Finding Similar Protein Structures Efficiently and Effectively

by

Xuefeng Cui

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

To assess the similarities and the differences among protein structures, a variety of structure alignment algorithms and programs have been designed and implemented. We introduce a low-resolution approach and a high-resolution approach to evaluate the similarities among protein structures. Our results show that both the low-resolution approach and the high-resolution approach outperform state-of-the-art methods.

For the low-resolution approach, we eliminate false positives through the comparison of both local similarity and remote similarity with little compromise in speed. Two kinds of contact libraries (ContactLib) are introduced to fingerprint protein structures effectively and efficiently. Each contact group from the contact library consists of one local or two remote fragments and is represented by a concise vector. These vectors are then indexed and used to calculate a new combined hit-rate score to identify similar protein structures effectively and efficiently.

We tested our ContactLibs on the high-quality protein structure subset of SCOP30, which contains $3,297$ protein structures. For each protein structure of the subset, we retrieved its neighbor protein structures from the rest of the subset. The best area under the ROC curve, archived by a ContactLib, is as high as 0.960. This is a significant improvement over 0.747, the best result achieved by the state-of-the-art method, FragBag.

For the high-resolution approach, our PROtein STructure Alignment method (PROSTA) relies on and verifies the fact that the optimal protein structure alignment always contains a small subset of aligned residue pairs, called a seed, such that the rotation and translation (ROTRAN), which minimizes the RMSD of the seed, yields both the optimal ROTRAN and the optimal alignment score. Thus, ROTRANs minimizing the RMSDs of small subsets of residues are sampled, and global alignments are calculated directly from the sampled ROTRANs. Moreover, our method incorporates remote information and filters similar ROTRANs (or alignments) by clustering, rather than by an exhaustive method, to overcome the computational inefficiency.

Our high-resolution protein structure alignment method, when applied to optimizing the TM-score and the GDT-TS score, produces a significantly better result than state-of-the-art protein structure alignment methods. Specifically, if the highest TM-score found by

TM-align is lower than 0.6 and the highest TM-score found by one of the tested methods is higher than 0.5, our alignment method tends to discover better protein structure alignments with (up to 0.21) higher TM-scores. In such cases, TM-align fails to find TM-scores higher than 0.5 with a probability of 42%; however, our alignment method fails the same task with a probability of only 2%.

In addition, existing protein structure alignment scoring functions focus on atom coordinate similarity alone and simply ignore other important similarities, such as sequence similarity. Our scoring function has the capacity for incorporating multiple similarities into the scoring function. Our result shows that sequence similarity aids in finding high quality protein structure alignments that are more consistent with HOMSTRAD alignments, which are protein structure alignments examined by human experts. When atom coordinate similarity itself fails to find alignments with any consistency to HOMSTRAD alignments, our scoring function remains capable of finding alignments highly similar to, or even identical to, HOMSTRAD alignments.

Acknowledgements

First, I would like to thank my supervisor, Dr. Ming Li, in whose steps I have been trying to follow in my research adventures. One of the most treasured pieces of advice I have received from Dr. Li is that we should not just focus on current problems, but we should discover and address critical future problems.

I would also like to thank my collaborators and friends, Shuai Cheng Li, Dongbo Bu, Xin Gao, Lin He, and Babak Alipanahi Ramandi, for the knowledge and skills I learned by working with them. They provided useful discussions in both my research area and other branches of Bioinformatics.

I would like to thank my Ph.D. thesis committee members, Dr. Yang Zhang, Dr. Brendan J. McConkey, Dr. Bin Ma, and Dr. Forbes Burkowski, for taking their precious time to review this thesis and provide valuable comments.

Last, I would like to thank my parents for their unconditional support and understanding. Without my mother taking care of my father in the hospital, I would not be able to finish this thesis on time.

Dedication

This thesis is dedicated to my parents.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Proteins function in living organisms as enzymes, antibodies, sensors, and transporters, among myriad other roles. The understanding of protein function has great implications for the study of biological and medical sciences. It has been widely accepted that protein function is largely determined by protein structure. Moreover, protein structures are more conserved than protein sequences. Therefore, protein structures are often aligned for their common substructures, for the purpose of discovering functionally or evolutionarily meaningful structure units.

To assess the similarities and the differences among protein structures, a variety of structure alignment algorithms and programs have been designed and implemented [40, 64, 29, 66]. We introduce a low-resolution approach and a high-resolution approach to evaluate the similarities among protein structures. These two approaches can be used either individually or together in protein studies, and our results show that both the low-resolution approach and the high-resolution approach outperform state-of-the-art methods. The fundamentals of protein structure alignment are further described in Chapter 2.

The low-resolution approach is useful to many protein studies. For example, when a new protein structure is experimentally determined, one interesting question is to annotate the protein function based on previously annotated proteins that are structurally similar to the newly determined protein. In this case, the low-resolution approach can be adopted to quickly scan all known protein structures and identify potential candidates of interest.

Because of the large number of annotated proteins, the speed is our primary concern for the low-resolution approach.

For some protein studies, a high-resolution approach is preferred over a low-resolution approach. For example, when we have a pair of proteins that potentially share common evolution history or functions, one interesting question is to identify the highly conserved regions and to access the similarities of the highly conserved regions. In this case, the high-resolution approach can be adopted to accurately align two protein structures and evaluate the similarities between them. The input protein structures are also optimally superimposed by the high-resolution approach so that one can visually check the similarities. Here, the accuracy is our primary concern for the high-resolution approach.

Previous protein structure studies, on high resolution protein structure data, show that the same type of bond lengths and bond angles fit Gaussian distributions well, with small standard deviations [21, 22]. The mean values of these Gaussian distributions have been widely used as ideal bond lengths and ideal bond angles in bioinformatics [28, 69, 50, 8]. However, we are not aware of any research done to evaluate how accurately one can model protein structures with ideal bond lengths, ideal bond angles and bond dihedral angles.

In Chapter 3, we introduce our protein structure idealization method that focuses on protein backbone structure idealization. We describe a fast $O(nm/\epsilon)$ dynamic programming algorithm to find an idealized protein backbone structure, which is approximately optimal according to our scoring function. Our scoring function evaluates not only the statistical energy, but also the similarity with the target structure. Thus, the idealized protein structures found by our method are guaranteed to be protein-like and close to the target protein structure.

We have implemented our protein structure idealization method and successfully idealized the $1,898$ high resolution protein structures with low sequence identities of the CULLPDB_PC30_RES1.6_R0.25 data set [72]. We demonstrated that idealized backbone structures always exist with small refinements and significantly better statistical energy. Thus, protein structures can be modeled accurately with ideal bond lengths, ideal bond angles and bond dihedral angles. We also applied our idealization method to refine protein pseudo-structures determined in NMR experiments.

A very large amount of data is currently available for protein structure studies, and the number of known protein structures in the Protein Data Bank has exceeded 90,000 [5]. A major drawback with current structure alignment programs is that they require a large amount of computational time, rendering them infeasible for pairwise alignments on large collections of protein structures. To overcome this drawback, a fragment alphabet learned from known structures has been introduced [7]. The method, however, considers local fragment similarity only, and therefore occasionally assigns high scores to structures that are similar only in local fragments.

In Chapter 4, we propose a novel approach for retrieving protein structures that are similar to a given protein structure in a large protein structure database. Our method eliminates false positives, through the comparison of both local similarity and remote similarity, with little compromise in speed. Two kinds of contact libraries (ContactLib) are introduced to fingerprint protein structures. Here, each contact group from the contact library consists of one local or two remote fragments. In current protein structure alignment methods [40, 29], a protein structure is usually represented by either 3-dimensional coordinates, which are difficult to index [46, 75], or a pairwise distance matrix, which has a significantly higher number of dimensions [32]. However, the result of our protein structure idealization experiment suggests to use a distance vector with a number of dimensions similar to that of the 3-dimensional coordinates. These vectors can be easily indexed and used to calculate a new combined hit-rate score to identify similar protein structures effectively and efficiently.

We tested our ContactLib method on the high-quality protein structure subset of SCOP30 [59], which contains $3,297$ protein structures. For each protein structure of the SCOP30 subset, we retrieved similar protein structures from the remaining SCOP30 subset. The performance was evaluated by the area under the receiver operating characteristic (ROC) curve. As a result, the best area under the ROC curve, archived by our ContactLib, is 0.960. This is a significant improvement, compared to 0.747, which is the best result achieved by FragBag [7].

Our ContactLib method can be seen as a fast low-resolution approach for the selection of potentially similar protein structures and for the elimination of dissimilar protein structures from a large collection of protein structures. Given a pair of protein structures that

are potentially similar, an accurate high-resolution approach to locate the similar regions and to evaluate the similarity between the protein structure pair is also desired. Before introducing our high-resolution approach, we studied a simplified version of the protein structure alignment problem, which provided the fundamental observation for our protein structure alignment method.

In Chapter 5, we introduce a simple and reliable method to optimally superimpose two protein structures, given the aligned residue pairs and the scoring function. State-of-the-art superimposition programs, TMscore [85] and LGA [83], begin looking for the optimal superimposition by calculating the rotation and translation (ROTRAN) that minimizes the RMSD of a highly conserved small subset (seed) of aligned residue pairs. Unlike TMscore and LGA, our superimposition method employs not only consecutive seeds containing contiguous aligned residue pairs, but also gapped seeds which do not require the aligned residue pairs to be contiguous. Moreover, a weighted RMSD is used with gapped seeds as refinement.

Our superimposition experiment on the HOMSTRAD database [57] and the CASP10 dataset [43] shows that the optimal protein structure alignment always contains a seed (highly conserved small subset) such that the rotation and translation, which minimizes the RMSD of the seed, yields both the optimal superimposition and the optimal alignment score. Moreover, when calculating the TM-scores [85] and the GDT-TS scores [83] between the predicted and the native structures of CASP10, our superimposition method is capable of finding TM-scores and GDT-TS scores that are similar to, or (possibly significantly) higher than, those found by TMscore [85] and LGA [83], respectively.

A variety of methods have been proposed for protein structure alignment. One major shortcoming in current structure alignment methods is in their inherent design, which begins with primary and secondary structure alignments and fragment similarity. Unlike current methods, our PROtein STructure Alignment (PROSTA) method is based on our observation that the optimal protein structure alignment always contains a seed such that the ROTRAN, which minimizes the RMSD of the seed, yields both the optimal ROTRAN and the optimal alignment score. Thus, ROTRANs are sampled by minimizing the RMSDs of seeds, and global alignments are calculated directly from the sampled ROTRANs. Moreover, our method incorporates remote information and filters similar ROTRANs (or align-

ments) by clustering, rather than by an exhaustive method, to overcome the computational inefficiency. The details of our PROSTA method are described in Chapter 6.

Our PROSTA method, when applied to optimizing the TM-score [85] and the GDT-TS score [83], produces significantly better protein structure alignments than current methods. Specifically, if the highest TM-score found by TM-align [87] is lower than 0.6 and the highest TM-score found by one of the tested methods is higher than 0.5, our alignment method tends to discover better protein structure alignments, with (up to 0.21) higher TM-scores. In such cases, TM-align fails to find TM-scores higher than 0.5 with a probability of 42%, while our alignment method fails the same task with a probability of only 2%. Comparing the GDT-TS scores found by our alignment method and those found by LGA [83], we find that our alignment method improves the GDT-TS scores by up to 0.44, and it also finds 156% more alignments with GDT-TS scores higher than 0.5. This could significantly improve the fold detection accuracy if the cut-off score of 0.5 is used as the threshold of fold detection.

Another major shortcoming in current protein structure alignment methods is in their scoring functions. These scoring functions tend to focus on atom structure similarity alone, and simply ignore other protein similarities, such as sequence similarity. Although atom structure similarity may be efficient in many cases, incorporating multiple protein similarities helps to improve alignment quality. In Chapter 7, we introduce a new protein structure alignment scoring function incorporating a variety of of protein similarities. In this initial study, we focus on incorporating atom coordinate similarity and sequence similarity into a more reliable alignment score.

Our results show that incorporating atom structure similarity and sequence similarity assists in finding better protein structure alignments that are (possibly significantly) more consistent with HOMSTRAD alignments [57], which are protein structure alignments examined by human experts. When atom structure similarity itself fails to find any alignments with any consistency to the HOMSTRAD alignments, our scoring function is nevertheless able to find alignments that are highly consistent with, or even identical to, the HOMSTRAD alignments. Thus, our result suggests the advantage of incorporating multiple protein similarities, instead of focusing on atom structure similarity, for the best alignment.

Some of the research described in this thesis has been published, while the remaining has been submitted to highly regarded journals. Specifically, the protein structure idealization method has been published in [13, 11]; the fast low-resolution similar protein structure retrieval method has been published in [14]; part of the protein structure alignment and the alignment scoring function has been published in [12]; the protein structure superimposition method has been submitted as an application note; and the protein structure alignment method has been submitted as an original paper.

Chapter 2

Fundamentals

Before introducing our low-resolution approach and our high-resolution approach to evaluate the similarities among protein structures, the fundamentals are described in this chapter. Specifically, *the protein structure idealization problem* is introduced in Section 2.2, *the neighbor protein structure retrieval problem* is introduced in Section 2.3, *the protein structure superimposition problem* is introduced in Section 2.4, *the protein structure alignment problem* is introduced in Section 2.5, and *the protein structure alignment scoring function* is introduced in Section 2.6.

2.1 Terminologies

Proteins are biological molecules performing different functions within living organisms. A protein complex contains one or multiple chains of amino acids connected by peptide bonds. The 3-dimensional structure of a single chain of the protein complex is called a protein tertiary structure, and the amino acid sequence of a single chain is called a protein primary structure. Unless specifically specified, a protein tertiary structure is referred to as a protein structure, and a protein primary structure is referred to as a protein sequence in this manuscript.

Protein structures were determined experimentally and stored in the protein data bank

(PDB) [5]. Recently, the number of known protein structures stored in PDB had exceeded 90,000. In PDB, each protein structure is represented by the coordinates of the atoms of the protein. Since different amino acids contain different numbers and types of atoms, it is not convenient to perform an all-atom similarity evaluation between two protein structures. Thus, researchers tend to focus on the backbone atoms. In this manuscript, the coordinates of the C_α atoms on the protein backbone are used as the representatives for amino acids when comparing protein structures.

In case that the aligned residue pairs of two protein structures are given, the similarity between the two protein structures can be evaluated by the the root-mean-square deviation (RMSD) defined as follows:

$$\text{RMSD} = \sum_{i=1}^{n} \sqrt{\frac{d_i^2}{n}}, \tag{2.1}$$

where d_i is the Euclidean distance between the representative C_α atoms of the i-th aligned residue pair, and n is the number of aligned residue pairs. In practice, the optimal RMSD can be calculated by Kabsch's method [38] or a quaternion-based characteristic polynomial method [71].

When the superimposition of two protein structures is given, the similarity between the two protein structures can be evaluated by the TM-score [85] defined as follows:

$$\text{TM-score} = \frac{1}{N} \sum_{i=1}^{n} \frac{1}{1 + d_i^2/d_0^2}, \tag{2.2}$$

where d_i is the Euclidean distance between the representative C_α atoms of the i-th aligned residue pair, d_0 is the normalization distance defined as $d_0 = 1.24(N-15)^{1/3} - 1.8$, n is the number of aligned residue pairs, and N is the normalization number defined on the number of residues of the protein structures. In practice, the optimal TM-score can be calculated by the Needleman-Wunsch dynamic programming algorithm [60].

Other than TM-score [85], GDT-TS score [83] can also be used to evaluate the similarity between two protein structures when the superimposition is given. Formally, the GDT-TS score is defined as follows:

$$\text{GDT-TS} = \frac{n_1 + n_2 + n_4 + n_8}{4N}, \tag{2.3}$$

where n_d is the number of aligned residue pairs such that the Euclidean distance between the representative C_α atoms of the aligned residue pair is less than d, and N is the normalization number defined on the number of residues of the protein structures. Similar to TM-score, the optimal GDT-TS score can be calculated by the Needleman-Wunsch dynamic programming algorithm [60].

2.2 Protein Structure Idealization

When studying the functions of a protein, it is crucial to know the three-dimensional structure consisting of the Cartesian coordinates of all the atoms of the protein. These atoms are bonded together by inter-atomic forces called chemical bonds. It has been observed that the bond lengths and angles of the same type assume a Gaussian distribution with a small standard deviation (STDEV) in high resolution protein structure data. Typically, the bond lengths on protein backbones have STDEVs between 0.019Å and 0.033Å while the bond angles on protein backbones have STDEVs between 1.5° and 2.7° [21, 22]. These results suggest the possibility for modeling protein structures with the mean values of bond lengths and angles, which are often referred to as *ideal values*.

Ideal bond lengths and angles have been widely used in nuclear magnetic resonance (NMR) protein structure determination [28] and in protein structure prediction [69, 50, 8]. Moreover, stereochemical restraints are also used in X-ray protein structure determination [44, 45]. In protein structure prediction, the main advantage for using ideal bond lengths and angles is a reduction in the search space for the target protein structure. Specifically, if the target protein has n amino acids, the number of N, C_α and C atoms on the backbone is $3n$; thus, the Cartesian search space for the idealized backbone structure has a degree of freedom of $9n$ [70]. However, if all bond lengths and angles have ideal values, the protein backbone structure can be represented by a series of bond torsion angles in the feasible bond torsion angle space. In this case, the degree of freedom is reduced to approximately one tenth of that in the Cartesian space [70].

Although ideal bond lengths and angles have been widely used and accepted, we are not aware of any research done to evaluate how accurately it is possible to model protein

structures with dihedral angles. This motivates us to solve what we call the *protein structure idealization problem*: given the coordinates of the target protein structure, find the coordinates of the optimal idealized protein structure. Here, an idealized protein structure is a protein structure with bond lengths and angles that are ideal with respect to a given scoring function; the function depends on the resultant structure's statistical energy, as well as its similarity with the target structure. Thus, the idealized protein structure is taken to be a protein-like structure that is close to the target protein structure.

We solve the protein structure idealization problem by idealizing the backbone structure first and then idealizing the side-chain structure. This approach is widely accepted because previous research suggests that the backbone conformation is archived before the side-chain conformations are archived [18]. In our work, Ω dihedral angles are rounded to be either 0° or 180°. Some discussions on the properness of idealizing Ω dihedral angles can be found in [23, 34].

We introduce a novel dynamic programming algorithm with a run-time complexity of $O(n/\epsilon^8)$, where ϵ is a small constant, to find the optimal idealized protein backbone structure according to our scoring function. In practice, we observed that it is unnecessary to remember the entire dynamic programming table. Thus, with a filtering technique, the run-time complexity is further reduced to $O(nm/\epsilon)$, where m is a constant integer.

Given the idealized protein backbone structure, idealized side-chain structures are determined using an exhaustive search which assumes that side-chain structures for different residues are independent from each other. The scoring function is similar to the one we used for backbone structure idealization. In practice, we observe that it is efficient to regenerate idealized structures that are similar to a given idealized structure. We also refine the idealized backbone and side-chain structures according to our scoring functions iteratively.

We use our algorithm to evaluate how accurately it is possible to model protein structures with dihedral angles. We idealize all the X-ray protein structures from PDB [5], which satisfy the high resolution and the low sequence identity constraints downloaded on June 6, 2008 [72, 73]. The results show that such idealized structures always exist and that they are very similar to the target structures, in terms of the root-mean-square deviation

(RMSD) of C_α or all atoms. Moreover, the idealized backbone structures tend to have dDFIRE statistical energy scores [88], which are significantly better than the target structures. The results support our conclusion that it is possible to model protein structures accurately with dihedral angles on all high resolution protein backbone structures.

One application of the protein structure idealization algorithm is for refining protein pseudo-structures, either determined in experiments or predicted by computers. We have demonstrated one such case for improving poor (Φ, Ψ) dihedral angles of protein structures determined by NMR. The result is also consistent with the previous experiment showing that the idealized structure has a small RMSD and better backbone statistical energy. In the conclusion, we discuss several potential applications for our protein structure idealization algorithm.

2.3 Neighbor Protein Structure Retrieval

While assessing the similarities and differences between protein structures is a common practice in structural biology, efficiently performing this comparison is critical in some applications. For example, once a new protein structure is determined, researchers often need to infer its function or evolution by studying proteins with similar structures. A few databases, such as SCOP [59, 9] and CATH [61], maintain hierarchical classifications of known protein structures. The need to obtain structures similar to the new protein from these databases motivates the *neighbor protein structure retrieval problem*: given a query protein structure and a database of protein structures, retrieve all the structures in the database that are similar to the query structure.

One intuitive solution to the neighbor protein structure retrieval problem is to align the query protein structure with every protein structure of the database using a pairwise protein structure alignment tool. One successful approach for pairwise protein structure alignment is to represent protein structures as 3D coordinates and to find the optimal residue mapping and orientation (rotation and translation) together, as stralign [1], CE [68], LGA [83], SSM [41], TM-align [87], and SPalign [81] proposed. An orientation-free approach is possible by encoding each structure as a 2D matrix of residue-residue interaction distances;

comparison between two structures can be performed by an "alignment" of their respective matrices, as proposed in DALI [32]. One drawback of adopting these approaches is inefficiency, especially when the protein structure database is large. Whether or not 3D coordinates or distance matrices are used, solving the pairwise protein structure alignment problem is time consuming. Thus, all of these pairwise protein structure alignment tools adopt heuristic approaches without a global optimality guarantee. Unfortunately, such heuristic approaches are still time-consuming [40, 2].

The concern for efficiency has prompted the use of 1D protein structure profiles, which often perform very well. In particular, the state-of-the-art method, FragBag [7], has been shown experimentally to be fast and accurate, on average. Specifically, FragBag represents a protein structure as a profile that contains counts of structure fragments in a fragment alphabet learned from known structures. Then, neighbor protein structures can be retrieved by comparing the profiles efficiently. One drawback, however, is that although FragBag is capable of delivering high average accuracy, its accuracy is sometimes significantly worse than average. This accuracy drop occurs when two structures are similar in many local fragments but differ significantly in their overall structure; since FragBag compares only local contacts, it fails to identify the large non-local discrepancy in these structures.

We present ContactLib, a contact group library that contains all known contact groups which are to be used as fingerprints of protein structures. FragBag [7] and local feature frequency profile (LFFP) [10] are two promising tools that are closely related. Our ContactLib is different from FragBag and LFFP in the following ways: (1) FragBag and LFFP are developed on general structure fragments, while ContactLib introduces both local and remote contact groups eliminating potentially weak contact groups; (2) FragBag and LFFP use 3D coordinates or 2D distance matrices, while ContactLib introduces 1D distance vectors that can be efficiently indexed; (3) FragBag and LFFP require a predefined word alphabet, while ContactLib avoids using a word alphabet and introduces some freedom of specifying similarity thresholds at runtime; (4) FragBag and LFFP use word frequency profiles and distance functions from the text information retrieval problem, while ContactLib introduces a combined hit-rate scoring function for the neighbor protein structure retrieval problem. Since the word alphabet of LFFP is not publicly available, we focus on comparing ContactLib and FragBag in our experiments.

We built two ContactLibs: ContactLib-9L, that models local contacts, and ContactLib-3R, that models remote contacts. Using one or both ContactLibs, we tested our method on the high-quality protein structure subset of SCOP30 [59, 9] containing 3,297 protein structures. For each protein structure, we retrieved its neighbor protein structures from those remaining.

According to the receiver-operating characteristic (ROC) curve analysis [24], the best area under the ROC curve (AUROC), archived by ContactLib, is as high as 0.960. This is a significant improvement when compared to 0.747, which is the best result achieved by FragBag [7]. Specifically, when ContactLib-3R is used, 75% of the AUROC's are higher than 0.936, and the lowest AUROC is 0.504. When ContactLib-9L is used, 75% of the AUROC's are higher than 0.823, and 3% of the AUROC's are lower than 0.5. However, when FragBag is used, 75% of the AUROC's are higher than 0.657, and 10% of the AUROC's are lower than 0.5. Therefore, the worst-case AUROC is significantly improved by using ContactLib, and ContactLib-3R is furthermore able to guarantee an AUROC higher than that achieved with a random method, which has an AUROC equal to 0.5.

2.4 Protein Structure Superimposition

Given the aligned residue pairs of two aligned protein structures and the scoring function, *the optimal protein structure superimposition problem* is defined as finding the optimal superimposition that maximizes the alignment score. The problem has many applications in computational structural biology. For example, to evaluate the quality of protein structure prediction methods, a TM-score [85] and a GDT-TS score [83] are calculated between each pair of the predicted and the native structures during the ninth Critical Assessment of protein Structure Prediction (CASP) [39]. To calculate these alignment scores, the optimal superimposition problem must first be solved [85, 83].

The optimal protein structure superimposition problem is challenging. However, if the root-mean-square deviation (RMSD) is used as the scoring function, the problem of finding the maximum set of aligned residue pairs within a RMSD threshold may be solved in $O(l^{11})$ time [52], where l is the number of aligned residue pairs. Although the problem

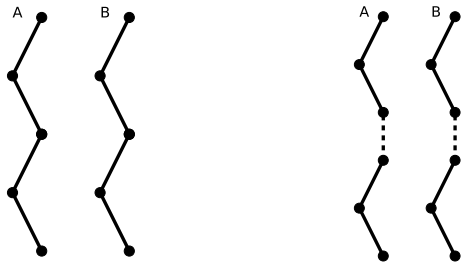
can be solved in polynomial time, implementation is computationally expensive. Thus, heuristic approaches have been adopted in state-of-the-art methods, TMscore [85] and LGA [83]. Specifically, initial *rotation and translation (ROTRAN)* is sampled by minimizing the RMSD of contiguous aligned residue pairs, and is refined iteratively by extending the number of aligned residue pairs, which are geometrically close after the superimposition, until convergence.

We introduce *PROSTA-super*, a simple and reliable method to solve the optimal protein structure superimposition problem. Our superimposition method relies on and verifies the fact that the optimal protein structure alignment always contains a small subset of aligned residue pairs, called a *seed*, such that the ROTRAN, which minimizes the RMSD of the seed, yields both the optimal ROTRAN and the optimal alignment score. Unlike current superimposition methods [85, 83], our superimposition method employs not only consecutive seeds that contain contiguous aligned residue pairs, but also employs gapped seeds that do not require the aligned residue pairs to be contiguous. Then, alignment scores are calculated directly from sampled ROTRANs. Thus, our superimposition method does not require seed assembly or seed extension. Moreover, weighted RMSDs of gapped seeds are used in the refinement step in our superimposition method.

Our protein structure superimposition method, PROSTA-super, has been implemented in C++ with OpenMP. Our result shows that our superimposition method is capable of finding TM-scores [85] and GDT-TS scores [83] that are similar to or (possibly significantly) higher than those found by TMscore [85] and LGA [83], respectively. Moreover, our method is significantly faster than LGA, and it has the added benefit of concurrent programming. One can also trade accuracy for speed by using fewer types of seeds in the first step or by reducing the number of refinement iterations in the second step.

2.5 Protein Structure Alignment

Protein structures are usually modeled as 3-dimensional coordinates of atoms. Thus, the alignment of two protein structures can be modeled as an optimization problem to minimize the distance between two protein structures after a specific rigid transformation (i.e.,



(a) Aligned local fragments (b) Aligned remote fragment pairs

Figure 2.1: Aligned local fragments and remote fragment pairs between protein structures A and B: each vertex represents a residue (the number of residues can be different); each solid edge represents a pseudo-bond between adjacent residues along the chain; and each dashed edge represents an arbitrary number of residues (possibly from different chains).

a rotation and a translation, referred to as *ROTRAN*). Given the ROTRAN, the alignment can be calculated by the Needleman-Wunsch dynamic programming algorithm [60]; and given the aligned residue pairs, the ROTRAN minimizing the root-mean-square deviation (RMSD) can be found by Kabsch's method [38] or a quaternion-based characteristic polynomial method [71].

Most existing methods for protein structure alignment are heuristic in nature due to the intensive computation in search of optimal aligned residues [1, 49, 68, 62, 41, 74]. For example, TM-align [87] creates an initial alignment through primary and secondary structure alignments and extracts an initial ROTRAN (or alignment) accordingly. Then, the ROTRAN is improved iteratively until convergence. This approach suffers from possibly unsatisfactory initial alignments and from a lack of optimality guarantees in the final result. TM-align was improved by the fragment-based approach in fr-TM-align [63], in which local structure alignments are computed and represented by the fragment alignments. A dynamic programming algorithm is then employed to assemble the fragment alignments and to generate a set of initial ROTRANs. Finally, ROTRANs are refined iteratively. However, this method only guarantees the quality of the local alignment rather than of the global alignment.

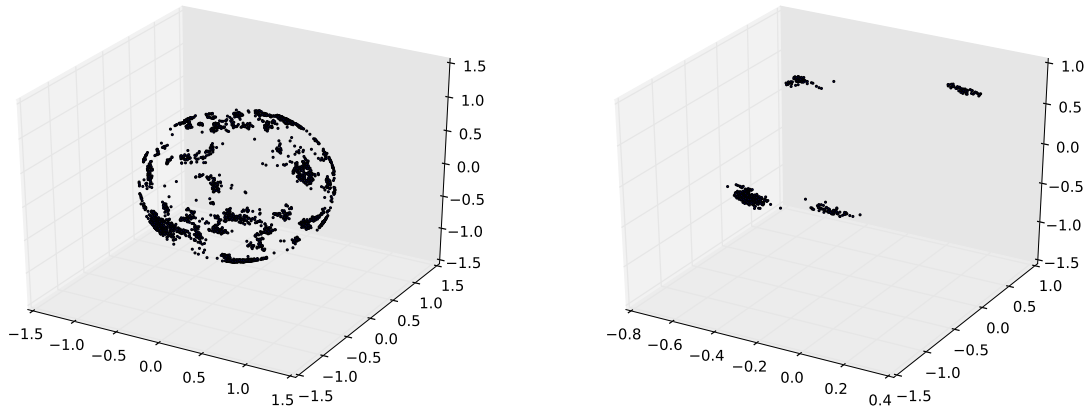
Our PROtein STructure Alignment (PROSTA, meaning simple in Polish) method is

based on the simple observation that similar protein structures share many small subsets of residues (or C_α atoms) such that the ROTRAN minimizing the RMSD of the small subset of residues is similar to the ROTRAN yielding the optimal protein structure alignment. Unlike fr-TM-align [63] that focuses on the small sets containing local fragments, we also consider the situation in which the small sets contain remote fragment pairs (or contacts). Here, a *remote fragment pair* is a pair of fragments that are distant within the same protein (as shown in Figure 2.1). The remote fragment pairs implicitly model the global topology, which is critical to the success of the protein structure alignment. Especially for structure alignments consist of mainly α-helices, the global topology becomes more important because local structures tend to be similar between α-helices.

Using remote fragment pairs is actually one of the reasons why DALI [32] becomes a successful protein structure alignment method. Instead of modeling protein structures with 3-dimensional coordinates of atoms, DALI models protein structures as $C_\alpha - C_\alpha$ distance matrices. First, highly similar 6-by-6 sub-matrices (between remote fragment pairs) are selected and assembled as initial alignments. Then, a Monte Carlo optimization is used as refinement. One advantage of using distance matrices is that it avoids superimposing. As a side effect, DALI does not yield a ROTRAN as our method and most existing alignment methods [**?**, **?**].

One critical computational challenge for DALI [32] is the large number of similar remote fragment pairs. If there are n residues in the query protein structure, the number of similar remote fragment pairs is $O(n^4)$. This is the main reason why existing alignment methods tend to avoid using remote fragment pairs. Moreover, the contact map overlap problem is NP-hard [26, 53]. To overcome the computational inefficiency, we choose to filter the ROTRANs by a fast star-like k-median clustering algorithm [3], rather than by using an exhaustive method. Since similar protein structures tend to have many local fragments or remote fragment pairs that have similar rotation matrices minimizing RMSDs, these rotation matrices tend to form a large cluster, as shown in Figure 2.2.

Clustering significantly improves the efficiency of our protein structure alignment method. Since the area of the rotation matrix space is a constant, and we assume that the maximum distance between two rotation matrices within a cluster is also a constant, the maximum number of clusters within the rotation matrix space is limited. This implies that the num-



(a) ROTRANs initially sampled

(b) ROTRANs of the four largest clusters

Figure 2.2: ROTRANs, before and after clustering, when aligning SCOP domains d3k2aa_ and d2cufa1: each ROTRAN is represented by a coordinate that is calculated by applying the rotation matrix of the ROTRAN on coordinate $(1, 0, 0)$; the four largest clusters include 19% of the initially sampled ROTRANs; and the optimal ROTRAN that maximizes the alignment score is located in the largest cluster, which includes 13% of those ROTRANs initially sampled.

ber of ROTRANs required to accurately identify large clusters is also limited. Therefore, only a limited number of ROTRANs are sufficient to identify the large cluster containing near optimal ROTRANs. Moreover, clustering also helps to eliminate refinements on similar protein structures because it is unnecessary to refine multiple alignments from the same cluster.

Some common techniques for generating initial alignments, which are shared by current protein structure alignment methods, are primary and secondary structure alignments and fragment alignment assembly or extension [40, 29]. However, our method does not rely on any of these techniques. Instead, ROTRANs minimizing the RMSDs of the small subset of residues are sampled, and global alignments are calculated directly from the sampled ROTRANs. Here, we introduce an initial alignment generation by a ROTRAN sampling from local fragments and remote fragment pairs, and an alignment refinement by a random ROTRAN sampling from six (possibly non-successive) aligned random residues. Our method relies on and verifies the assumption that similar protein structures tend to have small subsets of residues, such that the ROTRAN minimizing the RMSD of the small subset of residues is similar to the ROTRAN yielding the optimal protein structure alignment.

Experimental results suggest that both local fragments and remote fragment pairs show a significant contribution to finding higher TM-scores [85]. Specifically, if the highest TM-score found by TM-align [87] is lower than 0.6 and the highest TM-score found by one of the tested methods is higher than 0.5, our alignment method tends to discover better protein structure alignments, with (up to 0.21) higher TM-scores. In such cases, TM-align fails to find TM-scores higher than 0.5 with a probability of 42%, while our alignment method fails the same task with a probability of only 2%. This could significantly improve the accuracy of fold detection if the cutoff TM-score of 0.5 is used as the threshold of fold detection.

Comparing the GDT-TS scores [83] found by our alignment method to those found by LGA [83], we find that our alignment method consistently finds similar or higher GDT-TS scores than LGA does. Specifically, our alignment method improves GDT-TS scores by 0.06, on average, and by 0.44, in the best case. As a result, our alignment method finds 156% more alignments with GDT-TS scores higher than 0.5. Similar results are observed

comparing the GDT-TS scores found by our method and those found by SPalign [81]. Again, our protein structure alignment method is shown to be more reliable for finding high quality protein structure alignments.

2.6 Protein Structure Alignment Scoring Function

Proteins perform a diverse set of functions within living organisms, and it is widely accepted that protein function is largely determined by protein structure. Thus, protein structures are often aligned for their common substructures, to discover functionally or evolutionarily meaningful structure units. Current superimposition-based protein structure alignment methods share two common components: the superimposition method, which defines how the superimposition space is searched, and the scoring function, which evaluates the quality of protein structure alignments. Given the superimposed atom coordinates, the aligned residue pairs and the alignment score can be calculated by the Needleman-Wunsch dynamic programming algorithm [60]. Thus, the quality of the scoring function is critical to the success of the alignment method.

One limitation of state-of-the-art protein structure alignment scoring functions, specifically, the TM-score [85] and the LG-score [49], is that only atom coordinate similarity is taken into consideration, while other important protein similarities, such as sequence similarity, are ignored. It has been observed that many protein structure alignments, based only on atom coordinate similarity, are highly sensitive to conformational changes [64]. Thus, sequence similarity has recently been incorporated into protein structure alignment scoring functions [15, 74]. Here, we introduce a new scoring function that is capable of incorporating a variety of protein similarities, and focus on the impact of sequence similarity in this initial study.

Our results demonstrate that protein sequence similarity enables discovery of high quality protein structure alignments that are more consistent with HOMSTRAD alignments [57], which are curated by human experts. When atom coordinate similarity itself fails to find any alignments with any consistency to the HOMSTRAD alignments, our scoring function is nevertheless able to find alignments that are highly consistent with,

or even identical to, the HOMSTRAD alignments curated by human experts. Moreover, when the aligned protein structures contain mostly helices, TM-score [85], which involves only atom coordinate similarity, sometimes introduces residue shifting in the alignments. However, residue shifting tends to be avoided by using our scoring function, which involves both atom coordinate similarity and sequence similarity.

Chapter 3

How accurately one can model protein structures with dihedral angles?

In this chapter, we solve the protein structure idealization problem in two steps: (1) the backbone structure is idealized first using a novel dynamic programming algorithm optimizing our scoring function as described in Section 3.1; (2) the side-chain structure is idealized using an exhaustive search which assumes that side-chain structures of different residues are independent from each other as described in Section 3.2. We use our algorithm to evaluate how accurately one can model protein structures with dihedral angles in Section 3.3.1. The results demonstrate that it is possible to model protein structures accurately with dihedral angles on all high resolution protein backbone structures. We also demonstrate one application of the protein structure idealization algorithm to refine protein pseudo-structures either determined in experiments or predicted by computational methods in Section 3.3.2.

3.1 Protein Backbone Structure Idealization

Given the target protein backbone structure, we would like to find the optimal idealized backbone structure. For an idealized protein backbone structure, the coordinates of O, H and C_β backbone atoms can be calculated from the coordinates of N, C_α and C backbone atoms. Thus, we specifically describe how to generate coordinates of N, C_α and C atoms in this section. For simplicity, a protein backbone structure is always referred to as a structure unless strictly specified.

3.1.1 Idealized Backbone Structure Generation

Given the target structure, we would like to generate idealized structures fulfilling two generation goals. First, the idealized structures should be similar to the target structure. Second, each pair of idealized structures should be some distance away to avoid redundant computation. Furthermore, we are interested in generating as many of these idealized structures as possible.

Before describing how we fulfill the generation goals, we describe a simple distance metric to measure the distance between two sets of coordinates representing the target protein. Let P_i be a set of coordinates representing the target protein, and $P_i^j \in P_i$ be the coordinate of the j-th atom of the target protein. Thus, there is $P_i = \{P_i^1, P_i^2, ..., P_i^{3n}\}$, where n is the number of amino acids of the target protein. For simplicity, let P_0 always represent the target structure, and P_i represent a generated idealized structure for $i > 0$. Let $D(P_i^k, P_j^k)$ be the Euclidean distance between P_i^k and P_j^k. We describe the distance between P_i and P_j as the bottleneck distance:

$$D(P_i, P_j) = \max_k D(P_i^k, P_j^k). \tag{3.1}$$

Using this distance metric, we fulfill both generation goals by satisfying the following generation constraints:

$$\begin{cases} D(P_0, P_i) \leq r & \forall i > 0 \\ D(P_i, P_j) \geq \epsilon & \forall i, j > 0 \end{cases}. \tag{3.2}$$

The first generation constraint assumes that the accuracy of the coordinates of the target structure is reasonably good, and no-worse than error threshold r. If this constraint is satisfied, the distance between the target coordinate and any generated coordinates representing the same atom is upper bounded by r. Thus, it is reasonable for any generated idealized structure P_i to be considered similar to target structure P_0. If the second generation constraint is satisfied, for each pair of generated idealized structures, there exists a pair of coordinates, one from each structure representing the same atom, such that they are at least ϵ distance away from each other. Therefore, both generation goals are achieved.

These generation constraints suggest limiting the search space inside a sphere with radius r, and discretizing the search space with grids of size ϵ. When $\epsilon = 0.001$Å, the accuracy of X-ray crystallography [48] and PDB (protein database) format [65] is reached. Thus, this method is capable of generating all possible idealized structures at the accuracy of X-ray crystallography and PDB format.

Given the limited and discretized search space of each atom, one can generate idealized structure coordinates from the first atom to the last atom. For the first atom, an idealized coordinate lies within a sphere. Thus, the number of generated coordinates is bounded by $O(1/\epsilon^3)$. For each generated coordinate P_i^1 of the first atom, an idealized coordinate of the second atom lies on a ball surface with a constant distance to P_i^1. Thus, the number of generated coordinates is bounded by $O(1/\epsilon^2)$. For each generated coordinate pair (P_i^1, P_i^2) of the first two atoms, an idealized coordinate of the third atom lies on a circle with a constant distances to P_i^1 and P_i^2. Thus, the number of generated coordinates is bounded by $O(1/\epsilon)$. Similarly, the number of generated coordinates for any of the following atoms is also bounded by $O(1/\epsilon)$. Moreover, since we round Ω dihedral angles to either 0° or 180°, the coordinate of any C_α atom is unique and can be calculated from the coordinates of the previous three atoms.

Therefore, the total number of coordinates generated for all atoms is bounded by $O(1/\epsilon^{2n+4})$ by induction. Here, it is acceptable to assume that r is a constant because it is only related to the first atom. For subsequent atoms, we did not limit the search space to be inside the sphere with radius r as described above, and thus the actual number of generated coordinates should be much smaller in practice.

3.1.2 Idealized Backbone Structure Scoring Function

Given the generated idealized structures $\{P_i\}$, we need a scoring function $S_{BB}(P_i)$ to find the optimal idealized structure. The scoring function should evaluate not only the similarity between generated idealized structure P_i and target structure P_0, but should also evaluate the statistical energy of P_i, to ensure that P_i is protein-like. Thus, we define our scoring function as follows:

$$S_{BB}(P_i) = S_f(P_i) - D_\alpha(P_i, P_0) - D_\beta(P_i, P_0) - D_H(P_i, P_0) - D_{\Phi,\Psi}(P_i, P_0), \tag{3.3}$$

where $S_f(P_i)$ is the statistical energy score, $D_\alpha(P_i, P_0)$ is the root mean square deviation (RMSD) of C_α atoms, $D_\beta(P_i, P_0)$ is the RMSD of C_β atoms, $D_H(P_i, P_0)$ is the RMSD of the hydrogen and oxygen atoms participating in hydrogen bonds, and $D_{\Phi,\Psi}(P_i, P_0)$ is the angular RMSD of (Φ, Ψ) dihedral angles.

In our scoring function, the statistical energy is evaluated by a (Φ, Ψ) dihedral angle log-odd score as the statistical energy score $S_f(P_i)$. Specifically, we discretize the Ramachandran plot into grids of 360 by 360, and draw one plot for each type of amino acid. Then, we calculate the log-odd score $S_f(P_i^{1,t})$ of idealized structure $P_i^{1,t}$ of the first t atoms:

$$S_f(P_i^{1,t}) = \sum_{5 \le i \le t, A_i = C_\alpha} \log \frac{P_{AA_{i-3}}(\Phi_{i-3}, \Psi_{i-3})}{P_{null}(\Phi_{i-3}, \Psi_{i-3})}, \tag{3.4}$$

where one log-odd score is calculated at each C_α atom (by checking that atom type A_i is C_α) for the previous amino acid (represented by the previous C_α atom at $i-3$), $P_{AA_{i-3}}(\Phi_{i-3}, \Psi_{i-3})$ is the probability of the grid containing (Φ_{i-3}, Ψ_{i-3}) on the Ramachandran plot of amino acid type AA_{i-3}, and $P_{null}(\Phi_{i-3}, \Psi_{i-3})$ is the probability of the null model with a uniform distribution such that $P_{null}(\Phi_{i-3}, \Psi_{i-3}) = \frac{1}{360}\frac{1}{360}$.

Structure similarity is evaluated by other distance matrices in our scoring function. We use $D_\alpha(P_i, P_0)$ and $D_{\Phi,\Psi}(P_i, P_0)$ to $D_\beta(P_i, P_0)$ to serve as a distance metric to conserve the side-chain structure compatibilities. Here, $D_{\Phi,\Psi}(P_i, P_0)$ is defined as follow:

$$D_{\Phi,\Psi}(P_i, P_0) = \sqrt{\sum_{i \in \{C_\alpha \text{ atoms}\}} \frac{D(\Phi_i, \Phi_0)^2 + D(\Psi_i, \Psi_0)^2}{n}}, \tag{3.5}$$

where Φ_i and Ψ_i are the Φ and Ψ dihedral angles of the residue containing the i-th atom, $D(\Phi_i, \Phi_0) = \min(|\Phi_i - \Phi_0|, 2\pi - |\Phi_i - \Phi_0|)$, and $D(\Psi_i, \Psi_0) = \min(|\Psi_i - \Psi_0|, 2\pi - |\Psi_i - \Psi_0|)$. Moreover, we use $D_H(P_i, P_0)$ to serve as a distance metric to conserve the hydrogen bonds. Thus, some global dependencies are addressed implicitly by distance matrices $D_\beta(P_i^{1,t}, P_0^{1,t})$ and $D_H(P_i, P_0)$.

3.1.3 Dynamic Programming Algorithm

Theoretically, one can calculate scores for all generated idealized structures as similar structures always have similar scores. More formally, the method requires the assumption that $D(P_i, P_j) \leq \epsilon \implies |S_{BB}(P_i) - S_{BB}(P_j)| \leq \epsilon_s$, which is reasonable for small ϵ. Note that, since the total number of generated idealized structures is bounded by $O(1/\epsilon^{2n+4})$, this method is computationally expensive. Thus, we introduce a dynamic programming algorithm with a filtering technique to find the optimal idealized structure efficiently. For simplicity, let $P_i^{a,b} = \{P_i^a, P_i^{a+1}, P_i^{a+2}, \ldots, P_i^b\}$.

The dynamic programming algorithm has two assumptions. One assumption is that given two generated idealized structures $P_i^{1,t-1}$ and $P_j^{1,t-1}$, such that $D(P_i^{t-k,t-1}, P_j^{t-k,t-1}) \leq \epsilon$, there always exists a generated coordinate P_j^t for any generated coordinate P_i^t, such that $D(P_i^t, P_j^t) \leq \epsilon$. We observed that counter examples when $k \geq 5$ are rare, though they do exist theoretically.

The other assumption of the dynamic programming algorithm is that the scoring function satisfies the additive property, such that $S_{BB}(P_i^{1,t}) = S_{BB}(P_i^{1,t-k}) \oplus S_{BB}(P_i^{t-k+1,t})$, under some addition operators $\oplus$. First, the statistical energy score $S_f(P_i^{1,t})$ satisfies the additive property while the addition operators $\oplus$ is defined as follows:

$$\begin{aligned} S_f(P_i^{1,t}) &= S_f(P_i^{1,t-k}) \oplus S_f(P_i^{t-k+1,t}) \\ &= S_f(P_i^{1,t-k}) + S_f(P_i^{t-k+1,t}). \end{aligned} \tag{3.6}$$

Second, distance matrices $D_\alpha(P_i^{1,t}, P_0^{1,t})$, $D_\beta(P_i^{1,t}, P_0^{1,t})$, $D_H(P_i^{1,t}, P_0^{1,t})$ and $D_{\Phi,\Psi}(P_i^{1,t}, P_0^{1,t})$ satisfy the additive property because RMSD $D_{RMS}(P_i^{1,t}, P_0^{1,t})$ satisfies the additive property

while the addition operators $\oplus$ is defined as follows:

$$\begin{aligned} & D_{RMS}(P_i^{1,t}, P_0^{1,t}) \\ = \; & D_{RMS}(P_i^{1,t-k}, P_0^{1,t-k}) \oplus D_{RMS}(P_i^{t-k+1,t}, P_0^{t-k+1,t}) \\ = \; & \sqrt{\frac{D_{RMS}^2(P_i^{1,t-k}, P_0^{1,t-k})(t-k) + D_{RMS}^2(P_i^{t-k+1,t}, P_0^{t-k+1,t})k}{t}}. \end{aligned} \tag{3.7}$$

These two assumptions are fundamental to our dynamic programming algorithm. By induction, the first assumption implies that if $D(P_i^{t-k,t-1}, P_j^{t-k,t-1}) \leq \epsilon$, for any generated idealized structure $P_i^{t,n}$, there always exists a generated idealized structure $P_j^{t,n}$ such that $D(P_i^{t,n}, P_j^{t,n}) \leq \epsilon$. Recall that the scoring function assumes that $D(P_i^{t,n}, P_j^{t,n}) \leq \epsilon \implies |S_{BB}(P_i^{t,n}) - S_{BB}(P_j^{t,n})| \leq \epsilon_s$, and thus there is $S_{BB}(P_i^{t,n}) \approx S_{BB}(P_j^{t,n})$. If $S_{BB}(P_i^{1,t-1}) \geq S_{BB}(P_j^{1,t-1})$, there is approximately $S_{BB}(P_i) = S_{BB}(P_i^{1,t-1}) \oplus S_{BB}(P_i^{t,n}) \geq S_{BB}(P_j^{1,t-1}) \oplus S_{BB}(P_j^{t,n}) = S_{BB}(P_j)$. Therefore, if $D(P_i^{t-k,t-1}, P_j^{t-k,t-1}) \leq \epsilon$ and $S_{BB}(P_i^{1,t-1}) \geq S_{BB}(P_j^{1,t-1})$, there is no need to generate $P_j^{t,n}$ to find an approximately optimal solution.

Based on this observation, we developed a novel dynamic programming algorithm. Idealized structures are still generated as previously described, but the generation process is stopped for some idealized structures if we know it cannot lead us to the optimal one. First, the search space for each atom of the target protein is discretized to grids of size ϵ. When generating coordinates for atom t, if $P_i^{t-k+1,t}$ and $P_j^{t-k+1,t}$ are located in the same grids $G^{t-k+1,t} = \{G^{t-k+1}, G^{t-k+2}, G^{t-k+3}, \ldots, G^t\}$, we know that there is no need to continue the generation process on the lower scoring one of $P_i^{1,t}$ and $P_j^{1,t}$. Thus, we define the dynamic programming table $T_{BB}(t, G^{t-k+1,t})$ to be the optimal idealized structure for each observed grids $G^{t-k+1,t}$ as follows:

$$\begin{cases} T_{BB}(t, G^{t-k+1,t}) & = \max\limits_{i, G^{t-k}} T_{BB}(t-1, G^{t-k} \cup G^{t-k+1,t-1}) \oplus S_{BB}(P_i^t) \quad \forall t > k \\ T_{BB}(k, G^{1,k}) & = \max\limits_{i} S_{BB}(P_i^{1,k}) \end{cases}, \tag{3.8}$$

where $P_j^{t-k,t} \in G^{t-k,t}$ and $S_{BB}(P_j^{1,t-1}) \oplus S_{BB}(P_j^t) = S_{BB}(P_j^{1,t})$. Thus, the dynamic programming table can be calculated from the first atom to the last atom. Finally, the optimal idealized structure is the one with the highest score $\max_g G^{3n-k+1,3n}$.

The run-time complexity of our dynamic programming algorithm depends on the value of k. To keep all possible (Φ, Ψ) dihedral angles of the previous residue when generating

C_α atoms, we have to choose $k \geq 5$. For speed, we choose $k = 5$ in our implementation. In this case, the number of score calculations required to calculate $T_{BB}(t, G^{t-4,t})$ is no more than the maximum number of coordinates sampled for six consecutive backbone atoms. Recall that there are exactly two C_α atoms in six consecutive backbone atoms, and the Ω dihedral angle is rounded. Thus, the coordinate of one C_α atom can be calculated from the coordinates of the other C_α atom and the two atoms between them. For this reason, the maximum number of sampled coordinates is bounded by $O(1/\epsilon^8)$. Moreover, the number of score calculations required to calculate $T_{BB}(k, G^{1,k})$ is no more than the maximum number of possible coordinates sampled for five consecutive backbone atoms, which is also $O(1/\epsilon^8)$. Therefore, the run-time complexity of our dynamic programming algorithm is $O(n/\epsilon^8)$.

To increase the speed for the dynamic programming algorithm, we applied an additional filtering technique to remember only the highly scored idealized structures. Specifically, the algorithm only remembers the optimal idealized structure for the top m scored tail configurations instead of all possible conformations. Thus, the run-time complexity is reduced to $O(nm/\epsilon)$. This approach works well in practice because an optimal idealized structure with a long poorly scored fragment is rare. Thus, we assumed that the local quality of the idealized structure should be reasonably high (in the top m entries of the score list).

3.2 Protein Side-chain Structure Idealization

After the backbone structure of the target protein has been idealized, we begin to idealize the side-chain structures. When doing this, the idealized backbone structure is considered to be rigid. This approach is widely accepted because previous research suggests that the backbone conformation is formed before the side-chain conformations are formed [18]. After the side-chain idealization, we should have a complete idealized protein structure with all of the backbone and the side-chain structures idealized.

Protein side-chains suffer from low quality when determining protein structures. This is mainly because side-chains are not as stable as backbones, and they are more likely to have disorder problems than are backbones in crystals [48]. Thus, the target side-chain

structure might be a poor reference for defining the search space and for evaluating the structure similarity score for generated idealized side-chain structures. To address this, we perform an exhaustive search on the entire feasible torsion angle space, instead of the limited torsion angle space, around the target side-chain structure.

Our side-chain idealization method assumes that the side-chain conformations of different residues are independent of each other. Otherwise, all residues with dependencies have to be generated together and the run-time complexity increases exponentially in the number of atoms involved. Moreover, the $N_{\eta 1} - C_\zeta - N_\epsilon - C_\delta$ and the $N_{\eta 2} - C_\zeta - N_\epsilon - C_\delta$ torsion angles of arginine residues are rounded to be either 0° or 180°. Then, the degree of freedom of the search space for each residue is at most four and it is now practical to perform an exhaustive search for each residue independently.

To find the optimal idealized side-chain structure, we design a new scoring function involving the similarity among the generated idealized side-chain structures and the target side-chain structures, and the statistical energy of the generated idealized side-chain structures. Let P_0 be the target side-chain structure of some residue, and P_i for all $i > 0$ be a generated idealized side-chain structure of the same residue. Then, the scoring function $S_{SC}(P_i)$ is defined:

$$S_{SC}(P_i) = S_f(P_i) - D_{H'}(P_i, P_0) - D_\chi(P_i, P_0),$$

where $S_f(P_i)$ is the statistical energy score, $D_{H'}(P_i, P_0)$ is the root mean square divergence (RMSD) of all non-hydrogen atoms, and $D_\chi(P_i, P_0)$ is the RMSD of χ torsion angles.

In our scoring function, the statistical energy score $S_f(P_i)$ is defined as a simple χ torsion angle log-odd score, which is similar to the statistical energy score of our backbone scoring function. Moreover, the log-odd score is based on the popular backbone dependent rotamer library downloaded from Dunbrack's lab [67]. Certainly, other local statistical energy scores can be adopted here. Similar to the backbone scoring function, $D_{H'}(P_i, P_0)$ and $D_\chi(P_i, P_0)$ serve as distance metrics to conserve the side-chain structure.

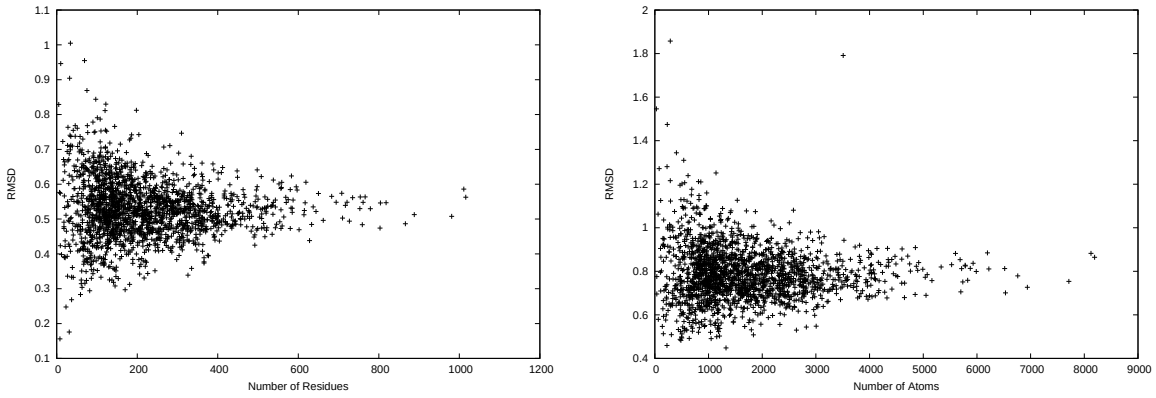
3.3 Results

To study the protein structure idealization problem and its applications, we implemente our protein structure idealization algorithm. In our implementation, we use the mean bond lengths and angles that had been reported in [22] as the ideal bond lengths and angles, respectively. When idealizing the protein backbone structure, we set the search space radius of an atom as $r = 1.6$Å and the discrete grid size as $\epsilon = r/5$. We find that $m = 50,000$ had a reasonable balance between speed and accuracy. When idealizing the protein side-chain structure, we set the search space of a rotamer dihedral angle to be within 3σ distance from the mean value, where σ is the STDEV of the rotamer dihedral angle, and we set the discrete grid size to be 10°. We also refine the idealized structure by iteratively reducing the search space and the discrete grid size by a constant factor of 0.5.

3.3.1 PDB Protein Structure Idealization

In this experiment, we addressed how accurately it is possible to model protein structures with dihedral angles. We idealized high resolution protein structures with low sequence identities of the CULLPDB_PC30_RES1.6_R0.25 data set [72, 73]. In fact, the CULLPDB_PC30_RES1.6_R0.25 data set is the complete set of X-ray protein structures in PDB [5] with a sequence identity cutoff of 30%, a resolution cutoff of 1.6Å, and an R factor cutoff of 0.25. In summary, the data set contains 1898 proteins with an average length of 227 residues, as downloaded on June 6, 2008.

To show that the idealized and the target backbone structures are very similar, we calculated the C_α-RMSD as shown in Figure 3.1(a). The C_α-RMSD is a popular distance metric to evaluate the backbone distance between two protein backbone structures. The result shows that most distances between the idealized and the target backbone structures are small with mean 0.53Å and STDEV 0.08Å. Specifically, the smallest C_α-RMSD reaches 0.16Å, and 90% of the C_α-RMSDs are smaller than 0.63Å. Moreover, the C_α-RMSD is upper bounded by 1.00Å, although the search space radius for each atom is set to be 1.6Å. This result is consistent with the result of checking (Φ, Ψ) dihedral angles, where the average difference between the idealized and the target (Φ, Ψ) dihedral angles



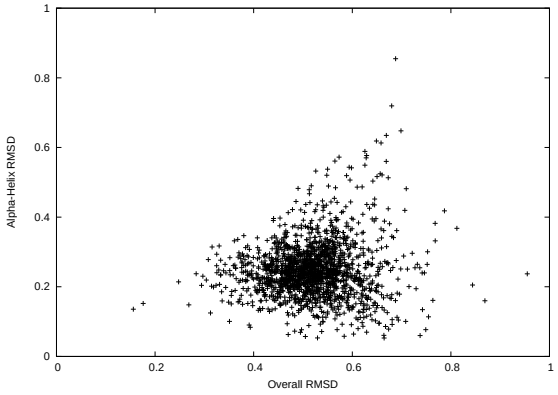
(a) C_α-RMSD (b) All-atom RMSD

Figure 3.1: RMSD between the idealized and the native backbone structures: (a) the C_α-RMSD has a mean of 0.53Å and a STDEV of 0.08Å; 90% of the C_α-RMSDs are smaller than 0.63Å; (b) the all-atom RMSD has a mean of 0.79Å and a STDEV of 0.13Å; and 90% of the all-atom RMSDs are smaller than 0.94Å.

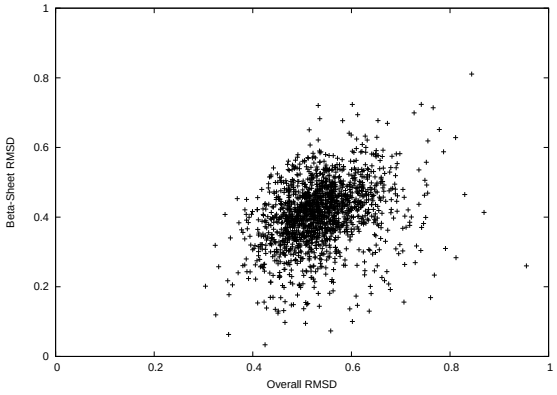
is as small as 0.08°. Therefore, it is possible to model protein backbone structures in CULLPDB_PC30_RES1.6_R0.25 accurately using only Φ and Ψ dihedral angles.

We studied the C_α-RMSD further in different regions of the target protein structures. In Figures 3.2(a) and 3.2(b), we see that the C_α-RMSD of the α-helix and the β-sheet regions are smaller than that of the complete protein by 0.28Å and 0.12Å, respectively. Indeed, these regions are more restricted because of using $D_H(P_i, P_0)$ to conserve hydrogen bonds of α-helices and β-sheets in our scoring function. We also observe that the C_α-RMSD of residues that are closer to the geometric center of a target protein structure is 0.13Å smaller on average than the C_α-RMSD of the other residues that are farther, as shown in Figure 3.2(c). Thus, the inner residues tend to be closer to the idealization state than are the outer residues. We did not observe any significant differences on the C_α-RMSD between the buried and the exposed regions.

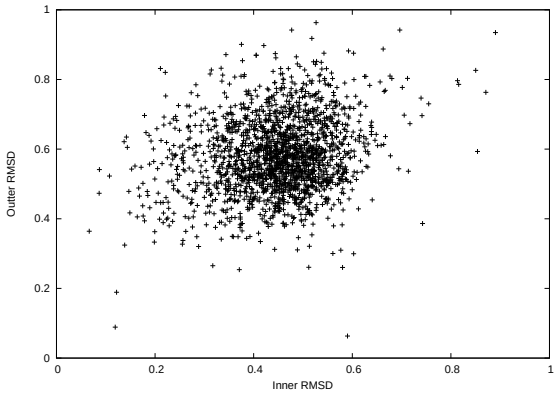
We also calculated the all-atom RMSD to show that the idealized and the target structures are very similar. In Figure 3.1(b), we see that most distances between the idealized and the target structures are small, with mean 0.79Å and STDEV 0.13Å. Moreover, the smallest all-atom RMSD reaches 0.45Å, and 90% of the all-atom RMSDs are smaller than



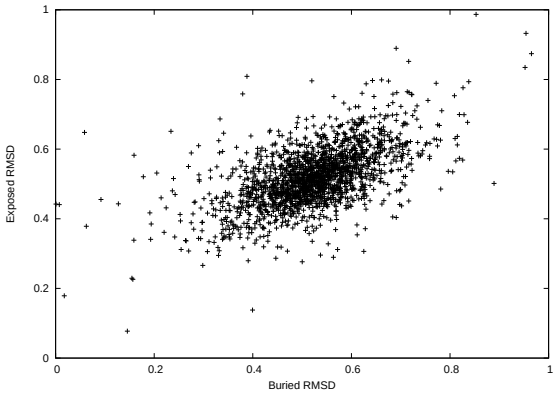
(a) All regions v.s. α-helix regions



(b) All regions v.s. β-sheet regions

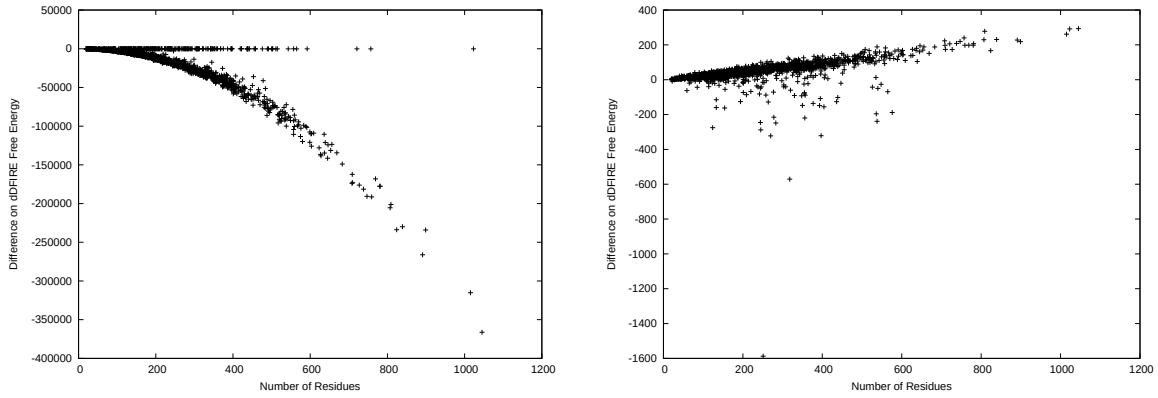


(c) Inner regions v.s. outer regions



(d) Buried regions v.s. exposed regions

Figure 3.2: C_α-RMSD in different regions of the target protein structure: (a,b) the C_α-RMSD of the α-helix and the β-sheet regions are smaller than that of the complete protein by 0.28Å and 0.12Åon average, respectively; (c) the C_α-RMSD of the residues that are closer to the geometric center of a target protein structure is 0.13Å smaller on average than that of the other residues; and (d) there is no significant difference on the C_α-RMSD between the buried and the exposed regions.



(a) Backbone statistical energy

(b) All-atom statistical energy

Figure 3.3: Difference on dDFIRE statistical energy values between the idealized and the native protein structures: (a) the dDFIRE statistical energy values of most idealized backbone structures are significantly better than those of the native backbone structures; (b) the dDFIRE statistical energy value is improved for 4.74% of the idealized protein structures by up to 1585, and is worsened slightly by 44 on average.

0.94Å. Note that both the C_α-RMSD and the all-atom RMSD between the idealized and the target structures tend to be stable when the target protein is long. Therefore, it is possible to model protein structures accurately with only Φ, Ψ, and χ dihedral angles.

The idealized backbone structures are also favored in terms of statistical energy. This is shown by checking the statistical energy differences between the idealized and the target protein backbone structures in Figure 3.3(a). Here, we calculate the statistical energy using dDFIRE [88], and observe that the dDFIRE statistical energy of most idealized backbone structures are significantly better than are those of the target backbone structures. For the rest without significant improvements, the difference is close to zero. This may be the result of some tight stereochemical restraints used in existing X-ray structure refinement programs [23, 34]. It is also interesting that the observed statistical energy improvements are clearly not independent from the protein length. The figure suggests that the statistical energy difference has a square dependence on the protein length.

After idealizing the side-chain structures, the statistical energy is either improved by

a relatively biger amount or worsened by a relatively smaller amount as shown in Figure 3.3(b). Unfortunately, in most cases, the statistical energy is worsened slightly but is still in a stable state with negative values. Again, here we used dDFIRE [88] to calculate the statistical energy. We observed that the dDFIRE statistical energy is improved for 90 or 4.74% of the idealized protein structures and is worsened slightly by 44 on average. Moreover, the dDFIRE statistical energy is improved by 1585 in the best case, and worsened by 293 in the worst case. The figure also suggests that the statistical energy difference has a linear dependence on the protein length.

Several side-chain prediction tools have been proven to predict accurate side-chain structures from native backbone structures [8, 42, 78, 79]. However, these tools do not perform well when predicting side-chain structures from predicted backbone structures. To address this, we compared the predicted side-chain structures given the native backbone structures and the predicted side-chain structures given the predicted backbone structures in terms of statistical energy. Here, we treat the idealized backbone structures of the CULLPDB_PC30_RES1.6_R0.25 data set as those which are best possibly predicted. Moreover, we used SCWRL4 [42] to predict side-chain structures and dDFIRE [88] to calculate statistical energies. The result shows that the statistical energy is worsened slightly by 43 if the predicted backbone structures are used. We do not think this difference is significant to side-chain prediction, and more experiments may show if this is conclusive.

Finally, we study the effects of idealization on hydrogen bonds. As shown in Table 3.1, we compare the number of hydrogen bonds detected by the DSSP program [37, 36]. Here, only differences of the most popular types of hydrogen bonds are included. We note that the total number of hydrogen bonds is increased by 1.59% or 0.012 per residue after idealization. Specifically, the effects of idealization on hydrogen bonds of β-bridges is minor, and the loss of the hydrogen bonds on α-helices is reasonably controlled under 1.48%. Interestingly, the idealized backbone structures have significantly more 2_7 ribbons. The reason behind this observation remains open.

In summary, we demonstrate that using dihedral angles with ideal bond lengths and angles is capable of modeling protein structures that are highly similar to the ones in CULLPDB_PC30_RES1.6_R0.25 [72, 73]. Since CULLPDB_PC30_RES1.6_R0.25 is the complete set of PDB protein structures satisfying the high resolution and the low sequence

Type	Count Difference	Percent Difference
Parallel Bridge	9	0.04%
Antiparallel Bridge	-211	-0.37%
2_7 Helix	7080	26.46%
3_{10} Helix	-1018	-2.35%
α Helix	-1644	-1.48%
π Helix	-82	-1.27%
All	5183	1.85%

Table 3.1: Difference on DSSP hydrogen bond before and after idealization: the effects of idealization on hydrogen bonds of β-bridges is minor, and the loss of the hydrogen bonds on α-helices is reasonably controlled.

identity constraints, it is reasonable to extend the conclusion to all protein backbone structures. A positive side effect is that idealization improves backbone statistical energy, while most hydrogen bonds are conserved.

3.3.2 NMR Protein Structure Refinement

In this experiment, we demonstrate an application of the protein structure idealization problem in NMR by idealizing 32 NMR protein structures. The NMR protein structures were randomly chosen from PDB [5] with a sequence identity cutoff of 30% and a gapless fragment length cutoff of 80 residues. In cases of multiple chains or models of some NMR protein structures, only the first chain from the first model is used in this experiment. This addition to the conclusion of the previous experiment shows that poor (Φ, Ψ) dihedral angles of the NMR protein structures are improved by idealizing them.

To demonstrate this, we compared the percentage of favored (Φ, Ψ) dihedral angles calculated by PROCHECK [47] in Table 3.2. After idealization, we see that 19 out of 32 NMR protein structures have more favored (Φ, Ψ) dihedral angles. Overall, the percentage is increased by 4.34% on average and 27.30% in the best case, which is closer to the minimum percentage of 90% expected in a good quality model [47].

PDB	Native	Ideal	Diff	PDB	Native	Ideal	Diff
1SSK	44.6%	71.9%	27.3%	2LBN	59.7%	77.6%	17.9%
2KQP	62.9%	80.0%	17.1%	1WPI	64.4%	81.4%	17.0%
1EXE	60.5%	76.7%	16.2%	2LNV	58.6%	72.4%	13.8%
1X6F	64.1%	73.1%	9.0%	2L6B	72.2%	81.1%	8.9%
2GFU	72.3%	80.4%	8.1%	1PC2	79.3%	87.4%	8.1%
2LMR	79.7%	87.0%	7.3%	2KA0	72.6%	78.3%	5.7%
2L3O	71.3%	76.9%	5.6%	1O1W	67.2%	72.1%	4.9%
2CQ9	78.3%	82.6%	4.3%	2RQA	72.0%	75.4%	3.4%
2D86	89.0%	92.1%	3.1%	1NTC	80.5%	83.1%	2.6%
2JZT	76.6%	79.0%	2.4%	2CZN	76.5%	76.5%	0.0%
1RCH	75.4%	74.6%	-0.8%	2JU1	77.1%	75.9%	-1.2%
2KV7	85.5%	84.2%	-1.3%	2JT2	83.6%	81.5%	-2.1%
2KYW	83.8%	81.1%	-2.7%	2OSR	82.7%	80.0%	-2.7%
2L6M	81.7%	78.5%	-3.2%	2CU1	81.1%	77.8%	-3.3%
1AJ3	93.3%	88.8%	-4.5%	1WI5	84.0%	78.0%	-6.0%
1NMW	85.0%	78.0%	-7.0%	2LBV	83.9%	74.8%	-9.1%

Table 3.2: Percentage of the favored (Φ, Ψ) dihedral angles of 32 NMR protein structures before and after idealization: 19 NMR protein structures have up to 27.30% more favored (Φ, Ψ) dihedral angles.

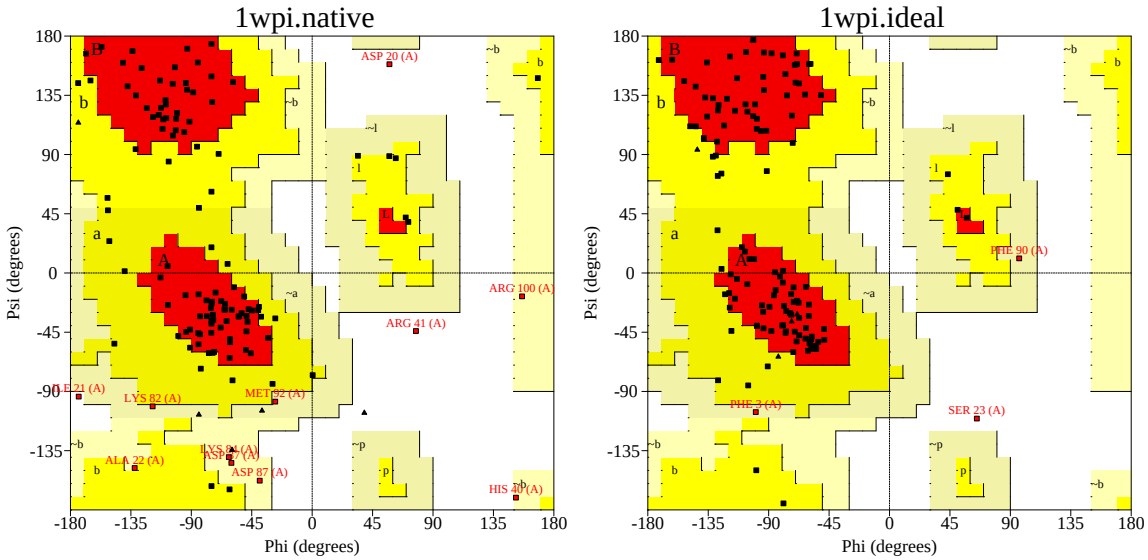


Figure 3.4: Ramachandran plots of the native structure (left) and the idealized structure (right) of NMR protein structure 1WPI: the native structure contains only 64.4% of (Φ, Ψ) dihedral angles in favored regions, while the idealized structure contains a significantly improved percentage of 81.4% of (Φ, Ψ) dihedral angles in favored regions.

Note that for those NMR protein structures that already have more than approximately 75% of favored (Φ, Ψ) dihedral angles, idealization harms the percentage by -0.85% on average. There are at least two reasons for this. First, our statistical energy score $S_f(P_i)$ is calculated from a data set that is different from the one used by PROCHECK. In fact, we used 1898 protein structures of the CULLPDB_PC30_RES1.6_R0.25 data set [72, 73], while PROCHECK used 118 protein structures, with a resolution cutoff of 2.0Å and an R factor cutoff of 0.20 [47]. Although the percentages of favored (Φ, Ψ) dihedral angles are decreased in Table 2, our statistical energy scores of proteins 1WI5, 1NMW, and 2LBV are increased by 0.22, 1.35, and 0.31, respectively, after idealization. Second, our implementation is trying to optimize our scoring function $S_{BB}(P_i)$, instead of optimizing only the statistical energy score. Thus, it is possible to see decreased statistical energy scores after idealization, especially when the target protein structure has a high percentage of favored (Φ, Ψ) dihedral angles.

Our conclusion is further supported by the case study of the NMR structure with PDB ID 1WPI. From the Ramachandran plots drawn by PROCHECK [47] in Figure 3.4, we find that (Φ, Ψ) dihedral angles tend to move towards favored regions. Specifically, the native structure contains only 64.4% of (Φ, Ψ) dihedral angles in favored regions, while the idealized structure contains a significantly improved percentage of 81.4% of (Φ, Ψ) dihedral angles in favored regions. Moreover, the native structure contains three (Φ, Ψ) dihedral angles that are not in any feasible areas of the Ramachandran plot. However, there is only one such case found in the idealized structure. Thus, two infeasible (Φ, Ψ) dihedral angles are fixed by the (Φ, Ψ) dihedral angle log-odd score. Here, we did not, but certainly can, implement a hard constraint to disallow any infeasible (Φ, Ψ) dihedral angles.

In summary, we have demonstrated that protein structure idealization can be used to improve poor (Φ, Ψ) dihedral angles of protein pseudo-structures. These protein pseudo-structures can either be predicted or be experimentally determined. More applications of the protein structure idealization problem will be studied.

3.4 Discussion

We have introduced the protein structure idealization problem and performed our first attempt to solve it. The experiment results show that idealized structures always exist with small changes on the coordinates. Furthermore, the idealized backbone structures have significantly better statistical energy and (Φ, Ψ) dihedral angle distributions. Therefore, protein structures can be modeled accurately with dihedral angles and ideal bond lengths and angles, and it is feasible to predict protein backbone and side-chain structures by searching the dihedral angle space.

Our protein structure idealization algorithm can also correct modelling errors of protein structures in PDB [5]. In fact, previous research indicates that many bond conformations and side-chain rotamers are likely incorrect in PDB, and it is useful to have an automated mechanism to fix these problems [33, 35]. Thus, we can address these problems by idealizing all protein structures in PDB with our protein structure idealization algorithm and using our specially tuned scoring functions.

The idealized version of the PDB [5] provides new protein structure references to study protein structures and functions. For example, we can rebuild fragment and rotamer libraries based on the idealized PDB. It would then be more intuitive to use the idealized fragment or rotamer libraries in the protein backbone or side-chain structure prediction algorithms searching the dihedral angle space. Thus, we expect to see some improvements of the accuracy of these algorithms with the idealized fragment and rotamer libraries. Therefore, we also provide a new approach for discovering unusual atoms and bonds by comparing the idealized and the original PDB structures. Although most of these unusual atoms and bonds are due to errors, we expect to discover some biochemical insights that assist in understanding protein functions.

Chapter 4

Fingerprinting Protein Structures Efficiently and Effectively

In this chapter, we present ContactLib, a complete contact group library defined in Section 4.1, which is to be used as fingerprints of protein structures. Specifically, we present two ContactLibs: ContactLib-9L, that models local contacts, and ContactLib-3R, that models remote contacts. According to the receiver-operating characteristic (ROC) curve analysis [24] in Section 4.2.1, the worst-case AUROC is significantly improved by using ContactLib, and ContactLib-3R is even able to guarantee an AUROC higher than a random method, which has an AUROC of 0.5. We also study the influence of secondary structure to the neighbor protein structure retrieval problem in Section 4.2.2 and the correlations among RMSD, the Euclidean distance between distance matrices and the Euclidean distance between distance vectors in Section 4.2.3.

4.1 ContactLib Neighbor Protein Structure Retrieval

In this section, we first define a *contact group*. Then, we build a comprehensive library of contact groups as fingerprints of all existing protein structures and we call such a contact group library *ContactLib*. We also propose an indexing technique for ContactLib, which

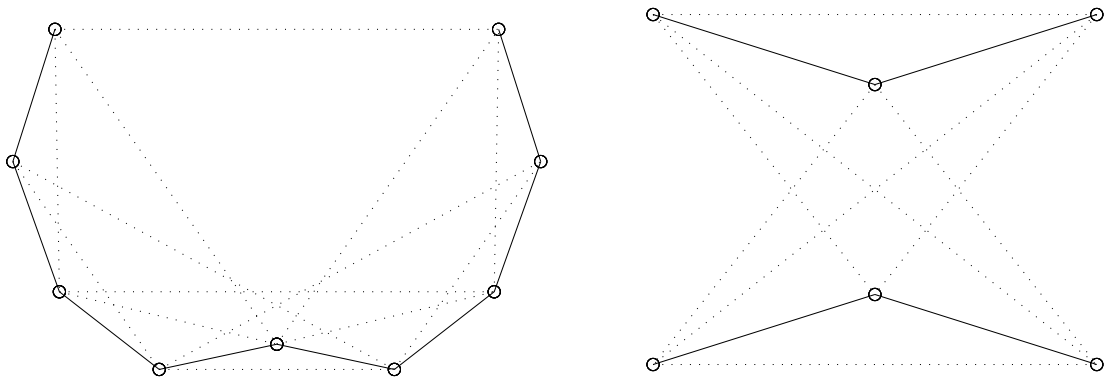
may be applied to neighbor contact group retrieval. Finally, we introduce a *combined hit-rate score* to retrieve neighbor protein structures.

A contact group refers to a small collection of residues that may have a high density of contacts among the residues. As two residues in contact should not be far apart, we require all residues are within a sphere. The position of each residue here is represented by its C_α atom. A local contact group models contacts within a protein structure fragment and a remote contact group could involve two or more structure fragments. Due to chemical and physical constraints within limited sphere space, it is rare for a contact group to contain a large number of fragments. For conciseness, we require a remote contact group to involve exactly two fragments. Hence, we define a *contact group* as a set of residues, represented by the respective C_α atoms, of either a single fragment with l_1 residues, called a *local contact group*, or a pair of fragments with l_2 residues, called a *remote contact group*, such that all the C_α atoms are located within a sphere of radius r.

Here, we set $l_1 = 9$ and $l_2 = 3$ as we find that they are sufficient to accurately model a local and a remote contact group, respectively. The fragment length of nine has also been used and shown to be the optimal fragment length to model protein structure fragments [69, 54]. Moreover, the radius of the sphere is set to be $r = 16$Å, so that it is large enough to capture most contacts. Then, we define a *ContactLib* as a contact group library containing local and/or remote contact groups in all protein structures of the search protein structure database.

We use the contact groups to fingerprint protein structures. To create an efficient and effective index of the ContactLib, we devise a strategy to represent a contact group by a low-dimensional vector. Before defining such a representation, we examine the number of dimensions or the degree of freedom of a contact group; that is, we want to know how many values are necessary to reconstruct a contact group.

We determine the dimension of a contact group as follows. As discussed in the previous chapter, a protein structure can be represented by ideal bond lengths, ideal bond angles and dihedral angles. The peptide dihedral angles (i.e. the Ω angles) are also rounded to either 0 or π. Since less than 2% of the Ω dihedral angles have a value closer to 0, it is treated as a rare case [22]. Hence, it is acceptable to use $\Omega = \pi$ as a good approximation,



(a) Local contact group (b) Remote contact group

Figure 4.1: Captured distances of local and remote contact groups: each circle represents a C_α atom, each solid line represents a pseudo bond between two adjacent C_α atoms (captured implicitly in our distance vector), and each dashed line represents a distance captured by our distance vector.

which results in the distance between two adjacent C_α atoms to be 3.8Å. If we connect any two adjacent C_α atoms by such a pseudo bond, the number of dihedral angles in this pseudo molecule of a local contact group is $l_1 - 3$, and the number of bond angles in it is $l_1 - 2$. Thereafter, the dimension required to represent a local contact group is $2l_1 - 5 = 13$. Similarly, the number of dihedral angles in the pseudo molecule of a remote contact group is $2l_2 - 3$, and the number of bond angles in it is $2l_2 - 2$, and the number of bond lengths between non-adjacent C_α atoms in it is 1. Thereafter, the dimension required to represent a remote contact group is $(2l_2 - 3) + (2l_2 - 2) + 1 = 4l_2 - 4 = 8$. Note that the number of dimensions is proportional to the number of residues in the contact group.

Given the desired number of dimensions, we create distance vectors to represent contact groups. Denote $D(a, b)$ as the distance between two points a and b. Given a local contact group of a single protein structure fragment $\{P_1, P_2, ..., P_{l_1}\}$, the distance vector is defined as

$$V_1 = \{D(P_i, P_{i+g}) \mid 1 \leq i,\ i + g \leq l_1,\ g = 2^k,\ k \geq 1\}.$$

For a remote contact group of a protein structure fragment pair $\{P_1^1, P_2^1, ..., P_{l_2}^1, P_1^2, P_2^2, ..., P_{l_2}^2\}$,

we define the distance vector as:

$$V_2 = \left\{ \begin{array}{l} D(P_i^1, P_{i+2}^1) \mid 1 \leq i,\ i+2 \leq l_2 \\ D(P_i^2, P_{i+2}^2) \mid 1 \leq i,\ i+2 \leq l_2 \\ D(P_i^1, P_{i+1}^2) \mid 1 \leq i < l_2 \\ D(P_i^2, P_{i+1}^1) \mid 1 \leq i < l_2 \\ D(P_{l_2}^1, P_1^2) \\ D(P_{l_2}^2, P_1^1) \end{array} \right\}.$$

Here, V_1 and V_2 have 13 dimensions and 8 dimensions, respectively. In addition, our definition of V_1 and V_2 covers different types of distances (as shown in Figures 4.1(a) and 4.1(b)). One critical feature of V_1 and V_2 is that if two contact groups have similar structures with low RMSD, they should have similar pairwise distances [32] and hence similar V_1 or V_2, as described in Section 4.2.3.

The number of similar contact groups shared by two proteins can be used as an indicator of their structure similarity. Here, we introduce an index to efficiently find all contact groups that are similar to a query contact group in ContactLib by using a 13-by-256 table of bit vectors for a local ContactLib and an 8-by-256 table of bit vectors for a remote ContactLib. Here, each row of the table represents a dimension of the distance vector. For each dimension of the distance vector, the value space is discretized into 256 bins, and each column represents a bin. Each element associated bin on the associated dimension for all contact groups of the ContactLib. Then, these tables can be effectively used to retrieve the set of contact groups in a particular bin along a given dimension. Contact groups in m consecutive bins along a particular dimension (or column) can be calculated by bitwise OR operations, and then contact groups in m consecutive bins along all dimensions (or rows) can be calculated by bitwise AND operations. Here, we carefully choose a parameter m, such that contact groups similar to the query contact group are within m bins from the query bins along each dimension.

To compare two structures, we introduce a *combined hit-rate score* to rank and select protein structures in the search database. We observed that, for a pair of similar protein structures, most of the contact groups for one structure tend to have similar contact groups from the other structure. Conversely, for a pair of dissimilar protein structures, the opposite

scenario was observed. These observations suggest a combined hit-rate score for a pair of protein structures, as the geometric mean of the similar contact group hit-rates of the two protein structures:

$$S = \sqrt{\frac{h_1}{n_1}\frac{h_2}{n_2}},$$

where h_1 is the number of hit contact groups for the first protein structure that have similar contact groups from the second protein structure, h_2 is the number of hit contact groups for the second protein structure that have similar contact groups from the first protein structure, n_1 is the number of contact groups for the first protein structure, and n_2 is the number of contact groups for the second protein structure.

In summary, we find all pairs of neighbor contact groups between the query protein structure and the search database using our indexes of ContactLib, and then we calculate the combined hit-rate score to rank and select protein structures in the search database. Let p be the number of contact groups in a query, q be the number of contact groups in the database, and N be the number of structures in the database. Recall that m is the number of consecutive bins that defines similarity on a dimension of the distance vector, and l is the dimension of the distance vector. For each query contact group, $O(m)$ bitwise OR operations and $O(l)$ bitwise AND operations are performed, and each bitwise OR or AND operation takes $O(q)$ time. Thus, the runtime complexity to find all similar contact group pairs between the query protein structure and the search database is $O(pq(lm+l))$, and the combined hit-rate scores can be calculated simultaneously. Moreover, the runtime complexity to rank structures according to the combined hit-rate scores is $O(N \log N)$. Therefore, the running time for our neighbor protein structure retrieval method is $O(pqm + N \log N)$. Here, the indexes can be pre-built, and the runtime complexity is not included.

4.2 Results

For performance analysis of our neighbor protein structure retrieval program, we used the high-quality protein structure subset of SCOP30 1.75B [59, 9] that has a minimum Summary PDB ASTRAL Check Index (SPACI) of 0.5. Here, we simply refer to this dataset as SCOP30. Then, we built the local contact group library, *ContactLib-9L*, and

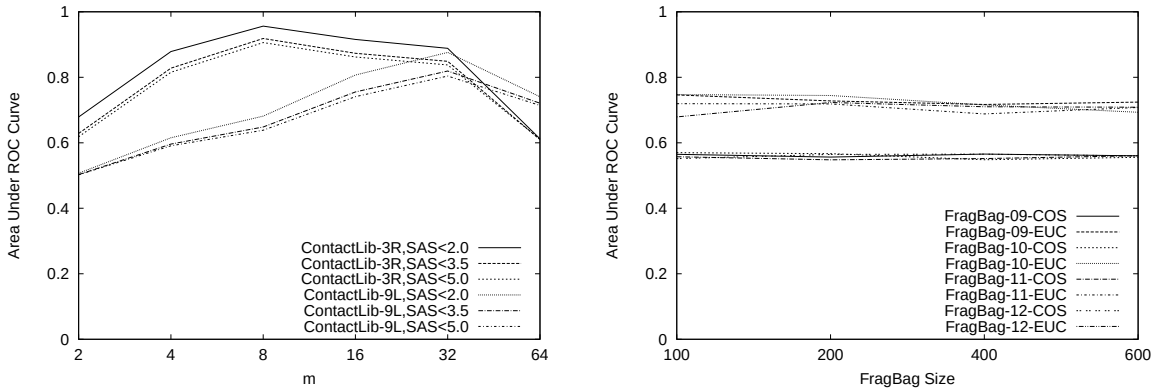
the remote contact group library, *ContactLib-3R*, of SCOP30. For each protein structure of SCOP30, we retrieved its neighbor protein structures from the rest of SCOP30. For reference, there are $3,297$ protein structures in SCOP30, $375,299$ local contact groups in ContactLib-9L, and $6,309,469$ remote contact groups in ContactLib-3R.

To find neighbor protein structures of each query protein structure, we used SCOP [59] and the best alignment found by six popular protein structure alignment tools: DALI [32], CE [68], LGA [83], SSM [41], TM-align [87] and SPalign [81]. Specifically, we considered two protein structures as neighbors if and only if both protein structures are from the same SCOP super-family and the best pairwise structure alignment has a structure alignment score (SAS) [40] below 2.0Å. Such neighbor protein structures tend to have globally similar structures and functional features, but do not necessarily have similar sequences. Since different SCOP levels and SAS thresholds produce similar conclusions, we focus on the above neighbor protein structure definition in this experiment. For the best alignments with SAS below 2.0Å, 50% are contributed by SPalign, 31% are contributed by LGA, and 16% are contributed by SSM.

The accuracy of neighbor protein structure retrieval is evaluated by the area under the receiver operating characteristic (ROC) curve (AUROC), which has been used in many research areas [24], including the protein structure alignment area [40, 7]. For instance, an AUROC of 0.9 means that a neighbor protein structure should be scored higher than a non-neighbor protein structure with a probability of 0.9, and a random method will have an AUROC equal to 0.5. When the query protein structure does not have any neighbor protein structures in SCOP30, the AUROC is not defined. Thus, such cases are eliminated in our analysis.

4.2.1 General ROC Curve Analysis

In this experiment, we demonstrate that ContactLib significantly outperforms FragBag for the neighbor protein structure retrieval problem in terms of AUROC. For ContactLib-9L and ContactLib-3R, we tested $m \in \{2, 4, 8, 16, 32, 64\}$ (recall that m is the number of neighboring bins we should use around the query bin along each dimension). For FragBag,



(a) ContactLib (b) FragBag

Figure 4.2: ROC curve analysis: (a) the highest average AUROC is 0.876 when the ContactLib-9L with $m = 32$ is used; the highest average AUROC is 0.956 when the ContactLib-3R with $m = 8$ is used; (b) the highest average AUROC is 0.747 when the FragBag with a Euclidean distance function, a fragment length of 10 and a bag size of 100 is used.

we tested the bag-of-words datasets of lengths between 9 and 12 and of all sizes from the FragBag website [7].

The AUROC of our combined hit-rate score, using ContactLib-9L and ContactLib-3R, are shown in Figure 4.2(a). We see that the best accuracy of ContactLib-9L is achieved when $m = 32$, where the average AUROC is 0.876. Moreover, the best accuracy of ContactLib-3R is achieved when $m = 8$, where the average AUROC is 0.956. Thus, the best result for ContactLib-3R is 9% more accurate on average than that for ContactLib-9L. This indicates that remote contacts carry critical information that is not carried by local contacts, and are capable of identifying neighbor protein structures more accurately.

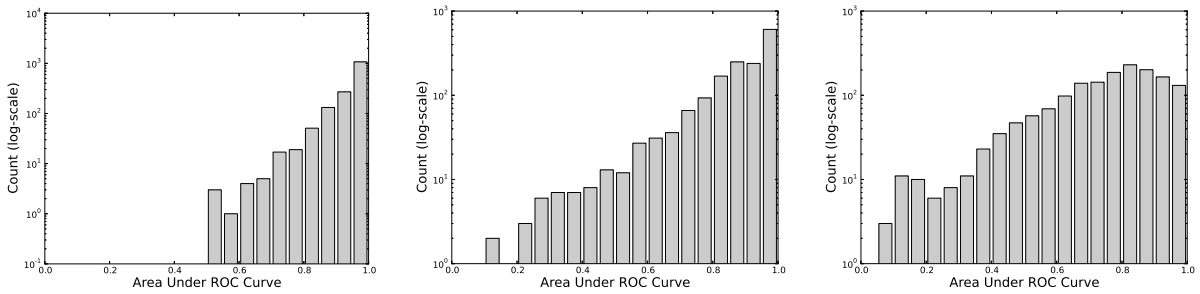
The AUROC of the neighbor protein structure retrieval defined on different SAS thresholds are also shown in Figure 4.2(a). Specifically, when the SAS threshold of 3.5Å is used, the best average AUROC's of ContactLib-3R and ContactLib-9L are 0.918 and 0.819, respectively; when the SAS threshold of 5.0Å is used, the best average AUROC's of ContactLib-3R and ContactLib-9L are 0.906 and 0.804. Moreover, the AUROCs of the neighbor protein structure retrieval defined on different SCOP levels are used in our exper-

iment but not shown here. Although the results are slightly different, our neighbor protein structure retrieval method, with either a local or a remote contact group library, is always capable of delivering high accuracies with high AUROC's.

We also combined ContactLib-9L and ContactLib-3R to retrieve neighbor protein structures from SCOP30 [59, 9]. This is done by linearly combining the score for ContactLib-9L with $m = 32$ and the score for ContactLib-3R with $m = 8$. When a weight of $1 : 16$ is used between ContactLib-9L and ContactLib-3R, the average AUROC is improved slightly to the highest value of 0.960. Thus, ContactLib-3R contributes more than ContactLib-9L to deliver more accurate results.

For comparison, we tested bag-of-words for FragBag [7] with different fragment lengths and bag sizes as shown in Figure 4.2(b). Different experiment settings, such as eliminating the query protein structures that do not have any neighbor protein structures in SCOP30, lead to a few new observations. First, the Euclidean distance function performs significantly more accurately than the cosine distance function. Moreover, the choice of FragBag, with different fragment lengths or different sizes, has no significant impact on the accuracy obtained. According to our results, the optimal FragBag is the one with a Euclidean distance function, a fragment length of 10 and a bag size of 100, that has an average AUROC of 0.747.

By comparing Figures 4.2(a) and 4.2(b), we find that our ContactLib outperforms FragBag [7] in terms of AUROC. This is further supported by looking at the AUROC distributions of ContactLib and FragBag in Figure 4.3. Specifically, when ContactLib-3R is used, 75% of the AUROC's are higher than 0.936, and the lowest AUROC is 0.504. When ContactLib-9L is used, 75% of the AUROC's are higher than 0.823, and 3% of the AUROC's are lower than 0.5. However, when FragBag is used, 75% of the AUROC's are higher than 0.657, and 10% of the AUROC's are lower than 0.5. Recall that a random method has an AUROC of 0.5. Although FragBag is capable of delivering good average accuracy, the worst case may not be acceptable for many accuracy sensitive applications. In our experiment, the worst-case AUROC is significantly improved by using ContactLib, and ContactLib-3R is even able to guarantee an AUROC, which is higher than a random method.



(a) ContactLib-3R (b) ContactLib-9L (c) FragBag

Figure 4.3: AUROC distributions (the AUROC of a random method equals to 0.5): (a) when ContactLib-3R is used, 75% of the AUROC's are higher than 0.936, and the lowest AUROC is 0.504; (b) when ContactLib-9L is used, 75% of the AUROC's are higher than 0.823, and 3% of the AUROC's are lower than 0.5; (c) when FragBag is used, 75% of the AUROC's are higher than 0.657, and 10% of the AUROC's are lower than 0.5.

In summary, the best accuracy is archived when ContactLib-3R with $m = 8$ is used. If only the top three ranked protein structures according to our combined hit-rate score are considered, there is a probability of 58% that we found at least one neighbor protein structure. The probability is increased to 73% when only the top 10 are considered. The excellent result suggests that ContactLib-3R can be used as a highly accurate and efficient filter to remove most unrelated protein structures while keeping many neighbor protein structures.

4.2.2 ROC Curve Analysis of all-α and all-β proteins

To understand the influence of secondary structure to the neighbor protein structure retrieval problem, we studied the AUROC of those all-α and all-β query protein structures in the previous section. From the 1574 query protein structures in the previous section, there are 157 all-α protein structures and 313 all-β protein structures.

The AUROC's of our neighbor protein structure retrieval with ContactLib-3R and ContactLib-9L for $m \in \{2, 4, 8, 16, 32, 64\}$ are shown in Figure 4.4. Comparing the AUROC's of all-α and those of all-β query protein structures, the AUROC's of all-α query

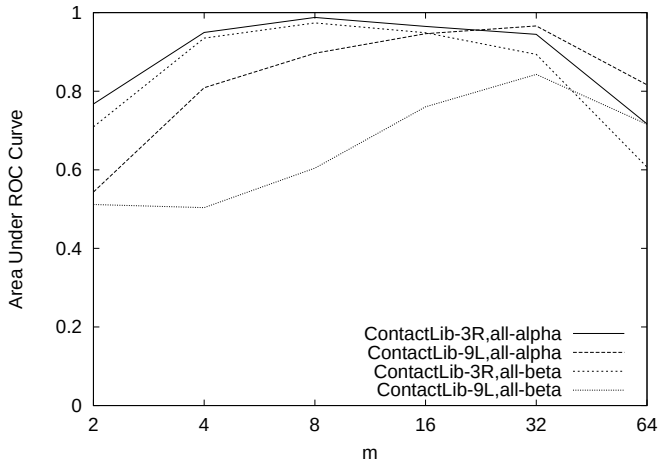


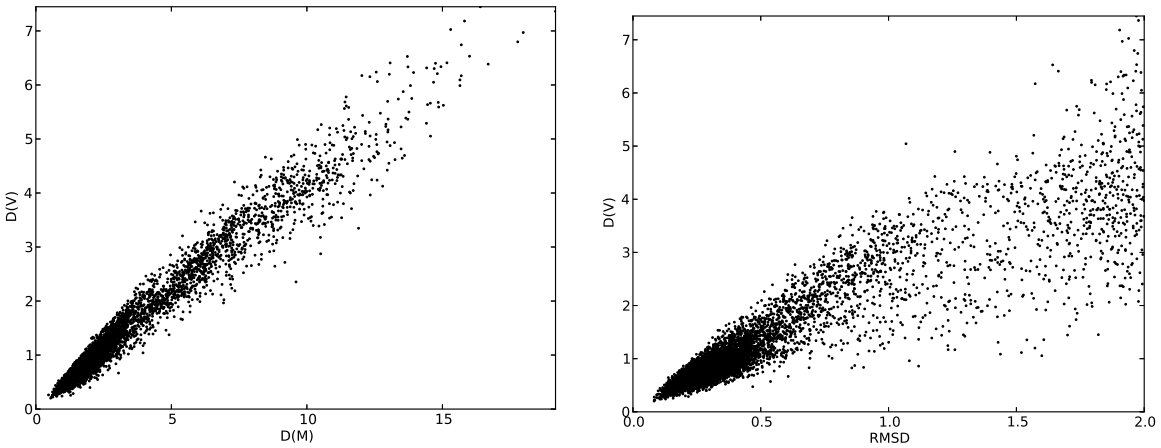
Figure 4.4: ROC curve analysis of all-α and all-β query protein structures: the AUROC's of all-α query protein structures tend to be higher than those of all-β query protein structures; the impact on the type of query protein structures is significantly smaller when ContactLib-3R is used than when ContactLib-9L is used.

protein structures tend to be higher. Comparing the AUROC's of ContactLib-9L and ContactLib-3R, the impact on the type of query protein structures is significantly smaller when ContactLib-3R is used. This is because our remote contact groups are also capable of modeling hydrogen bonds in α-helices. However, local contact groups are incapable of modeling hydrogen bonds in β-strands.

In summary, the neighbor protein structure retrieval problem for all-β query protein structures is more challenging than that for all-α query protein structures, and incorporating remote contact information is critical to produce accurate results consistently for all-α and all-β query protein structures.

4.2.3 Correlation Analysis of Distance Functions

In this experiment, we demonstrated that if two contact groups have similar distance vectors (defined in Section 4.1), they tend to have similar structures with low RMSDs. This was done by studying the correlations among RMSD, the Euclidean distance between



(a) $D(V)$ v.s. $D(M)$ (b) $D(V)$ v.s. RMSD

Figure 4.5: Correlation analysis among root mean square deviation (RMSD), $D(M)$ and $D(V)$ of local contact groups, where RMSD is less than 2.0Å, D is the Euclidean distance function, M is the distance matrix used by DALI and V is our distance vector: (a) the correlation coefficient is 0.98 between $D(V)$ and $D(M)$; (b) the correlation coefficient is 0.92 between $D(V)$ and RMSD.

distance matrices $D(M)$, and the Euclidean distance between distance vectors $D(V)$. The data shown in Figure 4.5 was collected from similar local contact groups, with RMSD less than 2.0Å, from 100 random pairs of proteins, such that each pair of proteins belonged to the same SCOP domain.

From Figure 4.5(a), we find a strong correlation between $D(V)$ and $D(M)$ for local contact groups with RMSD less than 2.0Å. This is also true for remote contact groups. Specifically, the correlation coefficients are 0.98 and 0.96, between $D(V)$ and $D(M)$ of local and remote contact groups, respectively. Therefore, our distance vector is as good as the distance matrix, which is used by the popular and successful pairwise protein structure alignment tool, DALI [32], to capture similar contact groups.

Although both of RMSD and pairwise distance matrix have been shown to be capable of capturing similarities between protein structures, they are not required to have strong correlations. This is also supported by our results. From Figure 4.5(b), we find that small

values of $D(V)$ suggest small values of RMSD among local contact groups. Specifically, the correlation coefficient is 0.92 between $D(V)$ and RMSD between local contact groups. However, neither $D(V)$ nor $D(M)$ has such strong correlations to RMSD between remote contact groups, and the correlation coefficients are approximately 0.6.

In summary, if two contact groups have similar distance vectors, they tend to have similar structures with low RMSDs. Specifically, for local and remote contact groups with RMSD less than 2.0Å, there is a strong correlation between the Euclidean distance between distance vectors and the Euclidean distance between distance matrices. Moreover, a small Euclidean distance between distance vectors suggests a small RMSD for local contact groups.

4.3 Discussion

In conclusion, we have shown that ContactLib is an effective and efficient neighbor protein structure retrieval method. Most importantly, ContactLib was able to maintain a consistent level of accuracy in our tests. The key to consistently retrieve accurate neighbor protein structures for all-β query protein structures is incorporating remote contact information in ContactLib. This is unmatched by existing neighbor protein structure retrieval method, FragBag [7].

Chapter 5

Superimposing Protein Structures Optimally

In Section 5.1, we introduce PROSTA-super, a simple and reliable method for optimally superimposing two protein structures, given the aligned residue pairs and the scoring function. Our superimposition method relies on and verifies the fact that the optimal protein structure alignment always contains a small subset of aligned residue pairs, called a seed, such that the rotation and translation (ROTRAN), which minimizes the RMSD of the seed, yields both the optimal ROTRAN and the optimal alignment score. This is demonstrated by finding both the optimal ROTRAN and the optimal alignment score of the pairwise protein structure alignments from the HOMSTRAD database [57] in Section 5.2.1 and the CASP10 dataset [43] in Section 5.2.2.

5.1 Method for Protein Structure Superimposition

Given the aligned residue pairs of a protein structure alignment, our protein structure superimposition method, called PROSTA-super, can be divided into two steps. First, the ROTRAN minimizing the RMSD is calculated for each seed. For each ROTRAN, the protein structures are superimposed and the alignment score is calculated. In the second

step, promising ROTRANs with high alignment scores are iteratively refined by calculating ROTRANs that minimizes the weighted RMSDs of different seeds. Each residue weight is set according to the TM-score [85] of the residue from the previous iteration, as proposed in SPalign [81]. Finally, the ROTRAN yielding the highest alignment score is kept as the optimal one.

TMscore [85] and LGA [83] samples a ROTRAN by calculating the RMSD of a *consecutive seed* containing contiguous aligned residue pairs. In our protein structure superimposition method, we take one step further and employ *gapped seeds*. Unlike a consecutive seed, a gapped seed does not require the aligned residue pairs to be contiguous. To simplify, we divide the gapped seeds into four categories. If there is a single gap in the middle of the gapped seed, it is called a *singly gapped seed*; if all the gaps of a gapped seed contain the same number of spaces, it is called an *constantly gapped seed*; if all the gaps except the one in the middle of a gapped seed contain the same number of spaces, it is called a *dual constantly gapped seed*; otherwise, the gapped seed is called an *arbitrarily gapped seed*.

In this study, we focus on seeds of size six that consist of small subsets of six aligned residue pairs. Formally, a size six seed can be defined as $S_1S_2S_3S_4S_5S_6$, where S_i is the index of an aligned residue pair. To simplify, let $D_i = S_{i+1} - S_i$ for $i \in \{1, 2, 3, 4, 5\}$. Hence, a consecutive seed has $D_1 = D_2 = D_3 = D_4 = D_5 = 1$; a singly gapped seed has $D_1 = D_2 = D_4 = D_5 = 1$ and $D_3 > 1$; an constantly gapped seed has $D_1 = D_2 = D_3 = D_4 = D_5 > 1$; a dual constantly gapped seed has $D_1 = D_2 = D_4 = D_5 > 1$, $D_3 > 1$ and $D_3 \neq D_1$; and an arbitrarily gapped seed does not have any of the above properties.

Currently, in our protein structure superimposition method, all consecutive seeds, singly gapped seeds, constantly gapped seeds, and dual constantly gapped seeds with all possible gap sizes are used in the first step. Only a limited number of randomly sampled arbitrarily gapped seeds are used in the second step. All above mentioned seeds can be used in either step of our superimposition method. However, we decided to use randomly sampled arbitrarily gapped seeds only for refinement because the large number of arbitrarily gapped seeds might be computationally expensive for performing an exhaustive approach. One can also trade accuracy for speed by using fewer types of seeds in the first step or by reducing the number of refinement iterations in the second step.

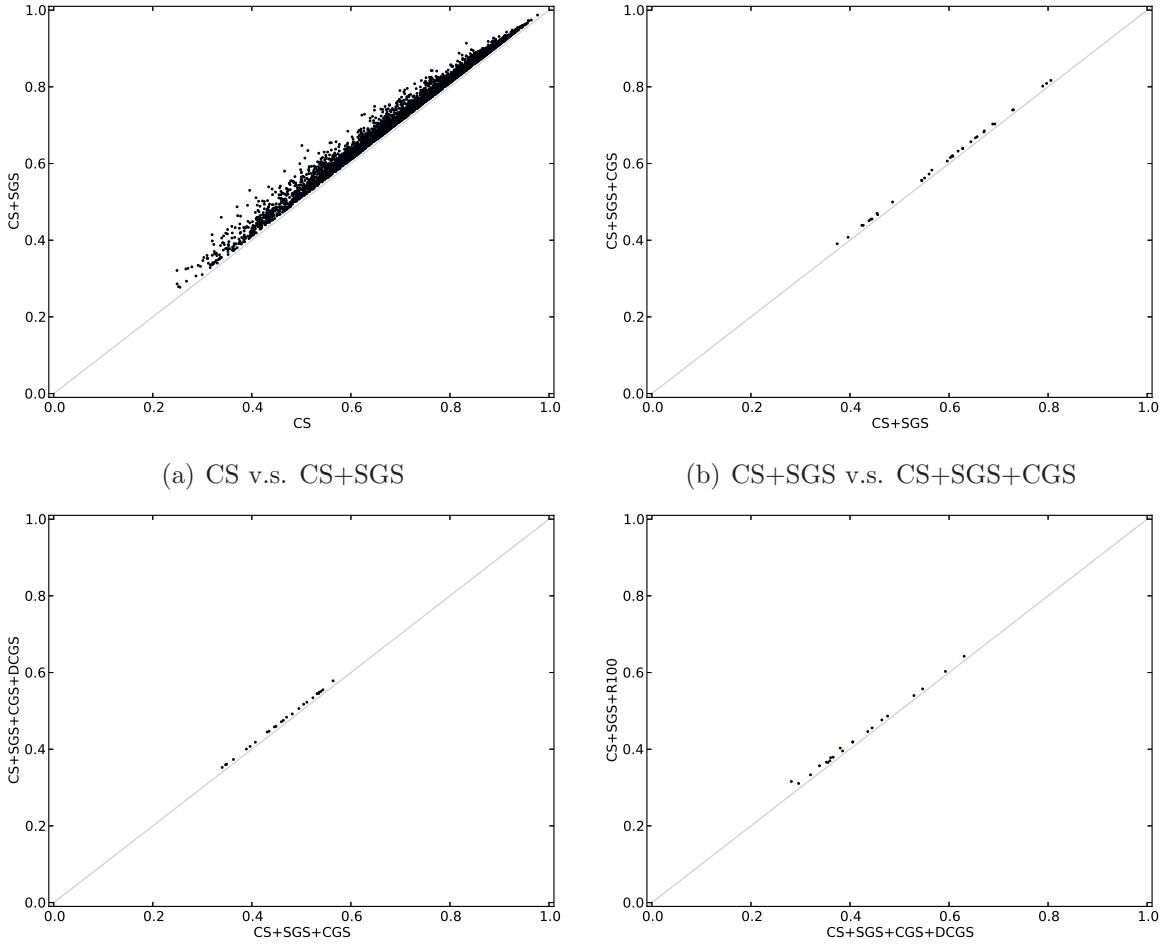
5.2 Results

Our simple protein structure superimposition method has been implemented in C++ with OpenMP, and we refer to it simply as PROSTA-super. The RMSD and the weighted RMSD are calculated using a quaternion-based characteristic polynomial [71]. Currently, our implementation uses size six seeds that consist of small subsets of six aligned residue pairs, and optimizes the TM-score [85], the GDT-TS score [83] and the GDT-HA score [83]. There is certainly the potential for other seed sizes and alignment scores in the future. In this study, we presume that two alignment scores are as good as each other, if the difference between the two alignment scores is less than 0.01. Our result verifies that there always exists a seed such that the ROTRAN minimizing the RMSD of the seed yields both the optimal ROTRAN and the optimal alignment score.

5.2.1 Superimposing Protein Structures Alignments for HOMSTRAD

To demonstrate the performance of our protein structure superimposition method, we calculate the optimal ROTRANs and the optimal alignment scores given the aligned residue pairs of the pairwise alignments abstracted from the HOMSTRAD database [57]. HOMSTRAD is a database of multi-structure alignments examined by human experts, and it contains $3,454$ homologous protein structures from $1,032$ protein families. After removing structures with alternative residues and duplicate indexes that might cause ambiguity, $8,444$ pairwise alignments abstracted from HOMSTRAD multi-structure alignments are included in this experiment.

Before studying the overall performance of our protein structure superimposition method, the contribution of each seed type is studied by calculating the highest TM-scores [85] before and after using each seed type. In Figure 5.1(a), the highest TM-scores calculated from consecutive seeds are compared to those calculated from consecutive seeds and singly gapped seeds. It can be seen that approximately half of the TM-scores are improved by at least 0.01 after using singly gapped seeds. This implies that singly gapped seeds contribute



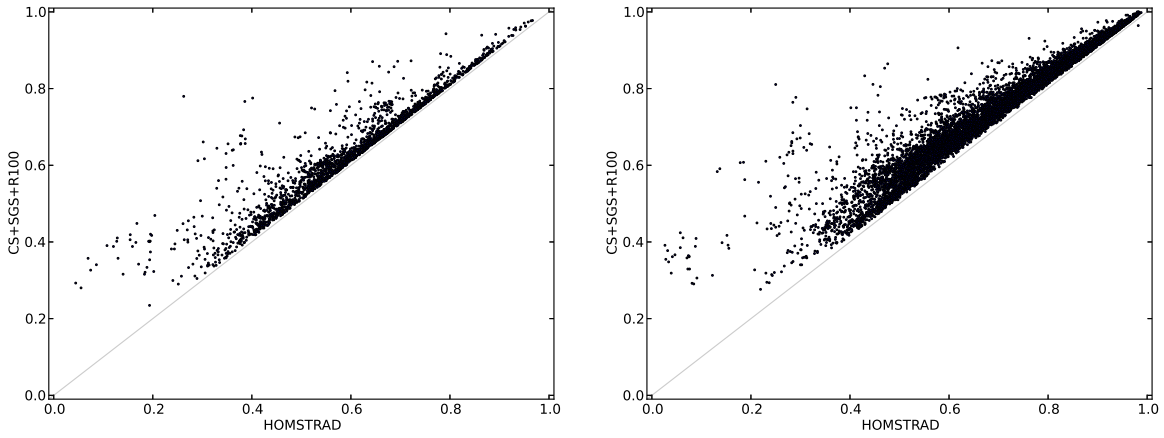
(a) CS v.s. CS+SGS

(b) CS+SGS v.s. CS+SGS+CGS

(c) CS+SGS+CGS v.s. CS+SGS+CGS+DCGS

(d) CS+SGS+CGS+DCGS v.s. CS+SGS+R100

Figure 5.1: Comparisons of the TM-scores calculated by PROSTA-super with consecutive seeds (CS), singly gapped seeds (SGS), constantly gapped seeds (CGS), dual constantly gapped seeds (DCGS) and 100 refinement iterations (R100) on the HOMSTRAD dataset (only cases yielding TM-scores higher than 0.2 and TM-score differences higher than 0.01 are shown): (a) after adding SGS, there are 275 cases with TM-score improvements > 0.05, and $4,017$ cases with TM-score improvements > 0.01; (b) after adding CGS, there are 35 cases with TM-score improvements > 0.01; (c) after adding DCGS, there are 25 cases with TM-score improvements > 0.01; and (d) after adding R100, there are 18 cases with TM-score improvements > 0.01.



(a) TM-score

(b) GDT-TS score

Figure 5.2: Comparisons of the alignment scores calculated directly from the HOMSTRAD alignments and those calculated by PROSTA-super with consecutive seeds (CS), singly gapped seeds (SGS) and 100 refinement iterations (R100) on the HOMSTRAD dataset (only cases yielding alignment scores higher than 0.2 and alignment score differences higher than 0.01 are shown): (a) there are 168 cases with TM-score improvements > 0.10, and $2,019$ cases with TM-score improvements > 0.01; (b) there are 725 cases with GDT-TS score improvements > 0.10, and $8,008$ cases with GDT-TS score improvements > 0.01.

critically to finding the optimal ROTRAN and the optimal alignment score, because the global topology of the protein structure is implicitly modeled by singly gapped seeds.

Adding constantly gapped seeds, dual constantly gapped seeds, and arbitrarily gapped seeds to our protein structure superimposition method only improves the TM-score [85] slightly as shown in Figures 5.1(b), 5.1(c) and 5.1(d). Similar results are observed if the highest GDT-TS scores [83] are calculated instead. One possible reason for the lack of significant improvement is that the consecutive seeds and the singly gapped seeds are sufficient for optimizing TM-scores for the high quality alignments for HOMSTRAD [57]. More significant TM-score improvements are expected for lower quality protein structures and lower quality protein structure alignments (as shown in Section 5.2.2).

The TM-scores [85] and the GDT-TS scores [83] calculated by our protein structure superimposition method are also compared to those calculated directly from the HOM-

STRAD alignments [57] in Figures 5.3(d) and 5.4(d), respectively. Interestingly, there are 168 cases with TM-scores improved by at least 0.10 and 725 cases with GDT-TS scores improved by at least 0.10. First, this suggests that the HOMSTRAD alignments can still be improved according to the TM-score and the GDT-TS score. More importantly, if the HOMSTRAD alignments examined by human experts are presumed to be optimal, this also demonstrates that there always exists a seed such that the ROTRAN, which minimizes the RMSD of the seed, yields both the optimal ROTRAN and the optimal alignment score.

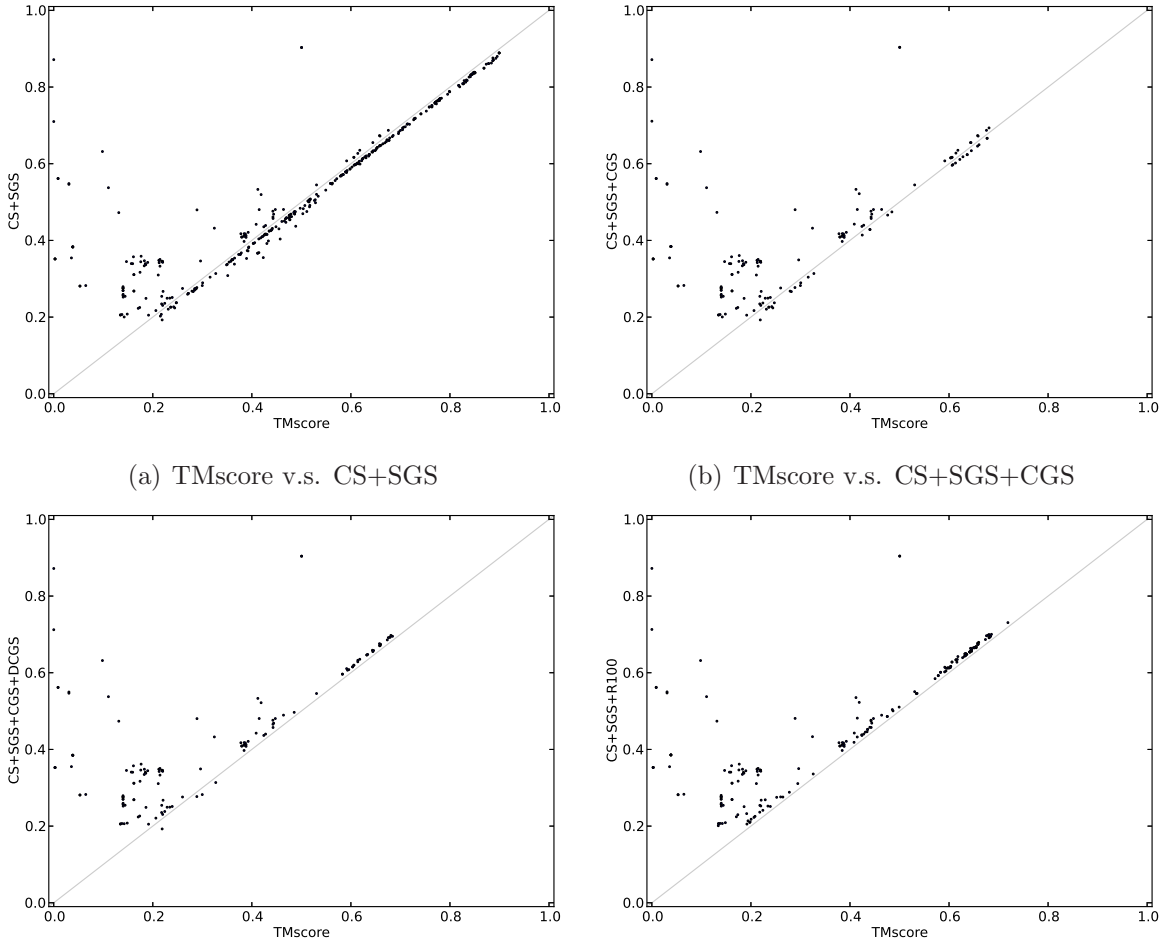
In summary, our results show that the optimal protein structure alignment always contains at least one seed such that the ROTRAN minimizing the RMSD of the seed yields both the optimal ROTRAN and the optimal alignment score. Moreover, remote information provides a critical contribution to finding the optimal ROTRAN and the optimal alignment score, because of its capability for modeling global protein structure topology.

5.2.2 Superimposing Protein Structures Alignments for CASP10

During the tenth community wide experiment on the Critical Assessment of Techniques for Protein Structure Prediction (CASP10) [43], a variety of types of alignment scores were calculated to evaluate the performance of each protein structure prediction group. The success of the performance evaluation process depends mainly on finding the optimal ROTRAN and the optimal alignment score. To demonstrate the performance of our protein structure superimposition method in case of low quality protein structures, the alignment scores between $26,156$ pairs of the predicted and native structures for CASP10 are calculated and compared, using our superimposition method, TMscore [85] and LGA [83].

First, the contribution of each type of seed in the case of low quality protein structures is studied by calculating the highest alignment scores before and after using each type of seed on the CASP10 dataset [43]. Again, singly gapped seeds make a critical contribution to finding the optimal ROTRAN and the optimal alignment score. Unlike in previous observations, constantly gapped seeds, dual constantly gapped seeds, and arbitrarily gapped seeds are more likely to improve the alignment scores, as shown in Figures 5.3 and 5.4.

The significance of the refinement step is also demonstrated by comparing Figures 5.4(c) and 5.4(d). Specifically, there are four cases in which the GDT-TS scores [83] are improved



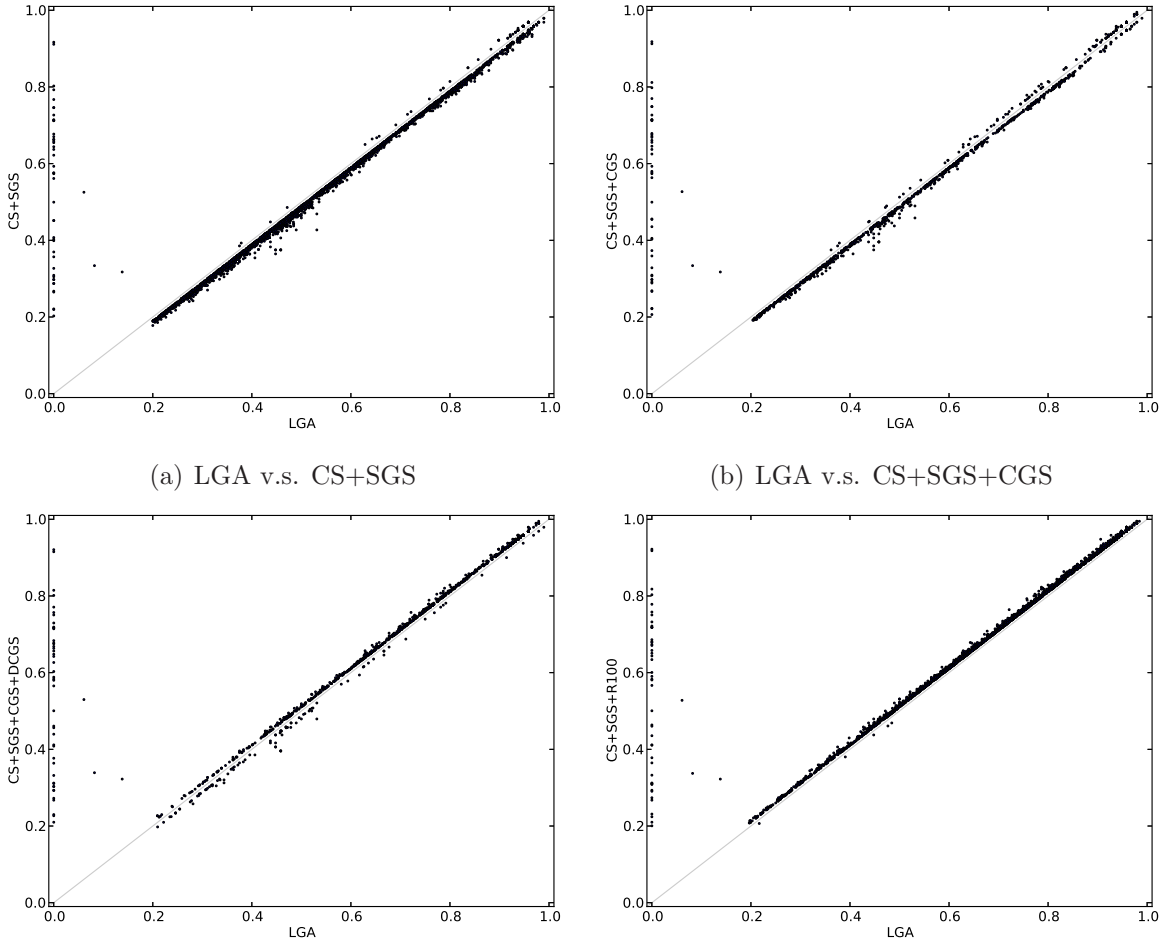
(a) TMscore v.s. CS+SGS

(b) TMscore v.s. CS+SGS+CGS

(c) TMscore v.s. CS+SGS+CGS+DCGS

(d) TMscore v.s. CS+SGS+R100

Figure 5.3: Comparisons of the TM-scores calculated by TMscore and PROSTA-super with consecutive seeds (CS), singly gapped seeds (SGS), constantly gapped seeds (CGS), dual constantly gapped seeds (DCGS) and 100 refinement iterations (R100) on the CASP10 dataset (only cases yielding alignment scores higher than 0.2 and alignment score differences higher than 0.01 are shown): (a,b,c,d) using CGS, DCGS and R100 improves the TM-scores calculated by PROSTA-super; and either with or without CGS, DCGS and R100, there are 66 cases with TM-score improvements > 0.10 comparing to the TM-scores calculated by TMscore.



(a) LGA v.s. CS+SGS

(b) LGA v.s. CS+SGS+CGS

(c) LGA v.s. CS+SGS+CGS+DCGS

(d) LGA v.s. CS+SGS+R100

Figure 5.4: Comparisons of the GDT-TS scores calculated by LGA and PROSTA-super with consecutive seeds (CS), singly gapped seeds (SGS), constantly gapped seeds (CGS), dual constantly gapped seeds (DCGS) and 100 refinement iterations (R100) on the CASP10 dataset (only cases yielding alignment scores higher than 0.2 and alignment score differences higher than 0.01 are shown): (a,b,c,d) using CGS, DCGS and R100 improves the GDT-TS scores calculated by PROSTA-super; and either with or without CGS, DCGS and R100, there are 54 cases with GDT-TS score improvements > 0.10 comparing to the GDT-TS scores calculated by LGA.

by at least 0.10, and there are 156 cases in which the GDT-TS scores are improved by at least 0.03, after 100 refinement iterations. However, it seems that the refinement step has only minor impact when calculating the highest TM-score [85], as shown in Figures 5.3(c) and 5.3(d). This demonstrates that the GDT-TS score is more difficult to optimize because the GDT-TS score is a discrete scoring function, while the TM-score is not.

After 100 refinement iterations, the alignment scores calculated by our protein structure superimposition method are similar to or (possibly significantly) higher than those found by TMscore [85] or LGA [83]. As shown in Figures 5.3(d) and 5.4(d), we found a case in which the TM-score [85] calculated by our superimposition method is 0.5619, while the TM-score calculated by TMscore is 0.0087. Moreover, we also found a case in which the GDT-TS score [83] calculated by our superimposition method is 0.5264, while the GDT-TS score calculated by LGA is 0.0609. The cases when TMscore or LGA does not return alignment scores because of PDB errors are also shown in the figure.

In summary, our protein structure superimposition method is capable of finding TM-scores [85] and GDT-TS scores [83] that are similar to or (possibly significantly) higher than those found by TMscore [85] and LGA [83]. This not only demonstrates that our superimposition method is reliable, but also demonstrates that it is robust to low quality protein structures, such as the predicted protein structures of CASP10 [43]. Thus, our superimposition method provides a reliable performance evaluation platform for CASP.

5.3 Discussion

We have introduced PROSTA-super, a simple and reliable method to optimally superimpose two protein structures, given the aligned residue pairs and the scoring function. Our superimposition method employs not only the consecutive seed containing consecutive residue pairs, but also gapped seeds containing gapped reside pairs. Thus, our superimposition method relies on and verifies the fact that the optimal protein structure alignment always contains a seed, such that the ROTRAN, which minimizes the RMSD of the seed, yields both the optimal ROTRAN and the optimal alignment score. Moreover, a weighted RMSD is used with gapped seeds as refinement in our superimposition method.

PROSTA-super has been implemented in C++ with OpenMP. Currently, our implementation uses size six seeds that consist of small subsets of six aligned residue pairs, and optimizes the TM-score [85], the GDT-TS score [83], and the GDT-HA score [83]. Our result shows that our superimposition method is capable of finding TM-scores [85] and GDT-TS scores [83] that are similar or (possibly significantly) higher than those found by TMscore [85] and LGA [83], respectively. Moreover, our method is significantly faster than LGA, and it has the added benefit of concurrent programming.

Chapter 6

Aligning Protein Structures Optimally

We introduce a simple PROtein STructure Alignment method, called PROSTA (meaning "simple" in Polish), for finding the optimal alignment that maximizes a given scoring function in Section 6.1. Our alignment method relies on and verifies the fact that the optimal protein structure alignment always contains a small subset of aligned residue pairs such that the rotation and translation (ROTRAN), which minimizes the RMSD of the subset of aligned residue pairs, yields both the optimal ROTRAN and the optimal alignment score. By comparing the TM-scores [85] calculated by our alignment method and state-of-the-art alignment methods in Section 6.2.1, our alignment method is shown to be more reliable in finding the optimal protein structure alignment. This is also supported by the experiment optimizing GDT-TS scores [83] instead of TM-scores in Section 6.2.2.

6.1 Method for Protein Structure Alignment

Given a protein structure alignment scoring function, finding the optimal alignment involves finding the optimal ROTRAN that maximizes the alignment score. Assume that there exists a near optimal ROTRAN that minimizes the RMSD of two small sets of C_α

atoms. We find the near optimal protein structure alignment by sampling ROTRANs in four steps: (1) ROTRANs are initially sampled from local fragment alignments and from remote fragment pair alignments; (2) noise ROTRANs are filtered out by clustering; (3) one representative alignment for each ROTRAN cluster is selected based on alignment scores; (4) the selected alignments are refined by random ROTRAN sampling.

First, an initial set of ROTRANs must be sampled. Here, the primary concern is to have several good candidates, instead of having a high signal-to-noise ratio, which is addressed in the next step. Finding good candidates is done by calculating the optimal ROTRAN that minimizes the RMSD between one or two fragments from each protein structure. When there is a single fragment from each protein structure, we call it local fragment. When there are two fragments from each protein structure, we call them a remote fragment pair. Here, we require the pair of remote fragments to be the same size and to be at least three residues away from each other to avoid modeling information redundant to the local fragments. In practice, a significantly large number of ROTRANs with the lowest RMSDs are kept for the next step, and the actual number of ROTRANs is selected empirically.

Since the initial set of ROTRANs may contain a great deal of noise, we try to filter out most of the noise with a star-like k-median clustering algorithm in the second step. Assuming that we know the maximum distance ϵ between the median of a cluster and any member of the same cluster, an approximate clustering is applied using a neighbor graph: each vertex represents a rotation matrix, and two vertices are connected if and only if the distance between them is at most ϵ. For each iteration, the vertex with the highest degree and its neighbors are grouped into a cluster, and are removed from the neighbor graph. The iteration repeats until either there are no vertices of degree higher than one or until the maximum number of clusters is reached. The unclustered ROTRANs are treated as noise. Similar approximate clustering algorithms have been used [86] and studied [3].

To complete the clustering algorithm, we need a distance function between ROTRANs. The Riemannian distance is a widely used distance metric measuring the length of the shortest geodesic curve between two rotation matrices [58]. Since the translation vector can be calculated by the rotation matrix and the weight centers of the aligned residues, we use Riemannian distances between rotation matrices to avoid using redundant information when clustering ROTRANs.

For each cluster, we find the representative alignment defined by the ROTRAN that yields the highest alignment score within the cluster. Since dynamic programming is computationally expensive, the number of clusters in the previous step must be carefully determined to avoid wasting computation on clusters of noise. After all alignment scores have been calculated, the top scored alignments are selected for the refinement step.

Finally, we refine the selected representative alignments by random ROTRAN sampling. Specifically, for each alignment to be refined, six aligned residue pairs are randomly selected from the alignment, the ROTRAN that minimizes RMSD of the aligned residue pairs is calculated, the alignment score of the alignment defined by the sampled ROTRAN is also calculated, and the previous steps are repeated until there is little improvement after $l_1 l_2$ iterations, where l_1 and l_2 are the number of residues of the two aligned protein structures.

6.2 Results

To demonstrate the reliability of our protein structure alignment method, PROSTA, we repeated the protein structure alignment experiment for TM-align [87] for the 200 non-homologous protein structures, which have sizes between 46 and $1,058$, and have a sequence identity cutoff of 30%. For each protein structure pair, the optimal alignment was calculated using our alignment method and state-of-the-art alignment methods, and then the alignment scores were compared. Here, we presume that two alignment scores are as good as each other, if the difference between the two alignment scores is less than 0.01. The result demonstrates that our alignment method is more reliable in finding the optimal protein structure alignment, and thus it is capable of discovering highly scored alignments missed by current alignment methods.

For the experiment settings, we decided to use local fragments of size 12, and remote fragment pairs of size 3, empirically. These experiment settings are called PROSTA-L12R3. For consistency, we selected $1,536$ local fragments of size 12 and $1,536$ remote fragment pairs of size three in the sampling step, used $\epsilon = 10^\circ$ in the clustering step, stopped clustering when 384 clusters were found, and selected 32 clusters in the refinement step in all experiments for this section.

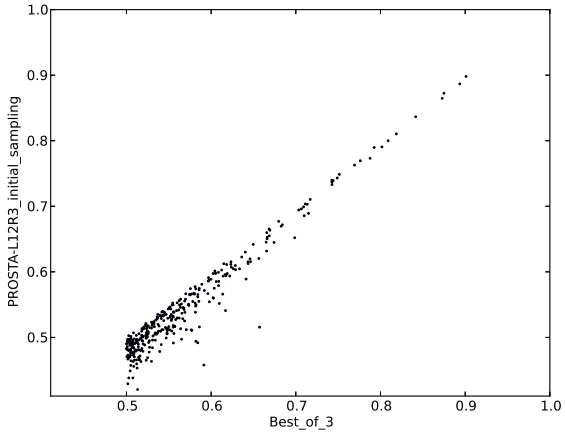
6.2.1 Alignment Quality Evaluation Using The TM-score

In this experiment, the optimal protein structure alignments, which maximize the TM-scores [85], are calculated using our alignment method, TM-align [87] and fr-TM-align [63]. Here, the TM-scores normalized by the smaller protein size are used. Since fr-TM-align does not support normalization by the smaller protein size, the TM-score normalized by the smaller protein size is recalculated based on the ROTRAN returned by fr-TM-align. To simplify, the highest TM-score found by the three methods is designated as the best-of-3 TM-score. Since biologists tend to be more interested in similar protein structures within the same protein fold, and the TM-score of 0.5 is a good approximate threshold for protein fold detection [80], only the 351 protein structure alignments with the best-of-3 TM-scores higher than 0.5 are included in this analysis.

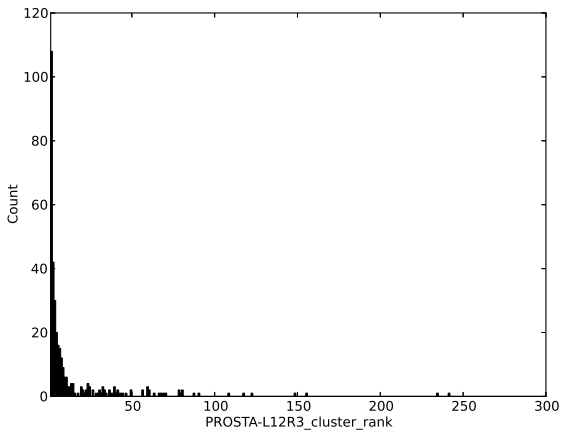
Before evaluating the performance of our protein structure alignment method as a black box, we would like to evaluate the performance for each step of our alignment method. The result does not only support the effectiveness of our alignment method, but also supports the appropriateness of our parameter settings, such as the local fragment size and the remote fragment pair size. In fact, the local fragment size of 12 has also been used by fr-TM-align [63].

In the first step of our alignment method, ROTRANs are sampled from both local fragment alignments and remote fragment pair alignments. As shown in Figure 6.1(a), 92% of the highest TM-scores [85] of the sampled ROTRANs are, at most, 0.05 lower than the best-of-3 TM-scores, and less than 1% of the highest TM-scores of the sampled ROTRANs are, at least, 0.1 lower than the best-of-3 TM-scores. This, combined with later results from this experiment, verifies that the initially sampled ROTRANs contain at least one ROTRAN that is suitable to be a starting point for finding the optimal ROTRAN.

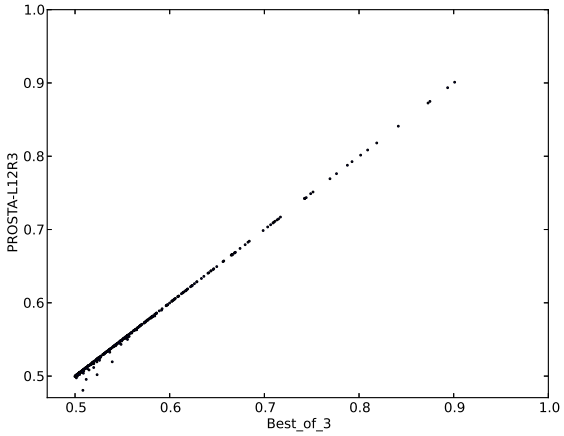
In the second step, the initially sampled ROTRANs are clustered. Figure 6.1(b) shows the rank of the first cluster that contains an optimal ROTRAN. Specifically, there exists at least one optimal ROTRAN in the largest cluster with a probability of 31%, and there exists at least one optimal ROTRAN in the largest 10 clusters with a probability of 75%. This demonstrates that an optimal ROTRAN tends to have many similar ROTRANs that minimize the RMSDs of local fragment alignments or remote fragment pair alignments,



(a) The best-of-3 TM-scores v.s. the PROSTA-L12R3 TM-scores (immediately after initial sampling)



(b) The rank distribution of the first cluster containing an optimal ROTRAN



(c) The best-of-3 TM-scores v.s. the PROSTA-L12R3 TM-scores

Figure 6.1: Performance for each step of PROSTA-L12R3: (a) after the initial ROTRAN sampling step, 92% of the highest TM-scores of the initially sampled ROTRANs are, at most, 0.05 lower than the best-of-3 TM-scores; (b) after the ROTRAN clustering step, there exists, at least, one optimal ROTRAN in the largest 10 cluster with a probability of 75%; after the cluster representative selection step (not shown in the figure), there exists at least one optimal ROTRAN in the clusters represented by the highest scored 10 alignments with a probability of 99%; (c) after the refinement step, all TM-scores are, at most, 0.03 lower than the optimal TM-score.

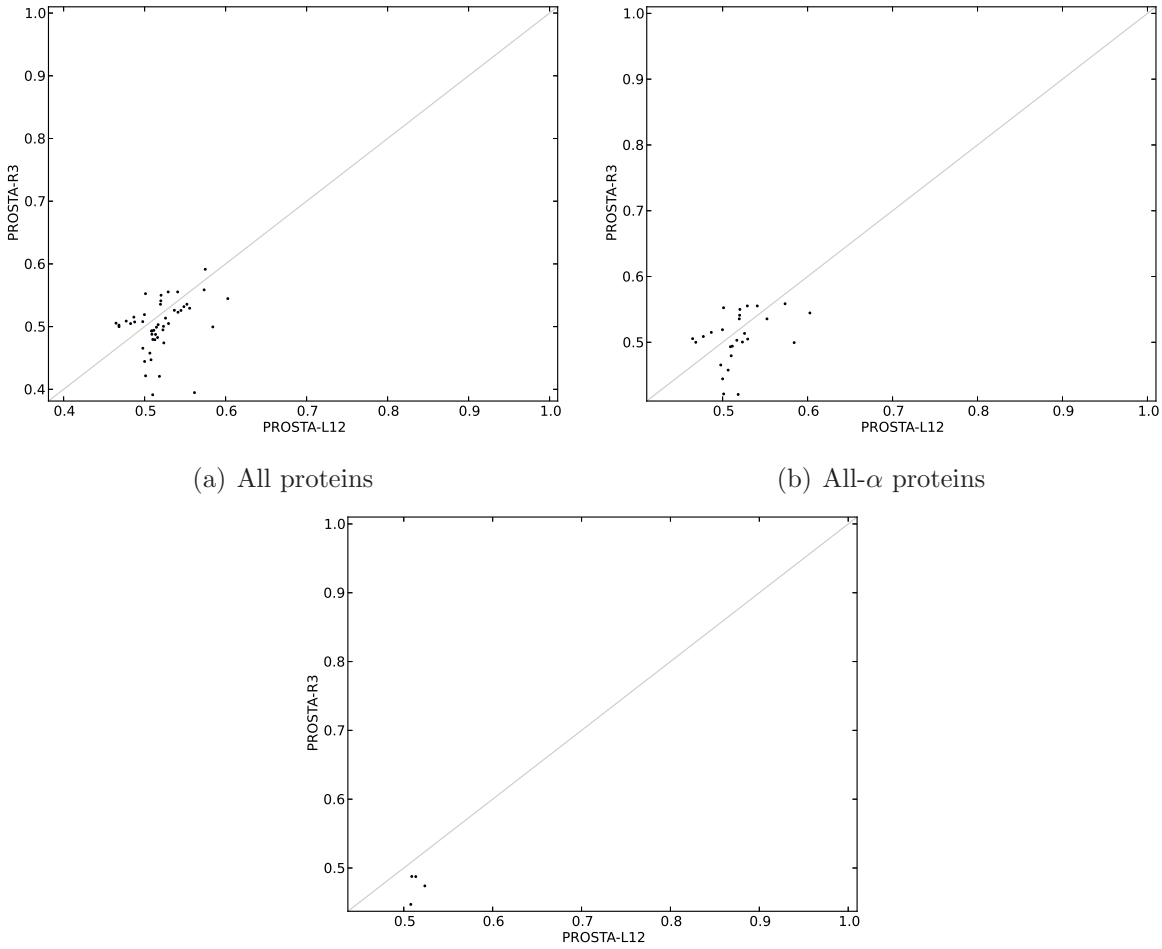
and that these ROTRANs tend to form a large cluster.

The representative alignment with the highest TM-score [85] is selected from each cluster in the third step. In our experiment, there exists at least one optimal ROTRAN in the cluster represented by the highest scored alignment with a probability of 85%, and there exists at least one optimal ROTRAN in the clusters, represented by the 10 highest scored alignments, with a probability of 99%. This demonstrates that the cluster representatives are capable of identifying an optimal cluster, which contains an optimal ROTRAN, efficiently and accurately.

Last, the selected alignments are refined by random ROTRAN sampling. Figure 6.1(c) shows the refined PROSTA-L12R3 TM-scores [85] and the best-of-3 TM-scores. It can be seen that the TM-scores are mostly similar, and the difference is always less than 0.03. This demonstrates that once an optimal cluster containing an optimal ROTRAN is identified, an optimal ROTRAN can be identified by random ROTRAN sampling from the cluster representative alignment.

To study the contributions made by using local fragments and using remote fragment pairs, we simplified our method to two variants: PROSTA-L12, that used only local fragments of size 12, and PROSTA-R3, that used only remote fragment pairs of size 3. For both PROSTA-L12 and PROSTA-R3, only 16 cluster representatives were used in the refinement step, and all other experiment settings remained the same as PROSTA-L12R3. Using PROSTA-R3, PROSTA-L12 and TM-align [87], we performed structure alignment between each pair of the 200 proteins in our dataset, and evaluated the alignment quality with TM-score [85]. Although either local fragments or remote fragment pairs are capable of finding high quality alignments that are comparable to or even better than those found by TM-align, neither local fragments nor remote fragment pairs are sufficient for finding optimal protein structure alignments.

Neither PROSTA-L12 nor PROSTA-R3 is sufficient for finding an optimal protein structure alignment. As shown in Figure 6.2(a), both PROSTA-L12 and PROSTA-R3 find similar TM-scores [85] when one of the TM-scores found by PROSTA-L12 and PROSTA-R3 is higher than 0.65. Among the 48 cases in which TM-scores differ by 0.01 or more, 32 TM-scores found by PROSTA-L12 are up to 0.17 higher, and 16 TM-scores found by

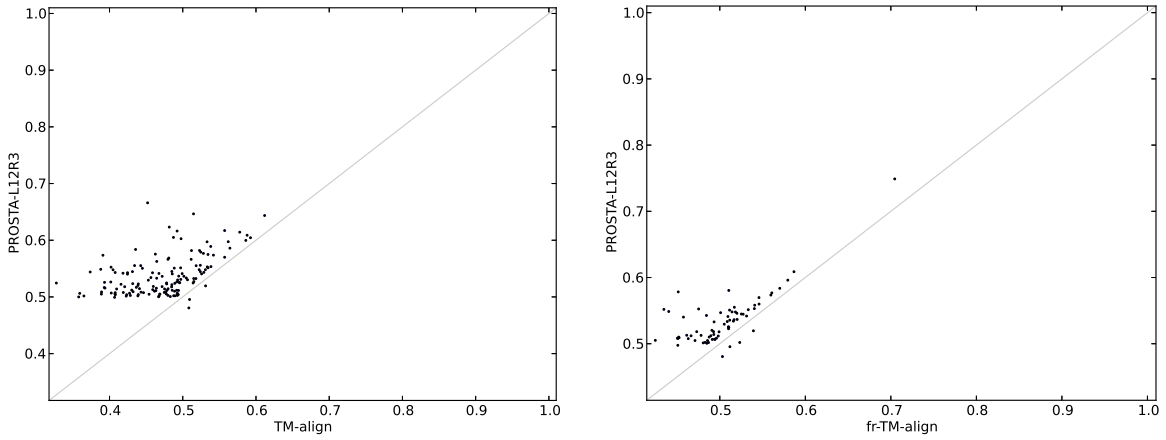


(a) All proteins

(b) All-α proteins

(c) All-β proteins

Figure 6.2: Comparisons of the TM-scores of PROSTA-L12 and PROSTA-R3 (only cases yielding TM-score differences higher than 0.01 are shown): (a) neither PROSTA-L12 nor PROSTA-R3 is sufficient for finding an optimal protein structure alignment; (b-c) due to its ability for modeling relationships of remote residues, PROSTA-R3 is capable of discovering alignments with higher TM-scores that cannot be discovered by PROSTA-L12, which is especially helpful for protein structure alignments involving multiple α-helices.



(a) TM-align v.s. PROSTA-L12R3

(b) fr-TM-align v.s. PROSTA-L12R3

Figure 6.3: Comparisons of the TM-scores of TM-align, fr-TM-align and PROSTA-L12R3 (only cases yielding TM-score differences higher than 0.01 are shown): (a) for the 285 cases in which the TM-score of TM-align is lower than 0.6 and the best-of-3 TM-score is higher than 0.5, TM-align fails to find TM-scores higher than 0.5 with a probability of 42%, while PROSTA-L12R3 fails the same task with a probability of only 2%; (b) compared to fr-TM-align, PROSTA-L12R3 finds 28 more cases in which TM-scores higher than 0.5.

PROSTA-R3 are up to 0.05 higher. Thus, both PROSTA-L12 and PROSTA-R3 contribute significantly for finding optimal protein structure alignments.

One advantage of PROSTA-R3 over PROSTA-L12 is the ability to model relationships of remote residues, and this is especially helpful for protein structure alignments involving multiple α-helices. Figures 6.2(b) and 6.2(c) shows the TM-scores [85] of alignments involving all-α proteins or all-β proteins (from the 351 test cases) respectively. For the cases that PROSTA-R3 works better than PROSTA-L12, the optimal alignment tends to contain multiple α-helices. This observation is not limited to all-α proteins, but also proteins containing α-helices and β-strands. This implies that the global topology is critical to the success of the protein structure alignment, and the global topology can only be modeled by remote information as we did for PROSTA-R3. Especially for structure alignments consist of mainly α-helices, the global topology becomes more important because local structures tend to be similar between α-helices.

Finally, we would like to demonstrate that PROSTA-L12R3 performs more reliably than current methods TM-align [87] and fr-TM-align [63] in finding an optimal protein structure alignment. Again, we aligned each pair of the 200 proteins in our dataset using the three methods. Then, we evaluated the alignment quality of each method with TM-score [85].

The improvements on TM-scores [85], found by PROSTA-L12R3 over those found by TM-align [87], are shown in Figure 6.3(a). We see that TM-scores found by PROSTA-L12R3 are mostly higher than those found by TM-align for the cases in which have TM-align TM-scores lower than 0.65. Specifically, PROSTA-L12R3 improves TM-scores by 0.03 on average and by 0.21 in the best case. Moreover, 10% of the TM-scores are improved by at least 0.1, 25% of the TM-scores are improved by at least 0.05, and no TM-scores are reduced by 0.03 or more.

If the highest TM-score [85] found by TM-align [87] is lower than 0.6 and the best-of-3 TM-score is higher than 0.5, PROSTA-L12R3 tends to discover better protein structure alignments with (possibly significantly) higher TM-scores. In our experiment, there are 285 such cases that form 81% of the 351 cases included in this analysis. In such cases, TM-align fails to find TM-scores higher than 0.5 with a probability of 42%, while PROSTA-L12R3 fails the same task with a probability of only 2%. This could significantly improve fold detection results.

Several interesting case studies are provided in Figures 6.4 and 6.5. Consistent with previous observations, PROSTA-L12R3 significantly improves TM-scores [85], compared to TM-align [87]. Consequently, new similar protein structures (or alignments) with TM-scores higher than 0.5 are discovered.

As shown in Figures 6.4(c) and 6.5(c), the superimposed protein structures of PROSTA-L12R3 and TM-align [87] share commonly aligned α-helices, but the global alignments are significantly different. Although the approximate subregion of the optimal alignment is found, the optimal alignment is still missed. This demonstrates one limitation of using local fragments to find the optimal alignment: an α-helix can be easily aligned (or overfitted) to another α-helix, and the overfitting causes the missing of the optimal alignment. In such cases, remote fragment pairs, which implicitly model the global topology, play a critical role on finding the optimal alignment.

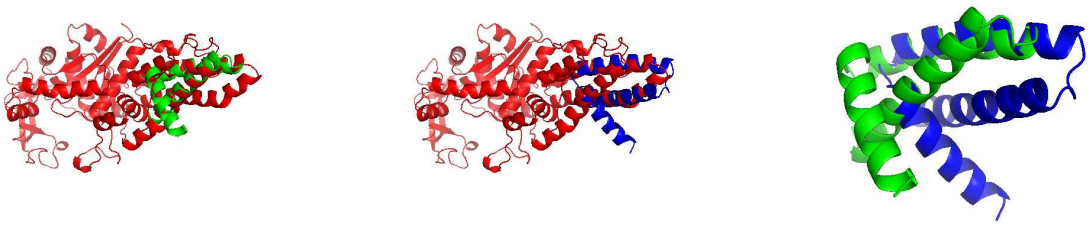
As shown in Figures 6.4(f) and 6.5(f), the superimposed protein structures of PROSTA-L12R3 and TM-align [87] are aligned to completely different regions of the red protein structures. This demonstrates one limitation of the initial alignment process without clustering: an initial sub-optimal alignment might not be the initial alignment with the highest alignment score, and thus a false alignment might be selected for the refinement process. After ROTRAN clustering, multiple ROTRANs (or alignments) from different clusters can be selected for the refinement process to avoid missing the optimal alignment.

In addition to comparison with TM-align [87], the TM-scores [85] found by PROSTA-L12R3 are also compared with those found by fr-TM-align [63], as shown in Figure 6.3(b). Note that TM-scores found by PROSTA-L12R3 are also mostly higher than those found by fr-TM-align for protein structure pairs that have fr-TM-align TM-scores lower than 0.65. Specifically, PROSTA-L12R3 improves TM-scores by up to 0.13, while it reduces TM-scores by at most 0.02. Moreover, PROSTA-L12R3 finds 45 more TM-scores that are higher than 0.5.

In summary, our alignment method performs more reliably in finding an optimal protein structure alignment with the highest TM-score [85] than do the current alignment methods, TM-align [87] and fr-TM-align [63]. Our alignment method, PROSTA-L12R3, not only can consistently find alignments with similar or higher TM-scores, but also can discover more alignments with TM-scores higher than 0.5. This promises better protein structure alignment and protein fold detection results. One, but certainly not the only one, reason for the success of our alignment method is using remote fragment pairs to implicitly model the global protein structure topology.

6.2.2 Alignment Quality Evaluation Using The GDT-TS Score

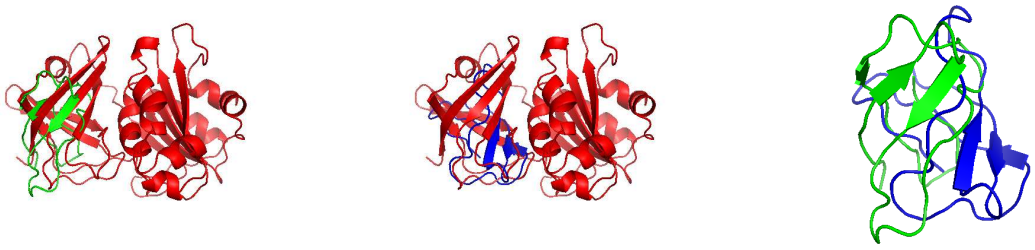
The GDT-TS score [83], in addition to the TM-score [85], is one of the most popular protein structure alignment scoring functions [**?**]. Thus, we repeated the previous experiment, but compared the GDT-TS scores found by our protein structure alignment method to those found by LGA [83] and SPalign [81]. Again, the highest GDT-TS score found by the three methods is referred to as the best-of-3 GDT-TS score, and only the 356 protein structure alignments with the best-of-3 GDT-TS scores higher than 0.5 are included in this analysis.



(a) 1A8H_ v.s. 1AIL_ (PROSTA-L12R3)

(b) 1A8H_ v.s. 1AIL_ (TM-align)

(c) 1AIL_ (PROSTA-L12R3) v.s. 1AIL_ (TM-align)

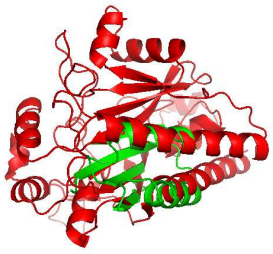


(d) 1A8P_ v.s. 1AFP_ (PROSTA-L12R3)

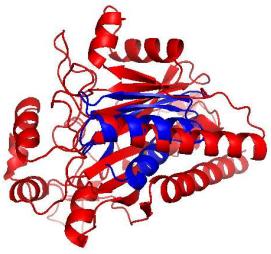
(e) 1A8P_ v.s. 1AFP_ (TM-align)

(f) 1AFP_ (PROSTA-L12R3) v.s. 1AFP_ (TM-align)

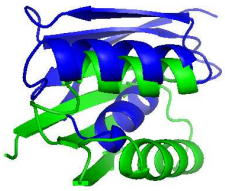
Figure 6.4: Comparisons of four protein structure alignments found by TM-align and PROSTA-L12R3: the query protein structure is shown in red; the target protein structure aligned by PROSTA-L12R3 is shown in green; the target protein structure aligned by TM-align is shown in blue; (a-c) PROSTA-L12R3: TM-score = 0.666, RMSD = 3.52, Sequence Identity = 0.11, Alignment Length = 64; TM-align: TM-score = 0.452, RMSD = 3.17, Sequence Identity = 0.04, Alignment Length = 49; (d-f) PROSTA-L12R3: TM-score = 0.525, RMSD = 2.96, Sequence Identity = 0.02, Alignment Length = 43; TM-align: TM-score = 0.327, RMSD = 2.80, Sequence Identity = 0.03, Alignment Length = 32.



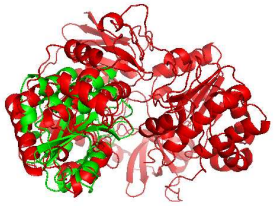
(a) 12ASA v.s. 1AW0_ (PROSTA-L12R3)



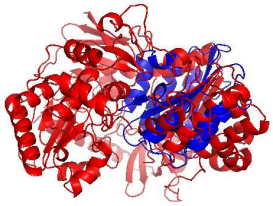
(b) 12ASA v.s. 1AW0_ (TM-align)



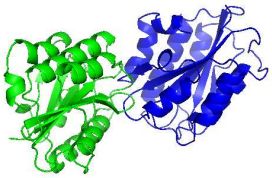
(c) 1AW0_ (PROSTA-L12R3) v.s. 1AW0_ (TM-align)



(d) 1AA6_ v.s. 1ATZA (PROSTA-L12R3)

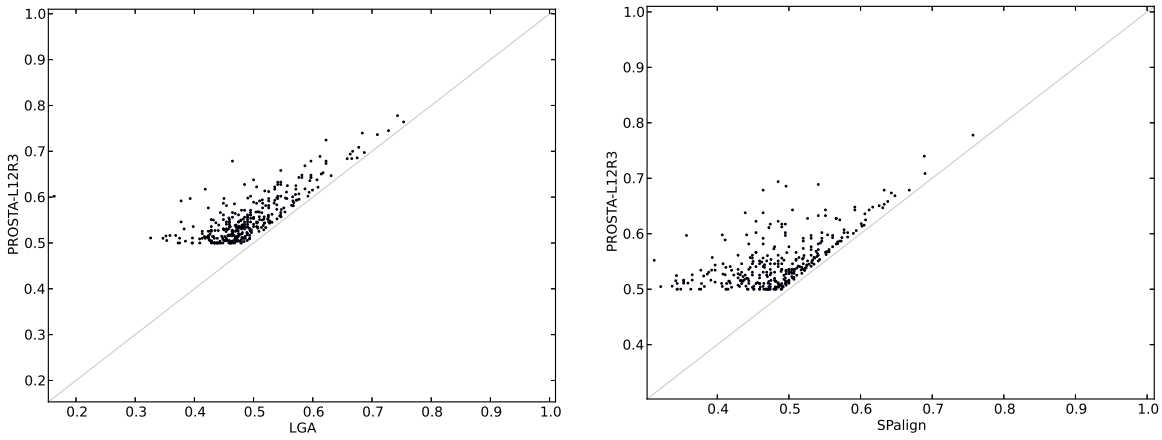


(e) 1AA6_ v.s. 1ATZA (TM-align)



(f) 1ATZA (PROSTA-L12R3) v.s. 1ATZA (TM-align)

Figure 6.5: Comparisons of four protein structure alignments found by TM-align and PROSTA-L12R3: the query protein structure is shown in red; the target protein structure aligned by PROSTA-L12R3 is shown in green; the target protein structure aligned by TM-align is shown in blue; (a-c) PROSTA-L12R3: TM-score = 0.502, RMSD = 5.42, Sequence Identity = 0.05, Alignment Length = 60; TM-align: TM-score = 0.365, RMSD = 4.04, Sequence Identity = 0.06, Alignment Length = 49; (d-f) PROSTA-L12R3: TM-score = 0.526, RMSD = 4.78, Sequence Identity = 0.10, Alignment Length = 142; TM-align: TM-score = 0.402, RMSD = 5.29, Sequence Identity = 0.10, Alignment Length = 125.



(a) LGA v.s. PROSTA-L12R3

(b) SPalign v.s. PROSTA-L12R3

Figure 6.6: Comparisons of the GDT-TS scores of LGA, SPalign and PROSTA-L12R3 (only cases yielding GDT-TS score differences higher than 0.01 are shown): (a) compared to LGA, PROSTA-L12R3 finds 217 (or 156%) more alignments with GDT-TS scores up to 0.44 higher than 0.5; (b) compared to SPalign, PROSTA-L12R3 finds 202 (or 131%) more alignments with GDT-TS scores up to 0.24 higher than 0.5.

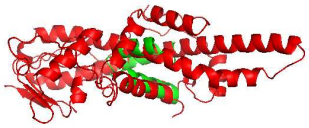
Comparing the GDT-TS scores [83] found by PROSTA-L12R3 and LGA [83] as shown in Figure 6.6(a), it can be seen that PROSTA-L12R3 consistently finds similar or higher GDT-TS scores than LGA. Specifically, PROSTA-L12R3 improves GDT-TS scores by 0.06 on average and by 0.44 in the best case. It is seen that 25% of the GDT-TS scores are improved by at least 0.08 and that 75% of the GDT-TS scores are improved by at least 0.03. As a result, PROSTA-L12R3 finds 217 (or 156%) more alignments with GDT-TS scores higher than 0.5. Similar results are observed comparing the GDT-TS scores found by PROSTA-L12R3 and SPalign [81], as shown in Figure 6.6(b). Therefore, our protein structure alignment method is further shown to be more reliable in finding high quality protein structure alignments.

Unlike previous observations, PROSTA-L12R3 tends to improve GDT scores [83] for protein structure alignments involving not only all-α proteins, but also all-β proteins. The improvement is even slightly better, on average, for all-β proteins than for all-α proteins, although the highest improvement is credited to an all-α protein structure alignment. Therefore, our alignment method is shown to be more reliable for finding high quality protein structure alignments, not limited to all-α protein structure alignments.

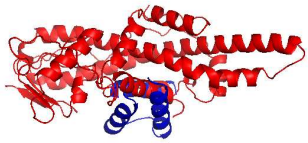
In summary, our method also performs more reliably in finding an optimal protein structure alignment with the highest GDT-TS score [83] than does LGA [83] and SPalign [81]. This conclusion holds for protein structure alignments involving not only α-helices but also β-strands.

6.3 Discussion

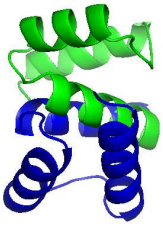
We have introduced PROSTA, our protein structure alignment method. Our result verifies the fact that the optimal protein structure alignment always contains a small subset of aligned residue pairs such that the ROTRAN, which minimizes the RMSD of the subset of aligned residue pairs, yields both the optimal ROTRAN and the optimal alignment score. Thus, our alignment method is not only reliable in finding the optimal alignment with the highest alignment score, but is also capable of discovering highly scored alignments missed by state-of-the-art alignment methods. This is the benefit of incorporating both



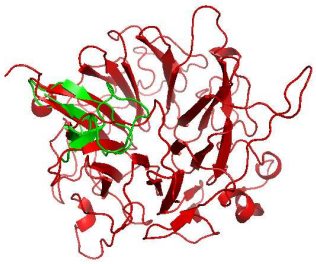
(a) 16VPA v.s. 1AKHA (PROSTA-L12R3)



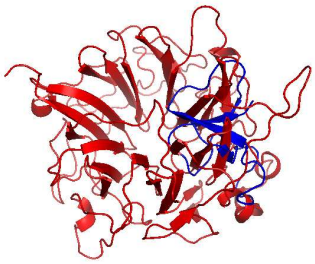
(b) 16VPA v.s. 1AKHA (LGA)



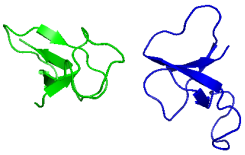
(c) 1AKHA (PROSTA-L12R3) v.s. 1AKHA (LGA)



(d) 1A12A v.s. 1ATX_ (PROSTA-L12R3)



(e) 1A12A v.s. 1ATX_ (LGA)



(f) 1ATX_ (PROSTA-L12R3) v.s. 1ATX_ (LGA)

Figure 6.7: Comparisons of four protein structure alignments found by LGA and PROSTA-L12R3: the query protein structure is shown in red color; the target protein structure aligned by PROSTA-L12R3 is shown in green color; the target protein structure aligned by LGA is shown in blue color; (a-c) PROSTA-L12R3: GDT-TS = 0.602, RMSD = 3.32, Sequence Identity = 0.09, Alignment Length = 45; LGA: GDT-TS = 0.163, RMSD = 3.38, Sequence Identity = 0.00, Alignment Length = 13; (d-f) PROSTA-L12R3: GDT-TS = 0.516, RMSD = 4.56, Sequence Identity = 0.06, Alignment Length = 34; LGA: GDT = 0.359, RMSD = 3.17, Sequence Identity = 0.00, Alignment Length = 28.

local fragments and remote fragment pairs in the alignment method.

Chapter 7

Evaluating Protein Structure Alignments Reliably

In Section 7.1, we introduce a new protein structure alignment scoring function incorporating several kinds of protein similarities. In this study, we focus on incorporating atom coordinate similarity and sequence similarity in a more reliable alignment score. As demonstrated in Section 7.2, although atom coordinate similarity may be efficient in many cases, sequence similarity helps to find better protein structure alignments that are (possibly significantly) more consistent with reference alignments examined by human experts.

7.1 Protein Structure Alignment Scoring Function

TM-score [85], based on LG-score [49], is one of the most successful protein structure alignment scoring functions. However, one limitation of both TM-score and LG-score is that they use only protein atom coordinate similarity while they ignore other protein similarities, such as sequence similarity. It has been observed that many protein structure alignments, based only on protein atom coordinate similarity, are highly sensitive to conformational changes [64]. This suggests the incorporation of other protein similarities, such as sequence similarity, in the protein structure alignment scoring function. Here, we introduce a new

scoring function incorporating many kinds of protein similarity as follows:

$$S = \frac{1}{L_r} \sum_{i \leq l} \frac{1}{1 + f_a(D_1(i), D_2(i), ..., D_n(i))},$$

where L_r is the reference protein size; l is the number of aligned residue pairs of the alignment; f_a is the weighted averaging function (e.g. arithmetic, geometric or harmonic average); $D_k(i)$ is the normalized distance of the i-th aligned residue pair using the k-th distance function; and n is the number of distance functions incorporated. If there is $n = 1$ and $D_1(i) = (d_i/d_0)^2$, where d_i is the distance between the C_α atoms of the i-th aligned residue pair and d_0 is a normalization factor, our scoring function is identical to the LG-score [49]. If there is also $d_0 = 1.24(L_r - 15)^{1/3} - 1.8$, our scoring function is identical to the TM-score [85]. Thus, LG-score and TM-score are two special cases of our scoring function.

In this study, we focus on the geometric average of the normalized C_α distance $D_1(i)$ and the normalized amino acid distance $D_2(i)$ as follows:

$$S = \frac{1}{L_r} \sum_{i \leq l} \frac{1}{1 + \sqrt[1+w]{D_1(i) D_2^w(i)}},$$

where w is a weighting factor. As with TM-score [85], we define the normalized C_α distance as

$$D_1(i) = (\frac{d_i}{d_0})^2,$$

where $d_0 = 1.24(L_r - 15)^{1/3} - 1.8$. Based on the popular BLOSUM62 matrix [30, 20], we define the normalized amino acid distance as

$$D_2(i) = 2^{-M(P_i,Q_i)} = 2^{-\lambda \log \frac{P(P_i,Q_i)}{P(P_i)P(Q_i)}} = (\frac{P(P_i)P(Q_i)}{P(P_i,Q_i)})^\lambda,$$

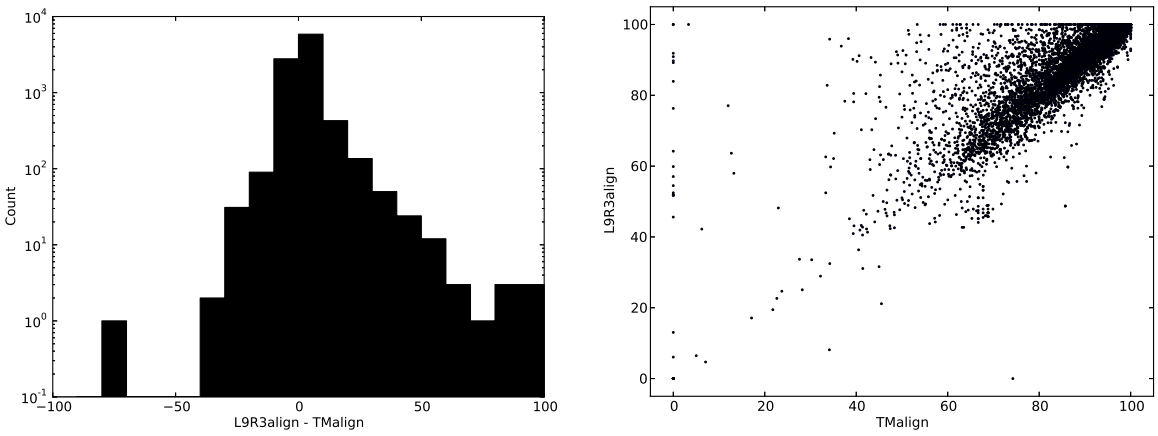
where M is the BLOSUM62 matrix, (P_i, Q_i) is the i-th aligned residue pair, λ is a scaling factor, $P(P_i, Q_i)$ is the probability of amino acid P_i aligning to amino acid Q_i, and $P(P_i)$ and $P(Q_i)$ are the probabilities of amino acid P_i and amino acid Q_i, respectively. Instead of using the default scaling factor λ, it is treated here as a parameter to control the rate of mutation.

An appealing property shared between TM-score [80] and our scoring function is that the prefered protein structure alignments tend to have scores higher than 0.5. If the C_α distance between the i-th aligned residue pair is prefered, there is $d_i < d_0$ and thus $D_1(i) < 1$. If the amino acid distance between the i-th aligned residue pair is prefered, there is $P(P_i, Q_i) > P(P_i)P(Q_i)$ and thus $D_2(i) < 1$. Then, for the i-th aligned residue pair, there is $D_1(i)D_2(i) < 1$ and thus $1/(1 + \sqrt[1+w]{D_1(i)D_2^w(i)}) > 0.5$. Therefore, if many prefered aligned residue pairs occur in the alignment, our protein structure alignment score tends to be higher than 0.5.

7.2 Results

In this experiment, we would like to show that our scoring function is capable of finding protein structure alignments that are significantly more consistent with alignments examined by human experts. Thus, we used protein structure alignments from the HOMSTRAD database [57] as a benchmark and compared the alignment quality of our PROSTA-L9R3 alignment optimizing our protein structure alignment scoring function with that of TM-align [87] optimizing TM-score [85]. Then, the quality of an alignment is evaluated by the F-score that is defined as the harmonic mean of the recall and the precision of the aligned residue pairs. Here, recall is the fraction of correctly predicted aligned residue pairs over the true aligned residue pairs, and precision is the fraction of correctly predicted aligned residue pairs over the predicted aligned residue pairs. Moreover, we chose the geometric average function with $\lambda = 0.25$ and $w = 1.9$, empirically.

The F-score differences between PROSTA-L9R3 and TM-align [87] are shown in Figure 7.1(a). Using PROSTA-L9R3, 47% of the F-scores are improved, and the average F-score is improved from 88% to 90% compared to using TM-align. Moreover, there are 663 PROSTA-L9R3 F-scores that are at least 10% higher and there are $1,342$ PROSTA-L9R3 F-scores that are at least 5% higher than the TM-align F-scores. For comparison, 31% of the TM-align F-scores are higher, and only 124 TM-align F-scores are at least 10% higher. Therefore, our scoring function tends to produce similar or higher F-scores compared to TM-score [85]



(a) F-score difference between PROSTA-L9R3 and TM-align

(b) TM-align F-score v.s. PROSTA-L9R3 F-score

Figure 7.1: Comparisons of the F-scores of the alignments found by PROSTA-L9R3 with our scoring function and TM-align with TM-score: 47% of the PROSTA-L9R3 F-scores are improved; 663 PROSTA-L9R3 F-scores are improved by at least 10%; and 1,342 PROSTA-L9R3 F-scores are improved by at least 5% compared to using TM-align.

The F-scores of the alignments found by PROSTA-L9R3 and TM-align [87] are compared in Figure 7.1(b). If the F-score of an alignment is higher than 90%, the alignment is considered to be highly consistent with HOMSTRAD alignments [57]. In total, TM-align finds $5,560$ such alignments, while PROSTA-L9R3 finds $6,114$ such alignments. Therefore, our scoring function finds 10% more alignments that are highly consistent with HOMSTRAD alignments.

Among the 34 pairs of protein structures that have TM-align [87] F-scores equal to zero, the PROSTA-L9R3 F-scores reach 36% on average. Specifically, two PROSTA-L9R3 F-scores equal to 100% and 19 PROSTA-L9R3 F-scores are higher than 50%. For the two cases that PROSTA-L9R3 F-scores are equal to 100%, the aligned protein structures contain a high percentage of helices, and TM-align shifts the HOMSTRAD alignment [57] by a few residues, which has also been previously observed [66]. Such shifting is difficult to avoid by evaluating only atom coordinate similarity. However, the shifting is avoided by our scoring function, involving both atom coordinate similarity and sequence similarity. There is also one pair of protein structures such that the PROSTA-L9R3 F-score is zero, while the TM-align [87] F-score is 74%. We have observed that such cases can be eliminated by using different weight parameters, and this problem will be addressed in our future work.

In summary, although atom coordinate similarity may be efficient in many cases, sequence similarity helps to find better protein structure alignments that are (possibly significantly) more consistent with HOMSTRAD alignments [57] examined by human experts. However, the weight parameters of our scoring function need to be carefully selected to eliminate the dependency between the alignment score and the protein size.

7.3 Discussion

Sequence similarity does aid in finding high quality protein structure alignments that are highly consistent with HOMSTRAD alignments [57] examined by human experts, even if atom coordinate similarity itself fails to do so. Our scoring function remains capable of modeling more types of protein similarities, such as the (ϕ, ψ) dihedral angle distance and the secondary structure distance, which might aid in finding higher quality protein struc-

ture alignments. Unknown protein domain length problems when aligning multi-domain proteins should also be addressed in the future as proposed by SPalign [81]. Moreover, the alignment quality can be further studied by checking self-consistency [66], and by simulating the SCOP fold detection [59].

Chapter 8

Discussion

We have introduced both a low-resolution approach (ContactLib) and a high-resolution approach (PROSTA) to assess the similarities and differences among protein structures. Our results show that both approaches outperform state-of-the-art methods. Unlike current methods that tend to focus on local information, remote information is used in both approaches. This critical difference is one reason for the success of our methods.

Both our low-resolution and high-resolution approaches are motivated from key observations. The low-resolution approach is based on the observation that ideal bond lengths and ideal bond angles can be used to accurately model protein structures. Thus, a low-dimension distance vector can be used to accurately model a contact group. The high-resolution approach is based on the observation that the optimal protein structure alignment always contains a small subset of aligned residue pairs such that the rotation and translation, which minimizes the RMSD of the subset of aligned residue pairs, yields both the optimal ROTRAN and the optimal alignment score. Thus, the optimal alignment can be calculated directly from the sampled subsets of aligned residue pairs.

Clustering is critical to our PROSTA method. First, since only a small number of sampled rotations and translations is sufficient for identifying large clusters, using clustering avoids alignment score calculation on most of the sampled noise rotations and translations. Second, since the alignments calculated from the same cluster tend to be highly similar and thus highly redundant, using clustering avoids alignment refinements on most of the

rotations and translations in the large clusters. This approach works well because good candidates tend to appear similar to each other, while noise candidates do not. For the same reason, clustering has also been used to distinguish good and noise predicted protein structures [51].

Our low-resolution and high-resolution approaches can also be combined together to archive high accuracy with high speed. To complete the experiments in Sections 6.2.1 and 6.2.2, the elapsed time required by PROSTA is approximately 4.5 hours on a workstation with dual Intel Xeon X5660 2.8GHz CPUs and dual Nvidia GeForce GTX 670 GPUs. Thus, each pairwise alignment took approximately 0.8 seconds on average. Considering that TM-align [87] requires approximately 45 minutes and LGA [83] requires approximately 10 hours, the runtime of PROSTA is approximately the average of fast and slow methods. In case that speed is critical, ContactLib can be used as preprocessing to filter the dataset for each query protein structure, and this preprocessing should be very fast (taking less than a second). If a keeper rate of 17% is used in the preprocessing, PROSTA would take approximately 45 minutes to complete the experiment, which is as fast as TM-align. A lower keeper rate can certainly be used to archive even higher speed.

8.1 Future Work

Our low-resolution approach can be improved in several ways. One possibility is to discover and study new types of contact groups. We will look for new definitions of distance vectors representing remote contact groups based on statistical energies [88]. Other than the atom-atom distances, the sequence similarities and the angles between certain (pseudo) bonds might also help to identify similar contact groups. Such bond angles have been previously used to calculate hydrogen bonds [37] and statistical energies [17].

We will also look for new applications for ContactLib. One promising application for ContactLib is the "structural BLAST" approach of PrePPI [16], whose performance depends mainly on the accuracy and on the speed of its neighbor protein structure retrieval. Moreover, ContactLib is also capable of finding neighbor protein structures if the query protein structure is only partially known in the process of protein structure prediction [84]

or determination [76]. Then, ContactLib may employ the incomplete $C_\alpha - C_\alpha$ pairwise distance matrix to find template candidates to enable it to predict or to determine the query protein structure.

Our PROSTA method has several limitations that will be addressed in the future. First, it would be interesting to study the impact of different clustering algorithms. For example, we observed clusters that are more suitable for a single-linkage clustering algorithm. Moreover, only the rotation of a ROTRAN is currently used in the clustering process, and using both the rotation and the translation might help to eliminate more noise. Second, it would be interesting to allow multiple ROTRANs from the same cluster for finding flexible pairwise structure alignments, as seen in FATCAT [82], and for finding flexible multi-structure alignments, as seen in Matt [56]. Third, the residue order in the alignment is consistent with that in the protein structure, and thus protein structure permutation is not allowed. However, proteins are flexible [77, 4], and protein structure permutation is important for studying protein folding and evolution [27, 31]. Moreover, the performance for aligning multi-domain protein structures [81] and multi-chain protein structures [6] remains open.

Our initial scoring functions are shown to be more accurate than current scoring functions, but are still subject to improvement. Recently, local superimposition-free scores have been used for evaluating the similarity between protein structures [43, 55]. Thus, a more accurate scoring function for our ContactLib method is promising. Similarly, a residue environment (including contacted residues) similarity score could be combined with the global alignment score in our scoring function. It would also be interesting to design a scoring function that is biased towards the inter-chain contacted residues or other residues that are specifically of interest to some researchers.

Our PROSTA method and scoring function can also be improved and applied to other structure studies of computational biology. For example, most structure alignment tools do not support aligning interaction interfaces between biological complexes involving DNAs and RNAs. One reason for this to happen is that they were not designed for the task because of their sequence order assumption. However, the sequence order might not be obeyed in the interaction interfaces. Especially, in case that multiple chains are involved in the interaction, the sequence order is not even defined. Thus, the Needleman-Wunsch dynamic programming algorithm [60] cannot be used safely. Possible solutions are the maximum

weighted bipartite matching method [25] and the integer programming method [19]. Then, interaction similarities can be added to our scoring function. This approach should be promising because remote fragment pairs could potentially model the interaction topology (or contact group) and become the key to the success of interaction interface alignment.

References

[1] Tatsuya Akutsu. Protein structure alignment using dynamic programing and iterative improvement. *IEICE TRANSACTIONS on Information and Systems*, 79(12):1629–1636, 1996.

[2] Zeyar Aung and Kian-Lee Tan. Rapid retrieval of protein structures from databases. *Drug Discovery Today*, 12(17-18):732–739, September 2007.

[3] Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Approximate clustering without the approximation. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1068–1077. Society for Industrial and Applied Mathematics, 2009.

[4] William S Bennett, Robert Huber, and JÜRgen Engel. Structural and functional aspects of domain motions in protein. *Critical Reviews in Biochemistry and Molecular Biology*, 15(4):291–384, 1984.

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

[6] T Andrew Binkowski, Shapor Naghibzadeh, and Jie Liang. Castp: computed atlas of surface topography of proteins. *Nucleic Acids Research*, 31(13):3352–3355, 2003.

[7] I. Budowski-Tal, Y. Nov, and R. Kolodny. FragBag, an accurate representation of protein structure, retrieves structural neighbors from the entire PDB quickly and accu-

rately. *Proceedings of the National Academy of Sciences*, 107(8):3481–3486, February 2010.

[8] A. A. Canutescu, A. A. Shelenkov, and R. L. Dunbrack. A graph-theory algorithm for rapid protein side-chain prediction. *Protein Science*, 12(9):2001–2014, September 2003.

[9] John-Marc Chandonia, Gary Hon, Nigel S Walker, Loredana Lo Conte, Patrice Koehl, Michael Levitt, and Steven E Brenner. The astral compendium in 2004. *Nucleic acids research*, 32(suppl 1):D189–D192, 2004.

[10] In-Geol Choi, Jaimyoung Kwon, and Sung-Hou Kim. Local feature frequency profile: a method to measure structural similarity in proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3797–3802, 2004.

[11] Xuefeng Cui, Shuai Cheng Li, Dongbo Bu, Babak Alipanahi, and Ming Li. Protein structure idealization: How accurately is it possible to model protein structures with dihedral angles? *Algorithms for Molecular Biology*, 8:5, 2013.

[12] Xuefeng Cui, Shuai Cheng Li, Dongbo Bu, and Ming Li. Towards reliable automatic protein structure alignment. In *WABI*, pages 18–32, 2013.

[13] Xuefeng Cui, Shuai Cheng Li, Dongbo Bu, Babak Alipanahi Ramandi, and Ming Li. How accurately can we model protein structures with dihedral angles? In *WABI*, pages 274–287, 2012.

[14] Xuefeng Cui, Shuai Cheng Li, Lin He, and Ming Li. Fingerprinting protein structures effectively and efficiently. *Bioinformatics*, page btt659, 2013.

[15] Noah M Daniels, Shilpa Nadimpalli, Lenore J Cowen, et al. Formatt: Correcting protein multiple structural alignments by incorporating sequence alignment. *BMC bioinformatics*, 13(1):1–8, 2012.

[16] Fabian Dey, Qiangfeng Cliff Zhang, Donald Petrey, and Barry Honig. Toward a structural blast: Using structural relationships to infer function. *Protein Science*, 2013.

[17] Guang Qiang Dong, Hao Fan, Dina Schneidman-Duhovny, Ben Webb, and Andrej Sali. Optimized atomic statistical potentials: assessment of protein interfaces and loops. *Bioinformatics*, 29(24):3158–3166, 2013.

[18] R. L. Dunbrack and F. E. Cohen. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Sci*, 6(8):1661–1681, August 1997.

[19] Joe Dundas, T Andrew Binkowski, Bhaskar DasGupta, and Jie Liang. Topology independent protein structural alignment. *BMC bioinformatics*, 8(1):388, 2007.

[20] Sean R Eddy et al. Where did the blosum62 alignment score matrix come from? *Nature biotechnology*, 22(8):1035–1036, 2004.

[21] R. A. Engh and R. Huber. Accurate bond and angle parameters for x-ray protein structure refinement. *Acta Crystallographica Section A*, 47:392–400, 1991.

[22] R. A. Engh and R. Huber. *Structure quality and target parameters*, volume F, chapter 18.3, pages 382–416. Wiley Online Library, 2006.

[23] P. R. Evans. An introduction to stereochemical restraints. *Acta Crystallographica Section D*, 63(1):58–61, Jan 2007.

[24] T. Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, June 2006.

[25] Mu Gao and Jeffrey Skolnick. ialign: a method for the structural comparison of protein–protein interfaces. *Bioinformatics*, 26(18):2259–2265, 2010.

[26] Deborah Goldman, Sorin Istrail, and Christos H Papadimitriou. Algorithmic aspects of protein structure similarity. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 512–521. IEEE, 1999.

[27] Weimin Gong, Margaret O'Gara, Robert M Blumenthal, and Xiaodong Cheng. Structure of pvuii dna-(cytosine n4) methyltransferase, an example of domain permutation and protein fold assignment. *Nucleic acids research*, 25(14):2702–2715, 1997.

[28] P. Gntert, C. Mumenthaler, and K. Wthrich. Torsion angle dynamics for nmr structure calculation with the new program dyana. *J. of Mol. Biol.*, 273(1):283–98, 1997.

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

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

[31] Jens Hennecke, Peter Sebbel, and Rudi Glockshuber. Random circular permutation of dsba reveals segments that are essential for protein folding and stability. *Journal of molecular biology*, 286(4):1197–1215, 1999.

[32] L. Holm and C. Sander. Protein Structure Comparison by Alignment of Distance Matrices. *Journal of Molecular Biology*, 233(1):123–138, September 1993.

[33] R. W. W. Hooft, G. Vriend, C. Sander, and E. E. Abola. Errors in protein structures. *NATURE*, 381(6580):272, MAY 23 1996.

[34] M. Jaskolski, M. Gilski, Z. Dauter, and A. Wlodawer. Stereochemical restraints revisited: how accurate are refinement targets and how much should protein structures be allowed to deviate from them? *Acta Crystallographica Section D*, 63(5):611–620, May 2007.

[35] R. P. Joosten, K. Joosten, S. X. Cohen, G. Vriend, and A. Perrakis. Automatic rebuilding and optimization of crystallographic structures in the Protein Data Bank. *Bioinformatics*, 27(24):3392–3398, December 2011.

[36] R. P. Joosten, T. A. H. te Beek, E. Krieger, M. L. Hekkelman, R. W. W. Hooft, R. Schneider, C. Sander, and G. Vriend. A series of PDB related databases for everyday needs. *Nucleic Acids Research*, 39(suppl 1):D411–D419, January 2011.

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

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

[39] Lisa Kinch, Shuo Yong Shi, Qian Cong, Hua Cheng, Yuxing Liao, and Nick V Grishin. Casp9 assessment of free modeling target predictions. *Proteins: Structure, Function, and Bioinformatics*, 79(S10):59–73, 2011.

[40] R. Kolodny, P. Koehl, and M. Levitt. Comprehensive Evaluation of Protein Structure Alignment Methods: Scoring by Geometric Measures. *Journal of Molecular Biology*, 346(4):1173–1188, March 2005.

[41] 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(12):2256–2268, 2004.

[42] G. G. Krivov, M. V. Shapovalov, and R. L. Dunbrack. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins: Structure, Function, and Bioinformatics*, 77(4):778–795, December 2009.

[43] Andriy Kryshtafovych, Bohdan Monastyrskyy, and Krzysztof Fidelis. Casp prediction center infrastructure and evaluation measures in casp10 and casp roll. *Proteins: Structure, Function, and Bioinformatics*, 2013.

[44] J. Kuszewski, A. M. Gronenborn, and G. M. Clore. Improving the quality of nmr and crystallographic protein structures by means of a conformational database potential derived from structure databases. *Protein Sci*, 5(6):1067–1080, 1996.

[45] J. Kuszewski, A. M. Gronenborn, and G. M.Clore. Improvements and extensions in the conformational database potential for the refinement of nmr and x-ray structures of proteins and nucleic acids. *Journal of magnetic resonance*, 125(1):171–177, March 1997.

[46] Y. Lamdan and H. Wolfson. Geometric hashing: a general and efficient model-based recognition scheme. In *ICCV*, 1988.

[47] R. A. Laskowski, M. W. MacArthur, D. S. Moss, and J. M. Thornton. *PROCHECK*: a program to check the stereochemical quality of protein structures. *Journal of Applied Crystallography*, 26(2):283–291, Apr 1993.

[48] E. Lattman, P. J. Loll, and P. Loll. *Protein crystallography: a concise guide*. Protein Crystallography. Johns Hopkins University Press, 2008.

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

[50] S. C. Li, D. Bu, J. Xu, and M. Li. Fragment-HMM: a new approach to protein structure prediction. *Protein Science*, 17(11):1925–1934, November 2008.

[51] Shuai Cheng Li, Dongbo Bu, and Ming Li. Clustering 100,000 protein structure decoys in minutes. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 9(3):765–773, 2012.

[52] Shuai Cheng Li et al. The difficulty of protein structure alignment under the rmsd. *Algorithms for Molecular Biology*, 8:1, 2013.

[53] Shuai Cheng Li and Yen Kaow Ng. On protein structure alignment under distance constraint. *Theoretical Computer Science*, 412(32):4187–4199, 2011.

[54] Mehdi Maadooliat, Xin Gao, and Jianhua Z Huang. Assessing protein conformational sampling methods based on bivariate lag-distributions of backbone angles. *Brief Bioinform*, 2012.

[55] Valerio Mariani, Marco Biasini, Alessandro Barbato, and Torsten Schwede. lddt: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728, 2013.

[56] Matthew Menke, Bonnie Berger, and Lenore Cowen. Matt: local flexibility aids protein multiple structure alignment. *PLoS computational biology*, 4(1):e10, 2008.

[57] Kenji Mizuguchi, Charlotte M Deane, Tom L Blundell, and John P Overington. Homstrad: a database of protein structure alignments for homologous families. *Protein science*, 7(11):2469–2471, 1998.

[58] Maher Moakher. Means and averaging in the group of rotations. *SIAM journal on matrix analysis and applications*, 24(1):1–16, 2002.

[59] Alexey G Murzin, Steven E Brenner, Tim Hubbard, and Cyrus Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995.

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

[61] 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 (London, England : 1993)*, 5(8):1093–1108, August 1997.

[62] Angel R Ortiz, Charlie EM Strauss, and Osvaldo Olmea. Mammoth (matching molecular models obtained from theory): an automated method for model comparison. *Protein Science*, 11(11):2606–2621, 2002.

[63] Shashi B Pandit and Jeffrey Skolnick. Fr-tm-align: a new protein structural alignment method based on fragment alignments and the tm-score. *BMC bioinformatics*, 9(1):531, 2008.

[64] Walter Pirovano, K Anton Feenstra, and Jaap Heringa. The meaning of alignment: lessons from structural diversity. *BMC bioinformatics*, 9(1):556, 2008.

[65] Rutgers and UCSD. Protein data bank contents guide. `http://www.wwpdb.org/documentation/format33/v3.3.html`, July 2011.

[66] M. I. Sadowski and W. R. Taylor. Evolutionary inaccuracy of pairwise structural alignments. *Bioinformatics*, 28(9):1209–1215, 2012.

[67] M. V. Shapovalov and R. L. Dunbrack. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure London England 1993*, 19(6):844–858, 2011.

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

[69] K. T. Simons, C. Kooperberg, E. Huang, and D. Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J. Mol. Biol.*, 268(1):209–225, 1997.

[70] E. G. Stein, L. M. Rice, and A. T. Brünger. Torsion-angle molecular dynamics as a new efficient tool for NMR structure calculation. *J. Magn. Reson.*, 124:154–164, 1997.

[71] Douglas L Theobald. Rapid calculation of rmsds using a quaternion-based characteristic polynomial. *Acta Crystallographica Section A: Foundations of Crystallography*, 61(4):478–480, 2005.

[72] G. Wang and R. L. Dunbrack. Pisces: a protein sequence culling server. *Bioinformatics*, 19(12):1589–1591, 2003.

[73] G. Wang and R. L. Dunbrack. Pisces: recent improvements to a pdb sequence culling server. *Nucleic Acids Research*, 33(Web-Server-Issue):94–98, 2005.

[74] Sheng Wang, Jianzhu Ma, Jian Peng, and Jinbo Xu. Protein structure alignment beyond spatial proximity. *Scientific reports*, 3:1–7, 2013.

[75] Haim J Wolfson and Isidore Rigoutsos. Geometric hashing: An overview. *Computational Science & Engineering, IEEE*, 4(4):10–21, 1997.

[76] Kurt Wüthrich. Protein structure determination in solution by nmr spectroscopy. *Journal of Biological Chemistry*, 265(36):22059–22062, 1990.

[77] Kurt Wüthrich and Gerhard Wagner. Internal motion in globular proteins. *Trends in biochemical sciences*, 3(4):227–230, 1978.

[78] J. Xu. Rapid protein side-chain packing via tree decomposition. In *Research in Computational Molecular Biology, Lecture Notes in Computer Science*, pages 423–439. Springer, 2005.

[79] J. Xu, F. Jiao, and B. Berger. A tree-decomposition approach to protein structure prediction. *Proc. IEEE Comput. Syst. Bioinform. Conf.*, pages 247–256, 2005.

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

[81] Yuedong Yang, Jian Zhan, Huiying Zhao, and Yaoqi Zhou. A new size-independent score for pairwise protein structure alignment and its application to structure classification and nucleic-acid binding prediction. *Proteins: Structure, Function, and Bioinformatics*, 80(8):2080–2088, 2012.

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

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

[84] Jingfen Zhang, Qingguo Wang, Bogdan Barz, Zhiquan He, Ioan Kosztin, Yi Shang, and Dong Xu. Mufold: a new solution for protein 3d structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 78(5):1137–1152, 2010.

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

[86] Yang Zhang and Jeffrey Skolnick. Spicker: A clustering approach to identify near-native protein folds. *Journal of computational chemistry*, 25(6):865–871, 2004.

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

[88] Hongyi Zhou and Yaoqi Zhou. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein science*, 11(11):2714–2726, 2002.