of iterations. Since the exact optimal values for these problems are unknown, we evaluate the performance of the solvers on the basis of the best solution obtained for different number of iterations.

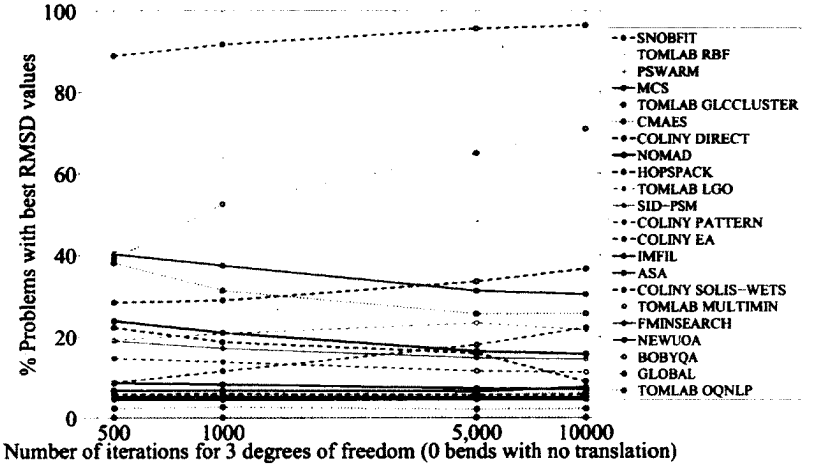


Figure 5.2: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained vs. number of iterations with RMSD objective for 3 degrees of freedom

Figures 5.2, 5.3, 5.4, and 5.5 represent the percentage of problems for which the 22 DFO solvers provided the best RMSD values vs different number of iterations for varying degrees of freedom. Figures 5.2 represents the protein structure alignment problems without allowing any flexibility within the proteins. The corresponding RMSD objective function is smooth due to the absence of any integer variables. For these problems with 3 and 6 degrees of freedom, the SNOBFIT solver provides better solutions than any other DFO solver for more than 80% of the alignment problems. This observation is also consistent with the utilization of the SNOBFIT solver in SAS-Pro for best performance.

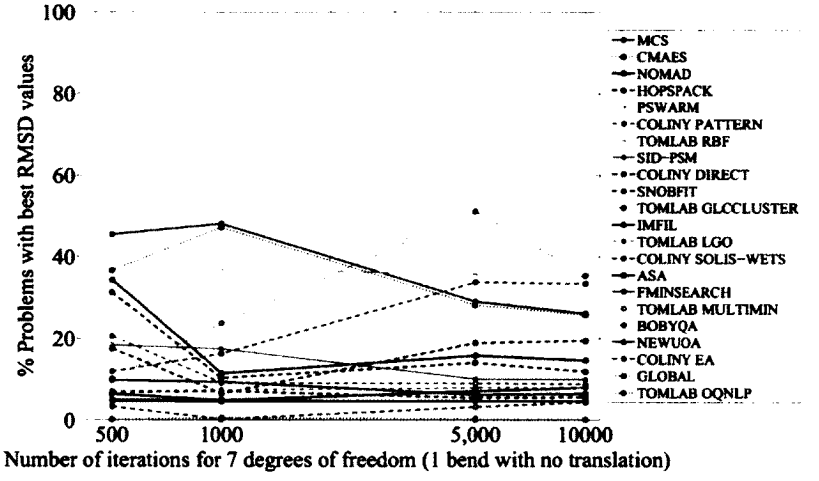


Figure 5.3: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained vs. number of iterations with RMSD objective for 7 degrees of freedom

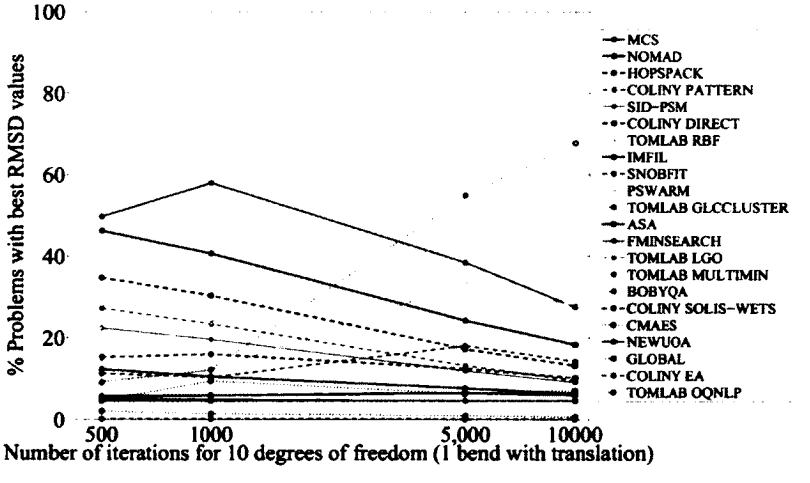


Figure 5.4: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained vs. number of iterations with RMSD objective for 10 degrees of freedom

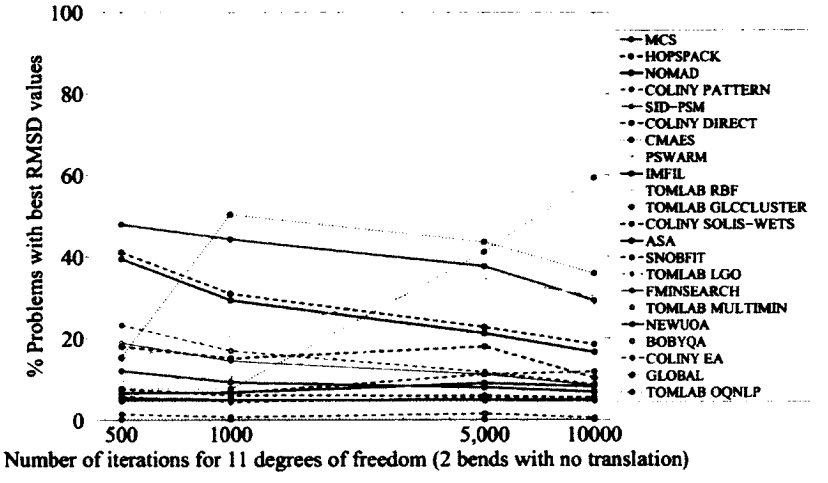


Figure 5.5: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained vs. number of iterations with RMSD objective for 11 degrees of freedom

The performance of TMOLAB/RBF and TOMLAB/GLCCLUSTER are the next best to SNOBFIT. This ranking of the DFO solvers is consistent with the observations of Rios and Sahinidis [RS11] for smooth optimization problems with low number of degrees of freedom.

Figures 5.3, 5.4, and 5.5 present the performance of the structure alignment problem with the inclusion of flexibility within the protein structures. The RMSD objective function here is nonsmooth due to the inclusion of discrete degrees of freedom representing the bend positions . For these problems with 7 or more degrees of freedom and nonsmooth objective function, the MCS, PSWARM, and CMA-ES solvers are observed to provide the best solution for the largest number of problems. However, for these large number of degrees of freedom, the differences in the performance of the DFO solvers are much smaller. This is also consistent with the observations of Rios and Sahinidis [RS11] for the case of nonsmooth optimization problems with increasing number of degrees of freedom.

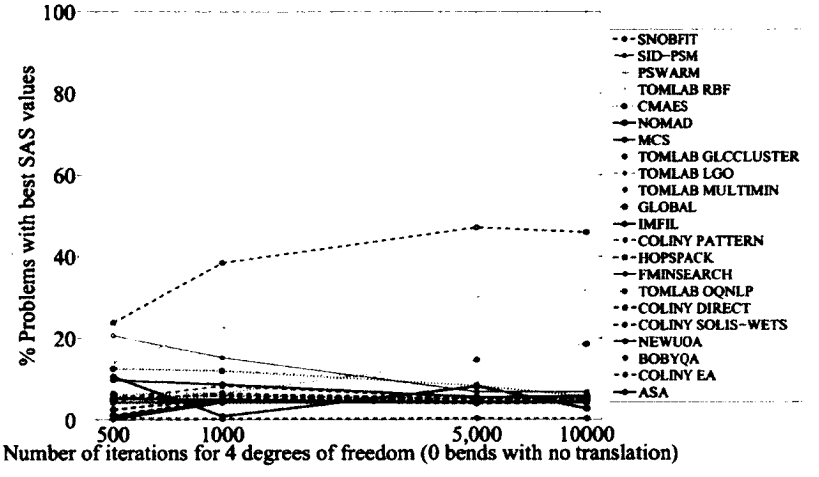


Figure 5.6: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained vs. number of iterations with SAS objective for 4 degrees of freedom

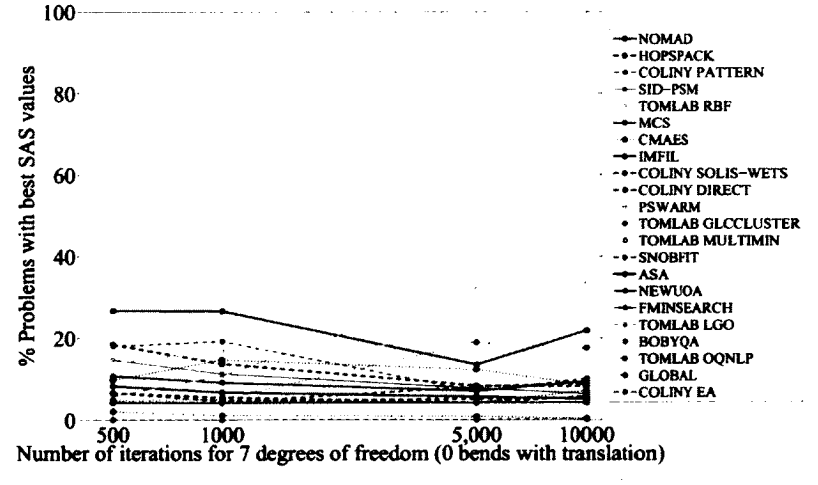


Figure 5.7: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained vs. number of iterations with SAS objective for 7 degrees of freedom

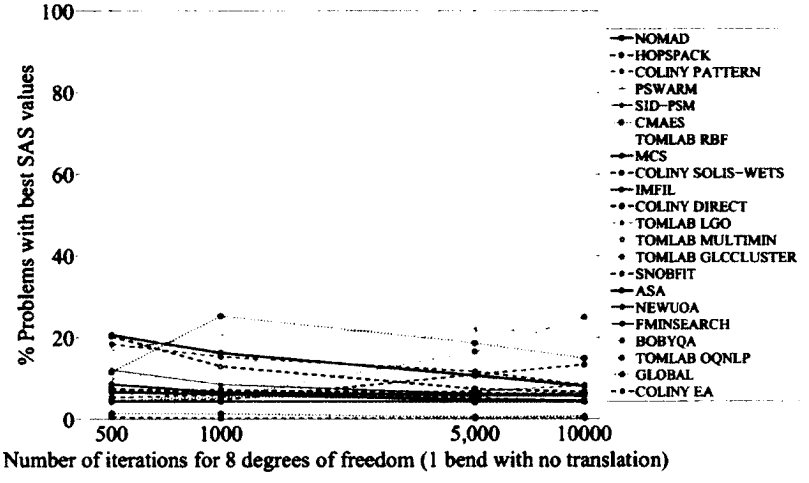


Figure 5.8: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained vs. number of iterations with SAS objective for 8 degrees of freedom

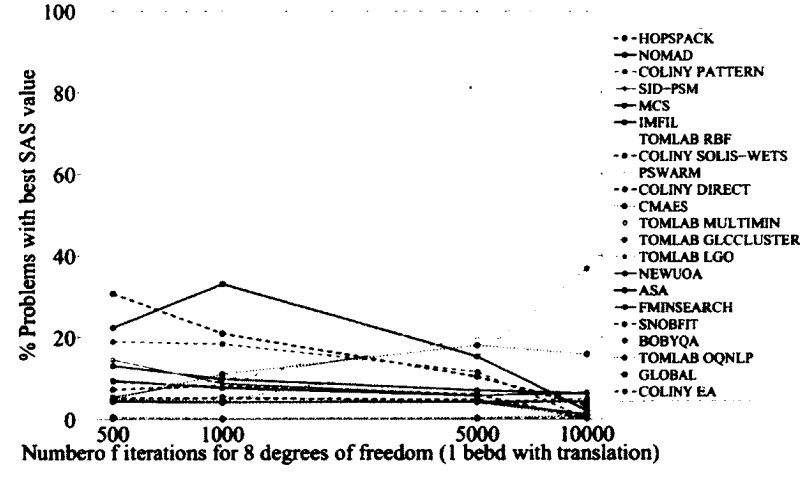


Figure 5.9: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained vs. number of iterations with SAS objective for 11 degrees of freedom

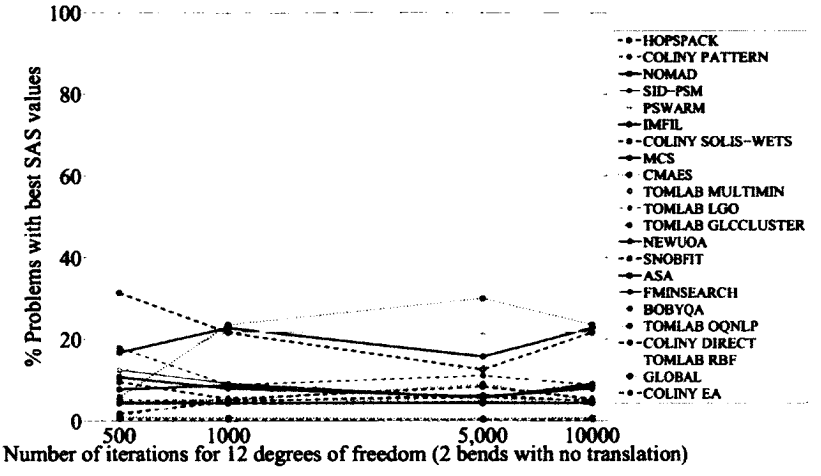


Figure 5.10: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained vs. number of iterations with SAS objective for 12 degrees of freedom

Figures 5.6, 5.7, 5.8, 5.9, and 5.10 represent the percentage of problems for which the 22 DFO solvers provided the best SAS values vs different number of iterations for varying degrees of freedom. The SAS objective function is nonsmooth due to the inclusion of r_m that takes integer values. Hence, optimizing the SAS function is more challenging than optimizing the RMSD function for identical problem specifications. Figures 5.6 and 5.7 show that even with the SAS objective, for a small number of degrees of freedom, the SNOBFIT solver provides better solutions than any other DFO solvers. However, the differences in the performances of the DFO solvers for are much smaller, indicating no single superior DFO solver. With larger number of degrees of freedom, the performance of the SNOBFIT solver deteriorates and the solvers TOMLAB/RBF, TOMLAB/GLCCLUSTER, PSWARM and CMA-ES show consistent good performance in obtaining near optimal solutions.

The SNOBFIT solver is based on a branch-and-fit approach that utilizes intelligent space branching techniques along with local quadratic fits for the

objective function. The solver utilizes a full stochastic quadratic model for the local fits of the objective function. Since this involves $O(n^6)$ operations for the linear algebra, this limits the number of variables that can be handled without excessive overhead. Hence for larger number of degrees of freedom, the SNOBFIT solver possibly times out before producing a good solution to the structure alignment problems. Stochastic solvers thus outperform the SNOBFIT solver for larger number of degrees of freedom.

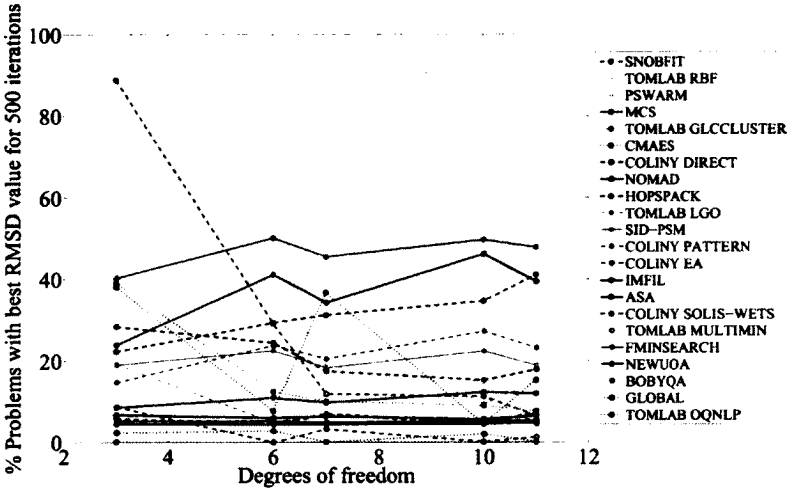


Figure 5.11: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained w.r.t. number of degrees of freedom with RMSD objective for 500 iterations

The comparison of different DFO solvers w.r.t. number of degrees of freedom for different number of iterations is shown in Figures 5.11, 5.12, 5.13, and 5.14. The objective function used here in the RMSD function. It is observed that amongst all the solvers, the MCS, CMA-ES, PSWARM and TOMLAB/GLCCLUSTER consistently provide the best performances for different number of iterations.

A similar analysis for the SAS objective function is presented in Figures 5.15, 5.16, 5.17, and 5.18. The figures indicate that most DFO solvers

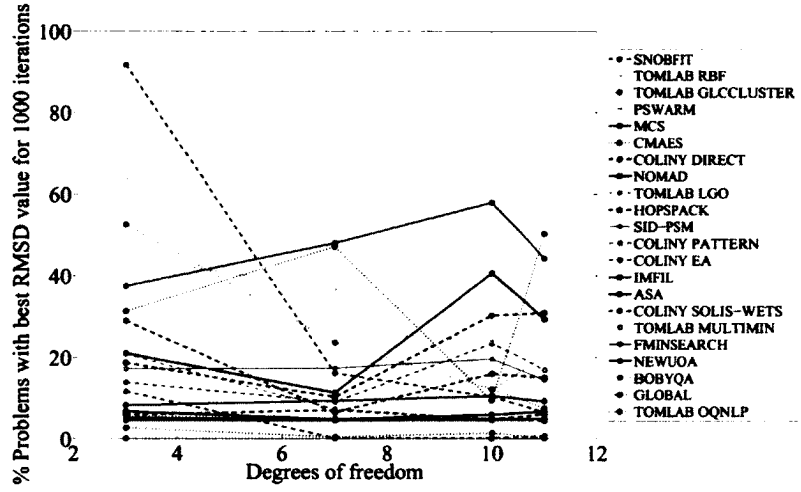


Figure 5.12: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained w.r.t. number of degrees of freedom with RMSD objective for 1000 iterations

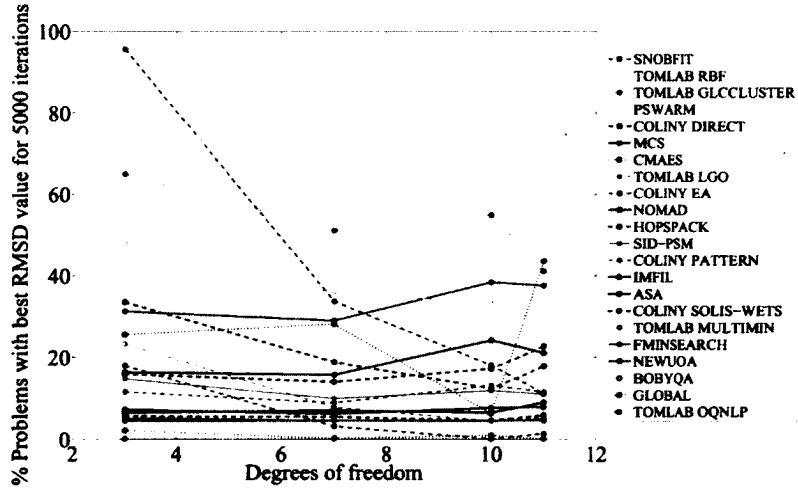


Figure 5.13: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained w.r.t. number of degrees of freedom with RMSD objective for 5000 iterations

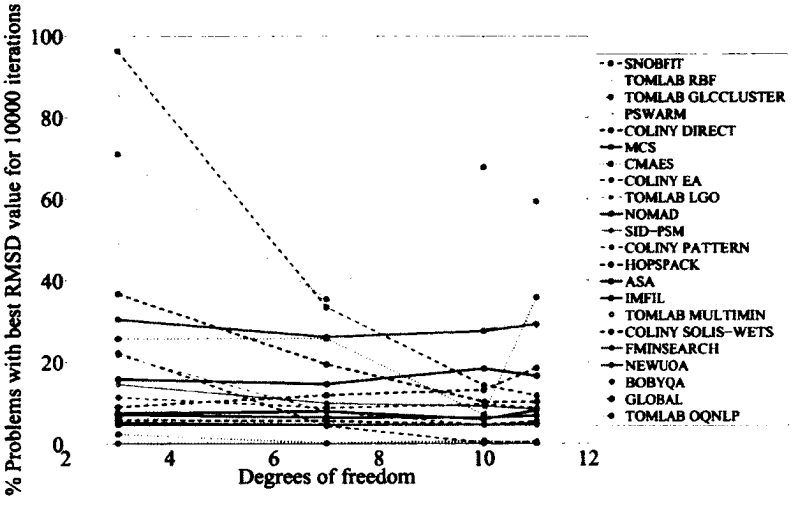


Figure 5.14: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained w.r.t. number of degrees of freedom with RMSD objective for 10000 iterations

perform similarly for different degrees of freedom and number of iterations. PSWARM and CMA-ES are observed to perform slightly better than other DFO solvers providing the best solution for about 20% of the problems. However, no one solver is observed to provide the best performance in all cases.

We further analyze the quality of solutions obtained from solving the flexibility model using the DFO solvers. It is observed that after the inclusion of the flexibility related degrees of freedom, none of the DFO solvers provide optimal solutions to the structure alignment problem within a limit of 10000 iterations. However, the near-optimal solutions provided by these solvers are observed to characterize similarity between the protein pairs accurately. Also, amongst the RMSD and SAS objective functions, the solutions obtained using the SAS function are observed to possess larger N_{frag} and lower SAS values, indicating more biologically relevant alignments. This may be true due to inclusion of r_m as a variable while using the SAS objective which allows sampling of a larger variety of alignments as compared to only one value of r_m utilized

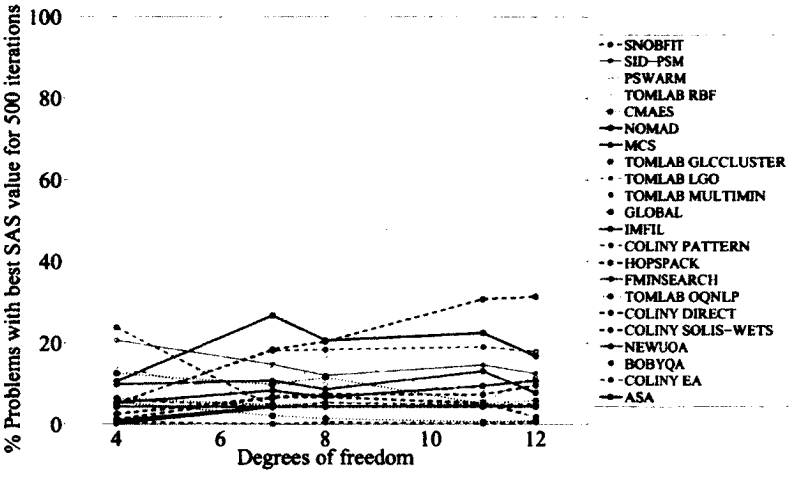


Figure 5.15: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained w.r.t. number of degrees of freedom with SAS objective for 500 iterations

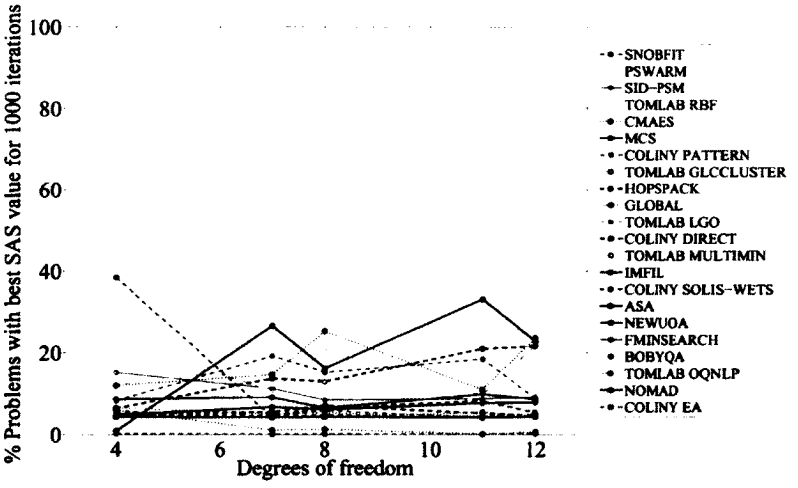


Figure 5.16: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained w.r.t. number of degrees of freedom with SAS objective for 1000 iterations

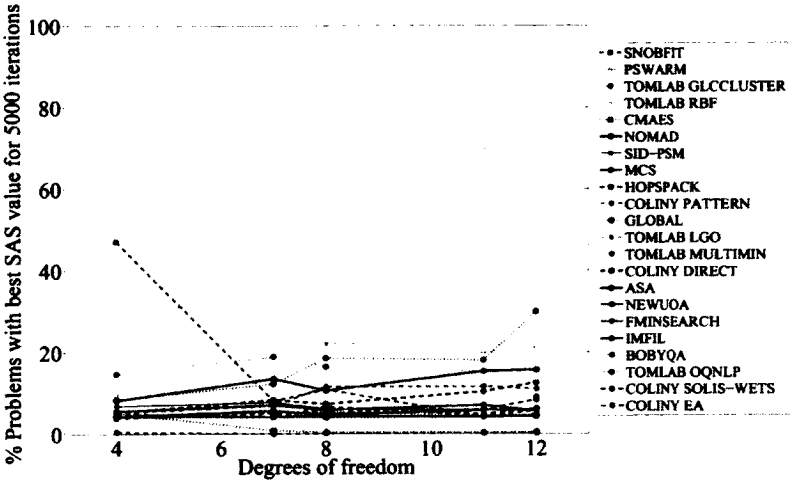


Figure 5.17: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained w.r.t. number of degrees of freedom with SAS objective for 5000 iterations

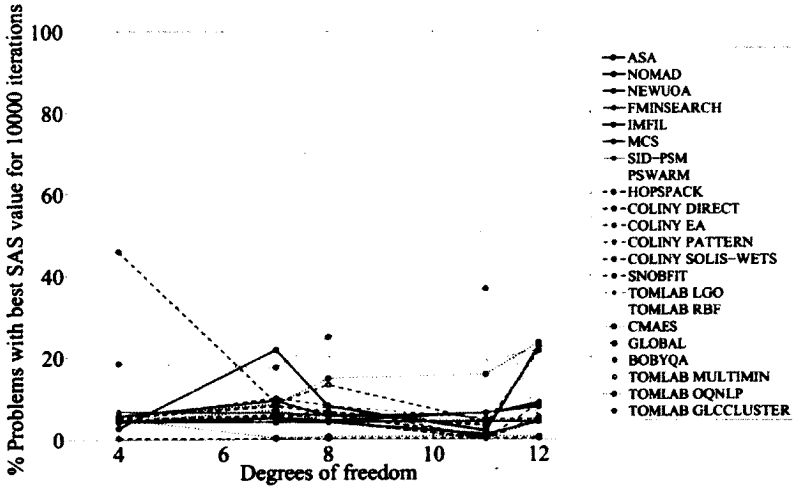


Figure 5.18: Comparison of performance of 22 DFO solvers. Number of problems for which best solution was obtained w.r.t. number of degrees of freedom with SAS objective for 10000 iterations

in the RMSD objective based analysis. Thus, for future computations, the flexible structure alignments are obtained using the SAS objective function.

5.3.2 RIPC data set

Based on the analysis of the Skolnick data set, we concluded that the use of SAS objective function provides more biologically relevant alignments than minimizing the RMSD function alone. We have thus evaluated the alignments in the RIPC data set by using the SAS objective function. Also, the analysis indicates that the performance of six DFO solvers–MCS, PSWARM, CMA-ES, SNOBFIT, TOMLAB/RBF, and TOMLAB/GLCCLUSTER, dominates the performance curves for different degrees of freedom and different number of iteration limits. Hence, we have compared the alignments obtained from only these six solvers.

Table 5.3 shows the results obtained for the ten alignment problems after the introduction of flexibility constraints. As observed here, for eight of the ten problems we now obtain 100% agreement with the reference alignments. For the remaining two, the alignment obtained is in much better agreement than SAS-Pro without any flexibility. We believe that not all problems present a 100% agreement with the reference alignment since the SAS-Pro with flexibility models are not solved to optimality by the DFO solvers due to increased complexity caused by additional degrees of freedom.

We observe that there is no one solver that provides the best solution for all the ten alignment problems. The best performance is obtained from the CMA-ES, PSWARM and SNOBFIT solvers. These solvers collectively provide the best solutions for nine of the ten problems with the best SAS values and best agreement with the reference alignments.

Further, as an illustrative example of the performance of the SAS-Pro tool

Protein	SAS-Pro		SAS-Pro with flexibility			
pair	SAS	% ref match	SAS	% ref match	solver	no. bends
1b5t-1k87	3.07	62.5	2.28	100	CMA-ES	1
1gsa-2hgs	4.1	100	2.8	100	CMA-ES	0
1jwy-1puj	8.23	37.5	7.5	50	PSWARM	1
1jwy-1u01	5.07	27.3	5.5	50	CMA-ES	2
1kia-1nw5	8.83	58.33	3.3	100	TOMLAB/RBF	1
1nkl-1qdm	5.85	100	3.5	100	SNOBFIT	0
1nls-2bqp	3.1	100	1.14	100	PSWARM	0
1nw5-2adm	5.5	53.8	4.5	100	CMA-ES	2
1qas-1rsy	2.72	100	1.09	100	PSWARM	1
1qq5-3chy	3.4	100	3.4	100	SNOBFIT	0

Table 5.3: Results for the RIPC data set for flexible protein structure alignment. SAS measures are in Å and % ref match represents % agreement with the reference alignment.

with flexibility, we present an alignment between the 1TOP and 2BBM proteins. These two proteins are known to be similar to each other through a conformational change resulting in two bends in the 1TOP protein. In Figure 5.3.2, we show the corresponding alignments after allowing zero, one and two bends in the flexible version of SAS-Pro. As observed in Figure 5.3.2(a), the alignment obtained with zero bends is an average alignment that minimizes the RMSD/SAS values without properly aligning the biologically relevant parts. After the introduction of one bend, Figure 5.3.2(b) indicates that the number of aligned residues increased considerably. However, a small part of the two proteins is still misaligned. After the inclusion of two bends, Figure 5.3.2(b) shows that most parts of the protein superimpose very well with each other, resulting in a very good alignment between the proteins that clearly depicts the similarity of the proteins.

(a) 0 bends **(b) 1 bend** **(c) 2 bends**

Figure 5.19: Alignment of 1TOP protein with 2BBM protein using SAS-Pro with flexibility. The alignment obtained after allowing (a) 0 bends, (b) 1 bend, and (c) 2 bends.

5.4 Conclusions

Our analysis on the performance of several DFO solvers indicates that, for up to 6 degrees of freedom, SNOBFIT provides the best solutions. As the number of degrees of freedom increases, the stochastic solvers CAM-ES and PSWARM, as well as the deterministic solvers MCS, TOMLAB/RBF and TOMLAB/GLCCLUSTER provide the best solutions among all the solvers. While these solvers are unable to provide optimal solutions within a limit 10000 iterations, the solutions obtained from these six solvers provide good quality biologically relevant alignments. Further, the alignments obtained using the SAS objective function are observed to produce alignments with larger fragment lengths than RMSD, thus providing more biologically relevant alignments as compared to the RMSD objective function.

Analysis of the RIPC data set concludes that the SAS objective function

produces good quality alignments for some of the flexible protein structure alignment problems included the data set. We obtain a 100% agreement with the reference alignment for eight out of ten problems, while improving the alignment for the remaining two problems to 50% as compared to 35% without flexibility. Thus, while the DFO solvers may not produce the optimal structure alignments, the near-optimal solutions obtained from the solvers provide a good estimate of the flexible structure alignments.

For rigid structural alignments, excellent quality alignments with very close to optimal solutions are obtained using the original SAS-Pro alignment tool. The flexibility option for SAS-Pro will provide considerably better alignments only in cases where flexibility-affected alignments are anticipated.

Chapter 6

Conclusions

In this chapter, we first present conclusions of the thesis and highlight our specific contributions chapter-wise. Then, we present recommendations for future work.

6.1 Thesis conclusions and contributions

In this thesis, we have studied two approaches to the protein structure alignment problem, especially addressing the challenges of computational performance, allowance of flexibility, and nonsequential alignments.

In Chapter 2, we presented a comprehensive review of the structure alignment tools developed in the past three decades. We classified these tools based on protein structure representations, and analyzed their strengths and limitations in terms of computational speed and alignment accuracy. We also examined the different similarity criteria used for evaluating quality of structure alignments. The main conclusions from this chapter are as follows:

- For sequential alignments, dynamic programming techniques have proven to be effective.

- For nonsequential alignments, continuous alignment methods are most promising.
- The problem of fast nonrigid and nonsequential alignment of proteins is a challenge in previous literature.

In Chapter 3 we enhanced the computational efficiency of the state-of-the-art structure alignment tool CMOS. The computational performance of CMOS is governed by effective reduction and bounding schemes. We introduced physical property information constraints as additional reduction schemes. These schemes introduce approximations in an otherwise exact algorithm but resulted in a five-fold reduction of the computational requirements of the CMOS algorithm. Furthermore, we demonstrate that this increase in computational efficiency leads to solutions of more complex structure alignment problems which were unsolvable earlier. The improved CMOS algorithm provides near-optimal solutions for over 80% problems in the Sokol and Skolnick data sets that were previously unsolved. Finally, the inclusion of the physical property constraints provided more biologically relevant alignments, and eliminated isolated residue matches.

In Chapter 4, we presented a novel reformulation of the protein structure alignment problem in the form of a single bilevel optimization problem that addresses the assignment of amino acid residues and the structure superposition of proteins simultaneously. This model allows for both sequential and nonsequential structure alignments. We demonstrated via computational experiments that the proposed model accurately captured similarities among the benchmark protein data set cases. Based on this model, we introduced the structure alignment tool SAS-Pro that employs derivative-free optimization techniques to obtain quickly near-optimal solutions. SAS-Pro provides superior alignments with lower RMSD values and larger lengths of alignments

when compared to the commonly used structure alignment tools CE, SSM, and STSA for over 50% problems in the Sokol and Skolnick data sets. Moreover, SAS-Pro exhibits excellent performance for the RIPC data, set which comprises of complex nonsequential structure alignment problems. For this set, SAS-Pro provides alignments with 100% agreement with the reference for eight of the 23 protein pairs. In this sense, it does better than the commonly used alignment tools CE, DALI, FATCAT, MATRAS, CA, SHEBA, SARF, and LGA. In addition, for the same data set, SAS-Pro produces a median agreement of 70%, which is better than these other tools, while matching the performance of the STSA algorithm. The computational requirements of SAS-Pro are low, requiring on average 1 CPU minute per protein pair on an Intel Quad Core 2.83 GHz processor with 6 GB RAM.

In Chapter 5, we expanded the scope of SAS-Pro through introduction of flexibility within protein structures under comparison. We performed an extensive computational analysis of 22 derivative-free optimization solvers in the context of this complex nonlinear nonsmooth optimization problem with added degrees of freedom to determine the most suitable solution approach. This extensive analysis provided systematic comparisons for the performance evaluation of solvers for optimizing black-box nonsmooth nonlinear objective functions, an area in which no previous comparisons have been presented for such a large collection of solvers on black-box models. The results demonstrated that SAS-Pro provides excellent quality alignments for a set of ten similar and flexible protein pairs.

In conclusion, the SAS-Pro approach provides an alignment tool capable of providing near-optimal nonsequential flexible protein structure alignments with low computational requirements. An implementation of SAS-Pro without the flexibility option is freely available online for download at

http://eudouxs.cheme.cmu.edu/saspro/SAS-Pro.html. We plan to introduce the flexibility alignment option in future releases of SAS-Pro.

6.2 Future directions

In this dissertation, we have proposed novel optimization solutions for addressing complex protein structure alignment problems. The following research directions can further improve our understanding of the structure alignment problem, and make further progress toward a more comprehensive structure alignment tool.

6.2.1 Enhancements to the CMOS algorithm

The increased computational efficiency of CMOS has increased the applicability of the algorithm to larger and more difficult alignment problems. However, the algorithm is still not efficient enough for an all-to-all comparison of PDB protein structures. The following strategies may be examined in this context.

- SSE expansion: The current SSE reduction scheme utilizes only $\alpha-$helices and $\beta-$strands for eliminating amino acid residue matches. More specific secondary structure types, such as different types of $\alpha-$helices, $\beta-$strands, bends, loops and coils, may also be utilized to develop a more comprehensive reduction scheme. Further, Kolodny et al. [KKGL02] has suggested a set of over 200 commonly found structural features in protein structures that may also be used as a secondary structure library to develop a more comprehensive reduction scheme.

- Upper bounding: It is observed in the CMOS algorithm that, while the lower bound obtained by CMOS reaches the optimal value in a small

number of iterations, the upper bound of CMOS is not tight enough, resulting in a large number of iteration for convergence. Developing better upper bounding techniques would result in a considerable improvement in the performance of the CMOS algorithm.

6.2.2 Applications and enhancements of the SAS-Pro tool

The SAS-Pro tool introduced in this thesis provides a generalized framework for complex protein structure alignment problems. It would be interesting to perform a comprehensive analysis of nonsequentiality and flexibility in the protein database [pdb] using the SAS-Pro model with suitable DFO solvers. Also, our current computational experiments provide excellent results with homologous protein pairs. It would be interesting to investigate the performance and utility of SAS-Pro towards identifying similarities amongst distantly related protein structures as well. Our current analysis indicates that a comparison of representative proteins from the the fold families in the SCOP database with the remaining proteins in the PDB database would require about 145 CPU years for a complete comparison. This may be further reduced considerably through intelligent choice of protein pairs to compare and parallelization of the comparison process.

The SAS-Pro framework may be further enhanced by addressing the following issues.

- Solution optimality: The objective function of the SAS-Pro model represents the value function of an integer program. A careful theoretical analysis of this function could be used to determine a sufficient domain resolution that could be utilized in the context of a branch-and-bound

algorithm for exact solutions.

- Parallel computation: Since the SAS-Pro approach is based on search techniques (DFO solvers), the algorithm may be parallelized for reducing the computational time requirements of the solver. Considerable computational gains may be obtained through parallelization over partitions of variable space of the problem or different values of the parameter r_m.

- Variable reduction: Pre-computed information, such as secondary structure types, known protein relationships, and information about bend positions, may also be incorporated in the algorithm for more biologically relevant alignments and improved computational performance.

Bibliography

[AF96] N. N. Alexandrov and D. Fischer. Analysis of topological and nontopological structural similarities in the PDB: new examples with old structures. *Proteins*, 1996.

[AGM90] S. F. Altschul, W. Gish, and W. Miller. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[AM08] R. Andreani and J. M. Martnez. Trust-region superposition methods for protein alignment. *IMA Journal of Numerical Analysis*, 28:690–710, 2008.

[AMMY08] R. Andreani, J. M. Martnez, L. Martnez, and F. Yano. Continuous optimization methods for structure alignments. *Math Programming Ser. B*, 112:93–124, 2008.

[AS97] S. F. Altschul and A. A. Schaffer. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

[ATG92] N. N. Alexandrov, K. Takahashi, and N. Go. Common spatial arrangements of backbone fragments in homologous and non-homologous proteins. *Journal of Molecular Biology*, 225:5–9, 1992.

[BBC06] S. Bhattacharya, C. Bhattacharyya, and N. Chandra. Projections for fast protein structure retrieval. *BMC Bioinformatics*, 7:S5–S17, 2006.

[BHB+07] D. Barthel, J. D. Hirst, J. Blazewicz, E. K. Burke, and N. Krasnogor. ProCKSI: A decision support system for protein (structure) comparison, knowledge, similarity and information. *BMC Bioinformatics*, 2007.

[CCI+04] A. Caprara, R. Carr, S. Istrial, G. Lancia, and B. Walenz. 1001 optimal PDB structure alignments: Integer programming methods for finding the maximum contact map overlap. *Journal of Computational Biology*, 11:27–52, 2004.

[CHK+02] B. Carr, W. Hart, N. Krasnogor, J. Hirst, E. Burke, and J. Smith. Alignment of protein structures with a memetic evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1027–1034, 2002.

[CL02] A. Caprara and G. Lancia. Structural alignment of large-size proteins via lagrangian relaxation. In *Proceedings of the International Conference on Computational Biology (RECOMB)*, pages 100–108, 2002.

[CL04] R. D. Carr and G. Lancia. Compact optimization can outperform separation: A case study in structural proteomics. *4OR: A Quarterly Journal of Operations Research*, 2:221–233, 2004.

[CLI00] R. D. Carr, G. Lancia, and S. Istrail. Branch-and-cut algorithms for independent set problems: Integrality gap and an application to protein structural alignment. Technical report, Sandia National laboratories, 2000. Sandia Report SAND2000-2171.

[DLM01] M. Dell'Amico, A. Lodi, and S. Martello. Efficient algorithms and codes for k-cardinality assignment problems. *Dis. App. Math.*, 110:25–40, 2001.

[FC96] A. Falicov and F. E. Cohen. A surface of minimum area metric for the structural comparison of protein. *Journal of Molecular Biology*, 258:871–892, 1996.

[FKR+70] S. T. Freer, J. Kraut, J. D. Robertus, H. T. Wright, and Ng. H. Xuong. Chymotrypsinogen: 2.5-A Crystal Structure, Comparison with α-Chymotrypsin, and Implications for Zymogen Activation. *Biochemistry*, 9:1997–2009, 1970.

[GK08] A. Guerler and E. W. Knapp. Novel protein folds and their nonsequential structural analogs. *Protein science*, 17:1374 – 1382, 2008.

[GKS93] A. Godzik, A. Kolinski, and J. Skolnick. Lattice representations of globular proteins: How good are they? *Journal of Computational Chemistry*, 14:1194–1202, 1993.

[GL96] M. Gerstein and M. Levitt. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. In D. States, P. Agarwal, T. Gaasterland, L. Hunter, and R. Smith, editors, *Proceedings of international conference on intelligent systems in molecular biology*, pages 59–67. AAAI Press, 1996.

[GMB96] J. F. Gibrat, T. Madej, and S. H. Bryant. Surprising similarities in structure comparison. *Current Opinion in Structural Biology*, pages 377–385, 1996.

[GPI99] D. Goldman, C. Papadimitriou, and S. Istrail. Algorithmic aspects of protein structure similarity. In *Proceedings of the 40th annual symposium on foundations of computer science (FOCS)*, pages 512–522. IEEE Computer society, 1999.

[Gra04] J. Gramm. A polynomial-time algorithm for the matching of crossing contact-map patterns. *IEEE/ACM transaction on computational biology and bioinformatics*, 1:171–180, 2004.

[GS94] A. Godzik and J. Skolnick. Flexible algorithm for direct multiple alignment of protein structures and sequences. *Computer applications in biosciences: CABIOS*, 10:587–596, 1994.

[GSK93] A. Godzik, J. Skolnick, and A. Kolinski. Regularities in interaction patterns of globular proteins. *Protein Engineering*, 6:801–810, 1993.

[HESF71] R. Huber, O. Epp, W. Steigemann, and H. Formanek. The Atomic Structure of Erythrocruorin in the Light of the Chemical Sequence and its Comparison with Myoglobin. *European journal of biochemistry*, 19:42–50, 1971.

[HN08] W. Huyer and A. Neumaier. SNOBFIT–Stable noisy optimization by branch and fit. *ACM Transactions on Mathematical Software*, 35:1–25, 2008.

[HP00] L. Holm and J. Park. Dalilite workbench for protein structure comparison. *Bioinformatics Applications Note*, 16:566–567, 2000.

[HS93] L. Holms and C. Sander. Protein-structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, 233:123–138, 1993.

[HS96] L. Holm and C. Sander. The fssp database: Fold classification based on structure-structure alignment of proteins. *Nucleic Acids Research*, 24:206–209, 1996.

[IAL04] V. A. Ilyin, A. Abyzov, and C. M. Leslin. Structural alignment of proteins by a novel topfit method, as a superimposition of common volumes at a topmax point. *Protein Science*, 13:1865–1874, 2004.

[JL07] B. J. Jain and M. Lappe. Joining softassign and dynamic programming for the contact map overlap problem, 2007.

[JO09] B. J. Jain and K. Obermayer. Structure spaces. *Journal of Machine Learning*, 10:2667–2714, 2009.

[KD82] J. Kyte and R. F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157:105–132, 1982.

[KH04] 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*, 60:2256–2268, 2004.

[KJ94] G. J. Kleywegt and A. Jones. Superposition. *CCP4/ESF-EACBM Newsletter Protein Crystallog.*, 31:9–14, 1994.

[KKGL02] Rachel Kolodny, Patrice Koehl, Leonidas Guibas, and Michael Levitt. Small libraries of protein fragments model native protein structures accurately. *Journal of Molecular Biology*, 323(2):297 – 307, 2002.

[KKL05] R. Kolodny, P. Koehl, and M. Levitt. Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *Journal of Molecular Biology*, 346:1173–1188, 2005.

[KLL04] R. Kolodny, N. Linial, and M. Levitt. Approximate protein structural alignment in polynomial time. *Proc. Natl. Acad. Sci. USA*, 101:12201–12206, 2004.

[KMSG+06] B. Kolbeck, P. May, T. Schmidt-Goenner, T. Steinke, and E. W. Knapp. Connectivity independent protein-structure alignment: a hierarchical approach. *Bioinformatics*, 7:510–529, 2006.

[KP04] N. Krasnogor and D. A. Pelta. Measuring the similarity of protein structures by means of the universal similarity metric. *Bioinformatics*, 20:1015–1021, 2004.

[Kra04] N. Krasnogor. Self generating metaheuristics in bioinformatics: The protein structure comparison case. *Genetic Programming and Evolvable Machines*, 5:181–201, 2004.

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

[LCWI01] G. Lancia, R. Carr, B. Walenz, and S. Istrail. 101 optimal PDB structure alignments: A Branch-and-cut algorithm for the maximum contact map overlap problem. In *Proceedings of the fifth annual international conference on Computational biology, Montreal, Quebec, Canada*, pages 193–202, 2001.

[LFM+10] P. D. Lena, P. Fariselli, L. Margara, M. Vassura, and R. Casadio. Fast overlapping of protein contact maps by alignment of eigenvectors. *Bioinformatics*, 26:2250–2258, 2010.

[LG98] M. Levitt and M. Gerstein. A unified statistical framework for sequence comparison and structure comparison. *Proc. Natl. Acad. Sci.*, 95:5913–5920, 1998.

[LKR03] J. Leluk, L. Konieczny, and I. Roterman. Search for structural similarity in proteins. *Bioinformatics*, 19:117–124, 2003.

[LKSD00] P. Lackner, W. A. Koppensteiner, M. J. Sippl, and F. S. Domingues. ProSup: A refined tool for protein structure alignment. *Protein Engineering*, 11:745–752, 2000.

[LLMA05] D. Lupyan, A. Leo-Macias, and R. Ortiz A. A new progressive-iterative algorithm for multiple structure alignment. *Bioinformatics*, 21:3255–3263, 2005.

[Mar00] A. C. R. Martin. The ups and downs of protein topology; rapid comparison of protein structure. *Protein Engineering*, 13:829–837, 2000.

[MBHC95] 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*, 247:536–540, 1995.

[McL82] A. D. McLachlan. Rapid comparison of protein structures. *Acta. Cryst.*, A38:871–873, 1982.

[MDBO98] K. Mizuguchi, C. M. Deane, T. L. Blundell, and J. P. Overington. HOMSTRAD: A database of protein structure alignments for homologous families. *Protein Science*, 7:2469–2471, 1998.

[MDL07] G. Mayr, F. S. Domingues, and P. Lackner. Comparative Analysis of Protein Structure Alignments. *BMC Structural Biology*, 7:50–64, 2007.

[MGB95] T. Madej, J. F. Gibrat, and S. H. Bryant. Threading a database of protein cores. *Proteins*, 23:356–369, 1995.

[MST09] J. Melvin, J. Sokol, and C. Tovey. Finding optimal solutions to large cmo instances. *working paper*, 2009.

[MVR+10] L. Mavridis, V. Venkatraman, D. W. Ritchie, N. Morikawa, R. Andonov, A. Cornu, N. Malod-Dognin, J. Nicolas, M. Temerinac-Ott, M. Reisert, H. Burkhardt, A. Axenopoulos, and P. Daras. Shrec'10 track: Protein models, 2010.

[NMK04] M. Novotny, D. Madsen, and G. J. Kleywegt. Evaluation of protein fold comparison servers. *Proteins: Structure, Function and Bioinformatics*, 54:260–270, 2004.

[NW70] S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48:443–453, 1970.

[OMJ+97] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton. CATH–A hierarchic classification of protein domain structures. *Structure*, 5:1093–1108, 1997.

[OSO02] A. R. Ortiz, C. E. M. Strauss, and O. Olmea. Mammoth (matching molecular models obtained from theory): An automated method for model comparison. *Protein Science*, 11:2606–2612, 2002.

[pdb] Protein Data Bank. `http://www.pdb.org`.

[PGK05] D. Pelta, J. R. Gonzalez, and N. Krasnogor. Protein structure comparison through fuzzy contact maps and the universal similarity metric. In *Proceedings of the 4th Conference of European Society for Fuzzy Logic and Technology - LFA*, pages 1124–1129, 2005.

[PGV08] D. A. Pelta, J. R. Gonzlez, and M. M. Vega. A simple and fast heuristic for protein structure comparison. *BMC Bioinformatics*, 9:161–176, 2008.

[PKBC+05] D. Pelta, N. Krasnogor, C. Bousono-Calzon, J. L. Verdegay, and E. Burke. A fuzzy sets based generalization of contact maps for the overlap of protein structures. *fuzzy sets and systems*, 1:171–180, 2005.

[Pul07] W. Pullan. Protein structure alignment using maximum cliques and local search, 2007.

[RGL+85] G. Rose, A. Geselowitz, G. Lesser, R. Lee, and M. Zehfus. Surface and inside volumes in globular proteins. *Science*, 229:834–838, 1985.

[Rio09] L. M. Rios. *Algorithms for derivative-free optimization.* PhD thesis, Department of Industrial and Enterprise Systems Engineering, University of Illinois, Urbana, IL, May 2009.

[RS11] L. M. Rios and N. V. Sahinidis. Derivative-free optimization: A review of algorithms and comparison of software implementations, 2011. Working Paper, `http://thales.cheme.cmu.edu/dfo`.

[RSWD09] J. Rocha, J. Seguraa, R. C. Wilson, and S. Dasgupta. Flexible structural protein alignment by a sequence of local transformations. *Bioinformatics*, 25:1625–1631, 2009.

[SB90] 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. *Journal of Molecular Biology*, 212:403–428, 1990.

[SB97] A. P. Singh and D. L. Brutlag. Hierarchical protein structure superposition using both secondary structure and atomic representations. In *Proceedings in International Conference of Intelligent Systems in Molecular Biology*, 1997.

[SB98] I. Shindyalov and P. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering*, 11:739–747, 1998.

[SB01] A. P. Singh and D. L. Brutlag. Protein structure alignment: A comparison of methods. *Nature Structural Biology*, 2001.

[SBS05] D. M. Strickland, E. Barnes, and J. S. Sokol. Optimal protein structure alignment using maximum cliques. *Informs:Operations Research*, 53:389–402, 2005.

[Shi07] T. Shibuya. Efficient substructure RMSD query algorithms. *Journal of computational biology*, 14:1201–1207, 2007.

[Shi10a] T. Shibuya. Fast Hinge Detection Algorithms for Flexible Protein Structures. *IEEE/ACM transactions on computational biology and bioinformatics*, 7:333–341, 2010.

[Shi10b] T. Shibuya. Searching protein three-dimensional structures in faster than linear time. *Journal of computational biology*, 17:593–602, 2010.

[SHL10] Y. Shibberu, A. Holder, and K. Lutz. Fast protein structure alignment. *Bioinformatics Research and Applications*, 6053:152–165, 2010.

[Sip82] M. J. Sippl. On the problem of comparing proteins: Development and applications of a new method for the assessment of structural similarities and polypeptide conformations. *Journal of Molecular Biology*, 156:359–388, 1982.

[SJS10] T. Shibuya, J. Jansson, and K. Sadakane. Linear-time protein 3-D structure searching with insertions and deletions. *Algorithms for Molecular Biology*, 5:7–14, 2010.

[SLL93] S. Subbiah, D.V. Laurents, and M. Levitt. Structural similarity of dna-binding domains of bacteriophage repressors and the globin core. *Current Biology*, 3:141–148, 1993.

[SNW02] M. Shatsky, R. Nussinov, and H. Wolfson. Flexible protein alignment and hinge detection. *Proteins: Structure, Function, and Bioinformatics*, 49:242–256, 2002.

[SW81] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

[SZB09] S. Salem, M. J. Zaki, and C. Bystroff. Iterative non-sequential protein structural alignment. *Journal of Bioinformatics and Computational Biology*, 7(3):571–596, 2009.

[SZB10] S. Salem, M. J. Zaki, and C. Bystroff. FlexSnap: Flexible Non-sequential Protein Structure Alignment. *Algorithms for Molecular Biology*, 5:12–24, 2010.

[TFO94a] W. R. Taylor, T. P. Flores, and C. A. Orengo. Multiple protein structure alignment. *Protein Science*, 3:1868–1870, 1994.

[TFO94b] W. R. Taylor, T. P. Flores, and C. A. Orengo. Multiple protein structure alignment. *Protein Science*, 3:1858–1870, 1994.

[TO89a] W. R. Taylor and C. A. Orengo. A holistic approach to protein structure comparison. *Protein Engineering*, pages 2505–519, 1989.

[TO89b] W. R. Taylor and C. A. Orengo. Protein structure alignment. *Journal of Molecular Biology*, 208:l–22, 1989.

[TVKL09] C. Tai, J. J. Vincent, C. Kim, and B. Lee. SE: an algorithm for deriving sequence alignment from a pair of superimposed structures. *Bioinformatics, The Seventh Asia Pacific Bioinformatics Conference*, 2009.

[uni] SwissProt/UniProtKB. `http://www.uniprot.org`.

[VG01] J. Viksna and D. Gilbert. Pattern Matching and Pattern Discovery Algorithms for Protein Topologies. *Algorithms in Bioinformatics, Lecture Notes in Comput. Sci Springer-Verlag*, 2149:98–111, 2001.

[VGV10] M. Veeramalai, D. Gilbert, and G. Valiente. An optimized TOPS+ comparison method for enhanced TOPS models. *Bioinformatics*, 11:138:151, 2010.

[WDK10] I. Wohlers, F. S. Domingues, and G. W. Klau. Towards optimal alignment of protein structure distance matrices. *Bioinformatics*, 26:2273–2280, 2010.

[WSHB98] T. D. Wu, S. C. Schmidler, T. Hastie, and D. L. Brutlag. Modeling and superposition of multiple protein structures using affine transformations: Analysis of the globins. *Pac. Sym. on Bio*, 1998.

[XJB07] J. Xu, F. Jiao, and B. Berger. A parameterized algorithm for protein structure alignment. *Journal of Computational Biology*, 14:564–577, 2007.

[XS06] W. Xie and N. V. Sahinidis. A branch-and-reduce algorithm for the contact map overlap problem. In A. Apostolico, C. Guerra, S. Istrail, P. Pevzner, and M. Waterman, editors, *RECOMB 2006, Lecture Notes in Computer Science*, volume 3909, pages 516–529, 2006.

[XS07] W. Xie and N. V. Sahinidis. A reduction-based exact algorithm for the contact map overlap problem. *Journal of Computational Biology*, 14:637–654, 2007.

[YG03] Y. Ye and A. Godzik. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, 19:246–255, 2003.

[YJL05] J. Ye, R. Janardan, and S. Liu. Pairwise protein structure alignment based on an orientation-independent backbone representation. *Journal of Bioinformatics and Computational Biology*, 2:699–717, 2005.

[YT06] J. Yang and C. Tung. Protein structure database search and evolutionary classification. *Nucleic Acids Research*, 34:36463659, 2006.

[ZFAS08] Z. Zhao, B. Fu, F. J. Alanis, and C. M. Summa. Feedback algorithm and web-server for protein structure alignment. *Journal of Molecular Biology*, 15:505–524, 2008.

[ZS05] Y. Zhang and J. Skolnick. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33:2302–2309, 2005.

Appendix A
Annotated bibliography

[AF96] N. N. Alexandrov and D. Fischer. Analysis of topological and nontopological structural similarities in the PDB: new examples with old structures. *Proteins*, 1996.

This paper describes an improvement of SARF tool, SARF2 which is based on a fast search of aligned fragments based on secondary structures. A dynamic programming refinement of the aligned fragments gives the optimal structural alignment.

[AGM90] S. F. Altschul, W. Gish, and W. Miller. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

The sequence alignment tool, BLAST, is described in detail in the paper.

[AM08] R. Andreani and J. M. Martnez. Trust-region superposition methods for protein alignment. *IMA Journal of Numerical Analysis*, 28:690–710, 2008.

This paper presents a trust-region method for optimizing the LG score maximization [GL96] for structure alignment. The algorithm uses an iterative scheme with alternate calculation of a suitable alignment by dynamic program and maximization of the LG score by optimizing a quadratic function estimation of the LG score around the alignment. The method requires lower computational resources than STRUCTAL and produces better quality solutions than STRUCTAL only in case of proteins with high level of structural similarity.

[AMMY08] R. Andreani, J. M. Martnez, L. Martnez, and F. Yano. Continuous optimization methods for structure alignments. *Math Programming Ser. B*, 112:93–124, 2008.

This paper presents a Gauss-Newton method for structural alignment which incorporates alignments with internal flexibility. The Gauss-Newton approach is used to optimize the non-smooth and non-continuous RMSD function for different alignments through an iterative method that alternates between obtaining a suitable alignment and identifying a suitable rotation-translation superposition transformation between the proteins. The algorithm performs at par with state-of-the-art alignment tools for proteins with high structural similarity.

[AS97] S. F. Altschul and A. A. Schaffer. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

Further improvements to BLAST, namely, GAP-BLAST and PSI-BLAST, are discussed in this paper.

[ATG92] N. N. Alexandrov, K. Takahashi, and N. Go. Common spatial arrangements of backbone fragments in homologous and non-homologous proteins. *Journal of Molecular Biology*, 225:5–9, 1992.

The paper presents the SARF alignment tool based on alignment of small protein fragments which are further joined together to obtain an optimal alignment. The smaller protein fragments are aligned through a continuous method [McL82] which aims at minimizing the RMSD value.

[BBC06] S. Bhattacharya, C. Bhattacharyya, and N. Chandra. Projections for fast protein structure retrieval. *BMC Bioinformatics*, 7:S5–S17, 2006.

The authors describe an alignment tool based on projection vectors obtained from distance matrices. Proteins are represented as projections with constant norm value and orthogonal property.

[BHB+07] D. Barthel, J. D. Hirst, J. Blazewicz, E. K. Burke, and N. Krasnogor. ProCKSI: A decision support system for protein (structure) comparison, knowledge, similarity and information. *BMC Bioinformatics*, 2007.

This paper presents an introduction to the ProCKSI platform which integrates different structure compari-

son tools like DALI, CE and TM-align as well as different similarity measures like USM and MAX-CMO on a single platform.

[CCI+04] A. Caprara, R. Carr, S. Istrial, G. Lancia, and B. Walenz. 1001 optimal PDB structure alignments: Integer programming methods for finding the maximum contact map overlap. *Journal of Computational Biology*, 11:27–52, 2004.

This paper presents a Lagrangian-relaxation-based branch-and-cut algorithm for the integer program formulation based on contact maps where the solution to the Lagrangian relaxation produces tight upper bounds. This led to the ability to solve problems with up to 1000 residue-long proteins, which is a considerable improvement over previous studies.

[CHK+02] B. Carr, W. Hart, N. Krasnogor, J. Hirst, E. Burke, and J. Smith. Alignment of protein structures with a memetic evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1027–1034, 2002.

In this paper, the authors suggest a memetic evolution algorithm the maximum contact map overlap formulation of the structure alignment problem. Basic genetic algorithm operators, such as crossover and mutation are used, along with some added local search strategies, including sliding, wiper and split motions. They compare the algorithm with LGA and a genetic algorithm on a set of 18 proteins, and observe that the algorithm does just as well as others.

[CL02] A. Caprara and G. Lancia. Structural alignment of large-size proteins via lagrangian relaxation. In *Proceedings of the International Conference on Computational Biology (RECOMB)*, pages 100–108, 2002.

This paper presents an introduction to the use of Lagrangian relaxation for obtaining bounds in the branch-and-cut algorithm [LCWI01]. The Lagrangian bounds are tighter than the bounds obtained using the meta-heursitics proposed in their previous work [LCWI01].

[CL04] R. D. Carr and G. Lancia. Compact optimization can outperform separation: A case study in structural proteomics. *4OR: A Quarterly Journal of Operations Research*, 2:221–233, 2004.

In this paper, the authors introduce of a compact integer program formulation of structure alignment represented by contact maps. This formulation provides a tighter linear programming relaxation than previous formulations [LCWI01].

[CLI00] R. D. Carr, G. Lancia, and S. Istrail. Branch-and-cut algorithms for independent set problems: Integrality gap and an application to protein structural alignment. Technical report, Sandia National laboratories, 2000. Sandia Report SAND2000-2171.

This report outlines a branch-and-cut algorithm for the integer program formulation based on contact maps.

[FC96] A. Falicov and F. E. Cohen. A surface of minimum area metric for the structural comparison of protein. *Journal of Molecular Biology*, 258:871–892, 1996.

In this paper, the authors describe a structure alignment method using an area minimization technique. Initially an alignment is found using dynamic programming for a similarity matrix based on triangulation distances. The alignment is then refined minimizing the area measure using conjugate gradient, downhill simplex and Powell's method of minimization. Tough the measure they use is observed to be applicable only in the same homologue family, in general their method is able to obtain the same clustering as FSSP and similar RMSD values for protein alignments.

[FKR+70] S. T. Freer, J. Kraut, J. D. Robertus, H. T. Wright, and Ng. H. Xuong. Chymotrypsinogen: 2.5-A Crystal Structure, Comparison with α-Chymotrypsin, and Implications for Zymogen Activation. *Biochemistry*, 9:1997–2009, 1970.

In this paper, experimental results depicting structural similarities between proteins with low sequence identity are presented, thus establishing importance of structure alignment.

[GK08] A. Guerler and E. W. Knapp. Novel protein folds and their nonsequential structural analogs. *Protein science*, 17:1374 – 1382, 2008.

This paper describes a protein structure alignment tool that provides nonsequential structure alignments.

[GL96] M. Gerstein and M. Levitt. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. In D. States, P. Agarwal, T. Gaasterland, L. Hunter, and R. Smith, editors, *Proceedings of international conference on intelligent systems in molecular biology*, pages 59–67. AAAI Press, 1996.

In this paper, the authors develop an iterative dynamic programming algorithm called STRUCTAL, which uses a similarity matrix based on the inter-residue distances of aligned proteins. The objective function used is a variation of the weighted RMSD with a gap penalty.

[GMB96] J. F. Gibrat, T. Madej, and S. H. Bryant. Surprising similarities in structure comparison. *Current Opinion in Structural Biology*, pages 377–385, 1996.

[GPI99] D. Goldman, C. Papadimitriou, and S. Istrail. Algorithmic aspects of protein structure similarity. In *Proceedings of the 40th annual symposium on foundations of computer science (FOCS)*, pages 512–522. IEEE Computer society, 1999.

The authors prove that the MAX-CMO problem for proteins is NP-hard and outline some special cases (self-avoiding walks) where the problem can be solved in polynomial time. They also enumerate some of the requirements a good similarity measure must have, which has been used as a benchmark since.

[Gra04] J. Gramm. A polynomial-time algorithm for the matching of crossing contact-map patterns. *IEEE/ACM transaction on computational biology and bioinformatics*, 1:171–180, 2004.

This paper introduces a polynomial time dynamic-programming-based algorithm for contact maps based alignment problems with special structures. The proposed approach solves a limited class of problems, but may be utilized as plausible building block for other algorithms.

[GSK93] A. Godzik, J. Skolnick, and A. Kolinski. Regularities in interaction patterns of globular proteins. *Protein Engineering*, 6:801–810, 1993.

In this paper a Monte Carlo based simulated annealing technique for finding protein structure alignment is described.

[HP00] L. Holm and J. Park. Dalilite workbench for protein structure comparison. *Bioinformatics Applications Note*, 16:566–567, 2000.

This paper introduces a web-based server for the state-of-the-art alignment tool DALI [HS93].

[HS93] L. Holms and C. Sander. Protein-structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, 233:123–138, 1993.

The paper describes in detail the algorithm DALI, which is one of the most widely used structural alignment tool. DALI uses a monte carlo based technique with distance matrix representation for finding alignments. The technique is quite fast (5-10 minutes per alignment), however there is no guarantee of global optimality of the solution obtained. In most cases, the solution obtained is a local solution.

[HS96] L. Holm and C. Sander. The fssp database: Fold classification based on structure-structure alignment of proteins. *Nucleic Acids Research*, 24:206–209, 1996.

This paper describes the FSSP database.

[IAL04] V. A. Ilyin, A. Abyzov, and C. M. Leslin. Structural alignment of proteins by a novel topfit method, as a superimposition of common volumes at a topmax point. *Protein Science*, 13:1865–1874, 2004.

In this paper the authors describe the TOPFIT method for structure alignment which uses vornoi representation of the proteins.In this method the tetrahedrals in the vornoi diagram are matched to provide an alignment, which is further improved by addition of more tetrahedral matches. The method is compared with DALI and CE and is observed to provide better RMSD values though the number of aligned residues is smaller.

[JL07] B. J. Jain and M. Lappe. Joining softassign and dynamic programming for the contact map overlap problem, 2007.

In this paper, the authors describe a method called soft-assign for solving the maximum common subgraph (MCS) problem. Softassign minimizes a continuous quadratic objective by using simulated annealing. The solution obtained is converted to a feasible solution for the MAX-CMO problem using dynamic programming.

This approach is fast and has been observed to provide solutions that are close to the optimal for most data sets.

[KH04] 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*, 60:2256–2268, 2004.

The paper presents the SSM alignment tool based on a secondary structure based graphical representation of proteins [SB97]. It utilizes both graph theory algorithms as well as continuous optimization algorithms. The results suggest that SSM performs comparable to DALI, CE and VAST with lesser computational requirements. It is one of the alignment tools used in the development of SCOP database.

[KKL05] R. Kolodny, P. Koehl, and M. Levitt. Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *Journal of Molecular Biology*, 346:1173–1188, 2005.

In this paper, the authors compare 6 structure alignment methods using geometric measures of comparison. They conclude that STRUCTAL, SSM and LSQMAN perform the best among the 6 methods STRUCTAL, SSM, LSQMAN, DALI, CE and SSAP.

[KP04] N. Krasnogor and D. A. Pelta. Measuring the similarity of protein structures by means of the universal similarity metric. *Bioinformatics*, 20:1015–1021, 2004.

This paper describes the universal similarity metric (USM) for structure comparison. USM evaluates similarity between two protein structures without actually finding an alignment. The measure is only an estimate of the similarity, which can be further improved through the use of alignment tools.

[Kra04] N. Krasnogor. Self generating metaheuristics in bioinformatics: The protein structure comparison case. *Genetic Programming and Evolvable Machines*, 5:181–201, 2004.

In this paper, the author proposes a memetic algorithm for solving the maximum contact map overlap problem. This is an improvement over the Carr et al. paper

through the introduction of memetic processes. The algorithm is marginally better than a genetic algorithm but clearly better than LGA.

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

This paper presents the DSSP tool developed to identify physical properties like secondary structures and hydrophobicity from the 3D structure and side-chain placement.

[LCWI01] G. Lancia, R. Carr, B. Walenz, and S. Istrail. 101 optimal PDB structure alignments: A Branch-and-cut algorithm for the maximum contact map overlap problem. In *Proceedings of the fifth annual international conference on Computational biology, Montreal, Quebec, Canada*, pages 193–202, 2001.

This paper introduces the integer program formulation based on contact maps with separation of clique inequalities. A branch-and-cut method is proposed, where at every node they add the most violated clique inequality is added to the relaxed integer program.

[LG98] M. Levitt and M. Gerstein. A unified statistical framework for sequence comparison and structure comparison. *Proc. Natl. Acad. Sci.*, 95:5913–5920, 1998.

The LG score similarity metric is introduced here. The LG score provides more meaningful alignments than the RMSD measure as it also accounts for introduced gaps.

[LKR03] J. Leluk, L. Konieczny, and I. Roterman. Search for structural similarity in proteins. *Bioinformatics*, 19:117–124, 2003.

This paper presents the VeaR alignment tool that is based on proteins represented as sequences of dihedral angle and radius of curvature for fragments of 5 residues. The alignment is obtained by performing a multiple sequence alignment on the sequences of dihedral angles and the log of the radii of curvature.

[LKSD00] P. Lackner, W. A. Koppensteiner, M. J. Sippl, and F. S. Domingues. ProSup: A refined tool for protein structure alignment. *Protein Engineering*, 11:745–752, 2000.

The paper introduces the ProSup alignment tool based on alignment of small fragments.

[Mar00] A. C. R. Martin. The ups and downs of protein topology; rapid comparison of protein structure. *Protein Engineering*, 13:829–837, 2000.

This paper presents the TOPSCAN alignment tool based on secondary structure representation obtained through the DSSP tool [KS83].

[MBHC95] 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*, 247:536–540, 1995.

This paper describes the methodology behind the creation of the SCOP database of proteins.

[McL82] A. D. McLachlan. Rapid comparison of protein structures. *Acta. Cryst.*, A38:871–873, 1982.

This paper describes a fast method for RMSD evaluation between protein structures.

[MDBO98] K. Mizuguchi, C. M. Deane, T. L. Blundell, and J. P. Overington. HOMSTRAD: A database of protein structure alignments for homologous families. *Protein Science*, 7:2469–2471, 1998.

The HOMSTRAD database for fold families is introduced in this paper. The COMPARER [SB90] structure alignment tool is used to obtain structure alignments on this platform.

[MGB95] T. Madej, J. F. Gibrat, and S. H. Bryant. Threading a database of protein cores. *Proteins*, 23:356–369, 1995.

This paper outlines the VAST method, in the context of identifying 'true positives' in a fold-recognition experiment. This employs fast search by SSE alignment, followed by Monte Carlo refinement at the level of backbone coordinates.

[MST09] J. Melvin, J. Sokol, and C. Tovey. Finding optimal solutions to large cmo instances. *working paper*, 2009.

The authors introduced a new data structure to address the issue of large memory requirements for the MAX-CLIQUE reformulation [SBS05]. They obtain a O(N)

improvement over the existing method in both time and space requirements.

[NMK04] M. Novotny, D. Madsen, and G. J. Kleywegt. Evaluation of protein fold comparison servers. *Proteins: Structure, Function and Bioinformatics*, 54:260–270, 2004.

The paper presents a review comparing web-based structure alignment tools on the basis of their performance as well as presentation. The review showed that different solvers had different performance levels in different test conditions and there is no server that can be considered as the best or worst.

[NW70] S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48:443–453, 1970.

This paper describes the basic dynamic programming algorithm for sequence alignment.

[OMJ+97] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton. CATH–A hierarchic classification of protein domain structures. *Structure*, 5:1093–1108, 1997.

In this paper the authors describe the classification of proteins into the CATH database. The structures are classified at 5 levels, namely C - class, A - architecture, T - topology, H - homologue family and S - sequence family. Each of these is an increasing level of classification. Thus every protein can be represented to be belonging to a CATHS number which is a 5-tuple.

[OSO02] A. R. Ortiz, C. E. M. Strauss, and O. Olmea. Mammoth (matching molecular models obtained from theory): An automated method for model comparison. *Protein Science*, 11:2606–2612, 2002.

In this paper, the MAMMOTH algorithm is described. The algorithm is based on a dynamic programming based alignment using similarity scores obtained from a RMSD minimization technique of McLachlan [McL82]

[PGK05] D. Pelta, J. R. Gonzalez, and N. Krasnogor. Protein structure comparison through fuzzy contact maps and the universal similarity metric. In *Proceedings of the 4th Conference of European Society for Fuzzy Logic and Technology - LFA*, pages 1124–1129, 2005.

In this paper, the Universal Similarity Metric (USM) is introduced. USM provides a measure of similarity without aligning the proteins. USM is defined using the fuzzy contact maps representation and provides a ball-park estimate for the similarity of two proteins. The clustering obtained from the USM of pairs of proteins is observed to be in agreement with the clustering observed in SCOP database.

[PGV08] D. A. Pelta, J. R. Gonzlez, and M. M. Vega. A simple and fast heuristic for protein structure comparison. *BMC Bioinformatics*, 9:161–176, 2008.

This paper presents a multi-start variable neighborhood search metaheuristic with pair addition and deletion heuristics for solving the MAX-CMO formulation. The heuristic is run for a fixed number of iterations and provides an approximate solution to the MAX-CMO problem. The proposed approach is observed to be computationally expensive.

[PKBC+05] D. Pelta, N. Krasnogor, C. Bousono-Calzon, J. L. Verdegay, and E. Burke. A fuzzy sets based generalization of contact maps for the overlap of protein structures. *fuzzy sets and systems*, 1:171–180, 2005.

Fuzzy contact maps have been introduced in this paper using multiple thresholds to capture different levels of residue interactions. A variable neighbor search (VNS) metaheuristic based algorithm is presented for alignment and protein clustering is performed on various data sets which is observed to be similar to clustering in the SCOP database.

[Pul07] W. Pullan. Protein structure alignment using maximum cliques and local search, 2007.

The authors present a heuristic local search method which provides an approximate solution for the MAX-CLIQUE reformulation [SBS05]. The approach is on order of magnitude faster than [SBS05] and provides the exact optimal solution for most problems in the tested data set.

[RSWD09] J. Rocha, J. Seguraa, R. C. Wilson, and S. Dasgupta. Flexible structural protein alignment by a sequence of local transformations. *Bioinformatics*, 25:1625–1631, 2009.

This paper discusses the use of a sequence of local rigid-body transformations for aligning proteins, resulting in different rotation and translation angles for different fragments of the proteins. The resulting alignment is non-rigid, providing a tool for comparing similar proteins with small number of bends.

[SB90] 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. *Journal of Molecular Biology*, 212:403–428, 1990.

This paper presents an introduction to the Comparer alignment tool based on a multi-level sequence representation of proteins. The alignment is obtained as a weighted sum of the sequence alignments found at each level. The sequence representations are based on primary, secondary and tertiary structure of proteins as well as physical properties such as H-bonds and dihedral angles. A more biologically relevant alignment of the proteins is obtained. However the method is observed to be computationally expensive.

[SB97] A. P. Singh and D. L. Brutlag. Hierarchical protein structure superposition using both secondary structure and atomic representations. In *Proceedings in International Conference of Intelligent Systems in Molecular Biology*, 1997.

This paper describes a method called LOCK for structure comparison. In this paper a protein structure representation as a vector of secondary structures is introduced. The method is compared with DALI and argued to have obtained better RMSD values.

[SB98] I. Shindyalov and P. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering*, 11:739–747, 1998.

In this paper the authors describe the method Combinatorial Extension (CE) for structural alignment. CE is a heuristic method based on 8-mer residue fragment comparison through distance matrix norms.

[SBS05] D. M. Strickland, E. Barnes, and J. S. Sokol. Optimal protein structure alignment using maximum cliques. *Informs:Operations Research*, 53:389–402, 2005.

The authors introduce the MAX-CLIQUE reformulation of structure alignment problem based on contact maps. A coloring algorithm is developed which utilizes the special graph structure for better performance. The method is useful only for aligning small proteins since the clique graph structure is memory intensive.

[Shi07] T. Shibuya. Efficient substructure RMSD query algorithms. *Journal of computational biology*, 14:1201–1207, 2007.

This paper presents a fast alignment search tool which identifies protein alignments with low RMSD values.

[Shi10a] T. Shibuya. Fast Hinge Detection Algorithms for Flexible Protein Structures. *IEEE/ACM transactions on computational biology and bioinformatics*, 7:333–341, 2010.

This paper presents a fast alignment search tool which identifies flexible protein alignments with very few gaps.

[Shi10b] T. Shibuya. Searching protein three-dimensional structures in faster than linear time. *Journal of computational biology*, 17:593–602, 2010.

This paper presents a fast alignment search tool which identifies protein alignments with no gaps.

[Sip82] M. J. Sippl. On the problem of comparing proteins: Development and applications of a new method for the assessment of structural similarities and polypeptide conformations. *Journal of Molecular Biology*, 156:359–388, 1982.

The author introduces the Dk procedure used to identify similarity between protein structures. The calculation of Dk values is fast and can be used to derive the level of similarity between two proteins.

[SJS10] T. Shibuya, J. Jansson, and K. Sadakane. Linear-time protein 3-D structure searching with insertions and deletions. *Algorithms for Molecular Biology*, 5:7–14, 2010.

This paper presents a fast alignment search tool which identifies protein alignments with a fixed number of gaps.

[SLL93] S. Subbiah, D.V. Laurents, and M. Levitt. Structural similarity of dna-binding domains of bacteriophage repressors and the globin core. *Current Biology*, 3:141–148, 1993.

The paper introduces an iterative dynamic programming utilized in the development of STRUCTAL alignment tool [GL96].

[SW81] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

In this paper the authors describe the Smith-Waterman algorithm for sequence alignment.

[SZB09] S. Salem, M. J. Zaki, and C. Bystroff. Iterative non-sequential protein structural alignment. *Journal of Bioinformatics and Computational Biology*, 7(3):571–596, 2009.

Thus paper describes the STSA alignment tool that provides nonsequential structure alignment by aligning small fragments of protein structures.

[SZB10] S. Salem, M. J. Zaki, and C. Bystroff. FlexSnap: Flexible Non-sequential Protein Structure Alignment. *Algorithms for Molecular Biology*, 5:12–24, 2010.

This paper presents the FlexSnap alignment tool that provides flexible nonsequential protein structure alignments. The algorithm is based in aligning small protein fragments with each other nonsequentially and non-rigidly. It is the only other algorithm other than our present work that provides both nonsequential and flexible alignments.

[TFO94a] W. R. Taylor, T. P. Flores, and C. A. Orengo. Multiple protein structure alignment. *Protein Science*, 3:1868–1870, 1994.

The authors present an novel approach where the SSAP method of structure alignment [TO89b] is combined with the multal method of multiple sequence alignment to provide a multiple structure-sequence alignment tool.

[TFO94b] W. R. Taylor, T. P. Flores, and C. A. Orengo. Multiple protein structure alignment. *Protein Science*, 3:1858–1870, 1994.

This paper describes a multiple protein structure alignment tool combining structure and sequence alignments.

[TO89a] W. R. Taylor and C. A. Orengo. A holistic approach to protein structure comparison. *Protein Engineering*, pages 2505–519, 1989.

The authors present an extension to the SSAP algorithm to incorporate physical aspects such as hydrogen bonding, solvent exposure and torsional angles with suitable weights to produce a more holistic comparison method. The method is tested on a group of remotely related α/β type proteins that share a common feature in their overall chain fold. The results indicate that the inclusion of hydrogen bonds, torsion angles and a measure of solvent exposure led to improvements in the more difficult comparisons. Consideration of amino acid properties, including hydrophobicity, had no beneficial effect.

[TO89b] W. R. Taylor and C. A. Orengo. Protein structure alignment. *Journal of Molecular Biology*, 208:1–22, 1989.

The paper introduces the SSAP alignment tool which is used commonly in the CATH database for fold classification.

[TVKL09] C. Tai, J. J. Vincent, C. Kim, and B. Lee. SE: an algorithm for deriving sequence alignment from a pair of superimposed structures. *Bioinformatics, The Seventh Asia Pacific Bioinformatics Conference*, 2009.

This paper presents a heuristic method, Seed extension (SE) tool based on distance matrix representation. The tool identifies fixed alignments or seeds based on minimal elements in rows and columns of the distance matrix, and extends the seed segments to obtain the alignment.

[VG01] J. Viksna and D. Gilbert. Pattern Matching and Pattern Discovery Algorithms for Protein Topologies. *Algorithms in Bioinformatics, Lecture Notes in Comput. Sci Springer-Verlag*, 2149:98–111, 2001.

The paper presents the TOPS alignment tool based on a graphical secondary structure representation of proteins.

[VGV10] M. Veeramalai, D. Gilbert, and G. Valiente. An optimized TOPS+ comparison method for enhanced TOPS models. *Bioinformatics*, 11:138:151, 2010.

This paper discusses the TOPS+ alignment tool that is an improved version of TOPS [VG01] through incor-

poration of directional connectivity and chirality in the secondary structure graphs.

[WSHB98] T. D. Wu, S. C. Schmidler, T. Hastie, and D. L. Brutlag. Modeling and superposition of multiple protein structures using affine transformations: Analysis of the globins. *Pac. Sym. on Bio*, 1998.

In this paper a superimposition method allowing shear transformation along with rotation and translation is described. The protein structures are represented as a sequence of radius of curvature of smaller fragments. By setting up an average structure for the globulin family, the fold family is quite accurately characterized.

[XJB07] J. Xu, F. Jiao, and B. Berger. A parameterized algorithm for protein structure alignment. *Journal of Computational Biology*, 14:564–577, 2007.

The paper outlines a polynomial time approximation scheme for comparing a contact map with another contact map or a distance matrix. The authors use a tree decomposition technique in conjunction with discretization of rotation angles to come up with an optimal alignment.

[XS06] W. Xie and N. V. Sahinidis. A branch-and-reduce algorithm for the contact map overlap problem. In A. Apostolico, C. Guerra, S. Istrail, P. Pevzner, and M. Waterman, editors, *RECOMB 2006, Lecture Notes in Computer Science*, volume 3909, pages 516–529, 2006.

In this paper, the authors present the CMOS branch-and-reduce algorithm for the MAX-CMO problem. A variety of optimality and feasibility-based fast reduction schemes are employed in this algorithm.

[XS07] W. Xie and N. V. Sahinidis. A reduction-based exact algorithm for the contact map overlap problem. *Journal of Computational Biology*, 14:637–654, 2007.

In this paper, the authors improved the CMOS algorithm through the incorporation of faster and better stronger reduction schemes. This led to a considerable speed-up over existing exact algorithms and made possible the solution of many previously unsolved alignment problems.

[YG03] Y. Ye and A. Godzik. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, 19:246–255, 2003.

This paper discusses the alignment of small protein fragments, allowing non-rigidity using a limited number bends and turns in the aligned fragments. The resulting algorithm provides for finding non-rigid alignments with limited number of bends and turns.

[YJL05] J. Ye, R. Janardan, and S. Liu. Pairwise protein structure alignment based on an orientation-independent backbone representation. *Journal of Bioinformatics and Computational Biology*, 2:699–717, 2005.

This paper describes an iterative method for structure alignment. The backbone of the protein is described as a set of angles and then dynamic programming is used to make the alignment, followed by transformation evaluation of the RMSD. The method is compared with DALI and LOCK and similar results are obtained.

[YT06] J. Yang and C. Tung. Protein structure database search and evolutionary classification. *Nucleic Acids Research*, 34:36463659, 2006.

The authors present the 3D-BLAST alignment tool that aligns proteins by pattern matching techniques. The 3D structure is modeled using a 23-letter structural alphabet developed employing the angles between the C-α atoms. The algorithm demonstrates low time requirements with moderate precision.

[ZFAS08] Z. Zhao, B. Fu, F. J. Alanis, and C. M. Summa. Feedback algorithm and web-server for protein structure alignment. *Journal of Molecular Biology*, 15:505–524, 2008.

The paper outlines an algorithm based on feedback for structure alignment, SLIPSA. Local alignments are stitched together to get a global alignment, which is then provided as a starting point to the algorithm, as a feedback. The method is time-intensive (almost 5-10 times that of DALI). The results indicate that in most cases the number of matched residues is larger with comparable RMSD or the number of matched residues is comparable with lower RMSD.

[ZS05] Y. Zhang and J. Skolnick. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33:2302–2309, 2005.

This paper discusses the TM-align alignment tool based on iterative dynamic programming on sequence representation of proteins based on secondary structures. The TM-score similarity metric that is a commonly used similarity metric is also defined in the paper.