EFFICIENT SEARCH AND COMPARISON ALGORITHMS FOR 3D PROTEIN BINDING SITE RETRIEVAL AND STRUCTURE ALIGNMENT FROM LARGE-SCALE DATABASES

A Dissertation

presented to

the Faculty of Graduate School

at the University of Missouri – Columbia

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

by

BIN PANG

Dr. Chi-Ren Shyu, Dissertation Supervisor

Dr. Dmitry Korkin, Dissertation Co-Supervisor

May 2013

The undersigned, appointed by the dean of the Graduate School, have examined the dissertation entitled

EFFICIENT SEARCH AND COMPARISON ALGORITHMS FOR 3D PROTEIN BINDING SITE RETRIEVAL AND STRUCTURE ALIGNMENT FROM LARGE-SCALE DATABASES

presented by Bin Pang,

a candidate for the degree of Doctor of Philosophy

and hereby certify that, in their opinion, it is worthy of acceptance.

_____

Dr. Chi-Ren Shyu

_____

Dr. Dmitry Korkin

_____

Dr. Gavin Conant

_____

Dr. Lesa Beamer

Date Approved:_____

ACKNOWLEDGEMENTS

First of all, I would like to extend my greatest gratitude to my co-advisors, Dr. Chi-Ren Shyu and Dr. Dmitry Korkin, who always provided professional direction and guidance for me through the whole duration of my doctoral study. Dr. Shyu has given me solid training of academic research and provided his support beyond the call of duty. Dr. Korkin has continued to share valuable domain knowledge of bioinformatics with me through many discussions. I also would like to thank my committee members, Dr. Lesa Beamer and Dr. Gavin Conant. They are generous and kind for providing so many invaluable suggestions for my research.

I also would like to thank my friends from the Medical and Biological Digital Library Research Lab who are always supportive. I am also grateful to the Shumaker Endowment for Bioinformatics, Informatics Institute, and University of Missouri Research Board for their continuous financial support of my research.

Finally, I would like to thank my dear mother, Mrs. Jinlan Liu, and my dear wife, Jingfen, who have given me great spiritual support. I also would like to thank my daughters, Caroline and Eileen, who have given me so much joy. Without them, my research and this dissertation would not have been possible. My honors and achievements are dedicated to all of these people.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

EFFICIENT SEARCH AND COMPARISON ALGORITHMS FOR 3D PROTEIN BINDING SITE RETRIEVAL AND STRUCTURE ALIGNMENT FROM LARGE-SCALE DATABASES

Bin Pang

Dr. Chi-Ren Shyu, Dissertation Supervisor

Dr. Dmitry Korkin, Dissertation Co-Supervisor

ABSTRACT

Finding similar 3D structures is crucial for discovering potential structural, evolutionary, and functional relationships among proteins. As the number of known protein structures has dramatically increased, traditional methods can no longer provide the life science community with the adequate informatics capability needed to conduct large-scale and complex analyses. A suite of high-throughput and accurate protein structure search and comparison methods is essential. To meet the needs of the community, we develop several bioinformatics methods for protein binding site comparison and global structure alignment. First, we developed an efficient protein binding site search that is based on extracting geometric features both locally and globally. The main idea of this work was to capture spatial relationships among landmarks of binding site surfaces and build a vocabulary of visual words to represent the characteristics of the surfaces. A vector model was then used to speed up the search of similar surfaces that share similar visual words with the query interface. Second, we developed an approach for accurate protein binding site comparison. Our algorithm provides an accurate binding site alignment by applying a two-level heuristic process which progressively refines alignment results from coarse surface point level to accurate residue atom level. This setting allowed us to explore different combinations of pairs of

corresponding residues, thus improving the alignment quality of the binding site surfaces. Finally, we introduced a parallel algorithm for global protein structure alignment. Specifically, to speed up the time-consuming structure alignment process of protein 3D structures, we designed a parallel protein structure alignment framework to exploit the parallelism of Graphics Processing Units (GPUs). As a general-purpose GPU platform, the framework is capable of parallelizing traditional structure alignment algorithms. Our findings can be applied in various research areas, such as prediction of protein interactions and screening in drug discoveries. The uniqueness of our methods is its ability to manipulate the big data of protein structures by integrating the feature- and alignment-based search and comparison approaches together, thus enabling more efficient and accurate analysis of large-scale protein structures for the research community.

CHAPTER ONE

INTRODUCTION

1.1. Motivations

The great demand for an efficient and accurate comparison algorithm for three-dimensional (3D) protein structures has continued to rise due to the dramatic increase in protein structural data and the role of protein structures in biological findings [1]. The number of known protein structures in the primary structural database, the Protein Data Bank (PDB), had reached 80,086 (~225,090 protein chain structures) as of 10 March 2013, and this number is expected to continue growing at a high rate. Given a large-scale database of protein structures, the main goal of structure comparison is either to find proteins that are structurally similar to a given protein (i.e., one-against-all comparison) or to build various connectivities among proteins by performing exhaustive comparisons on the whole database (i.e., all-against-all comparison). The results of structural comparison are useful in discovering potential structural, evolutionary, and functional relationships among these proteins and have significant impact on structure-based drug design [2], protein-protein docking [3], and other biological findings [4].

Protein structure comparisons can be generally classified into two categories: global and local comparisons. The global structure comparison tries to superimpose most of the corresponding backbone atoms from two proteins, while the local comparison seeks to find common substructures between two proteins irrespective of similarities of protein global structures. In this dissertation, the local structure

comparison refers to the match of protein binding sites (PBS). In a protein complex, the binding site corresponds to a region of residues that are in close spatial proximity and can interact with residues from another subunit (chain or domain) [5]. The interaction between two subunits, also known as protein-protein interaction (PPI), plays a significant role in controlling biological processes and determining corresponding functions [6,7,8].

To provide global or local structure comparisons, one cluster of approaches is based on alignment of global or local structures to provide accurate comparison between two structures [9,10,11]. These approaches are usually computationally expensive, which makes them infeasible in coping with very large datasets. To accelerate this process, another cluster is to compare structures with extracted features without explicit alignments [12,13,14]. While this approach can deal with large datasets, a fine-level alignment is still needed to find accurate correspondences between two structures. Hence, the structure biology community needs informatics tools that can offer a full solution for fast retrieval and accurate alignment of global and local protein structures.

1.2. Summary of Contributions

In this dissertation, a suite of novel methods is proposed to quickly identify globally or locally similar protein structures from a large dataset and provide accurate alignments. The proposed methods integrate algorithms from information retrieval, computer vision, computer graphics, parallel computing, and structure biology to tackle computational biology challenges. The contributions of this dissertation are summarized as follows:

- Provides an efficient information retrieval algorithm for fast binding site search (PBSword). The method includes (i) a feature-based description of binding site surfaces by integrating both global and local geometric features and (ii) fast comparison algorithms for binding sites by utilizing a novel "visual words" representation generated by clustering a large set of feature descriptors.
- Provides an accurate alignment algorithm to support superimposition of protein binding site (PBSalign). The algorithm combines surface and structure information and progressively refines alignments of binding sites. A "filter and refine" paradigm can be developed to perform accurate alignment of binding site for a short list obtained from PBSword.
- Creates a parallel computing platform for large-scale protein global structure alignments (*ppsAlign*). The platform utilizes the massively parallel computing power of GPU (Graphic Processor Unit) to support residue-level alignment and optimizes the algorithm from the levels of architecture, memory layout and instruction.
- Disseminates computational tools and data to the scientific community. The software package includes (i) feature database of candidate binding sites and (ii) standalone or web-based software for global and binding site comparison of protein structures.

1.3. Thesis organization

This dissertation is organized as follows. Chapter 2 surveys literature on recent research related to protein structure comparison. Chapter 3 introduces our feature-based

method for the local protein binding site search. Chapter 4 explains our alignment methods for accurate protein binding site comparisons. Chapter 5 describes our GPU-based alignment algorithm for protein global structure comparisons. System design of PBSword server is given in Chapter 6. Finally, we summarize this dissertation and discuss possible future works in Chapter 7.

CHAPTER TWO

LITERATURE REVIEW

This chapter surveys existing work in protein global structure comparisons, GPU-accelerated structure alignments, and protein binding site structure comparisons.

2.1. Protein Global Structure Comparisons

The purpose of global comparison is to align two protein structures using their overall structural similarities such that potential functional and evolutional relationship can be discovered. The global comparisons can be classified into two categories: (i) alignment-based methods and (ii) structural alphabet-based methods. The alignment-based methods utilize the comparison of residues or fragments to build initial correspondences among residues which are further optimized by various procedures, such as Monte-Carlo, combinational search, and Dynamic Programming (DP). These methods can provide accurate alignments at the residue level but are usually computationally expensive. To accelerate the structural alignment, the structural alphabet-based approaches have been developed. These solutions map the protein structures into 1D sequences of structural alphabets and then use various sequence alignment methods to align two structures [15,16]. The structural alphabets are usually obtained by clustering geometrically similar fragments of proteins from a selected dataset. Generally, the structural alphabet-based approaches exhibit good efficiency; however, this is often achieved at the cost of loss of topological details, which could

lead to lower accuracy than the traditional alignment-based comparison methods or could be unsuitable to perform residue-level alignment.

2.1.1 Alignment-based Methods

DALI (Distance Alignment) [17] is a popular structural alignment method which performs pair-wise comparison based on a distance matrix, which is calculated using C_α coordinates of a pair of residues. The distance matrix is decomposed into sub-matrices to simplify the alignment complexity. Finally, a Monte Carlo procedure is deployed to generate a similarity score which is defined in terms of equivalent intra-molecular distances.

CE (Combinatorial Extension) [18] aligns two protein structures based on the combinatorial extension of an alignment path. The alignment path is defined by aligned fragment pairs (AFPs) in which structures show certain similarity in their local geometry. Different combinations of AFP can be used to represent possible continuous alignment paths, which could be further used to produce an optimal alignment by extending along the aligned path.

TM-align [19] initially superimposes protein structure pairs by TM-score [20] matrix and Dynamic Programming. Several seed alignment results are obtained after the initial superimposition, which are iteratively refined until the alignment becomes stable. The alignment with the highest TM-score is selected as the final result. In a benchmarking study [19], TM-align is reported to improve in both speed and accuracy over DALI and CE. An extension of TM-align, Fr-TM-align [21], is proposed to improve accuracy of TM-align by using an exhaustive search procedure to generate

initial seed alignments. Though showing a higher TM-score in comparison to TM-align, Fr-TM-align [21] is computationally more expensive than TM-align (~12 times slower than TM-align).

MAMMOTH [22] uses the similar idea of AFP as CE to decompose protein structures into seven-residue fragments. The similarity score between two fragments is calculated using a unit-vector RMS (URMS) method [23]. These scores are stored in a similarity matrix, and a Dynamic Programming method is used to optimize residue-residue correspondence and find the optimal residue alignment.

SABERTOOTH [24] represents protein structure with a new concept of profile which can characterize global structure connectivity of each residue as well as reveal relationships between structure and sequence. Compared to existing methods, SABERTOOTH can simultaneously handle multiple scenarios of alignment, such as sequence-to-sequence and sequence-to-structure comparisons.

SPalign [25] aligns two protein structures by optimizing a similarity score– SPscore. Different from the existing similarity scores, SPscore is independent of protein size so that aligned regions with different sizes can be compared. The alignment method is similar to that of TM-align [19] which iteratively refines seed alignments to obtain an optimal result with the Dynamic Programming method.

2.1.2 Alphabet-based Methods

Kolodny et al. [26] introduced a library of 20 representative fragments (i.e., structural alphabet) by applying a clustering algorithm on a set of continuous residues

extracted from protein backbone and showed quality of fit (i.e., accuracy) of these fragments to their native structures. In a later study, both local and global sequences of structural alphabets are used to perform protein classification, and higher accuracy was achieved [27].

Brevern et al. [28] proposed a set of structural alphabet (or protein blocks) based on five-residue fragment. Each fragment is encoded using dihedral angles of residues, which are further clustered to form a representative set of 16 protein blocks.

In 2006, Yang and Tung [16] developed 3D-Blast which characterizes patterns of the backbone fragments with 23 structural alphabet and then uses them to represent protein structure as 1D sequence. A database of alphabet sequence is constructed and the traditional BLAST [29], originally designed for comparing primary biological sequence, is employed as a search method to find the longest common substructures from the alphabet sequence database.

YAKUSA [15] utilized protein backbone internal coordinates (alpha angles) to describe protein structure as a sequence of structural alphabets. YAKUSA searches for the longest common substructures existing between a query structure and every structure in the database using a similar method as BLAST. Finally, YAKUSA selects the compatible substructure pairs for the query and database structures based on a similarity score.

Budowski-Tal, Nov, and Kolodny created FragBag [30] which describes a protein backbone by a collection of structural alphabets proposed in [26]. Each protein is

then represented as a vector of frequency (or histogram) of structural alphabets. Finally, the similarity of two structures is evaluated by the similarity of their vectors.

Tyagi et al. [31] utilized the structure alphabet proposed in [28] to convert protein structure into a 1D sequence and defined a substitution matrix of the alphabet. The 1D sequence comparison is performed using the Dynamic Programming approach. The experimental results showed this approach to be comparable to the alignment-based comparison methods such as DALI.

2.2. GPU-based Protein Alignments

To accelerate the protein structure comparison, in addition to the approach of using 1D alphabet-based alignment, another approach was developed to parallelize traditional algorithms using a cluster or grid environment consisting of thousands of computing nodes [32,33]. These approaches can fulfill the desires of efficiency and accuracy but require high-performance computing environments which are energy-consuming and may not be accessible to all biologists.

With the increase in performance and programmability of many-core GPUs [34], more and more bioinformatics applications have been deployed on GPUs and have shown promising results in terms of speedup over the corresponding conventional CPU implementations.

Liu et al. [35] implemented a GPU-based Smith-Waterman algorithm [36] for pair-wise DNA sequence alignment. The experimental results showed that on a NVIDIA GeForce 6800 GTO GPU card, the proposed approach allows speedups of more than one

order with respect to optimized CPU implementation on Intel Pentium 4 at 3.0 GHz with 1 GB RAM (Random Access memory). Later, the efficiency of sequence alignments was continuously improved in [37,38,39,40].

Vouzis and Sahinidis developed GPU-BLAST (Basic Local Alignment Search Tool) [41] to accelerate NCBI-BLAST [29]. When directly porting from the source codes of NCBI-BLAST, GPU-BLAST maintains the same input and output interface while producing identical results. In comparison to the CPU-based NCBI-BLAST running on a six-core Intel Xeon host CPU at 2.67 GHz with 12GB RAM, GPU-BLAST can achieve a speedup between 3 and 4 on a NVIDIA Fermi C2050 GPU card.

Stivala et al. [42] utilized simulated annealing (SA) to develop a protein substructure searching algorithm, SA Tableau Search, to find structural motif at level of secondary structure element (SSE). The GPU implementation on NVIDIA Tesla C1060 achieves up to 34 times speedup over the CPU implementation on AMD Quad Core Opteron at 2.3 GHz with 32 GB RAM. It is worth mentioning that from the literature, the SA Tableau Search is the first attempt to apply GPU in protein structure comparison at the SSE level.

2.3. Protein Binding Site Comparisons

Different from the protein global comparisons, protein binding site comparisons focus on local similarity. Early research studies in this area compared binding sites by aligning the corresponding C_α atoms from different protein complexes using global structural alignment tools [43,44,45]. This group of methods works well when sequence and structure are well conserved. However, as shown in an earlier work [46], the

limitation of global protein structure alignment is that it cannot usually produce accurate interface alignment since the true interface may not be the first priority for a global structure alignment method.

To overcome this issue, several alignment-based methods have been developed specifically for the comparison of protein binding sites or protein-protein interfaces. Comparing protein binding sites at 3D level is challenging because the residues of a binding site are not always sequential in nature, resulting in a large searching space for possible alignments. To date, only a handful of methods are available for the binding site or protein-protein interface alignments, which is in sharp contrast to the overall structure alignment methods developed in the last three decades [47]. The alignment-based methods can provide accurate correspondence between two binding sites but are usually computationally expensive.

Feature-based methods, on the other hand, make comparisons based on feature descriptors, which may represent structural and/or geometric properties of the binding site. Typically, the structural descriptors are distributions of distances between different types of functional atoms of the binding sites [48], whereas the geometric descriptors include spin-image [49], 3D-Zernike [13], shape distribution [12], and moment invariants [50]. After generating surface features, two binding sites can be compared using various metric functions in the feature space to perform high-throughput surface comparisons without explicit alignments. The existing feature-based methods mainly focus on protein-ligand binding sites, which have not been extensively applied or evaluated on protein-protein binding site classification and related applications. This is

important as protein-protein binding sites are known to have some unique characteristics, such as relatively large areas and flat binding site surfaces [51].

2.3.1 Alignment-based Methods

I2ISiteEngine [52] calculates the surface properties of protein binding site at selected pseudo-center, which is defined as a site residue belonging to one of the following functional groups: hydrogen-bond donor (DO), hydrogen-bond acceptor (AC), mixed donor/acceptor (ACDO), hydrophobic aliphatic (AL) and aromatic (AO). The matching of interface is based on hashing of almost congruent triangles defined by triplets of pseudo-centers. Finally, a hierarchical scoring scheme (low- and high resolution) is applied simultaneously to both sides of the interface to find out the best solutions. Similar to I2ISiteEngine, PFS (Protein Functional Surfaces) [53] and SURFCOMP [54] utilize global shape and local physicochemical properties of binding site to match protein-ligand binding sites.

Galinter [10] compares similarity of two protein-protein interfaces based on the geometry and type of non-covalent interactions. Each interface is modeled as a graph and maximum common subgraph [55] is used to align two interfaces. Evaluated on a pilot dataset introduced in [52], Galinter shows similar alignment quality as I2I-SiteEngine for homologous complexes. ProBiS [56] also represents the protein interface as a graph and utilizes the graph theory to compare two interfaces. The difference is ProBiS clusters surface residues based on the same functional groups as that used in I2ISiteEngine.

iAlign [9] was developed to align residues from protein-protein interfaces based on a score of residue similarity and an iterative Dynamic Programming algorithm. The score could be TM-score [19] or an extension of TM-score which integrates the numbers of residue contacts of two interfaces. The experimental results show iAlign can achieve higher accuracy than I2I-SiteEngine based on the classification of protein-protein interfaces. CMAPi [57] searches the optimal alignment of two interfaces utilizing the Dynamic Programming method and contact map of interface residues which is a 2D matrix defined as the minimum distance between all heavy atoms on the interfaces.

2.3.2 Feature-based Methods

Merelli et al. [49] identified correspondences between points of two protein surfaces according to the similarity of spin-image, which is defined as a 2D image describing the local topology of a surface point. All these candidate pairs are then filtered, using specific similarity thresholds, and clustered to provide final match.

Sael et al. [13] characterized surface points based on 3D Zernike descriptors, which provided a compact representation of a given property defined on a protein surface. Similarity measure of two descriptors was calculated using Euclidean distance.

Das, Kokardekar and Breneman [12] described each protein-ligand binding site surface using property-encoded shape distributions (PESD). Similarity between the PESDs can then be treated as a measure of similarity between two binding sites.

Sommer et al. [50] utilized moment invariants as a descriptor to characterize protein-protein binding sites. The descriptor is based on shape features and can be used to compare the binding sites independently from the sequence similarities.

Finally, Table 1 summarizes the typical methods of our survey. The column 'Method' lists the name and reference of these methods which are classified into three categories (i.e., global structure alignments, alignment-based, and feature-based protein binding site comparisons) and marked using different background colors. The methods from row 'DALI' to '*ppsAlign*' belong to the category of global structure alignment. The row '*ppsAlign*' is a GPU-based platform for global structure alignment. The methods from row 'I2ISiteEngine' to 'PBSalign' are from the category of alignment-based protein binding site comparison. The methods from 'Merelli et al.' to 'PBSword' are from the category of feature-based protein binding site comparisons. In the table, the methods '*ppsAlign*', 'PBSalign', and 'PBSword' are the key research results presented in this dissertation. The column 'Type' include three categories, 'Global', 'Local', and 'Local/Feature', which corresponds to the global structure alignment, alignment-based, and feature-based binding site comparisons, respectively. The column 'Hardware' specifies hardware configuration for performance evaluation. The column 'Dataset' is a brief description of dataset used for the evaluation. The columns 'Accuracy' and 'Efficiency' show experimental results of the methods.

Table 1: Summary of global and local protein structure comparisons.

METHODS	TYPE	HARDWARE	DATASET	ACCURACY	EFFICIENCY
DALI [17]	Global	Sparc-1	representative protein chains	n/a	5-10 min. for protein pairs
CE [18]	Global	Ultra Sparc II 248Mhz	representative protein chains	n/a	20 sec. for 100 Protein structures
TM-align [19]	Global	Intel 1.26 GHz Pentium III	10,515 representative protein chains	TM-score of CE: 0.44 DALI: 0.47 TM-align: 0.51	4 times faster than CE and 20 times faster than DALI
Fr-TM-align [21]	Global	AMD 2 GHz Opteron	10,515 representative protein chains	Improve TM-score by 9%	12 times slower than TM-align
MAMMOTH [22]	Global	n/a	Predicted protein models	Correlated better than other tools	n/a
SABERTOOTH [24]	Global	Intel 2.8 GHz Xeon	525 protein structures	Comparable accuracy as DALI, CE, and MAMMOTH	Faster than DALI and CE, but slower than MAMMOTH
SPalign [25]	Global	AMD 1.8GHz Opteron	representative protein chains	Improve TM-score by 4.6%	Similar as TM-align
ppsAlign	**Global**	**NVIDIA Tesla C2050 GPU and AMD 1.8GHz Opetron CPU**	**representative protein chains**	**Comparable accuracy as TM-align, Fr-TM-align and MAMMOTH**	**36-fold speedup over TM-align, 65-fold speedup over Fr-TM-align, and a 40-fold speedup over MAMMOTH**
I2ISiteEngine [52]	Local	Intel 3.0GHz Xeon	representative protein interfaces	n/a	28 sec
Galinter [10]	Local	3.0 GHz CPU	Dataset from I2ISiteEngine	Comparable accuracy as I2ISiteEngine on homologous dataset	138.5 sec
CMAPi [57]	Local	n/a	representative protein interfaces	Better accuracy than its previous version	n/a
iAlign [9]	Local	AMD 2.4GHz Opteron	Biologically related protein interfaces	Better accuracy than I2ISiteEngine	89-fold speedup over I2ISiteEngine
PBSalign	**Local**	**AMD 1.8GHz Opetron CPU**	**Homologous and non-homologous protein interfaces**	**Better accuracy than iAlign**	**Slower than iAlign and I2ISiteEngine**
Merelli et al. [49]	Local/Feature	Intel Core 2 processors	representative protein-ligand interfaces	Better accuracy than ZDOCK and Rosetta DOCK for two-thirds cases	Seconds to 10 min
Sael et al. [13]	Local/Feature	n/a	representative protein-ligand interfaces	Cluster proteins into function-related groups	0.05 ~ 50 sec
Das et al. [12]	Local/Feature	Intel 8 core 2.66 GHz	representative protein-ligand interfaces	79.5% for top match	0.0056 sec
Sommer et al. [50]	Local/feature	n/a	representative protein-protein interfaces	40% for top match	n/a
PBSword	**Local/Feature**	**AMD 1.8GHz Opetron CPU**	**Dataset from [50]**	**76% for top match**	**0.0016 sec**

CHAPTER THREE

EFFICIENT PROTEIN BINDING SITE SEARCH

In this chapter, we extend the text comparison method from the Information Retrieval (IR) area and propose a novel approach, PBSword, for characterizing protein binding sites with a collection of "visual words" such that similar binding sites from the database can be efficiently and accurately identified. The methods and results of this chapter have been published in *Bioinformatics* journal [58]

3.1. Methods

The framework of PBSword is shown in Figure 1. The inputs include a query binding site and a database of protein binding sites. The outputs are similarity scores between the query and database binding sites. The workflow of PBSword consists of the following four steps. First, we select feature points of each database binding site surface and extract corresponding geometric features. Second, a visual vocabulary is built by clustering a huge number of feature point descriptors collected from the entire database of binding sites. Third, according to its descriptor, each feature point is associated with the nearest visual word from the visual vocabulary. This allows each binding site to be represented by the corresponding distribution of visual words. The above processes for the database binding sites are performed off-line. For the query binding site, we follow similar steps to generate its visual word representation. Finally, pairs of binding sites are compared by a scoring function which calculates the similarity between the two visual word vectors.

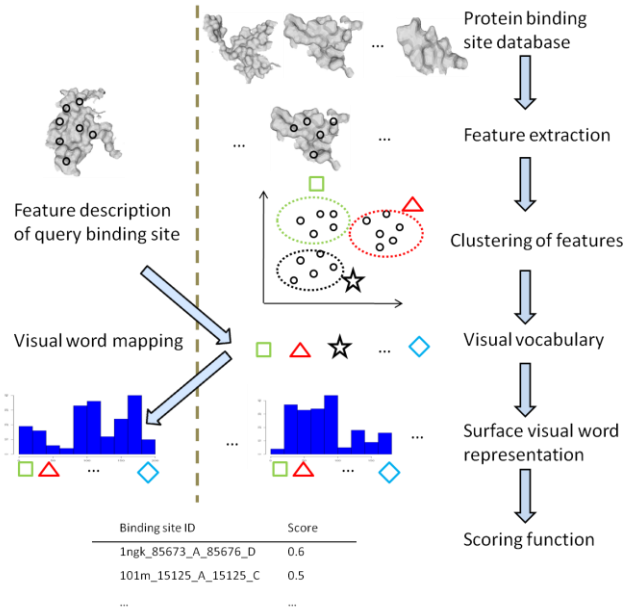

Figure 1: Framework of PBSword. The inputs include a query binding site and a database of binding sites. The outputs are similarity scores between the query and the database binding sites. For each binding site, geometric features are extracted and assigned to the nearest visual word from a vocabulary which is generated by a huge set of features collected from the entire database. Thus, each binding site can be represented by a histogram of occurrence of visual words. Two binding sites can be compared by the similarity of two visual words vectors.

Comparing protein binding sites is a more challenging problem than comparing protein sequences or structures since the residues of a binding site are not always sequential in nature [45]. Hence, a binding site cannot be represented by a string of residues from the N- to C- terminus of a protein chain. To solve this problem, we first use the surface features to represent the whole binding site. Then, by projecting the feature vectors into a visual word from the vocabulary, we can use the occurrence of visual words to describe the surface, which alleviates the need for sequential relationships between two feature points. As a feature-based method, PBSword shares some attractive properties with other similar methods in this category, such as being sequence- and structure- independent and alignment-free. The uniqueness of PBSword is

in the way it represents the binding sites as a collection of visual words, which can be used not only in the development of a compact representation for a family of protein binding sites but also in the construction of an inverted index for fast retrieval of geometrically similar binding sites from a large dataset.

3.1.1 Surface generation

For a protein complex, we use the MSMS program [59] to generate a triangulated mesh for each of its interacting subunits and set the density and probe radius to 2.0 points/Å^2 and 1.4 Å, respectively. Since we are only interested in the binding regions, for each protein mesh, we retain only those surface points that are within a distance of 4Å from the surface of its binding partner [52]. A triangle is selected when its three vertices are all retained in the interaction region. In this dissertation, we define a face, as used in the computer graphics community, to represent the surface of a triangle.

3.1.2 Feature descriptor

In our approach, feature points are defined as the centers of faces (or triangles), and a number of sample faces are selected using the procedure proposed in [60] to improve computational efficiency. In this method, we first calculate the area of each face on the surface and store it as an array of cumulative areas. Then, we generate a random number ranging from 0 to the total area, and select the face corresponding to that value in the array of cumulative areas.

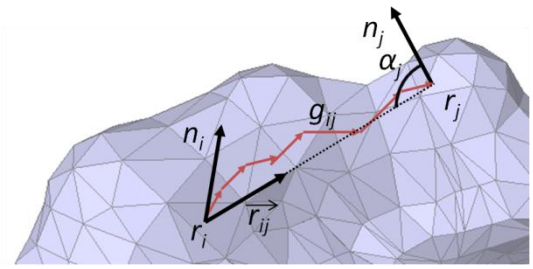

Figure 2: Feature descriptors for a protein binding site. Two sample points (i and j) are shown on the binding site surface. n_i (or n_j) and r_i (or r_j) represent the normal vector and 3D coordinate of the i-th (or j-th) face center, respectively. The geodesic distance g_{ij} is calculated between r_i and r_j, and the angle α_j is calculated between n_j and a unit vector pointing from r_i to r_j.

We further develop an approach based on the representation of shape contexts, originally introduced in [61] for 2D object matching as a feature descriptor for each feature point. Before calculating features, we normalize the scale of binding site mesh by the maximum geodesic distance from all pair-wise distances within each binding site. Assuming binding site A has N faces and M ($M<<N$) sample faces, for face F_i, we use n_i and r_i (see Figure 2) to represent the normal vector and 3D coordinate of the i-th face center, respectively. For a given sample face F_i (i=1,…,M) and another face F_j ($j \neq i$, j=1,…,N) on the binding surface, we first calculate the geodesic distance g_{ij} between r_i and r_j using Dijkstra's shortest path algorithm [62] and angle α_j between n_j and a unit vector pointing from r_i to r_j. Then, a feature descriptor W_i (or shape context) for the sample face F_i is calculated to quantitatively measure the distributions between F_i and the remaining N-1 face centers in logarithmic geodesic distances bins and angles bins, which are set to 10 and 8, respectively. By construction, the feature vector at a given face center is invariant under translation and scaling. In total, we extract M features $\{W_i^A, i \in \{1,\ldots,M\}\}$, and say that binding site A is characterized by its feature descriptor W^A. In the experiments, M is empirically set to 200.

3.1.3 Visual vocabulary construction

The feature descriptors from different binding sites are clustered to construct a vocabulary of visual words, where each word in the vocabulary is quantitatively described by the corresponding cluster. We empirically identify a number of clusters based on the standard square-error partitioning method, *k*-means [63]. In this step, a vocabulary $V = \{V_1, \ldots, V_k\}$ of size k is obtained. Descriptors sharing the same center will be represented by the same word.

Unlike the vocabulary of a text corpus whose size is relatively fixed, the size of a vocabulary for protein binding sites is controlled by the number of feature clusters generated by *k*-means. With a small vocabulary, the visual word is not very discriminative because dissimilar feature points can be mapped to the same visual word. As the vocabulary size increases, the feature is expected to be more discriminative; however, it becomes possible for similar feature points to be mapped to different visual words. As there is no standard way to select the appropriate size of a vocabulary, we empirically select k based on the experiment for various vocabulary sizes (see Section 3.2 Results).

3.1.4 Surface visual words representation

The surface visual words representation consists of two parts: (i) Single-word Frequency Profile (SFP) and (ii) Pair-word Frequency Profile (PFP).

We first compute the frequency of occurrence of each individual visual word for a binding site A. Given a vocabulary V, for each sample face $i \in$ A with descriptor W_i^A,

we calculate the Euclidean distance between W_i^A (i=1,…,M) and each visual word V_j (j=1,…,k). We then associate face i with the index of the visual word with the smallest distance from the vocabulary V.

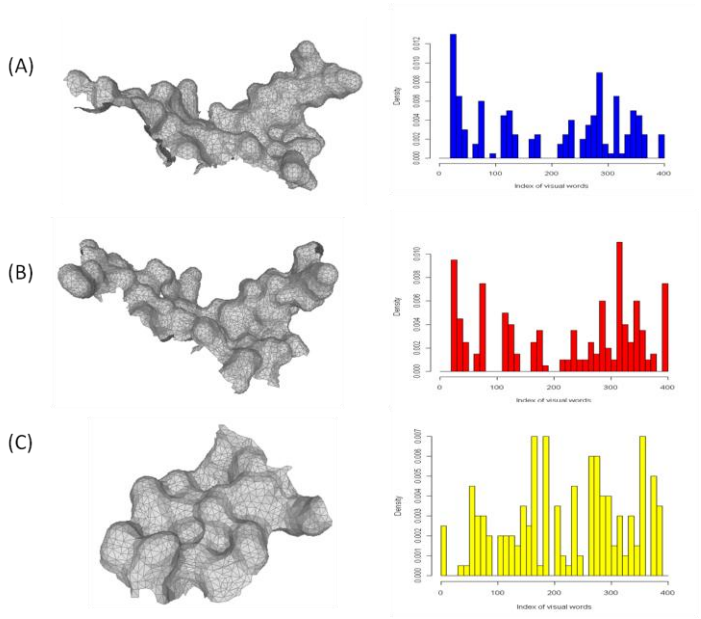

Figure 3: Examples of surface single-word frequency profiles (SFP).(A) Binding site 1m3d_78535_B_78538_C. (B) Binding site 1t60_106525_V_106528_W. (C) Binding site 1cer_29989_R_39901_O.The binding sites (A) and (B) are from same SCOPPI group (ID: d.169.1.6_1) while the binding site (C) is from another SCOPPI group (ID: c.2.1.3_25). The labels of x and y axes of histogram are "Index of visual words" and "Density", respectively.

The single-word frequency profile (SFP) of binding site A is a k×1 vector P^A=(p_1^A,…,p_k^A), which is defined as the frequency of each word in A. An example of the SFP for three binding sites with k = 400 is shown in Figure 3. As these binding sites are from the database SCOPPI [64], we use the same identifier as [50] to name each binding site: <PDB-ID>_<SCOP-domain of the binding site>_<Chain-ID of the binding site>_<SCOP-domain of the binding partner>_<Chain-ID of the binding partner>. The group ID for each binding site is labeled as: <SCOP-family of the binding site

>_<cluster ID within the SCOP family>. In Figure 3, the binding sites 1m3d_78535_B_78538_C and 1t60_106525_V_106528_W are from same SCOPPI group (ID: d.169.1.6_1) while the binding site 1cer_29989_R_39901_O is from another SCOPPI group (ID: c.2.1.3_25). From the figure, we can see that the SFPs of 1m3d_78535_B_78538_C and 1t60_106525_V_106528_W are very similar whereas they are starkly different from the SFP of 1cer_29989_R_39901_O.

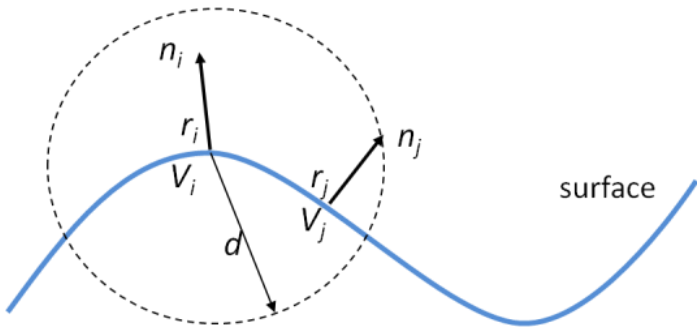

Figure 4: Calculation of a pair-word frequency profile (PFP).Two sample points (*i* and *j*) are shown on the binding site surface. n_i (or n_j) and r_i (or r_j) represent the normal vector and 3D coordinate of the *i*-th (or *j*-th) face center, respectively. The geodesic distance g_{ij} is calculated between r_i and r_j, and the angle α_j is calculated between n_j and a unit vector pointing from r_i to r_j.

The pair-word spatial-sensitive frequency profile (PFP) of binding site A, extended from [65] for texture analysis of images, is based on the idea that the surface of a binding site can be characterized spatially by measuring the local distribution of pairs of visual words that are within a given distance d in a given direction θ for a sample point. Given a sample point with a visual word index V_i, the frequency $P_{d,\theta}(i, j)$ is calculated by accumulating the occurrences of a pair of visual words that have visual word indices of visual word (V_i, V_j) and are located within a distance d in direction θ (see Figure 4). The following measures are computed for each binding site surface:

$$Energy = \sum_{(i,j)} P_{d,\theta}(i,j)^2$$

$$Entropy = \sum_{(i,j)} -P_{d,\theta}(i,j)\log P_{d,\theta}(i,j)$$

$$Homogeneity_1 = \sum_{(i,j)} \frac{P_{d,\theta}(i,j)}{1+|i-j|}$$

$$Homogeneity_2 = \sum_{(i,j)} \frac{P_{d,\theta}(i,j)}{1+|i-j|^2}$$

$$Contrast = \sum_{(i,j)} |i-j|^2 P_{d,\theta}(i,j)$$

$$Correlation = \sum_{(i,j)} \frac{(i-\mu)(j-\mu)P_{d,\theta}(i,j)}{\sigma^2}$$

$$ClusterTendency = \sum_{(i,j)} (i+j-2\mu)^2 P_{d,\theta}(i,j)$$

where $\mu=\Sigma_{(i,j)} iP_{d,\theta}(i,j)$ and $\sigma=\Sigma_{(i,j)}(i-\mu)^2 P_{d,\theta}(i,j)$. Here, the distance d is defined as 50% of the maximum Euclidean distance between the coordinates of all visual words on the binding sites, and the direction θ is defined as the angle between the two corresponding normal vectors. As there is no particular purpose required to retain the θ dependence, we compute the above mentioned measures with θ = 0°, 45°, 90°, 135°, and 180° and take the average over these angles.

3.1.5 Scoring function

For a given query binding site A, the scoring function is used to assign a similarity score of a retrieved binding site B from the database. We first define the score of SFP as follows:

$$S_{SFP}(A,B) = 1 - COS^{-1}\left(\frac{< P^A, P^B >}{\| P^A \| \bullet \| P^B \|}\right),$$

where P^A and P^B are the SFPs for binding sites A and B, respectively. $<P^A, P^B>$ is the inner product of P^A and P^B, and $||P^A||$ and $||P^B||$ are the norms of P^A and P^B, respectively.

In addition, we define a score for PFP as follows:

$$S_{PFP}(A,B) = 1 - \frac{\| T^A - T^B \|}{\arg\max_i (\| T^A - T^i \|)},$$

where T^A and T^B are the PFPs for binding sites A and B, respectively. The distance $\|T^A - T^B\|$ is normalized by the maximum distance between A and the database binding sites T^i (i=1,…,L) where L is the size of database.

Finally, we also look at the ratio of areas between two binding sites A and B. The intuition is that if two binding sites display a significant difference in surface area, they will not be geometrically similar. We define the scoring function regarding area of binding sites as follows:

$$S_{Area}(A,B) = \frac{\min(R^A, R^B)}{\max(R^A, R^B)},$$

where R^A and R^B are areas of binding sites A and B, respectively.

An aggregated similarity function is defined to include the above three scoring functions to provide a final similarity score:

$$S_{BS}(\text{A, B}) = w_1 \times S_{SFP} + w_2 \times S_{PFP} + w_3 \times S_{Area},$$

where w_1, w_2, and w_3 are used to weight the contributions from the three similarity terms.

3.2. Results

To demonstrate the significance of this work, we compare the performance to current methods in terms of accuracy and efficiency. The current methods include an alignment-based method, iAlign [9], which is designed for protein-protein interface comparison based on the local substructures of subunits, and a feature-based method,

Moment Invariants (MI) [50], which describes the geometric shape of a binding site using a feature vector based on moment invariants.

We apply PBSword, iAlign, and MI to a non-redundant dataset from SCOPPI [64] to evaluate classification and retrieval performance for protein-protein binding sites. During the experiment, a query dataset is selected and used to retrieve similar binding sites from the entire database.

To evaluate classification accuracy, we use a general metric, the correct classification rate (CCR), which is defined as follows:

$$\text{CCR} = \frac{\text{The number of correctly classified binding sites}}{\text{The total number of test binding sites}}.$$

In addition, for each query, we identify the top results whose SCOPPI group matches the query's group. The ranks of these "top results" are then accumulated with summary statistics reported.

We also the use AUC (Area Under Curve) in the ROC (Receiver Operator Characteristics) curve [66] to measure how well each method identifies the other binding sites from the query's group. An AUC of 1.0 would correspond to perfect classification, which would rank the binding sites from the same group as the query's before all other binding sites, whereas an AUC of 0.5 would be expected for a random classifier.

3.2.1 Protein binding site classification and retrieval with SCOP 1.69

The dataset used in this experiment, denoted as D_1, is a non-redundant dataset of protein binding sites extracted from SCOPPI 1.69 and has been used to evaluate the

performance of MI. Dataset D_1 consists of 2,819 protein binding sites clustered into 501 groups, as determined in [50]. The query dataset, denoted as Q_1, includes 224 binding sites from 53 groups that are selected from D_1 by applying a structural alignment tool, TM-align [19] to ensure that the similarity score (i.e., TM-score) among the binding sites within one group was greater than 0.45. For a more detailed description of D_1, see [50].

Table 2: CCR of PBSword and MI for Q_1 and D_1.*

TOP	**1**	**5**	**10**	**1%**
PBSword	0.76	0.88	0.91	0.95
MI	0.40	-	-	-

*The result of MI is from caption of Fig.1 in [50].

We tested our method with a vocabulary size (k) of 400 on all protein binding sites from Q_1. As each binding site in D_1 belongs to a group of geometrically similar binding sites, the purpose of our evaluation was to measure how well we can correctly classify members from the same group. For a query binding site, we rank each binding site in D_1 (with the query binding site excluded) based on similarity score. In a perfect scenario, the query binding site would be classified into the same group as the top-ranked binding site. Table 2presents the CCR performance comparison of PBSword and MI for Q_1 and D_1. Intuitively, the optimal accuracy of classification is100% CCR. Our classification results show that when using only the top rank, PBSword achieves a 76% CCR, which is a significant improvement over the 40% CCR reported by MI in its original paper [50]. We also investigated the performance of PBSword by examining up to the top 1% (=28) of ranked results. In these situations, if one protein binding site from the same group as query can be found in a certain range, the query binding site is

regarded as correctly classified. As shown in Table 2, CCRs are 88% and 91% when the top 5 and top 10 ranks are considered, respectively.

Table 3: Summary statistics for best hits from PBSword and MI for Q_1 and D_1.*

	MIN	**1ST Q**	**2ND Q**	**3RD Q**	**MAX**
PBSword	1	1	1	1	587
MI	1	1	2	18.3	1902

*The results of MI are from [50].

Table 3 shows summary statistics for the best hits from PBSword and MI for Q_1 and D_1. The columns "1st Q", "2nd Q", and "3rd Q" correspond to the first, second, and third quartile, respectively. Here, the quartiles are values that divide a set of data into four equal parts. From the table, we can see that the third quartile of PBSword is still 1, as opposed to 18.3 for MI, which represents a significant improvement. In the worst case, PBSword finds a binding site from the same group as the query at rank 587 (top 20%).

Table 4: Summary statistics for the AUC from PBSword and MI for Q_1 and D_1.*

	MIN	**1ST Q**	**2ND Q**	**3RD Q**	**MAX**
PBSword	0.66	0.94	0.99	1	1
MI	0.30	0.94	0.98	0.99	1

*The results of MI are from [50].

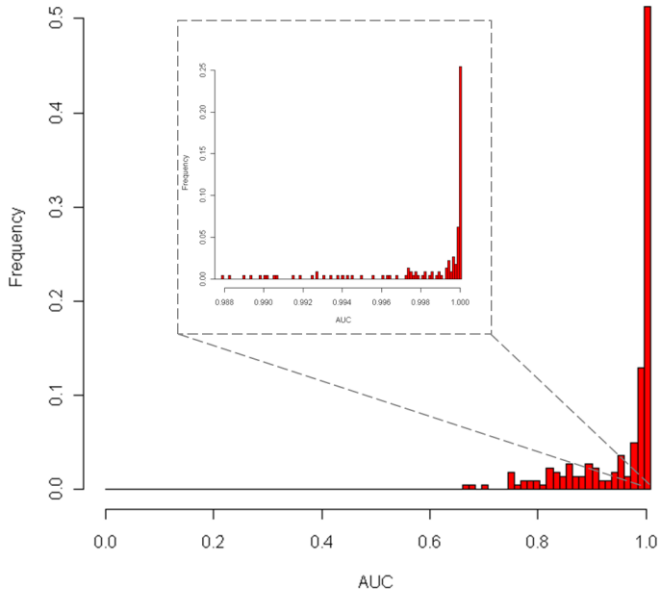

Figure 5: Histogram of AUC for the query dataset Q_1.For easy comparison with Fig. 1(B) in [50] which shows the histogram of AUC of MI on the same dataset, we use similar settings to plot the histogram: the *x* axis is broken into bins of 0.012 and frequency in the *y* axis is normalized with size of Q_1. The insert illustrates the frequency of AUC within the first bin.

Table 4 shows summary statistics for the AUC of PBSword and MI for Q_1 and D_1. From the table, we can see PBSword also out-performs MI in terms of AUC. In the worst case, PBSword can achieve an AUC of 0.66, which is better than a random classifier (0.5). Figure 5 presents the histogram of AUC values for PBSword. With our method, for 25% of the queries, we can identify all the members from the same group without any false positives, which significantly outperforms MI (5%) and improves by 20%.

In addition to comparing the feature-based method MI, we also perform the same experiments using the alignment-based method, iAlign, on Q_1 and D_1. Unfortunately, iAlign cannot always find all the alignments of binding sites from the dataset; hence, to

facilitate the comparison between PBSword and iAlign, we constructed a reduced query dataset, Q_1', by removing those groups from the query dataset that contained the binding sites iAlign could not find. This results in the exclusion of 10 groups (ID: a.2.11.1_3, c.1.9.2_7, c.1.14.1_1, c.48.1.2_2, d.8.1.1_2, d.8.1.1_5, d.19.1.1_36, d.58.33.1_6, d.153.1.4_13, and f.21.1.2_15) from the query dataset Q_1, left 188 binding sites clustered into 43 groups.

Table 5: CCR of PBSword and iAlign for Q_1' and D_1.

TOP	**1**	**5**	**10**	**1%**
PBSword	0.77	0.88	0.92	0.95
iALign	0.82	0.99	1	1

Table 6: Summary statistics for the best hits from PBSword and iAlign for Q_1' and D_1.

	MIN	**1ST Q**	**2ND Q**	**3RD Q**	**MAX**
PBSword	1	1	1	1	587
iAlign	1	1	1	1	6

Table 7: Summary statistics for the AUC from PBSword and iAlign for Q_1' and D_1.

	MIN	**1ST Q**	**2ND Q**	**3RD Q**	**MAX**
PBSword	0.75	0.98	0.99	1	1
iAlign	0.69	1	1	1	1

Table 5 presents the CCR performance comparison between PBSword and iAlign for the reduced query dataset Q_1'. As can be seen, PBSword achieves a 77% CCR, which is relatively close to the 82% CCR of iAlign. We further investigate the CCR performance by examining up to the top 1% of ranked results. By using the top

five results, PBSword can achieve a CCR of 88%, which is comparable to the CCR using top result from iAlign (82%). Thus, on this dataset, instead of performing one-against-all structural alignment using iAlign, we can achieve a similar classification accuracy by checking the top five results (0.1% of the entire data-base of binding sites) generated by PBSword. Table 6 presents the summary statistics for the best hits from these two methods. Though it is noted that the worst rank of the best hit from PBSword is 587, which is worse than iAlign, PBSword still filters out about 80% of the dissimilar binding sites. By performing iAlign on the remaining 20% of binding sites in the database, we can not only achieve a similar classification accuracy as iAlign, but can also save about 80% in computing resources compared to a one-against-all alignment using iAlign. Table 7 shows the summary statistics for the AUC. In the worst case, PBSword obtains an AUC of 0.75, which is better than that of iAlign (0.69). From these experimental results, we can see that PBSword provides an efficient way to filter out geometrically dissimilar binding sites and obtain a short list of similar ones, which can be sent to the existing alignment-based methods for refinement.

3.2.2 Protein binding site classification and retrieval with SCOP 1.75

We also evaluate performance of PBSword using a nor-redundant dataset selected from SCOPPI 1.75, which is denoted as D_2. For the dataset D_1, protein structures can be downloaded from the Protein Quaternary Structure (PQS) server [67]. However, since the PQS server had not been updated since August 2009 and the SCOPPI 1.75 server did not provide detailed information of protein structure, we retrieved the protein structures of D_2 from the DOMMINO server [68] by comparing

protein's PDB ID, SCOP family, sequence, and binding site residues. The dataset D_2consisted of 2,090 protein binding sites clustered into 475 groups. Query dataset, denoted as Q_2, included 249 binding sites from 55 groups, which were selected from D_2 by ensuring that (i) difference of binding site residue number within one group was less than 50% and (ii) TM-score among the binding sites within one group was greater than 0.45. The group members with TM-score less than or equal to 0.45 were discarded from the group. The dataset D_2, though derived from an incremental release of SCOPPI 1.69, shared only 7.4% common binding sites with D_1. This is because for each release of SCOPPI, the non-redundant interface datasets were generated for each family when new structures were added [64]. Dataset D_2 and Q_2 can be downloaded from the PBSword homepage (http://pbs.rnet.missouri.edu/).

Table 8: CCR of PBSword for Q_2 and D_2.

TOP	**1**	**5**	**10**	**1%**
PBSword	0.84	0.93	0.95	0.96

We tested our method with a vocabulary size (k) of 400 on all protein binding sites from Q_2. For a query binding site, we ranked each binding site in D_2 (with the query binding site excluded) based on the similarity score. Our classification results show that when using only the top rank, PBSword achieves an 84% CCR. We also assessed the performance of PBSword by examining up to the top 1% (=21) of ranked results. As shown in Table 8, CCRs are 93% and 95% when the top 5 and top 10 ranks are considered, respectively.

Table 9: Summary statistics for best hits from PBSword for Q_2 and D_2.

	MIN	1ST Q	2ND Q	3RD Q	MAX
PBSword	1	1	1	1	115

Table 9 shows summary statistics for the best hits from PBSword for Q_2 and D_2. From the table, we can see that the third quartile of PBSword is still 1. In the worst case, PBSword finds a binding site from the same group as the query at rank 115 (top 6%).

Table 10: Summary statistics summary for the AUC from PBSword for Q_2 and D_2.

	MIN	1ST Q	2ND Q	3RD Q	MAX
PBSword	0.89	0.99	0.99	1	1

Table 10 shows summary statistics for the AUC of PBSword for Q_2 and D_2. Figure 6 presents the histogram of AUC values for PBSword. With our method, for 34% of the queries, we were able to identify all the members from the same group without any false positives.

3.2.3 Efficiency

We measured the average response time for 188 binding sites from Q_1' to evaluate the efficiency of PBSword. The experiments were conducted on a Linux Fedora server with AMD Opteron dual-core 1000 series processors and 8GB RAM. With PBSword, each query took0.31 second for one-against-all score calculations. For iAlign, however, each query took1,016 seconds to scan the entire database D_1. Note that we have excluded the CPU time spent on generating the surface and calculating the visual words, as it can be performed off-line during the preprocessing stage. An efficiency comparison with I2ISiteEngine, another alignment-based method, was not performed in this dissertation, though we note that I2ISiteEngine has been evaluated elsewhere [9]

where experimental results showed that iAlign can achieve about an 89-fold speedup over I2ISiteEngine.

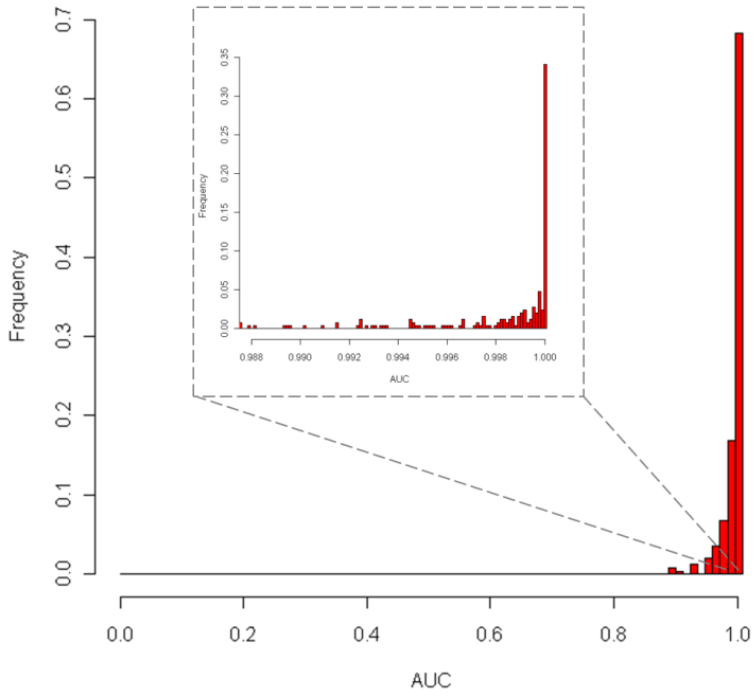

Figure 6: Histogram of AUC for the query dataset Q_2.The *x* axis is broken into bins of 0.012 and frequency in the *y* axis is normalized with size of Q_2. The insert illustrates the frequency of AUC within the first bin.

3.2.4 Parameter selection

The performance of our method is heavily dependent on the vocabulary size (*k*). To study the influence of k on the results, we carried out three experiments on D_1 with *k*=200, 400, and 600. The corresponding CCR of top rank is 69%, 76%, and 74%, respectively. As such, we selected *k*=400 as our default settings.

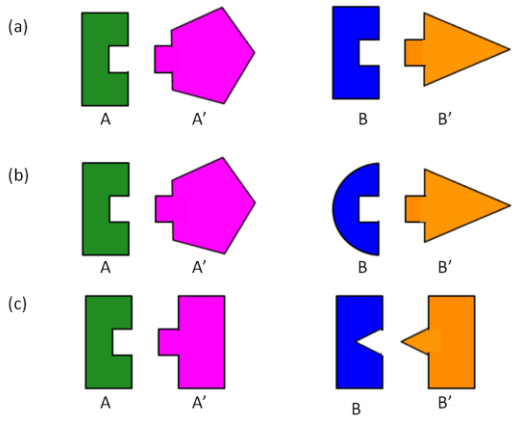

Figure 7: Schematic representations of the three case studies. The protein chains A (green) and A' (magenta) are from the query protein while B (blue) and B' (orange) are from the database protein. (**a**) A and B are from same SCOP family whereas A' and B' are from different families. The binding site surface on A is geometrically similar to that on B. (**b**) A, A', B, and B' are all from different SCOP families but the two binding site surfaces are similar. (**c**) A, A', B, and B' are all from same SCOP family, but two interactions have dissimilar binding surfaces which correspond to the different functions carried out by each protein complex.

3.3. Case Studies

We have shown that PBSword can identify geometrically similar protein binding sites from the same SCOP family based on surface shape features. As the molecular shape has long been recognized as a key factor in protein-protein interactions, we further investigate whether PBSword can discover non-trivial biological connections among proteins from a geometric perspective. In this section, we first considered whether our approach can help investigate the relationships between the geometrically similar shapes of protein binding sites participating in an interaction and the functions carried out by the interactions. Specifically, we used our approach to first retrieve geometrically similar binding sites for a 'seed' binding site A, and then selected the top-ranked binding site B to analyze the functional similarity between the corresponding proteins. The binding partners of A and B are denoted as A' and B', respectively. We considered two

cases: (i) A and B are from same SCOP family, whereas A' and B' are from different families; and (ii) A, A', B, and B' are all from different SCOP families. We then studied the relationships between the shapes of protein binding sites and functional diversity within a SCOP family. Intuitively, proteins from the same family are expected to be structurally similar and have related functions. Discovering a protein binding site from such a family would not be very biologically significant, since the binding sites from the structurally similar proteins are expected to be similar and clustered together. What would be more interesting would be the discovery of two protein binding sites which are from the same family, but have dissimilar geometric shapes and different molecular functions. For this study, we considered another case (iii) where A, A', B, and B' are from the same SCOP family but belong to different functional groups. In this case, the binding site B is not the top-ranked, but B has the highest ranking result from the same family. The schematic representations for the three cases are shown in Figure 7.

To study the structure-function relationship based on the geometric features of the binding site, we selected another non-redundant dataset of protein-protein interfaces generated by all-against-all interface comparisons of protein complexes from PDB using I2ISiteEngine [69]. This dataset, denoted as D_3, consisted of 604 protein-protein interfaces clustered into 59 groups. In each group, interface members shared a sequence identity of less than 50%. As the binding site in this dataset is defined on the protein chain, we used<PDB-ID>_<Chain ID of the binding site><Chain ID of the binding site partner> to represent a protein binding site.

In the first case, the protein binding site 4sgb_EI was selected as a query to retrieve similar binding sites from D_3 with our method. The top result was 1sgd_EI from the same group (see Figure 8A). The protein chains 4sgb_E and 1sgd_E are all from the prokaryotic pro-teases family, while the chain 4sgb_I and 1sgd_I belong to the plant proteinase inhibitors family and ovomucoid domain III-like family, respectively. The alignment results of TM-align and iAlign are shown in Figure 8B and Figure 8C, which show that two protein chain pairs can be well aligned. The TM-score of the two aligned protein structures was 0.99, and the IS-score of the aligned interfaces was 0.64, which is statistically significant [9,70]. The binding site surfaces with the chemical environments (i.e., electrostatic potential), shown in Figure 8D, are also similar. In this case, 4sgb_I and 1sgd_I belong to different families, but can bind to the equivalent sites of homologous 4sgb_E and 1sgd_E, respectively. This type of interactions, also known as convergently evolved interaction motifs, would be very biologically significant as it can be used to discover rules for interactions and design ligand [71].

In the second case, we select protein binding site 1b99_AD. In D2, 1b99_AD has been classified into a group including 1l0o_AB, 1l3b_AD, 1e7p_AD, 1gtt_BC, and 1iun_AB [69]. The top result from PBSword is 1tmk_BA, which is from another group. The proteins 1b99 and 1tmk are from the nucleoside diphosphate kinase family and nucleotide and nucleoside kinases family, respectively. However, both proteins are kinase and mainly composed of α-helices and β-strands (see Figure 9A). The sequence alignment between the 1b99 and 1tmk binding chains shows low similarity. The global structure alignment using TM-align is given in Figure 9B, which shows that the residues from the binding chains 1b99_D and 1tmk_A cannot be aligned together. Hence, we

obtain a TM-score of 0.25, which is not statistically significant. In contrast, iAlign can find partial residue correspondences between the two interfaces (see Figure 9C) with an IS-score of 0.24 (p-value = 0.42×10^{-2}). As a comparison, the binding site surfaces of 1b99_AD and 1tmk_BA are shown in Figure 9D. Despite the sequence and global structure conservation being low, the binding site surfaces of 1b99_AD and 1tmk_BA are geometrically similar, which is effectively captured by PBSword.

In the third case, we select the family of C-type lectin domains (SCOP ID d.169.1.1). In 1993, Drickamer first classified proteins in this family into seven groups and showed that such classification can capture functional similarities between proteins [72]. The classification was later revised [73,74], and currently this family has 17 groups. The spatial relationships of protein binding sites in this family were analyzed in [75], and significant diversity was found. In this case, protein binding site 1k9i_68344_B68349_G from this family, which is not included in the query dataset Q_1, was selected and compared to the sites from D_1. In D_1, binding site 1bv4_42381_B42383_C (see Figure 10A) from the same family is recognized as having a similar interface type and is classified into the same group (ID: d.169.1.1_21) according to the sequence and structure alignment methods employed by SCOPPI. With our approach, the binding site 1bv4_42381_B42383_C was ranked as 2,334. The structure alignments of TM-align and iAlign are shown in Figure 10B and Figure 10C, respectively. Since these two bind sites belong to same family, there is a significant similarity from TM-align (TM-score = 0.86). However, the IS-score from iAlign is only 0.11 (p-value = 0.84). The binding site surfaces, shown in Figure 10D, are remarkably dissimilar. The functional groups of 1k9i and 1bv4 are also different. According to the

classification of functional groups in [74], protein 1k9i belongs to the asialoglycoprotein and DC receptors group while protein 1bv4 belongs to the collectins group. From this case, we can see that the surface dissimilarity detected by PBSword can be potentially used to discover functional diversity between two proteins from the same family.

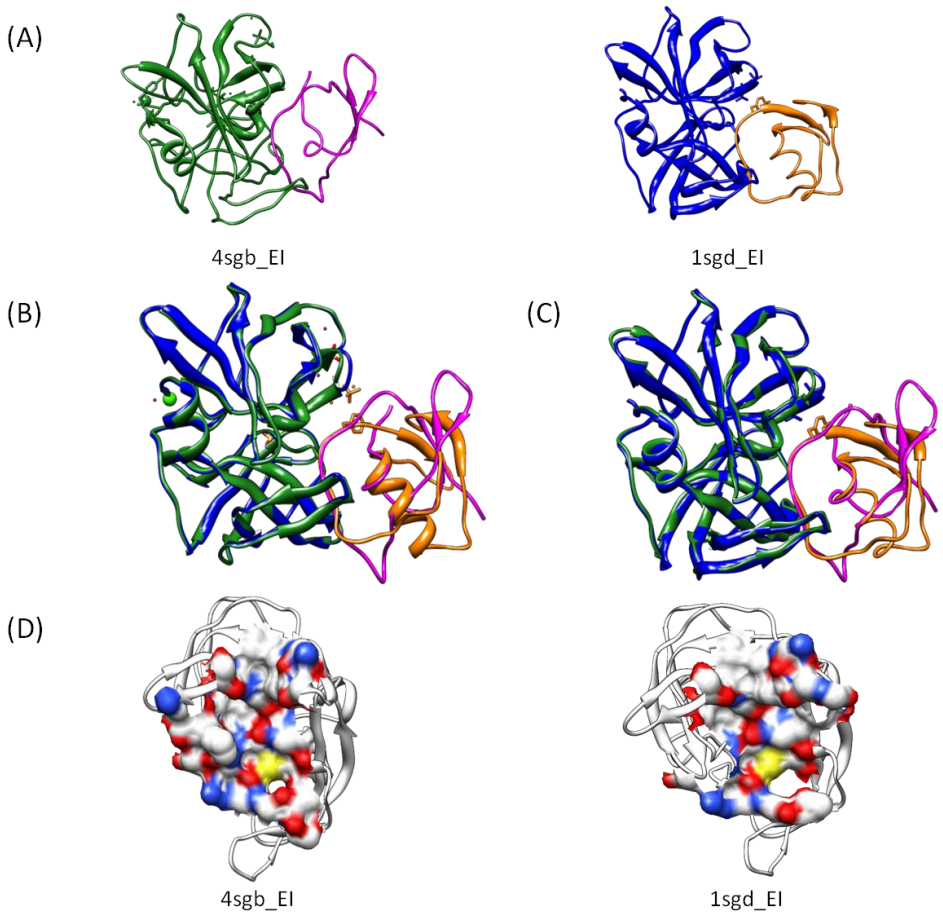

Figure 8: Case study of protein binding site 4sgb_EI.(**A**) Protein structures of 4sgb_EI (chain E-green, chain I-magenta) and 1sgd_EI (chain E-blue, chain I-orange). The protein chains 4sgb_E and 1sgd_E are both from the prokaryotic proteases family, while the chains 4sgb_I and 1sgd_I belong to the plant proteinase inhibitors family and ovomucoid domain III-like family, respectively. (**B**) Global structure alignment of 4sgb_EI and 1sgd_EI using TM-align (TM-score = 0.99). (**C**) Local substructure alignment of 4sgb_EI and 1sgd_EI using iAlign (IS-score = 0.64). (**D**) Binding site surfaces with the chemical environments (i.e., electrostatic potential) of 4sgb_EI and 1sgd_EI. In this case, 4sgb_I and 1sgd_I belong to different families, but can bind to the equivalent sites of homologous 4sgb_E and 1sgd_E, respectively. This type of interaction is also known as convergently evolved interaction motifs.

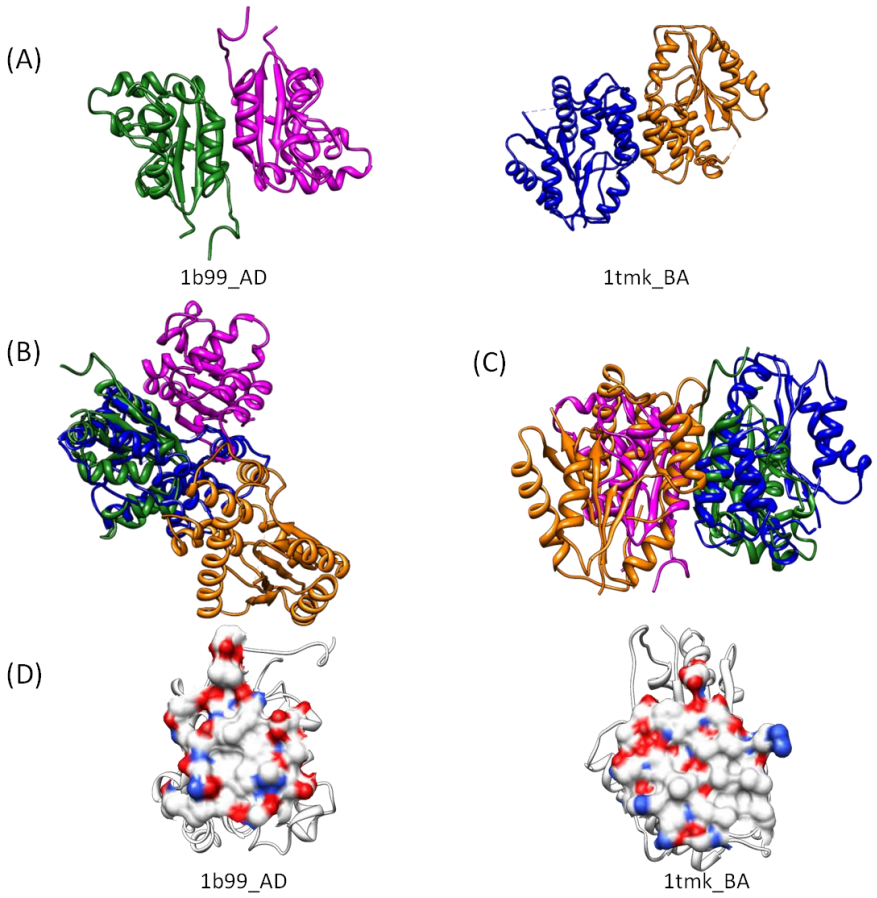

Figure 9: Case study of protein binding site 1b99_AD. (**A**) Protein structures of 1b99_AD (chain A-green, chain D-magenta) and 1tmk_BA (chain B-blue, chain A-orange). The proteins 1b99 and 1tmk are from the nucleoside diphosphate kinase family and nucleotide and nucleoside kinases family, respectively. (**B**) Global structure alignment of 1b99_AD and 1tmk_BA using TM-align (TM-score = 0.25). (**C**) Local substructure alignment of 1b99_AD and 1tmk_BA using iAlign (IS-score = 0.24). (**D**) Binding site surfaces with the chemical environments (i.e., electrostatic potential) of 1b99_AD and 1tmk_BA. In this case, despite the global structure conservation being low, the binding site surfaces of 1b99_AD and 1tmk_BA are geometrically similar, which is effectively captured by PBSword.

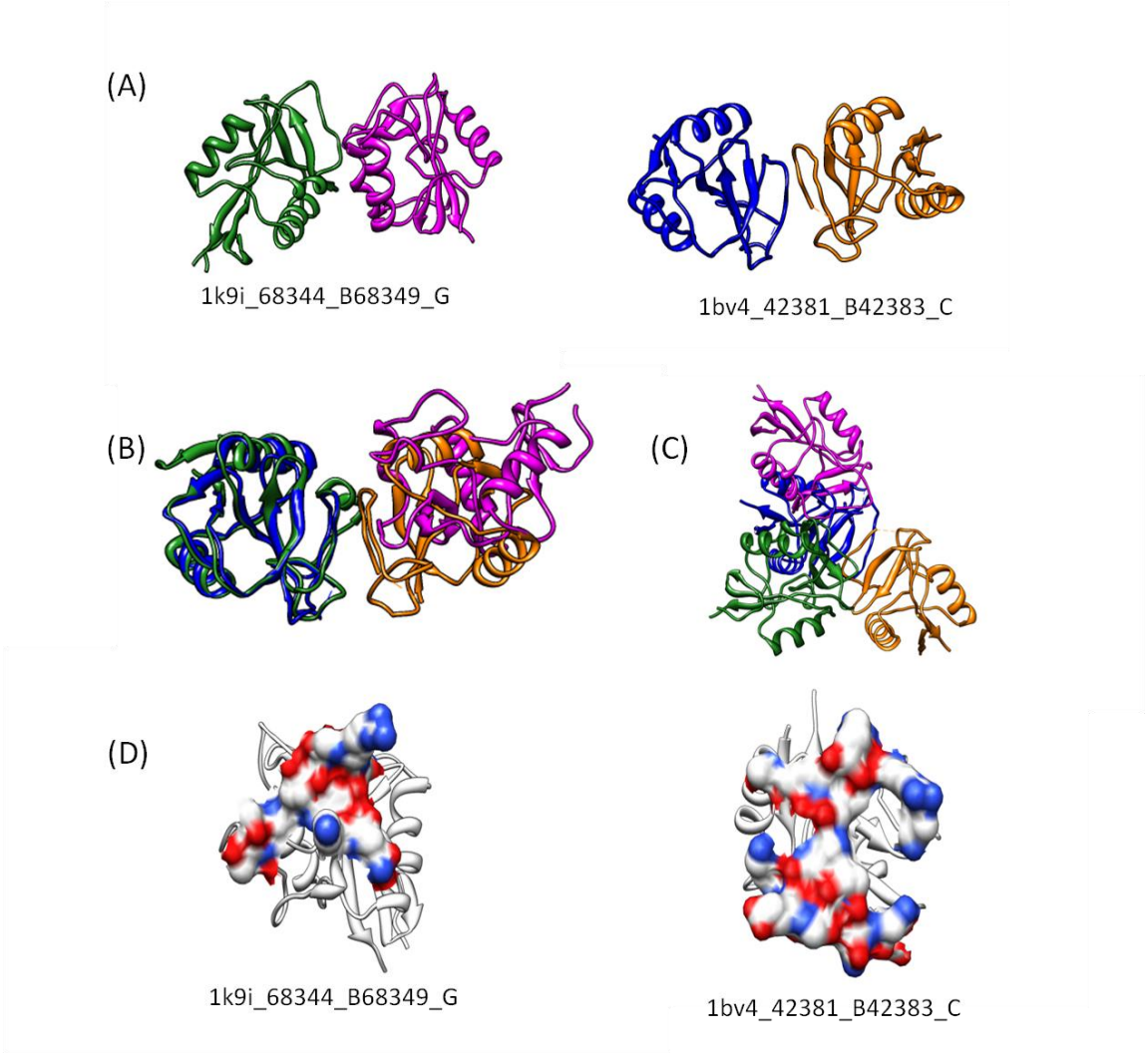

Figure 10: Case study of C-type lectin domains.(**A**) Protein structures of 1k9i_68344_B68349_G (chain B-green, chain G-magenta) and 1bv4_42381_B42383_C (chain B-blue, chain C-orange). Both proteins are from the family of C-type lectin domains (**B**) Global structure alignment of 1k9i_68344_B68349_G and 1bv4_42381_B42383_C using TM-align (TM-score = 0.86). (**C**) Local substructure alignment of 1k9i_68344_B68349_G and 1bv4_42381_B42383_C using iAlign (IS-score = 0.11). (**D**) Binding site surfaces with the chemical environments (i.e., electrostatic potential) of 1k9i_68344_B68349_G and 1bv4_42381_B42383_C. In this case, despite the proteins being from same family, the binding site surfaces are remarkably dissimilar and belong to different functional groups.

CHAPTER FOUR

ACCURATE PROTEIN BINDING SITEALIGNMENT

In this chapter, PBSalign (Protein-protein Binding Site alignment) is introduced for sequence-independent local alignment of protein binding sites. One application of PBSalign is to perform accurate alignments for a selected list generated by PBSword (see Chapter 3) screening on a binding site database. Hence, PBSalign and PBSword can work together to provide an efficient and accurate approach for large-scale binding site comparison.

4.1. Methods

The framework of PBSalign is shown in Figure 11. The input of PBSalign is a pair of PPIs: I_A={BS_A^1, BS_A^2} and I_B={BS_B^1, BS_B^2}. Each protein-protein interface (PPI) consists of two binding sites (*BS*) which could come from the same or different fold(s). Without loss of generality, we assume BS_A^1 and BS_B^1 are binding site pairs for comparison and use BS_A and BS_B to represent these two sites in the following sections. During the alignment, BS_A will be fixed, and BS_B will be rotated and translated towards BS_A as a rigid body. The outputs include residue correspondences and similarity score of alignment. Similar as the global structure alignment, PBSalign aims to determine an alignment between residues of two given binding sites such that functional and evolutionary relationships between them can be identified.

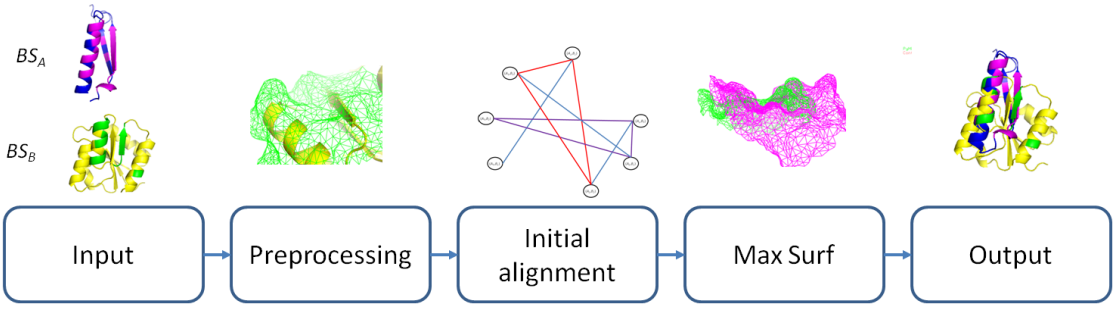

Figure 11: Framework of PBSalign.

PBSalign mainly consists of three steps (see Figure 11): (i) preprocessing, (ii) initial alignment, and (iii) MaxSurf. In the preprocessing step, a binding site surface is generated and various surface properties are calculated. In the initial alignment step, similarity of properties is used to find potential correspondences between two binding site surfaces and a list of seed alignments is generated by the correspondences. Finally, the seed alignments are refined using MaxSurf, an algorithm developed to find maximal overlapping surface of two binding sites while minimizing RMSD (Root Mean Square Deviation) of alignment.

4.1.1 Preprocessing

The workflow of preprocessing is illustrated in Figure 12, which includes (i) surface generation, (ii) properties calculation, and (iii) feature point and region selection.

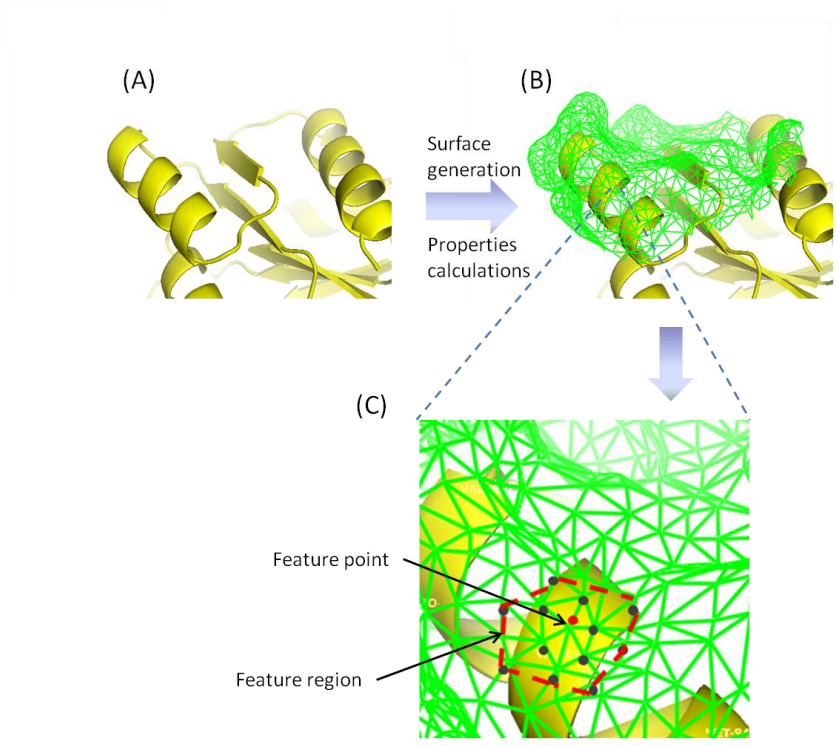

Figure 12: Illustration of the preprocessing method of PBSalign. From the input protein binding site structure (**A**), a triangulated mesh (**B**) is first generated using the MSMS program and various properties are calculated. (**C**) Feature point, represented using point in red color, is selected based on the distance to the binding site residue and feature region, which is growing around the feature point with radius r=4Å. For clarity, only one feature point and region is shown.

Surface Representation

For a given protein complex, we use the MSMS program [59] to generate a triangulated mesh for each of its interacting subunits and set the density and probe radius to 1.0 point/Å^2 and 1.4 Å, respectively. Since we are only interested in the binding regions, for each protein mesh, we retain only those surface points that are within a distance cutoff from the surface of its binding partner. A triangle is selected when its three vertices are all retained in the interaction region. We represent binding site surface as a connected and non-directed graph G=(V, E) where each node $v \in V$ represents a

surface point and E includes all and only the edges (v_i, v_j) such that point v_i and v_j are adjacent on the surface.

Surface Properties

Each node $v \in V$ is associated with a local value of three properties: shape index, $si(v)$, electrostatic potential, $esp(v)$, and hydrophobicity, $hyd(v)$. The shape index of v, $si(v)$, is defined using the maximum (k_1) and minimum (k_2) local curvature, which is given as follows [76]:

$$si(v) = \frac{1}{2} - \frac{1}{\pi} arctan \frac{k_1(v) + k_2(v)}{k_1(v) - k_2(v)}.$$

It takes a value in the interval [0, 1] which is further divided into nine categories each corresponding to a well-known shape, such as dome and saddle [76]. The initial values of potential in v, $esp(v)$ and hydrophobicity, $hyd(v)$, are calculated using the software package VASCo [77], which are standardized and discretized into six categories by PBSalign.

Feature Point and Region

To find potential correspondences between two binding sites, we should specify some feature points on the surfaces. In PBSalign, a feature point corresponds to a binding site residue and is defined as the surface point which has the smallest Euclidean distance to the C_α atom of the corresponding residue. In the following sections, we will use the feature point and binding site residue interchangeably.

For a feature point v_i, we grow a feature region R_i which is centered at the feature point and covers surface points with Euclidean distance < 4Å (see Figure 12). The feature region can be represented using

$$R_i=\{\mathbf{c}(R_i), \mathbf{n}(R_i), \mathbf{s}(R_i)\},$$

where $\mathbf{c}(R_i)$ represents the coordinates of feature point, $\mathbf{n}(R_i)$ is the normal vector of feature point, and a region signature set $\mathbf{s}(R_i)$ contains three signatures of the region R_i for the shape index, electrostatic potential, and hydrophobicity properties, respectively. Each signature is represented by a histogram and a region signature set has a collection of all histograms. The number of histogram bins is set to the discretization categories for the characteristics of the three signatures, which are empirically set to nine, six, and six for *si*, *esp*, and *hyd*, respectively. The above process is repeated for each binding site residue.

4.1.2 Initial Alignment

We use the feature regions to match the binding sites BS_A and BS_B by seeking a set of common feature regions on two binding sites and an alignment transformation that brings BS_B close to BS_A. For each feature region on BS_A, we first find all corresponding feature regions on BS_B and then extract subsets of consistent region correspondences. The initial correspondence set is then filtered by several pruning algorithms, to be discussed shortly, and the final set of matching regions is found using a maximal clique detection algorithm [78]. In this step, we keep all the consistent sets of matching feature regions, which are then used as seed alignments for further refinement by the next step, MaxSurf. An overview of initial alignment is illustrated in Figure 13.

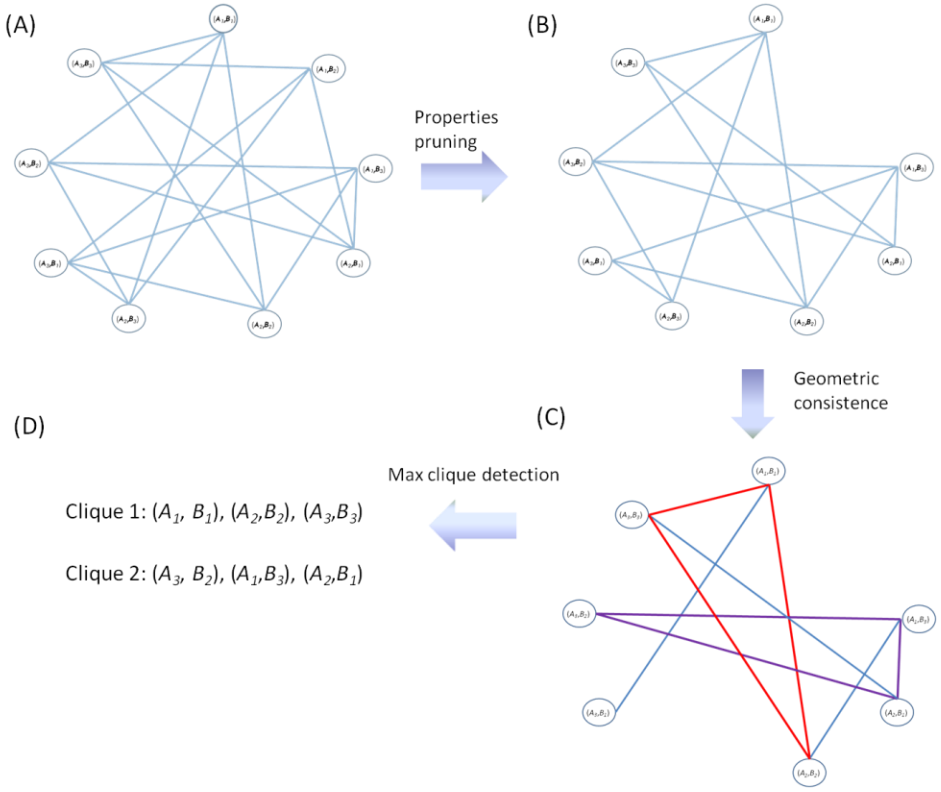

Figure 13: Example of the initial alignment method of PBSalign. Suppose the binding site BS_A and BS_B have three feature points, denoted as $\{A_i\}$ and $\{B_i\}$, i=1,..,3, respectively. (**A**) A product graph is generated. (**B**) After applying the properties pruning, vertices (A_2, B_2), whose region properties are incompatible, is removed from the graph. (**C**) Geometric consistence is used to filter out invalid correspondences and maximum clique is detected (thick lines in red and magenta colors). (**D**) Residue correspondences or seed alignments are generated.

Product Graph

Given the two binding sites, BS_A and BS_B, and their graph representations, G_A=(V_A, E_A)and G_B=(V_B, E_B), the product graph of G_A and G_B, G_P=(V_P, E_P), is constructed by inserting every pair of feature points, $v_i \in V_A$ and $v_j \in V_B$, from two binding sites into its vertices set V_P. Edges are drawn between every two vertices if they do not share a common feature point.

Properties Pruning

For each node of V_P, correspondence between two feature points is established through their compatibility of associated regions. Two feature regions are compatible if a certain similarity is observed in their region properties. Let p = (A, B) be a potential region correspondence (i.e., a node in the product graph G_P) between the binding sites BS_A and BS_B. We assess the compatibility of p by comparing the region signatures of A and B using the following similarity score of two feature regions:

$$S_R(p) = 1 - COS^{-1}\left(\frac{<\boldsymbol{s}(R_A), \boldsymbol{s}(R_B)>}{\|\boldsymbol{s}(R_A)\| \cdot \|\boldsymbol{s}(R_B)\|}\right).$$

Here, $\mathbf{s}(R_A)$ and $\mathbf{s}(R_B)$ are the signature for regions A and B, respectively. $<\mathbf{s}(R_A), s(R_B)>$ is the inner product of $\mathbf{s}(R_A)$ and $\mathbf{s}(R_B)$, and $\|\mathbf{s}(R_A)\|$ and $\|\mathbf{s}(R_B)\|$ are the norms of $\mathbf{s}(R_A)$ and $\mathbf{s}(R_B)$, respectively. A region correspondence p is considered further only if $S_R(p) > \varepsilon_R$ (=0.6) holds true; otherwise we discard it from the product group G_P. In PBSalign, similarity scores of properties shape index, electrostatic potential, and hydrophobicity are applied sequentially to filter out incompatible feature points or regions correspondences.

Geometric Consistency

Because the initial set of correspondences after the properties pruning might be quite large, we further examine the geometric consistency and relational constraints of two connected nodes in G_P to remove outliers. These processes make the next step of maximal clique detection run faster, but do not affect the correctness of our algorithm.

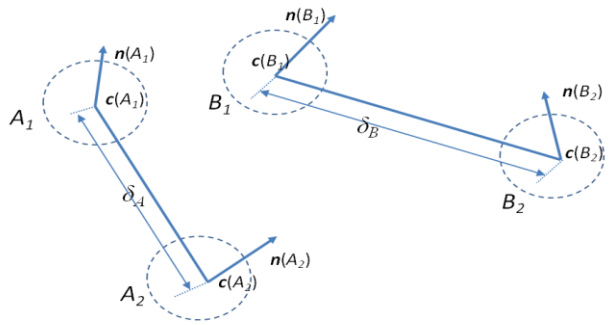

Figure 14: Example of geometric consistency. A_1(or A_2) and B_1(or B_2) are feature regions from binding site A and B. **c** and **n** represents coordinate and normal vector of feature point.

We call a pair of region correspondences, e.g.,p_1 = (A_1, B_1) and p_2 = (A_2, B_2), geometrically consistent (see Figure 14) if the Euclidean distance between the feature points' position δ_A=||**c**(A_1)−**c**(A_2)|| and δ_B=||**c**(B_1)−**c**(B_2)|| are of similar length, i.e., $|\delta_A-\delta_B|<\varepsilon_d$ (=2Å based on empirical observations), and the angles between corresponding pairs of vectors $\angle$(**n**(A_1),**n**(A_2))and $\angle$(**n**(B_1), **n**(B_2)) do not differ more than a certain threshold ε_α(=30° based on empirical observations). In the product graph G_P, edges between p_1 and p_2 are removed if the geometric consistence does not hold.

Maximal Clique Detection

Having applied all the filters, the size of the potential correspondences is reduced so that it is possible to search for cliques in it. We use the algorithm of Bron and Kerbosch [78] to find all cliques with a minimum clique size of three. For each clique we generate a rigid body transformation based on all region correspondences, and we then calculate RMSD and the number of aligned surface feature points (N_p) of this transformation. All the cliques are sorted according to the ratio of RMSD/N_p in ascending order and up to N_{seed} (=20) cliques are selected as initial alignments for further

refinements. The binding site BS_B is then transformed and rotated towards BS_A using the results of the initial alignments.

4.1.3 MaxSurf

In this step, each seed alignment from the previous step is further refined by searching maximal surface overlapping through an iterative method, ICP (Iterative Closest Point) [79]. The final alignment is generated by a refinement process.

Iterative Closest Point (ICP)

The outputs of the initial alignments may not be the optimal transformation, but provide a coarse initialization for ICP to refine the rigid transformation. After applying ICP to the binding sites, the output is used to transform the binding site BS_B towards BS_A. As the ICP algorithm can only provide a local minimum of alignment error, the binding site pairs are sent to the next step for further refinement.

Refinement

In this step, we refine the matched binding site surfaces on the basis of the results recognized in the previous stage. For each seed alignment, we utilize the resulting product graph from the previous step. Secondly, we calculate the Euclidean distance between each residue on the binding site BS_A and its nearest neighbor from the binding site BS_B, then sort the distances in descending order and select 70^{th} percentile of the observed distance as a cutoff to prune feature point correspondences (vertices from the graph). Secondary structure of a residue is optionally applied in this step, which is calculated using C_α coordinate of five neighbor residues [19]. Finally, we utilize

maximum clique detection [80] to obtain a set of residue correspondences whose C_α atom distance is less than 4Å.

The output of maximum clique detection consists of a string of aligned residue pairs which are sequentially ordered from their N to their C terminal. We further extend the alignments to include as many unaligned binding site residues as possible using a greedy approach. The fixed binding site, BS_A, is selected as our starting point. We define a fragment as a set of at least two adjacent residues in the binding site BS_A. For fragment F, let F_h denote the first residue and F_t denote the last residue of the fragment. The extension procedure for the fragment F is described as follows. First, we begin searching unaligned residues from F_h towards the N terminal. An unaligned residue is selected (if any) and the distance between its nearest unaligned residue from the BS_B is calculated. If the distance is < 4Å, the residue pairs are marked as aligned and the next unaligned residue towards the N terminal is selected until an aligned residue or beginning of binding sites is approached. Second, a similar searching process is started from F_t towards the C terminal. Finally, we use the q-Score [81] to rank each valid seed alignment and select the top one as the final output. The valid alignment of PBSalign means the residue correspondence is unique and sequential.

4.1.4 Time Complexity

For clear description, we define the number of feature points on the binding site as m. In the preprocessing step, the properties are calculated for m feature points, which needs a complexity of O(m). In the second step, we define the number of correspondences of each feature point as n for each feature point. Hence, we need to

perform an n comparison of signatures for pruning and the complexity is O(mn). We define g as the number of vertices of the product graph, and the time complexity for detecting maximal cliques is O($3^{g/3}$) [82]. In the third step, ICP is implemented using a k-d tree [83], and its time complexity is O($b_A \log b_B$) where b_A and b_B are the numbers of vertices on the binding site surface BS_A and BS_B, respectively. It can be seen that the total time complexity of first step is O(mn)+O($3^{g/3}$)+O($b_A \log b_B$).

4.2. Results

In this section, we compare performance of PBSalign and the concurrent methods, which have publicly accessible software packages, including I2I-SiteEngine [52], iAlign [9], and Galinter [10]. Since the performance comparisons of Galinter/I2I-SiteEngine and I2I-SiteEngine/iAlign were conducted somewhere else [9,10] and iAlign had the best performance, we selected iAlign for benchmarking the performance of PBSalign. The datasets consisted of homologous and non-homologous binding sites, as well as different types of protein complexes:

•The first dataset, denoted as D_1, is composed of homologous PPIs which was originally constructed to evaluate performance of iAlign [9].

•The second dataset, denoted as D_2, is taken from Table 1 in [43], which was used to study structurally similar binding sites coming from different protein folds (i.e., non-homologous).

In the following sections, we used<PDB-ID>_<Chain ID of the binding site><Chain ID of the binding site partner> to represent a binding site.

Similar to the structural alignments, the performance comparison of binding sites alignments could be based on alignment quality or alignment accuracy. The alignment quality was evaluated using the geometric match measures, such as the RMSD of the superimposed structures. While the alignment accuracy, also known as the discrimination problem, measures the accuracy with which an alignment method classifies a pair of protein structures into a similar/dissimilar category [84,85] defined using the similarity of folds (e.g., SCOP [86]) or PPI (e.g., SCOPPI [64] and SCOWLP [87]). However, recent studies [14,43] show that some proteins or binding sites from different folds may have very similar structures. Hence, we only compare different alignment methods based on the alignment quality.

When evaluated using the geometric match, the goal of structural alignment is to minimize the RMSD of the aligned region. However, as the RMSD depends on the number of aligned residues, the RMSD values associated with alignments of different lengths cannot be compared. To overcome this issue, we used the geometric match measures which simultaneously consider both factors, such as similarity index (SI), match index (MI), and structural alignment score (SAS). These measures, which have also been used in a comprehensive evaluation of protein structure alignment method [88], are defined as follows:

$$SI = \frac{RMSD \times \min(L_A, L_B)}{N_e},$$

$$SAS = \frac{RMSD \times 100}{N_e},$$

$$MI = 1 - \frac{1+N_e}{\left(1+\frac{RMSD}{\omega_0}\right)(1+\min(L_A, L_B))},$$

where N_e is the number of aligned residues, ω_0 is a normalizing factor and set to 1.5 [88], and L_A and L_B are the lengths of binding site BS_A and BS_B. The units of SI and SAS are Å. MI takes values between 0 and 1. For these measures, lower values correspond to better alignments.

4.2.1 Database D_1

The dataset D_1 consists of biologically related PPIs. Here, a PPI pair is said to be biologically related if they are from same SCOP superfamily and share a certain level of similarity. Hence, PPI pairs from this dataset are homologous. For details about selecting PPI pairs, see [9].

For a given PPI pair, $I_A=\{BS_A^1, BS_A^2\}$ and $I_B=\{BS_B^1, BS_B^2\}$, iAlign first considers two possible ways of alignment. One is to align BS_A^1 to BS_B^1 and BS_A^2 to BS_B^2, and the other is to align BS_A^1 to BS_B^2 and BS_A^2 to BS_B^1. Then, iAlign selects the one whose score is the best. The final output of iAlign consists of two lists of residue correspondences, each of which is related to a pair of binding sites from different subunits. In contrast to iAlign, PBSalign is designed specifically for binding site comparison, which can designate binding site pairs for comparison. Hence, the problem is how to select one binding site pair aligned by iAlign and compare its alignment with that of PBSalign. Our solution is to pick the binding site pair which has lower SAS score to form a testing dataset for PBSalign. During the experiments, the geometric match measures are calculated on the same binding site pairs.

Table 11: Average (standard deviation) values of SI, SAS, and MI with iAlign and PBSalign for dataset D_1.

METHODS	SI	SAS	MI
iAlign	2.45(0.88)	7.15(3.25)	0.62(0.09)
PBSalign	2.12(0.76)	6.23(3.04)	0.61(0.10)

A comparison of the SI, SAS, and MI values obtained by iAlign and PBSalign is summarized in Table 11. As can be seen, PBSalign results in structural alignments with better SI, SAS, and MI values in comparison to iAlign. The detailed distributions of SI, SAS and MI values are given in Figure 15. We also tested the statistical significance of the observed difference in the mean values of SI, SAS, and MI using the paired t-test. The difference in the mean values of SI, SAS, and MI was found to be statistically significant (p-value $<<0.001$).

We further analyzed the relative improvement in various geometric match measures obtained by PBSalign or iAlign. The measure difference is defined as (measure(PBSalign) – measure(iAlign)). For example, the measure difference of SI is given as dSI=(SI(PBSalign) – SI(iAlign)). Similarly, we defined the difference for SAS (dSAS) and MI (dMI). The detailed distribution of dSI, dSAS, and dMI values are shown in Figure 16.

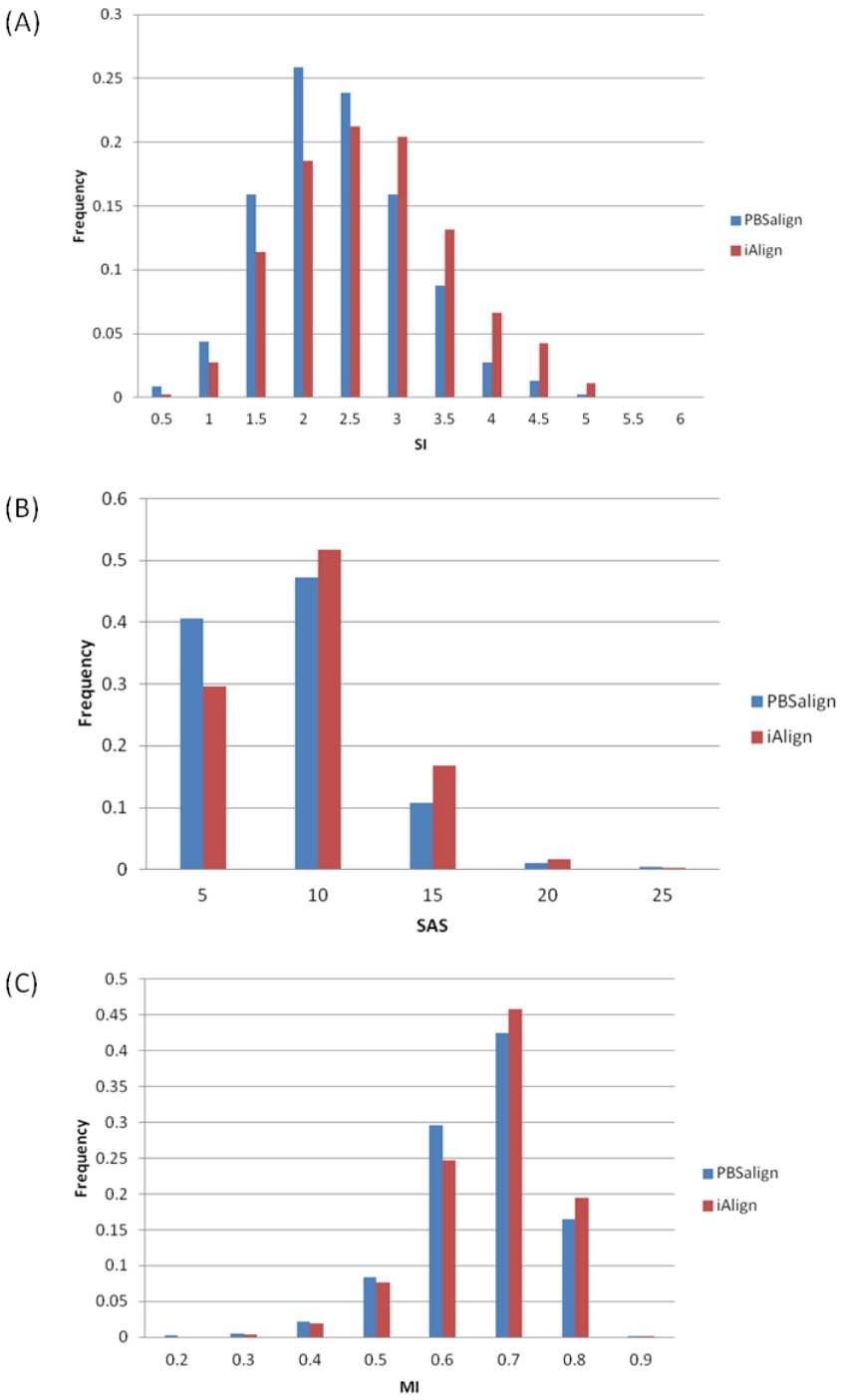

Figure 15: Histogram of (**A**) SI, (**B**) SAS, and (**C**) MI for dataset D_1.

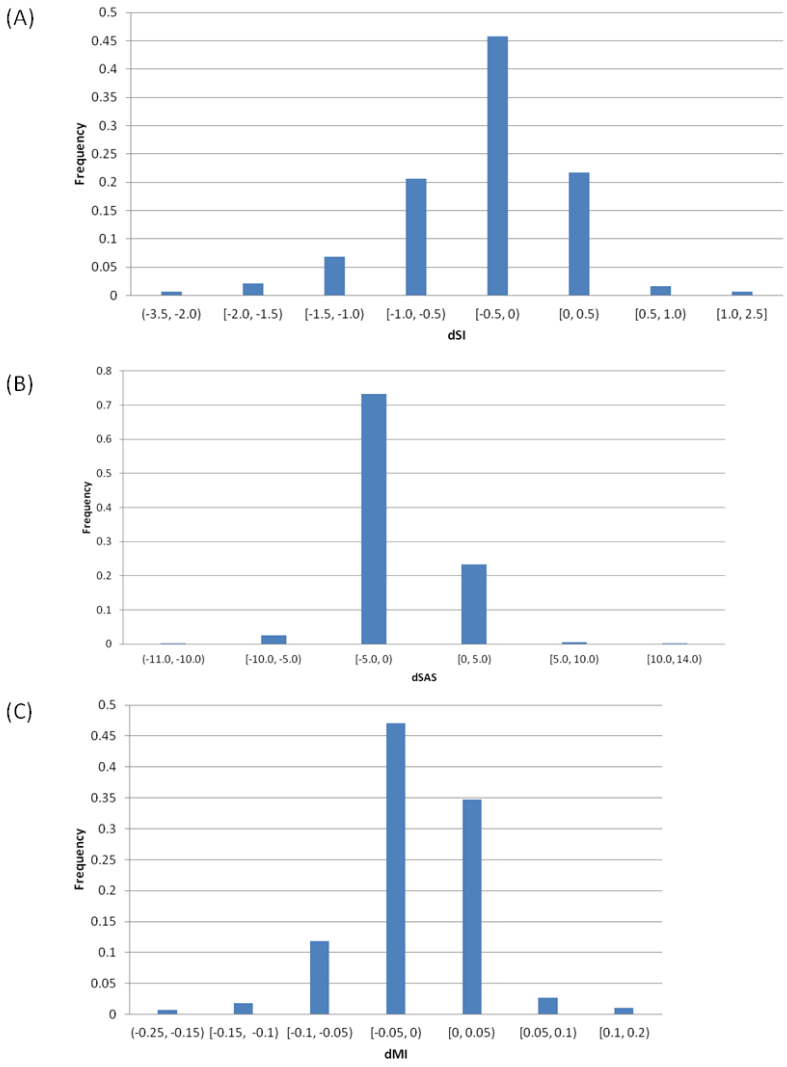

Figure 16: Histogram of (**A**) dSI, (**B**) dSAS, and (**C**) dMI for dataset D_1.

For these measure differences, it is difficult to identify the exact value which shows a significant change in the alignment quality. Based on the experimental results, we have empirically considered a |dSI|>-0.5 as a significant improvement imposed by iAlign (i.e., dSI>0.5) or PBSalign (i.e., dSI<-0.5) and a dSI between -0.5 and 0.5 as not so significant. Similarly, we define |dSAS| > 1.5 and |dMI| > 0.05 as significant improvement for SAS and MI, respectively. Table 12 shows summary of alignment

improvement based on SI, SAS, and MI values. From the table, we can see that PBSalign results in alignments with better SI (by ~30%), SAS (by ~26%), and MI (by ~14%) in comparison to iAlign.

Table 12: Comparison of iAlign with PBSalign for dataset D_1 using SI, SAS, and MI measures.

	% BINDING SITE PAIRS WHERE	
MEASURES	PBSALIGN IS BETTER	IALIGN IS BETTER
SI	30	2
SAS	26	3
MI	14	4

4.2.2 Dataset D_2

The dataset D_2 consists of 69 binding sites from 10 groups. Members of each group had similar binding sites on one side of their interfaces, but the partner proteins were different. During the experiments, we performed pair-wise alignments among members from the same group. Hence, totally $k\times(k\text{-}1)/2$ times of alignments are needed for each group where k is the size of group.

Different from the dataset D_1, the dataset D_2 has designated similar binding site pairs. For some cases, iAlign cannot find the same matching binding site pairs as that of the datasets because of different definitions of similarity score and strategies to select paired binding sites. Hence, we simply excluded those pairs from the dataset D_2 according to the alignment results of iAlign. Finally, we obtained54 binding site pairs which are structurally similar but non-homologous.

Table 13: Average (standard deviation) values of SI, SAS, and MI with iAlign and PBSalign for dataset D_2.

METHODS	SI	SAS	MI
iAlign	2.95(1.39)	19.20(9.89)	0.68(0.10)
PBSalign	2.06(1.07)	13.50(8.46)	0.63(0.10)

We first calculated average and standard values of SI, SAS, and MI for the alignments generated by iAlign and PBSalign, which are shown in Table 13. From this table, we can see that PBSalign achieved better geometric match measures compared with iAlign. The detailed distributions of the SI, SAS and MI values are shown in Figure 17. Next, we tested the statistical significance of the observed difference in the mean values of SI, SAS, and MI, using a paired t-test. The difference in the mean SI, SAS, and MI was found to be statistically significant (p-value <<0.001).

We further analyzed the relative improvement in various geometric match measures by PBSalign. Similar to the dataset D_1, we defined |dSI|> 0.5, |dSAS| > 1.5, and |dMI| > 0.05 as an significant improvement for SI, SAS, and MI, respectively. Table 14 shows the summary of measure differences. From this table, we can see that PBSalign results in alignments with better SI (by ~56%), SAS (by ~76%), and MI (by ~46%) in comparison to iAlign. The detailed distribution of dSI, dSAS, and dMI values are shown in Figure 18.

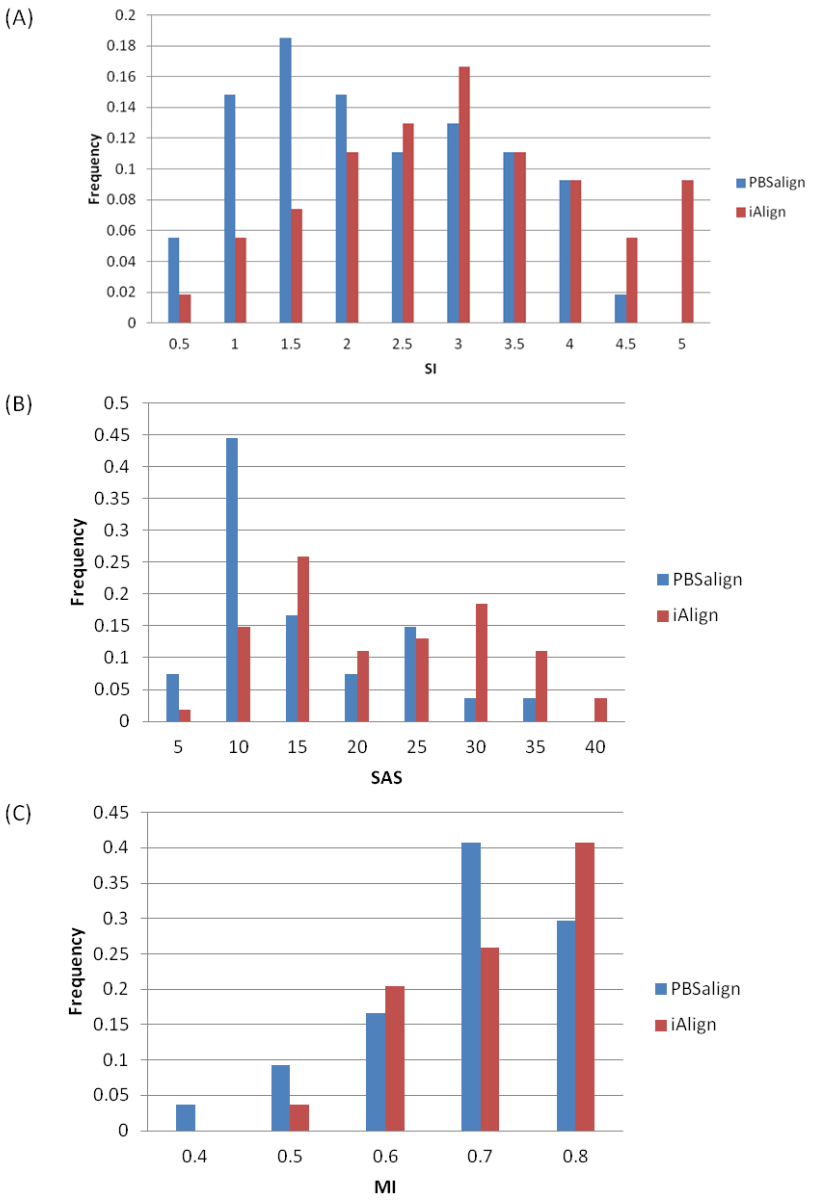

Figure 17: Histogram of (**A**) SI, (**B**) SAS, and (**C**) MI for dataset D_2.

Table 14: Comparison of iAlign with PBSalign for the dataset D_2 using SI, SAS, and MI measures.

	% BINDING SITE PAIRS WHERE	
MEASURES	PBSALIGN IS BETTER	IALIGN IS BETTER
SI	56	2
SAS	76	6
MI	46	9

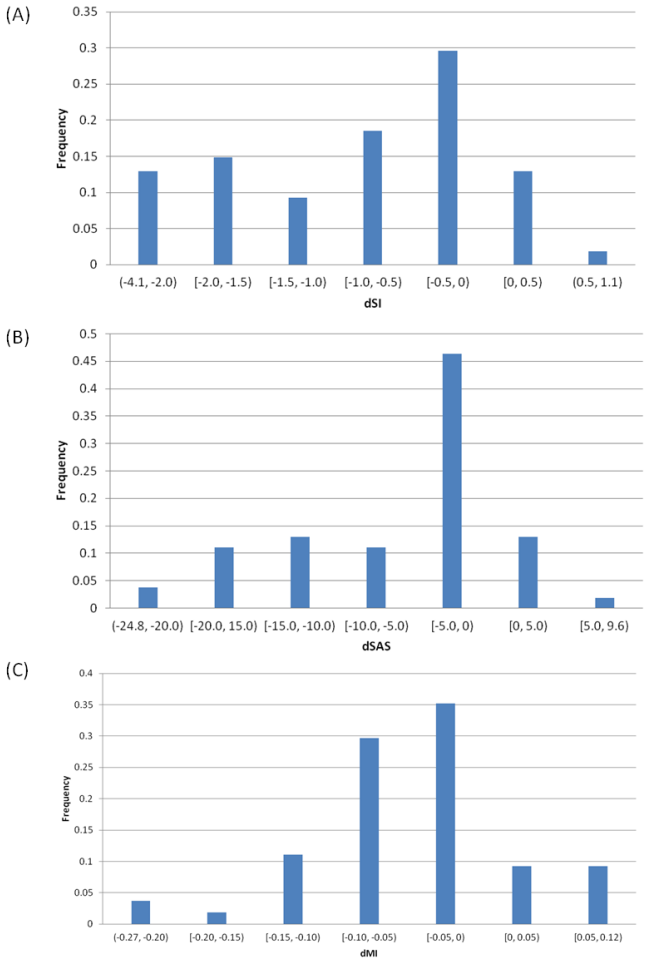

Figure 18: Histogram of (A) dSI, (B) dSAS, and (C) dMI for dataset D_2.

CHAPTER FIVE

PARALLEL PROTEIN GLOBAL STRUCTURE ALIGNMENT

To accelerate protein global structure alignment, we present *ppsAlign*, a parallel protein structure alignment framework designed and optimized to exploit the parallelism of GPUs. *ppsAlign* is a high-performance protein structure alignment tool designed to tackle the computational complexity issues associated with protein structural data. The solution presented in this chapter allows large-scale structure comparisons to be performed using massive parallel computing power of GPU. The methods and results of this chapter have been published in *BMC Research Notes* [89].

5.1. Methods

The framework of *ppsAlign* is shown in Figure 19. The inputs include a target protein and a protein database $\Lambda=\{P_1, P_2, \ldots, P_n\}$. The outputs are structure alignments between the target protein and each database protein. The online alignment starts with a generation of some initial sets of matched fragments and corresponding alignments. Then, the initial alignments are extended and refined using Dynamic Programming to obtain the final results. Specifically, the *ppsAlign* algorithm consists of five steps: 1) Index-based matched fragment set (MFS) search is utilized to find the maximal N_{seed} seed MFS' between the target protein and each database protein; 2) Fragment-level alignment is used to assemble the MFS' and generate initial alignments; 3) Residue-level alignment is used to refine the initial alignments to residue alignments; 4) Maximal alignment search is used to find a transformation that can best superimpose the entire

target protein over each database protein based on the obtained residue alignments; 5) Final assessment is performed to calculate z-Score and evaluate statistical significance of alignments. Steps 1) and 5) are executed on the CPU core, while steps 2) ~ 4), the most time-consuming parts of *ppsAlign*, are implemented as GPU kernels and iteratively executed on GPU for N_{iter} times.

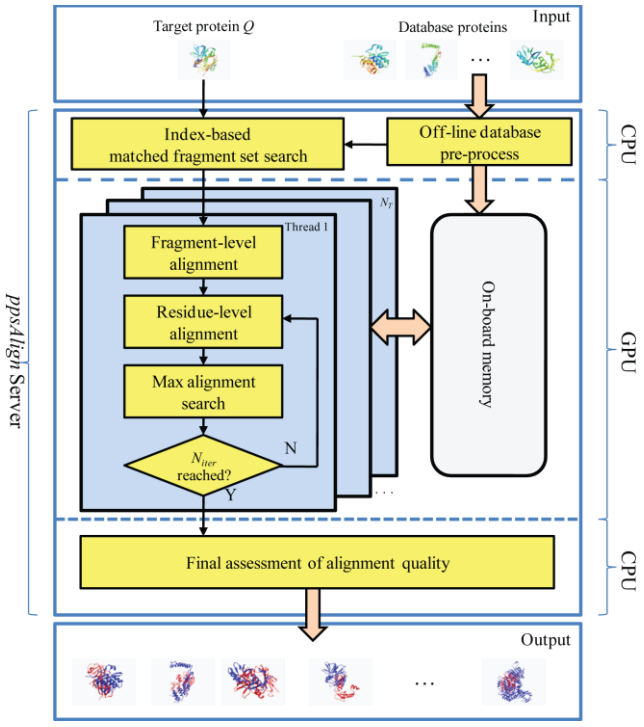

Figure 19: Framework of *ppsAlign*. The framework consists of both GPU- and CPU-based processes. The input includes a target protein and database proteins. The output contains all the structural alignment results between the target protein and each database protein.

The GPU kernels are developed using CUDA (Compute Unified Device Architecture) programming model [34]. During the alignment, the protein structures and intermediate results from each GPU kernel are stored in GPU's on-board memory, such as read-only constant memory, read-only texture memory, and read-write global

memory. Generally, the constant and texture memory have limited capacity but high access rate compared to the global memory. For an overview of GPU architecture and CUDA model, readers are referred to [34,90]. To facilitate the search of structurally similar fragments from the protein database, *ppsAlign* has an off-line component that pre-processes substructures from the entire protein database and builds an indexing tree to allow fast retrievals.

5.1.1 Index-based Matched Fragment Set Search

The purpose of this CPU-based step is to quickly find all possible matched fragment sets (MFS') between the target protein and each database protein for further refinement based on an information retrieval (IR) approach which goes beyond the capability of the traditional "bag of words" [30] concept by introducing spatial relationships among these fragments. Let $Q=\{q_1,q_2,\ldots, q_{L_Q}\}$ and $P=\{p_1, p_2,\ldots, p_{L_P}\}$ be a target protein with L_Q residues and a database protein with L_P residues, respectively. Here, q and p represent 3D coordinates of the C_α atoms. A fragment f is a set of L_f (=8) continuous residues with the direction from N terminal to C terminal along the protein backbone. A MFS includes two non-empty subsets, F_Q and F_P, which contain an order of fragments that conforms to some criteria of structural similarity between Q and P, respectively. The fragments in an MFS were used to generate a rough alignment between Q and P in the fragment-level alignment.

The MFS search utilizes the substructure mapping method of the Index-based Substructure Alignment algorithm [91] developed by the authors to retrieve similar fragments from the database proteins. In this method, substructures of the database

proteins, extracted by a large set of pairs of windows along the backbones, are indexed off-line by an indexing tree in which similar substructures are clustered into the same leaf node, denoted by t_i^A, and one substructure is selected as representative for each leaf node. Such representative structures preserve certain topological information, both locally and globally, from two disjoint substructures with various ranges of distances. Similarly, substructures in the target protein *Q* are indexed by an indexing tree in which each leaf node is denoted by t_j^Q. The representative substructure of each t_j^Q was used to search the indexing tree of our database and a list of best matched t^A was returned. For simplicity, we used *t* to denote t_j^Q and t^A. The database proteins that have substructures in t^A can be found by an inverted index. Such a database protein, *P*, can be represented by an order of substructures, denoted by Ω_t, occurring in *t*. Likewise, the protein *Q* can be represented by an order of substructures, denoted by Ω_t^Q, occurring in *t*. As substructures identified by the same *t* are similar, they can be used as "anchors" for rough alignments. For detailed explanation of the substructure mapping method, readers are referred to [91].

In *ppsAlign*, substructures are further projected into fragments as follows: if any residue of a substructure from Ω_t^P (or Ω_t^Q) is located in a fragment, the fragment is selected and added to F_P (or F_Q). The fragment subsets F_P and F_Q are used to construct an MFS between the protein *Q* and *P*. After searching all t^Q, we can obtain all possible MFS' between *Q* and database proteins, if any. In this step, if the algorithm cannot find any MFS for a database protein, all the fragments from *Q* and the database protein are selected to form a MFS.

After searching MFS, a filtering process was called to remove redundant MFS'. Then, the non-redundant MFS' between Q and each database protein were ranked according to scoring function S_{MFS} and the top N_{seed} sets selected. The scoring function is defined as follows:

$$S_{MFS} = w_1 \cdot \frac{N_Q}{N_f^Q} + w_2 \cdot \frac{N_P}{N_f^P} + w_3 \cdot \frac{\min(N_Q, N_P)}{\max(N_Q, N_P)},$$

where N_Q and N_P denote the cardinality of F_Q and F_P in an MFS, respectively. $N_f^Q = \lceil L_Q/L_f \rceil$ and $N_f^P = \lceil L_P/L_f \rceil$ are the numbers of fragments in the target protein and a database protein, respectively. The third term of the above scoring function is used to favor MFS' which have comparable N_Q and N_P. The values w_1, w_2, and w_3 are used to weight the contributions from the three terms.

The data needed by *ppsAlign* in order to compute the alignments on GPU are: structures of the protein Q and of the database proteins, and MFS'. To allow efficient processing, those data must be judiciously laid out on the GPU memories. Specifically, the database structures are transferred to the texture memory before execution. The MFS' are transferred from CPU memory to GPU global memory as inputs to the fragment-level alignment. Finally, the structure of protein Q is stored in the constant memory, which has smaller capacity but lower access latency compared to the texture memory.

5.1.2 Fragment-level Alignment

In this step, the fragments in each MFS are assembled to obtain initial alignments using Dynamic Programming (DP). For a given MFS, the DP algorithm first sorts the

fragments from F_Q and F_P according to their locations in Q and P. Then, it computes the similarity score $S_f(i, j)$ of each fragment pair for $1 \leq i \leq N_Q$ and $1 \leq j \leq N_P$ using the following recurrence:

$$S_f(i,j) = max \begin{cases} S_f(i-1, j-1) + S_f(i,j) \\ S_f(i, j-1) + G_f \\ S_f(i-1, j) + G_f \end{cases},$$

where G_f is gap penalty and S_f is based on the inverse cosine distance of fragment's feature vector. Given a fragment pair, A and B, and their corresponding feature vectors D_A and D_B, S_f is calculated as follows:

$$S_f = 1 - \cos^{-1}\left(\frac{<D_A, D_B>}{||D_A|| \cdot ||D_B||}\right),$$

where $<D_A, D_B>$ is the inner product of D_A and D_B, and $||D_A||$ and $||D_B||$ are the norm of D_A and D_B, respectively. In the current implementation, features only use Euclidean distance of each residue pair for fast calculation. The main reason for using feature distance as an approximate measure of fragment similarity is the need for simple control paths due to the SIMT (Single Instruction, Multiple Thread) computing mode of the GPU [34]. Traditional methods usually calculate RMSD and find an optimal transformation using the Kabsh algorithm [92], which contains complex control flows and is therefore not suitable for the SIMT mode. This step provides a rough alignment result which must be refined by the residue-level alignment.

Algorithm 1 Fragment-level alignment

```
// GPU kernel for fragment similarity score calculation
for i = 1 to ⌈L_Q /L_f⌉ do
    for j = 1 to ⌈L_P /L_f⌉ do
        Calculate S_f using Equation (3)
    end for
end for
Batch number N_batch ← ⌈N_F/N_T⌉
for all batches do
    // GPU kernel for Dynamic Programming
    Initialize each thread using MFS
    for i = 1 to N_Q do
        for j = 1 to N_P do
            Calculate S_f(i, j) using Equation (2)
            Store S_f and directions in GPU global memory
        end for
    end for
    // GPU Kernel for back tracing
    Start from the largest score in last column/row
    Back tracing using direction matrix to obtain alignment paths
    Copy fragment alignments paths from GPU to CPU memory
end for
```

Figure 20: Algorithm of fragment-level alignment. Fragment-level alignment consists of three GPU kernels. The first kernel performs the computation of the fragment scores. The second kernel implements the Dynamic Programming algorithm, and the third one performs the back tracing.

GPU Computation for Fragment-level Alignment

The pseudo-code in Figure 20 describes the fragment-level alignment. The algorithm splits the computation into three GPU kernels. The first kernel performs the computation of the fragment scores S_f by assigning a database protein to each thread. This kernel performs all-against-all fragment comparisons and writes similarity scores into the GPU global memory.

The second GPU kernel implements the DP algorithm, whereas the third one performs back tracing. The total number of threads N_T that can run concurrently on the GPU is mainly limited by the global memory capacity of the GPU (in this phase each

thread requires approximately 10kB of memory). Suppose that the total number of MFS between Q and all database proteins is N_F. If $N_F > N_T$, the overall MFS' will be divided into $N_{batch} = \lceil N_F / N_T \rceil$ batches. *ppsAlign* sequentially schedules each batch to run on GPU. In each batch, the DP is first executed as a GPU kernel and each thread corresponds to a MFS. Then, the GPU kernel for the back tracing is called to obtain alignment paths for each MFS. When a batch terminates, *ppsAlign* transfers the output (i.e., alignment path for each MFS) from the GPU memory to CPU memory. After aggregating the outputs from all batches, *ppsAlign* first performs filtering to remove redundant alignments, and then assembles all the fragments along the alignment paths to form residue alignments which will be further refined by the residue-level alignment.

It is critically important to effectively utilize the limited memory resources of the GPU. Our GPU memory allocation scheme is exemplified in Figure 21. The MFS' are stored in a 2D block of size ($N_T \times N_S$) where N_S is the maximal size of all MFS'. Each thread of the DP kernel fetches a MFS to initialize its setting. The score and direction matrices are stored in a separate 3D memory block of size ($N_Q \times N_P \times N_T$), where N_Q and N_P represent the maximal number of fragments from the target protein and all the database proteins, respectively. The alignment paths are then stored in a 2D block of size ($N_P \times N_T$). In *ppsAlign*, multiple GPU memory accesses are coalesced into a single transaction whenever possible. This fragment-level alignment process provides a selection of seed fragments which are likely to be successful in accurate alignment. Only approximately 1.6% of the total execution time is spent in this phase.

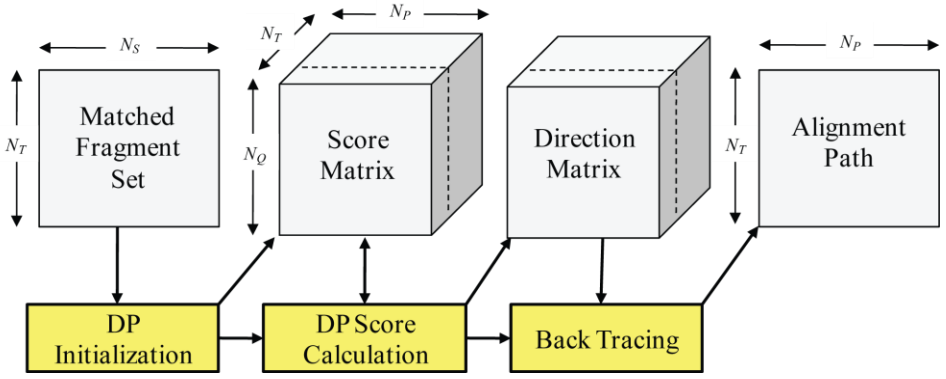

Figure 21: GPU global memory layout of fragment-level alignment. The Matched Fragment Sets (MFS') are stored in a 2D block. The score and direction matrices are stored in a separate 3D memory block. The alignment paths are stored in a 2D block.

Residue-level Alignment

The results of fragment-level alignment are then refined by a residue-level alignment process. Such a refined alignment result is an ordered set $R=\{(q_i, p_i) \mid q_i \in Q', p_i \in P'\}$, where $Q' \subseteq Q$ (target protein) and $P' \subseteq P$ (database protein).

In this step, a rigid-body transformation (rotation and translation) T that minimizes the RMSD of R is first calculated. Then, the transformation T is used to superimpose all the residues from Q over P. Finally, the DP algorithm is used to find an alignment path between Q and P similar to the fragment-level alignment. In the DP, the gap penalty G_r is set to 0 and the residue similarity score S_r uses the scoring function from TM-align [19]. However, our framework can be configured to use any suitable residue-level scoring function [47].

As we mentioned previously, the complex control flows present in the traditional method for computing T (e.g., Kabsch algorithm [92]) make it unsuitable for the SIMT computing model of GPU. To address this issue, we implemented and optimized a fast

algorithm using quaternion-based characteristic polynomial (QCP) [93], gRMSD-QCP, to determine the transformation *T* on GPU. In the gRMSD-QCP kernel, coordinates of residues from two protein structures were first written into the GPU global memory and origin of coordinate was moved to the center of coordinates for each protein. Then, we calculated the inner-product of two coordinate matrices, which is used by QCP for RMSD calculation. The work flow of gRMSD-QCP is relatively simple, and therefore, amenable to efficient GPU implementation.

GPU Computation for Residue-level Alignment

The GPU implementation of residue-level alignment starts with loading coordinates of residues from *R* to the GPU global memory. Next, the gRMSD-QCP kernel is invoked to calculate the transformation *T* which is also written into the GPU global memory. Finally, a DP kernel is called to find residue alignments which are transferred into the CPU memory after the kernel terminates.

As in the fragment-level alignment phase, the residue-level alignments are divided into batches according to the memory requirement of the threads. After all the batches are executed, *ppsAlign* aggregates the outputs of residue alignment *R*, which are used in the next step for searching the maximal alignment.

5.1.3 Maximal Alignment Search

The maximal alignment search is used to find the largest subset $M \subseteq R$ such that the score of the residue alignment *R*, denoted by S_a, is maximized. Because finding the largest subset *M* is extremely time-consuming, a heuristic and approximate algorithm, MaxSub [94], has been developed to solve this problem. In *ppsAlign*, a variant of

MaxSub, gMaxSub, is designed to parallelize the search process on the GPU. In the current implementation of *ppsAlign*, S_a is defined using the TM-score [19].

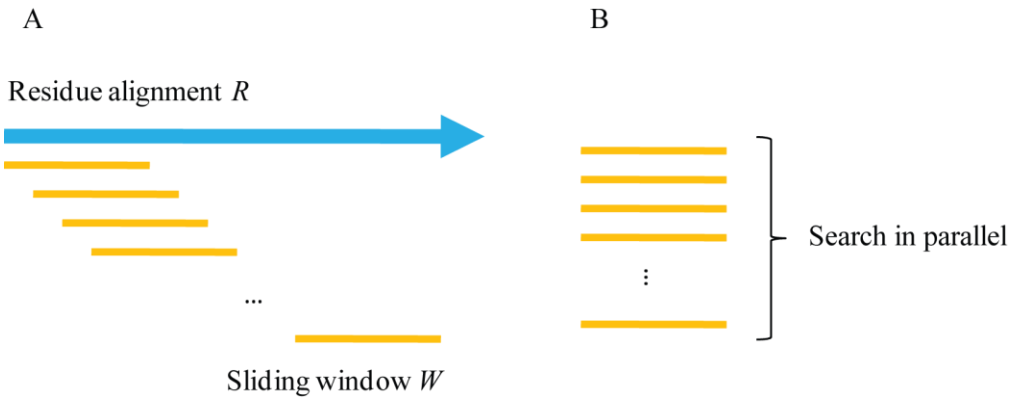

Figure 22: Comparison of MaxSub and gMaxSub. The original MaxSub algorithm on CPU searches the largest subset by shifting a window W along the residue alignment *R*. The gMaxSub searches the maximal alignment by concurrently dispatching each calculation of *W* to different GPU threads.

GPU Computation for Maximal Alignment Search

The input of this step is the alignment *R* from the residue-level alignment which has L_R aligned residue pairs. The original MaxSub algorithm on CPU searches the largest subset *M* by shifting a window *W* of size L_W along *R* (see Figure 22A). This leads to (L_R-L_W+1) shift operations which are candidates for parallelization. Then, gMaxSub searches the maximal alignment by concurrently dispatching each calculation of *W* to different GPU threads (see Figure 22B).

Algorithm 2 Maximal alignment search

```
1: Load residue alignment R to GPU global memory
2: Calculate shifting window W for each R
3: N_w ← total number of windows
4: Batch number N_batch = ⌈N_w/N_T⌉
5: for all batches do
       // GPU kernel for gMaxSub
6:     W = ∅
7:     for i = 1 to N_MS do
8:         Call kernel gRMSD-QCP to calculate transformation T
9:         Superimpose residues in R with T
10:        for each residue pair in R do
11:            if distance < 4.0 Å then
12:                Add the residue pair to W
13:            end if
14:        end for
15:    end for
16:    Calculate score S_a
17: end for
18: Select W with maximal S_a as the largest subset M for R
```

Figure 23: Algorithm of maximal alignment search. For each search, the gRMSD-QCP kernel is invoked to calculate the transformation of superimposed residues with each window. The residue pair is added into the window if its distance is below a cutoff.

Figure 23 describes a pseudo-code of gMaxSub. First, for each residue alignment R between Q and P, (L_R-L_W+1) windows are generated. Second, the gRMSD-QCP kernel is invoked to calculate the transformation T for the residue pairs within each W and then T is used to superimpose residues from Q over P in R. Third, residue pair $(q_i, p_i) \in R$ is added into W if its distance is below a cutoff (4.0Å) after the superimposition. The above two steps (i.e., gRMSD-QCP and window extension) are iteratively executed for N_{MS} times. Fourth, the last W is assigned to M and S_a is calculated.

As in previous phases, the maximal alignment searches are divided into batches. After all the batches are executed, *ppsAlign* aggregates the outputs of subset and selects the one with the largest S_a as the largest subset M. The transformation T associated with the largest subset M is used to superimpose all the residues from Q over P and the

residue pair whose distance is below a cutoff (4.0Å) is selected to form a new residue alignment *R*.

After gMaxSub terminates, if the current iteration number < N_{iter}, the residue alignment *R* will be first filtered to remove redundant alignments from the same database protein and then sent to the residue-level alignment for further refinement; otherwise, *R* will be used as input for the next step of final assessment.

Final Assessment of Alignment Quality

After structure alignments are computed on GPU, the residue alignments *R* are transferred from the GPU memory to CPU memory. We use PSI (percentage of structural similarity), defined as the percentage of residue pairs from *R* with distance below 4.0Å, to score the alignment quality. We also assess the statistical significance of the alignments through z-Score of the PSI, which is given as follows:

$$\text{z-Score} = \frac{\text{PSI} - \mu_{\text{PSI}}}{\sigma_{\text{PSI}}},$$

where μ_{PSI} and σ_{PSI} denote mean and standard deviation of PSI for a given protein chain length, respectively. The parameters μ_{PSI} and σ_{PSI} are obtained using a method similar to [22], leading to the following settings: $\mu_{PSI}=375.64 \cdot k^{-0.5295}$ and $\sigma_{PSI}=99.67 \cdot k^{-0.5885}$. Here, *k* is the minimum chain length between target and database proteins.

5.2. Results

In this section, *ppsAlign*'s performance is compared to concurrent methods in terms of alignment quality and computational efficiency. We evaluated *ppsAlign* using an NVIDIA Tesla C2050 GPU card equipped with 448 cores at 1.15GHz and 3GB

global memory. The concurrent methods included TM-align [19], Fr-TM-align [21], and MAMMOTH [22], which share similar computational framework as *ppsAlign*. As DALI [17] and CE [18] have been exhaustively evaluated elsewhere [19], this dissertation does not include these approaches in the description of experiments. Software packages of these methods were downloaded from their official websites and evaluated on a Linux personal computer with AMD Opetron dual-core 1000 series processor at 1.8GHz and 8GB RAM.

The main purpose of structure alignment was to maximize the number of aligned residues (N_e) while minimizing the RMSD of the aligned residues denoted by cRMSD. To eliminate the size dependence of cRMSD on N_e, in this dissertation we use a normalized measure of cRMSD, $RMSD_{100}$, to evaluate the alignment quality. $RMSD_{100}$ is calculated as follows [95]:

$$RMSD_{100} = \frac{cRMSD}{1+ln\sqrt{\frac{N_e}{100}}},$$

which corresponds to the cRMSD value expected when two protein structures are100 residues long.

To evaluate efficiency, execution time was measured on a dataset in which the protein's chain length ranged from 80 to 500 residues extracted from ASTRAL 1.75 database [96] with sequence identity<40% (ASREAL40). The database protein chain length was determined by the global memory capacity on the GPU card. However, this limitation is not severe as 98.5% ASTRAL40 protein chains have less than 500 residues. We expect that the advancement of GPU technology will solve this memory limitation

issue in the near future so that the *ppsAlign* algorithm can handle protein chains longer than 500 residues. Currently, we can handle structures larger than 500 residues in one of the following two ways: 1) by sending the alignment tasks to our CPU-based algorithm and 2) if resource allows, by using another GPU card to align the remaining 1.5% of large structures. Although the algorithm can also handle small protein chains below 80 residues (~16% of ASTRAL40), we do not use them for our testing because they have relatively simple topologies [70].To efficiently utilize global memory of GPU card, the entire database proteins are sorted according to the chain length and then divided into two small datasets: 1) D_1, which includes 6,569 proteins in the range [80, 250) residues selected from ASTRAL40 according to the length distribution of proteins, and 2) D_2, which includes 1,912 proteins in the range [251, 500) residues. The target dataset includes 100 proteins which are randomly selected in the range [80, 250) from ASTRAL40. For each target protein, a one-against-all alignment is performed with all database proteins and a total of 100×(6,569+1,912)=848,100 non-homologous protein pairs are compared during the experiment.

5.2.1 Scalability of ppsAlign

There are two critical parameters for *ppsAlign*, namely the maximal number of iteration (N_{iter}) and the maximal number of MFS (N_{seed}). Intuitively, when increasing N_{iter} or N_{seed}, *ppsAlign* will often obtain better alignment quality but the execution time will be significantly lengthened. To verify this, we preliminarily investigated the performance of different settings using a small target dataset of 17 proteins and the dataset D_1 in terms of $RMSD_{100}$. The experimental results of $RMSD_{100}$ with N_{iter}={3, 5, 7} and N_{seed}={10, 30, 50, 70} are shown in Figure 24, which illustrates that *ppsAlign* has

decreased $RMSD_{100}$ when N_{iter} and/or N_{seed} is increasing. This figure can be used as a guideline for parameter selection of *ppsAlign*. For a fair comparison of efficiency improvement from *ppsAlign* to a concurrent method, we selected a combination of N_{iter} and N_{seed} that achieved comparable alignment quality.

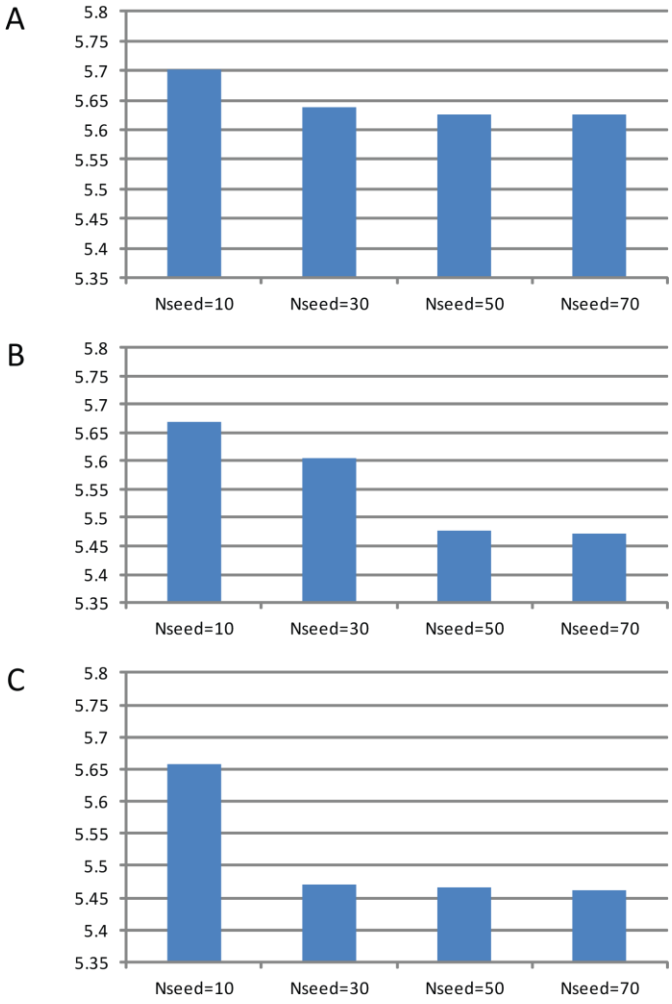

Figure 24: Performance comparison of *ppsAlign* with different settings of N_{seed} and N_{iter}. *ppsAlign* is running on NVIDIA Tesla C2050 GPU card with a small target dataset of 17 proteins. The parameter settings of *ppsAlign* are N_{iter}={3, 5, 7} and N_{seed}={10, 30, 50, 70}. (A) N_{iter}= 3. (B) N_{iter} = 5. (C) N_{iter}=7.

Table 15: Average execution time of TM-align, CPU-based ppsAlign, and *ppsAlign* with parameter settings (N_{iter}=3 and N_{seed}=20).

DATASET	METHODS	RMSD100	EXEC TIME(S)	SPEEDUP OF PPSALIGN
D_1	*ppsAlign*	5.7	64	-
	CPU-based *ppsAlign*	5.7	1596	24.9
	TM-align	5.7	2170	33.9
D_2	*ppsAlign*	5.3	41	-
	CPU-based *ppsAlign*	5.3	899	21.9
	TM-align	5.3	1597	39.0
Total	*ppsAlign*		105	-
	CPU-based *ppsAlign*		2495	23.8
	TM-align		3767	35.9

5.2.2 Speedup over TM-align and CPU-based ppsAlign

In this experiment, *ppsAlign* is executed with a parameter setting of N_{iter}=3 and N_{seed}=20 which results in a comparable $RMSD_{100}$ to TM-align and the CPU version of *ppsAlign*. Table 15 summarizes the alignment quality, average execution time, and corresponding speedup. *ppsAlign* achieves speedups of 23.8 and 35.9 compared to CPU-based *ppsAlign* and TM-align, respectively.

Table 16: Average execution time of Fr-TM-align and *ppsAlign* with parameter settings (N_{iter}=6 and N_{seed}=30).

DATASET	METHODS	RMSD100	EXEC TIME(S)	SPEEDUP OF PPSALIGN
D_1	*ppsAlign*	5.4	326	-
	Fr-TM-align	5.4	19849	60.9
D_2	*ppsAlign*	5.1	224	-
	Fr-TM-align	5.1	15729	70.2
Total	*ppsAlign*		550	-
	Fr-TM-align		35578	64.7

5.2.3 Speedup over Fr-TM-align

Since Fr-TM-align performs more iterations to improve its alignment quality over TM-align, we increased both iteration and seed numbers of *ppsAlign* algorithm to

achieve a comparable alignment quality with Fr-TM-align. The experimental results of $RMSD_{100}$, average execution time, and corresponding speedup with N_{iter}=6 and N_{seed}=30 are shown in Table 16. *ppsAlign* achieves speedup of 64.7 compared to Fr-TM-align with the same alignment quality.

Table 17: Average execution time of MAMMOTH and ppsAlign with parameter settings (N_{iter}=1 and N_{seed}=8)

DATASET	METHODS	RMSD100	EXEC TIME(S)	SPEEDUP OF PPSALIGN
D_1	*ppsAlign*	6.3	10	-
	MAMMOTH	10.3	470	47.0
D_2	*ppsAlign*	5.9	8	-
	MAMMOTH	9.2	255	31.9
Total	*ppsAlign*		18	-
	MAMMOTH		725	40.3

5.2.4 Speedup over MAMMOTH

In the last experiment, we used the same dataset to compare the performance of *ppsAlign* and MAMMOTH. Different from TM-align and Fr-TM-align, MAMMOTH was originally developed for the purpose of large-scale comparisons with high efficiency at the cost of the reduction of alignment quality. Because of its high speed, MAMMOTH was used as a benchmark for maximal speed on the CPU platform in [24]. The experimental results of $RMSD_{100}$, average execution time, and corresponding speedup with N_{iter}=1 and N_{seed}=8 are shown in Table 17. *ppsAlign* achieved speedup of 40.3 compared to MAMMOTH and higher alignment quality.

5.3. Discussion

The framework of *ppsAlign* is a general-purpose GPU platform for protein structure alignment which could take many concurrent methods, such as TM-align [19]

and Fr-TM-align [21], into the parallelized algorithm design. An important innovation in our approach is the creation of a unique design to manage resources of the GPU architecture. First, an intelligent decomposition of the application in kernels characterized by different parallelization strategies is provided. In the existing methods for GPU-based sequence alignment mentioned previously, a pair-wise comparison is either assigned to a thread (i.e., inter-task parallelization) or corporately performed by a block of threads (i.e., intra-task parallelization) [35,37]. However, since the workflow of structure alignment is more complicated than that of sequence alignment, neither the inter- nor the intra- task parallelization can efficiently exploit the GPU computing power. Therefore, *ppsAlign* utilizes a hybrid inter- and intra- task parallel model. In particular, each task (i.e., pair-wise structural comparison) is divided into several independent seed alignments. Each seed alignment is assigned to a different thread (inter-task parallelization), whereas each block executes one or more pair-wise comparisons (intra-task parallelization). Second, a smart design of memory layout and memory access patterns are developed, with the former allowing an effective use of the memory capacity at the different levels of the GPU memory hierarchy, while the latter minimizes the memory bandwidth requirement of the application. Third, several efficient algorithms for avoiding complex control flow on GPU are proposed to take advantage of the SIMT nature of GPU. For instance, a feature-based measure is used to compute similarity of fragments at the fragment-level alignment thereby avoiding time-consuming RMSD calculation at the initial stage of structure alignment.

One of the major ways in which *ppsAlign* differs from other methods is its ability to implement protein structure alignment at the residue level on GPU. Recently, GPU-

enhanced algorithms have gained increasing attention in bioinformatics. One of the major steps was a GPU implementation of a one-against-all sequence comparison using the Smith-Waterman algorithm [37,38]. With these methods, a sequence database search can be performed resulting in a list of similarity scores, although these methods do not provide the detailed alignment information of the best hits [40]. To provide detailed residue-residue correspondence, GPU-BLAST [41] was developed which allowed acceleration of the NCBI-BLAST search, achieving a speedup between 3 and 4 on an NVIDIA Tesla C2050 GPU card. Another approach to protein sequence that uses backtracking on GPU to construct an alignment of residues was proposed [40].Compared to sequence alignments, the implementation of GPU structure alignment has been the more challenging task because some routines (e.g., RMSD calculation) can cause severe divergence among GPU threads and decrease GPU performance. One of the first structure comparison methods implemented on GPU, SA Tableau Search [42], aligned protein substructure at the secondary structure level, that is by aligning secondary structure elements while not aligning structures at the residue level. To the best of our knowledge, *ppsAlign* is the first protein structure comparison platform for GPU that provides residue level structural alignment.

The substantial contribution of *ppsAlign* is that it provides a high-performance computing platform for the research community. An alternative solution to accelerate the protein structure alignment is to install more CPU computing cores in a single machine. However, using more CPU cores in a single machine necessitates upgrading the main board and memory accordingly, which could decrease the price/performance ratio. In contrast, installing a GPU card into a PCIe (Peripheral Component Interconnect

Express) slot does not require extra cost and more GPU cards can be installed into one PCIe slot by a switch. In this dissertation, an NVIDIA Tesla C050 GPU card was utilized to evaluate performance, which has also been used in GPU-BLAST [41]. Though it is a high end product of NVIDIA, we expect its price will drop in the near future due to market demand in the gaming industry.

CHAPTER SIX

WEB-BASED SYSTEM

To share our research results with the research community, we have developed a publicly accessible web-based system for efficient protein local binding site searches (PBSword). The accurate protein binding site alignment (PBSalign) and GPU-based protein global alignment (*ppsAlign*) are released as executable software packages, which can be downloaded from http://pbs.rnet.missouri.edu/. The design of PBSword server has been published in *Nucleic Acids Research* [97].

6.1. PBSword Server

PBSword server (http://pbs.rnet.missouri.edu/PBSword.php) is developed to provide the community with a web service for searching similar protein binding sites in terms of 'visual words' (see Chapter 3).The key features of PBSword server are as follows: (i) The binding site comparison method introduces a novel feature extraction algorithm and online database indexing; (ii) The database of the binding site is based on the interactions between domains which are defined using the latest SCOP version [98]; (iii) For each retrieved binding site from the database, a 3D view of structure and surface, as well as physico-chemical properties are presented; (iv) The efficiency has been significantly enhanced to meet the requirements of large-scale protein binding site database searching.

The system architecture of PBSword server, as shown in Figure 25, contains four modules: (i) database management and pre-processing; (ii) query interfaces; (iii) search

engine; and (iv) retrieval results visualization. A system tutorial can be viewed at the PBSword web site.

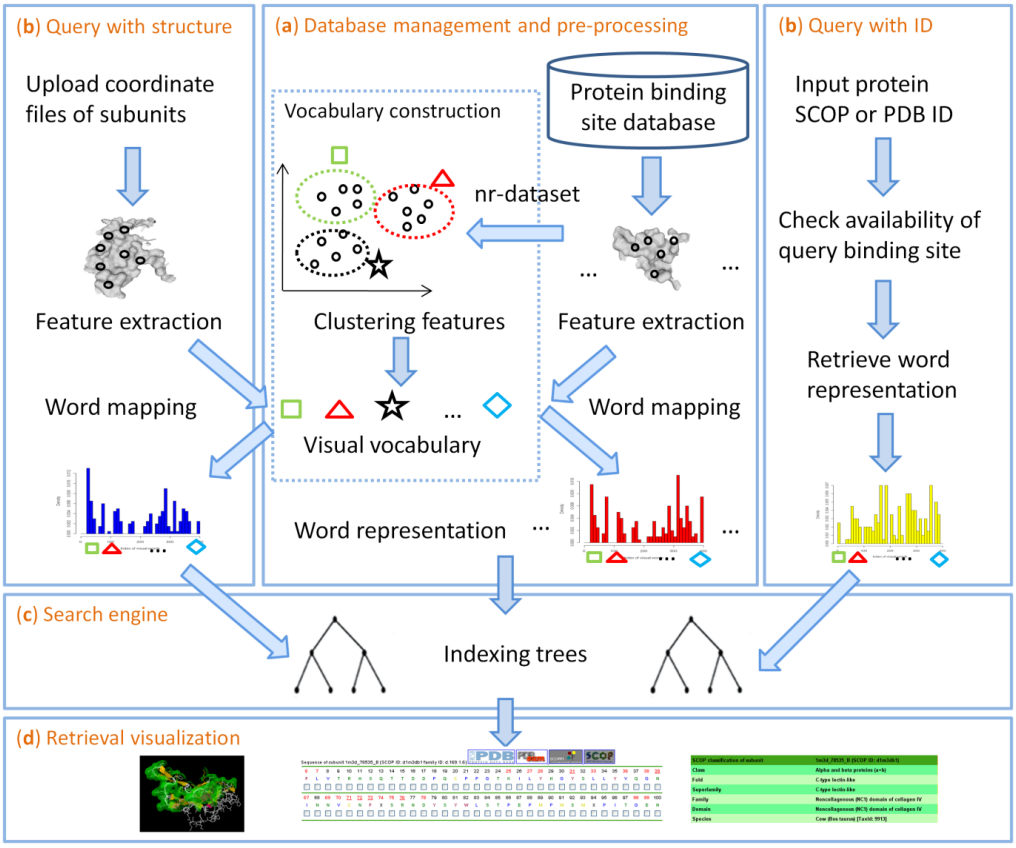

Figure 25: PBSword server architecture. (a) The database management and pre-processing module is responsible for feature extraction, visual vocabulary construction, and word representation of the database binding sites, which can be performed off-line. (b) The query interface modules provide friendly interfaces in an Internet browser to allow users to input protein ID or upload protein structure. (c) The search engine module organizes the word representation of the database binding site into indexing tree and returns *n* nearest neighbors for a query binding site in real-time. (d) The retrieval visualization module shows a 3D structure/surface view, sequence, and properties of retrieved binding sites.

6.1.1 Database management and pre-processing

The database of PBSword contains domain-domain binding sites of known protein structures. The structural data are extracted from Protein Data Bank (PDB) [99]. If a PDB entry has more than one structure model, the first model is used in the database's current implementation. For domain assignment, the most recent release (June 2009) of the manually curated SCOP database is used. For each PDB structure, each pair of determined subunits (i.e., domains) is analyzed to determine whether they interact with each other using the following definition. If any atom of a residue in one protein subunit is within 6Å of any atom of a residue in another protein subunit, the two residues are determined as the contact pair residues. Currently, the entire PBSword database contains 194,322 redundant binding sites selected from 3,123 SCOP families. Two non-redundant databases, denoted as NR40 and NR60, are constructed using sequence similarity of 40% and 60%, respectively.

The workflow of database pre-processing consists of the following three steps (see top-middle block of Figure 25). First, we select feature points from each database binding site surface and extract corresponding geometric features. Second, a visual vocabulary is built by clustering a large number (approximately 7×10^5) of feature point descriptors collected from a non-redundant (nr) dataset. The nr-dataset is selected from the entire database by applying a cutoff of 40% sequence identity for each SCOP family using CD-HIT [100]. The clustering method is *k*-means and each feature cluster is represented by a representative, which is regarded as a visual word and used to form the final vocabulary. The size of vocabulary is determined by *k*, which is set to 1,000 in the PBSword server. Third, according to its descriptor, each feature point from the database

binding site surface is associated with the nearest visual word from the vocabulary. This allows each binding site to be represented by the corresponding distribution of visual words. It is noted that the above processes for the database binding sites are performed off-line. The web server users are encouraged to access our most recent (2012) article in *Bioinformatics* [58] to find out more about the history and features of our service.

6.1.2 Query interfaces

There are two types of query methods, 'query by structure' and 'query by ID', as shown in the top-left and top-right blocks of Figure 25, respectively. Using an Internet browser, a user can upload a new protein structure in PDB format or provide a protein ID contained in a PBSword database to find similar protein binding sites. The target database could be (i) redundant; (ii) NR40; or (iii) NR60.

For the query by structure search, we follow the similar steps as the database binding sites to extract its features, map the features to the nearest visual word, and generate the visual word representation. The word representation of the query binding site is then sent to the search engine.

For the query by protein ID search, users can provide (i) SCOP IDs for the interacting subunits or (ii) PDB ID and chain ID for the subunit under investigation. For the second option, chain ID of interacting partner is optional. In that case, PBSword will search the database to find matched binding sites and allow the user to select one from the matched list. After the query binding site is selected, the corresponding word representation is then sent to the search engine.

6.1.3 Search engine

When the redundant PBSword database is selected as target, the on-line binding site search is performed on two customized indexing trees to avoid time-consuming one-by-one feature similarity calculation for the two query methods, namely query by ID and query by 3D structure. In this case, the query protein binding site can be represented by a data point in the visual word (or feature) space populated by the database binding sites as mentioned in the previous two subsections. Thus, searching similar binding site from the database is analogous to the identification of *n* nearest neighbors in the feature space. Such a search can be completed in log(*N*) time, where *N* is the total number of binding sites in the database. When the NR40 or NR60 database is selected, binding site similarity and associated z-Score are calculated for each database binding site.

6.1.4 Retrieval results visualization

The visualization of retrieval results includes seven parts: (i) structure and surface display, (ii) ranked list, (iii) sequence, (iv) SCOP classification, (v) properties of binding site, (vi) properties of each binding site residue, and (vii) property statistics of a SCOP family. The properties of binding site mainly include accessible surface area (ASA), polarity, hydrophobicity, hydrogen bonds, planarity, and gap index, which are originally defined in [101,102]. For completeness, we briefly introduce the calculation of these properties as follows.

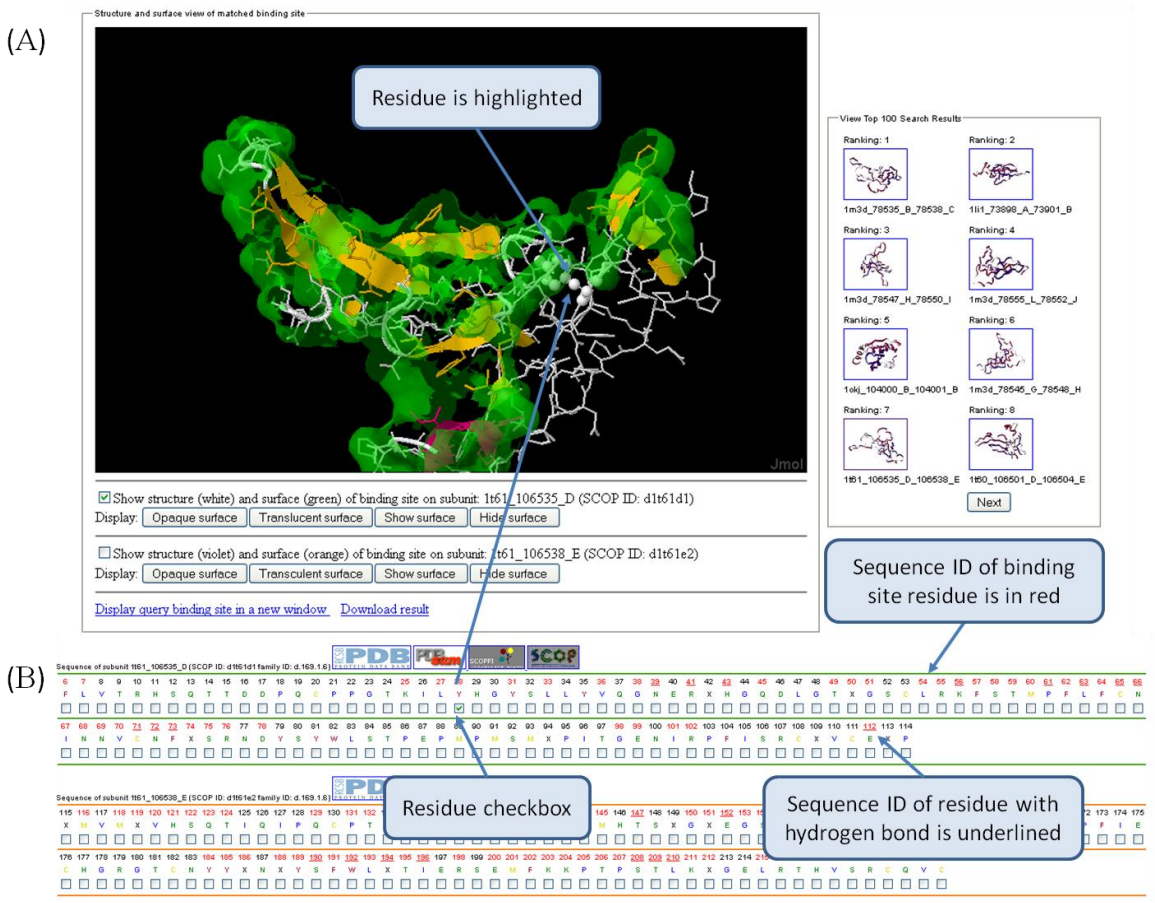

Figure 26: PBSword retrieval results visualization. (**A**) The top-left panel shows a 3D structure and surface view of a selected result protein binding site from the ranked list in the top-right panel. Users can click on the buttons and checkboxes in the top-left panel to select binding site and its partner as well as display modes of surface. (**B**) The sequence panel shows sequence information of subunit pairs. Each column in the panel corresponds to an amino acid of protein subunit, which consists of three rows. First row represents residue sequence numbers. For binding site residue, its number is in red font. For the residue with intermolecular hydrogen bonds, its numbers are underlined. Second row contains the residue name. The third row contains the residue check box. By clicking a checkbox, corresponding residues will be shown in the top-left structure view.

The ASA of binding site, ASA_{bs}, is calculated as

$$ASA_{bs} = ASA_1 - ASA_2,$$

where ASA_1 and ASA_2 are the ASAs of the subunit before and after its interacting partner presents, respectively. The ASA is calculated using the NACCESS, an implementation of the method proposed in [103]. A residue is defined as binding site residue if it loses 1.0Å^2 of ASA after subunit partner presents. The polarity of binding site is defined as [101]

$$\text{polarity} = (ASA_{polar}/ASA_{bs})\times 100,$$

where ASA_{polar} represents the difference of ASA of polar atoms before and after interacting partner presents. The hydrophobicity is measured using method proposed in [102]. The number of hydrogen bonds is calculated using the program HBPLUS [104]. The planarity is defined as the RMSD (Root Mean Squared Deviation) between all binding site atoms and a best-fit plane through all the binding site atoms, which is calculated using the PRINCIP program from the SURFNET package [105]. The gap index is defined as [101]

$$\text{gap index} = \text{gap volume}/ASA_{bs},$$

where gap volume is a measure of the closeness of the interface between the two subunits and calculated using the SURFNET package [105].

The retrieval results for an example query binding site 1m3d_78535_B_78538_C are shown in the top-left panel of Figure 26A. In PBSword, we use the identifier same as

[50] to name each binding site: <PDB-ID>_<SCOP- domain of the binding site>_<Chain-ID of the binding site>_<SCOP-domain of the binding partner>_<Chain-ID of the binding partner>. Accordingly, each subunit is defined as <PBD_ID>_<SCOP-domain ID>_<Chain-ID>. For each query, a set of 100 top-ranked binding sites is returned to the user, eight at a time for each page. To visualize the search results, a 3D structure and surface view of the top-retrieval result is displayed to the user. The user can select any of the ranked results from the top-right panel. The top-left panel in Figure 26A presents the structure and surface view of the top-ranked result, 1t61_106535_D_106538_E, which is generated by clicking on the thumbnail image on the top-right panel. In addition, the users can (i) select to show/hide structures of two subunits by clicking the checkboxes and (ii) specify different display themes of the binding site, such as opaque/translucent surface, by clicking on the buttons. The ranked list of protein binding sites can be downloaded from the result pages.

The sequence panel (Figure 26B) shows the sequence information of a subunit and its partner. For easy identification, the binding site residues are shown in red font and the residues with intermolecular hydrogen bond are underlined. The users can use the 'residue checkbox' under the residue to interact with the 3D structure view shown in Figure 26A. Clicking on the 'residue checkbox' will highlight one designated residue. Hyperlinks pointing to the protein's corresponding entry in PDB, PDBSum [106], SCOPPI [64], and SCOP [98] are also provided.

The SCOP classification panel (Figure 27A) shows the description of corresponding SCOP class, fold, superfamily, family and species for two subunits. The

properties of binding site and its interacting partner are shown in Figure 27B, including the number of binding site residues, ASA, percentage of ASA, percentage of polarity, percentage of hydrophobicity, planarity, number of hydrogen bonds, and gap index. By clicking on the hyperlink of SCOP family at the row 'Statistics of family,' the user can view the histogram and summary statistics of each property by SCOP family (see Figure 27D). The properties of each binding site residue, shown in Figure 27C, include ASA of all atoms and polar atoms for a specific residue and percentage of ASA against the entire binding site ASA, as well as the number of intermolecular hydrogen bonds. The family statistics panel (Figure 27D) shows the statistics summary of properties of binding sites belonging to a SCOP family, including total number of binding sites and amino acids compositions as well as the mean (standard deviation) and histogram of binding site properties. The properties include ASA, percentage of polarity, percentage of hydrophobicity, planarity, hydrogen bonding, and gap index. In this panel, hydrogen bonding is defined as the number of hydrogen bonds per 100 $Å^2$ ASA.

For a search with query ID, PBSword retrieval results can be generated in real-time. For the query with protein structures, however, the system will usually take minutes to generate surface and extract features, which is dependent on the size of the query binding site. Our system provides two options for the users: (i) PBsword server will return a session ID for the query along with an estimated execution time after the query protein structure has been uploaded. The user can then bookmark the link of the session ID and check the resulting page a few minutes later. (ii) If the user is willing to provide an email address when the query protein structure is uploaded, PBSword server will send ranked results to the user's email account.

(A)

SCOP classification of subunit	1t61_106535_D (SCOP ID: d1t61d1)
Class	Alpha and beta proteins (a+b)
Fold	C-type lectin-like
Superfamily	C-type lectin-like
Family	Noncollagenous (NC1) domain of collagen IV
Domain	Noncollagenous (NC1) domain of collagen IV
Species	Cow (Bos taurus) [TaxId: 9913]

(B)

Properties of binding site on subunit	1t61_106535_D (SCOP ID: d1t61d1)
Num of residues	45
ASA(Å^2)	1906.7
%ASA	25.0
%Polarity	33.3
%Hydrophobicity	40
Planarity	5.0
Num of hydrogen bonds	16
Gap index	1.3
Statistics of family	d.169.1.6

(C)

Properties of each binding site residue on subunit							1t61_106535_D (SCOP ID: d1t61d1)
Chain	ID	AA	ASA	%ASA	Polar	%Polar	Bonds
D	6	PHE	49.05	2.572	14.14	0.742	-
D	7	LEU	38.59	2.024	0	0	-
D	25	LYS	5.5	0.288	5.18	0.272	-
D	27	LEU	35.53	1.863	0	0	-
D	28	TYR	3.75	0.197	0	0	-
D	31	TYR	17.65	0.926	8.38	0.439	-
D	33	LEU	7.66	0.402	0	0	-
D	36	VAL	22.1	1.159	0	0	-
D	38	GLY	25.63	1.344	10.16	0.533	-

(D)

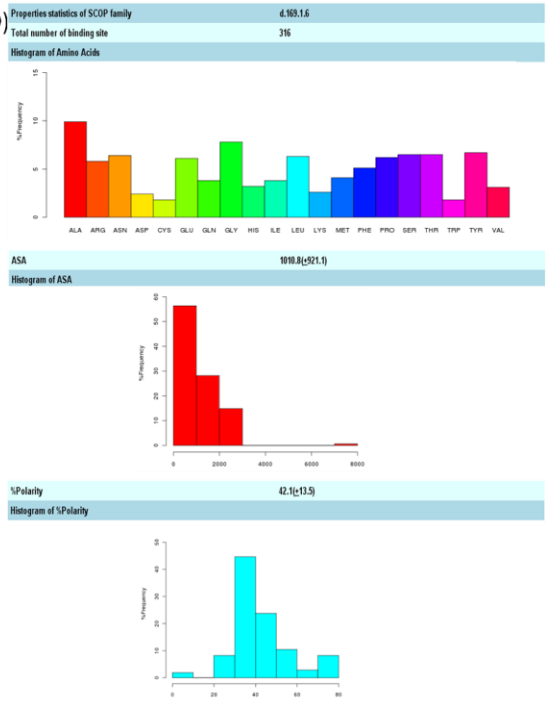

Figure 27: PBSword retrieval results of binding site properties. (**A**) The SCOP classification panel shows the subunit's classification, including class, fold, superfamily, family, and species. (**B**) The site properties panel shows values of various physic-chemical properties of binding sites, including number of residues, ASA, percentage of ASA, percentage of polarity, percentage of hydrophobicity, planarity, number of hydrogen bonds, and gap index. In addition, users can click on the hyperlink of SCOP family at the row "Statistics of family" to view the statistics summary of these properties for those binding sites belonging to same SCOP family. (**C**) The residue properties panel shows detailed properties for each binding site residue, including ASA, percentage of ASA, ASA of polar atoms, percentage of polar residue, and number of hydrogen bonds. (**D**) The family statistics panel shows the statistics summary of binding sites from a SCOP family, including number of binding sites and amino acids compositions as well as the mean (standard deviation) and histogram of binding site properties. The properties include ASA, percentage of polarity, percentage of hydrophobicity, planarity, hydrogen bonding, and gap index. Here, hydrogen bonding is defined as the number of hydrogen bonds per 100 Å2 ASA. Due to the page limitation, we only show subset of each panel.

CHAPTER SEVEN

CONCLUSIONS AND FUTURE WORK

We have presented a suite of methods (PBSword, PBSalign, and *ppsAlign*) for protein global and binding site structure comparisons. Among them, PBSword and PBSalign are designed to provide a pipelined process for search and comparison of protein binding site, while *ppsAlign* is designed as a parallel GPU computing platform for accelerating global structure alignment. These methods are expected to be utilized by the research community to investigate the structural, functional and evolutional relationships among proteins from large-scale structure and binding site databases.

7.1. PBSword

PBSword is a novel method for protein binding site characterization and comparison based on the distribution of visual words of surfaces. The proposed method complements existing alignment-based approaches in the analysis of protein-protein interactions. The method is applied to evaluate the classification and retrieval performance of protein-protein binding sites and is compared to an alignment-based method (iAlign) and to a feature-based method using moment invariants. The results show that PBSword can achieve comparable classification accuracy to the alignment-based methods with greatly improved efficiency.

7.2. PBSalign

PBSalign is a new method for explicitly comparing binding sites based on the geometric and physicochemical properties of local surfaces. PBSalign uses features extracted from the binding site surface to generate initial seed alignments which are further refined to produce accurate structural alignment. Our experimental results demonstrate that PBSalign can capture similarities of homologous and non-homologous protein binding sites accurately and provide alignments with better geometric match measures as compared to iAlign for a larger part of alignments in the selected datasets.

The alignment of PBSalign is based on the geometric and physicochemical features of binding site surfaces, which is different from iAlign relying only on the structural information. Hence, PBSalign is complementary to the existing methods by taking more properties into account and introducing novel algorithms for binding site surface alignments.

PBSalign is an alignment-based method for comparing binding sites. As we discussed in the Chapter 2 Literature Review, another type is the feature-based binding site comparison which utilize features (e.g. shape descriptors) to provide a fast comparison of binding sites without explicit alignment of residues. PBSword is this type of algorithm, which compares a pair of given binding sites by measuring similarities in their overall shapes. However, it does not output results of residue correspondences. PBSalign overcomes the problem of obtaining residue correspondences, which is essential to judge the quality of alignment. It is noted that when integrated with PBSword, PBSalign can quickly search a large-scale database, filter out geometrically

dissimilar binding sites, and provide accurate structural alignments only for the similar ones.

7.3. ppsAlign

*ppsAlign*is designed for large-scale protein structure alignment using GPUs. *ppsAlign* employs an index-based search procedure to find seeds of matched fragment sets, and then iteratively refines the seeds with fragment- and residue- level alignments. We provide an in-depth comparison of *ppsAlign* against several concurrent CPU-based methods. Our experimental results show that *ppsAlign* can achieve significant speedup over its CPU implementation, TM-align, Fr-TM-align, and MAMMOTH on a single NVIDIA Tesla C2050 GPU.

The framework of *ppsAlign* is a general-purpose platform for protein structure alignments on GPU. With this platform, we can parallelize existing algorithms (e.g., TM-align and Fr-TM-align) on GPU and utilize the massive parallel computing power of GPU to achieve high-throughput structural comparisons without sacrificing alignment quality.

7.4. Future Work

7.4.1 PBSword

Our future work of PBSword includes (i) the development of a more comprehensive scoring function for the surface comparison that takes into consideration physicochemical properties, and (ii) extension to the protein-ligand binding site comparison and retrieval.

7.4.2 PBSalign

PBSalign currently finds residue correspondences based on the maximal clique detection algorithm, which is known to be NP-hard. To accelerate the execution time, we will parallelize PBSalign on our general-purpose platform of protein structure alignment on GPU by dispatching each refinement of seed alignment to a computing core of GPU.

7.4.3 ppsAlign

As *ppsAlign* is designed for general-purpose global protein structure alignment, the future work of *ppsAlign* will be porting other CPU-based alignment tools to GPU and evaluating performance.

7.5. Biological Applications

First application of our methods is protein binding site prediction, which is critical for understanding functions of protein and identifying targets for therapeutics development. Our methods, PBSword and PBSalign, can be integrated to realize a 'template-based' prediction. First, binding site candidates or surface patches are identified based on the physicochemical and geometric features of query protein surfaces. Second, each patch is searched against our database of protein binding site using PBSword and a short list of similar sites is generated. As PBSword is dependent on the local surface features of binding site; this step is especially useful for remote similarities which cannot be directly revealed by conservation of the sequence or binding site structures. Finally, after performing PBSword screening, PBSalign is

applied for the filtered binding sites and accurate alignment results can be used to rank each patch and select protein binding sites.

Second application is studying human influenza protein-protein interaction network based on the structural properties. The protein-protein interaction network and structure information can provide valuable insight into analyzing molecular mechanism of disease [107]. An interaction network of human - influenza A virus (IAV) interactions has been constructed in [108] and a couple of hub IAV and human proteins with a higher degree are identified. To identify potential targets of IAV protein, PBSword and PBSalign can be applied to search human protein database and return top ranked human proteins.

BIBLIOGRAPHY

1. Zarembinski TI, Hung LW, Mueller-Dieckmann HJ, Kim KK, Yokota H, et al. (1998) Structure-based assignment of the biochemical function of a hypothetical protein: a test case of structural genomics. Proc Natl Acad Sci U S A 95: 15189-15193.
2. Zhang C, Lai L (2011) Towards structure-based protein drug design. Biochem Soc Trans 39: 1382-1386, suppl 1381 p following 1386.
3. Halperin I, Ma B, Wolfson H, Nussinov R (2002) Principles of docking: An overview of search algorithms and a guide to scoring functions. Proteins 47: 409-443.
4. Shin DH, Hou J, Chandonia JM, Das D, Choi IG, et al. (2007) Structure-based inference of molecular functions of proteins of unknown function from Berkeley Structural Genomics Center. J Struct Funct Genomics 8: 99-105.
5. Bahadur RP, Zacharias M (2008) The interface of protein-protein complexes: analysis of contacts and prediction of interactions. Cell Mol Life Sci 65: 1059-1072.
6. Shoemaker BA, Panchenko AR (2007) Deciphering protein-protein interactions. Part I. Experimental techniques and databases. PLoS Comput Biol 3: e42.
7. Shoemaker BA, Panchenko AR (2007) Deciphering protein-protein interactions. Part II. Computational methods to predict protein and domain interaction partners. PLoS Comput Biol 3: e43.
8. Phizicky EM, Fields S (1995) Protein-protein interactions: methods for detection and analysis. Microbiol Rev 59: 94-123.
9. Gao M, Skolnick J (2010) iAlign: a method for the structural comparison of protein-protein interfaces. Bioinformatics 26: 2259-2265.
10. Zhu H, Sommer I, Lengauer T, Domingues FS (2008) Alignment of Non-Covalent Interactions at Protein-Protein Interfaces. PLoS ONE 3: e1926.
11. Shulman-Peleg A, Nussinov R, Wolfson HJ (2004) Recognition of functional sites in protein structures. J Mol Biol 339: 607-633.
12. Das S, Kokardekar A, Breneman CM (2009) Rapid comparison of protein binding site surfaces with property encoded shape distributions. J Chem Inf Model 49: 2863-2872.
13. Sael L, La D, Li B, Rustamov R, Kihara D (2008) Rapid comparison of properties on protein surface. Proteins 73: 1-10.
14. Yin S, Proctor EA, Lugovskoy AA, Dokholyan NV (2009) Fast screening of protein surfaces using geometric invariant fingerprints. Proceedings of the National Academy of Sciences 106: 16622-16626.
15. Carpentier M, Brouillet S, Pothier J (2005) YAKUSA: a fast structural database scanning method. Proteins 61: 137-151.
16. Yang JM, Tung CH (2006) Protein structure database search and evolutionary classification. Nucleic Acids Res 34: 3646-3659.
17. Holm L, Sander C (1993) Protein structure comparison by alignment of distance matrices. J Mol Biol 233: 123-138.
18. Shindyalov IN, Bourne PE (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Eng 11: 739-747.

19. Zhang Y, Skolnick J (2005) TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic Acids Res 33: 2302-2309.
20. Zhang Y, Skolnick J (2004) Scoring function for automated assessment of protein structure template quality. Proteins 57: 702-710.
21. Pandit SB, Skolnick J (2008) Fr-TM-align: a new protein structural alignment method based on fragment alignments and the TM-score. BMC Bioinformatics 9: 531.
22. Ortiz AR, Strauss CE, Olmea O (2002) MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. Protein Sci 11: 2606-2621.
23. Kedem K, Chew LP, Elber R (1999) Unit-vector RMS (URMS) as a tool to analyze molecular dynamics trajectories. Proteins: Structure, Function, and Bioinformatics 37: 554-564.
24. Teichert F, Bastolla U, Porto M (2007) SABERTOOTH: protein structural alignment based on a vectorial structure representation. BMC Bioinformatics 8: 425.
25. Yang Y, Zhan J, Zhao H, Zhou Y (2012) A new size-independent score for pairwise protein structure alignment and its application to structure classification and nucleic-acid binding prediction. Proteins 80: 2080-2088.
26. Kolodny R, Koehl P, Guibas L, Levitt M (2002) Small libraries of protein fragments model native protein structures accurately. J Mol Biol 323: 297-307.
27. Le Q, Pollastri G, Koehl P (2009) Structural alphabets for protein structure classification: a comparison study. J Mol Biol 387: 431-450.
28. Brevern AGd, Hazout SA (2000) Hybrid Protein Model (HPM): A Method to Compact Protein 3D-structure Information and Physicochemical Properties. Proceedings of the Seventh International Symposium on String Processing Information Retrieval (SPIRE'00): IEEE Computer Society. pp. 49.
29. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25: 3389-3402.
30. Budowski-Tal I, Nov Y, Kolodny R (2010) FragBag, an accurate representation of protein structure, retrieves structural neighbors from the entire PDB quickly and accurately. Proc Natl Acad Sci U S A 107: 3481-3486.
31. Tyagi M, Gowri VS, Srinivasan N, de Brevern AG, Offmann B (2006) A substitution matrix for structural alphabet based on structural alignment of homologous proteins and its applications. Proteins 65: 32-39.
32. Pekurovsky D, Shindyalov IN, Bourne PE (2004) A case study of high-throughput biological data processing on parallel platforms. Bioinformatics 20: 1940-1947.
33. Shah AA, Folino G, Krasnogor N (2010) Toward high-throughput, multicriteria protein-structure comparison and analysis. IEEE Trans Nanobioscience 9: 144-155.
34. Nickolls J, Buck I, Garland M, Skadron K (2008) Scalable Parallel Programming with CUDA. Queue 6: 40-53.
35. Liu W, Schmidt B, Voss G, Muller-Wittig W (2007) Streaming Algorithms for Biological Sequence Alignment on GPUs. IEEE Transactions on Parallel and Distributed Systems 18: 1270-1281.

36. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147: 195-197.
37. Liu Y, Maskell DL, Schmidt B (2009) CUDASW++: optimizing Smith-Waterman sequence database searches for CUDA-enabled graphics processing units. BMC Res Notes 2: 73.
38. Manavski SA, Valle G (2008) CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment. BMC Bioinformatics 9 Suppl 2: S10.
39. Schatz MC, Trapnell C, Delcher AL, Varshney A (2007) High-throughput sequence alignment using Graphics Processing Units. BMC Bioinformatics 8: 474.
40. Blazewicz J, Frohmberg W, Kierzynka M, Pesch E, Wojciechowski P (2011) Protein alignment algorithms with an efficient backtracking routine on multiple GPUs. BMC Bioinformatics 12: 181.
41. Vouzis PD, Sahinidis NV (2011) GPU-BLAST: using graphics processors to accelerate protein sequence alignment. Bioinformatics 27: 182-188.
42. Stivala AD, Stuckey PJ, Wirth AI (2010) Fast and accurate protein substructure searching with simulated annealing and GPUs. BMC Bioinformatics 11: 446.
43. Keskin O, Nussinov R (2007) Similar binding sites and different partners: implications to shared proteins in cellular pathways. Structure 15: 341-354.
44. Keskin O, Tsai CJ, Wolfson H, Nussinov R (2004) A new, structurally nonredundant, diverse data set of protein-protein interfaces and its implications. Protein Sci 13: 1043-1055.
45. Tsai CJ, Lin SL, Wolfson HJ, Nussinov R (1996) A dataset of protein-protein interfaces generated with a sequence-order-independent comparison technique. J Mol Biol 260: 604-620.
46. Nagano N, Orengo CA, Thornton JM (2002) One fold with many functions: the evolutionary relationships between TIM barrel families based on their sequences, structures and functions. J Mol Biol 321: 741-765.
47. Hasegawa H, Holm L (2009) Advances and pitfalls of protein structural alignment. Curr Opin Struct Biol 19: 341-348.
48. Sander O, Domingues FS, Zhu H, Lengauer T, Sommer I. Structural Descriptors of Protein-Protein Binding Sites; 2008. Imperial College Press, London. pp. 79-88.
49. Merelli I, Cozzi P, D'Agostino D, Clematis A, Milanesi L (2011) Image-based surface matching algorithm oriented to structural biology. IEEE/ACM Trans Comput Biol Bioinform 8: 1004-1016.
50. Sommer I, Muller O, Domingues F, Sander O, Weickert J, et al. (2007) Moment invariants as shape recognition technique for comparing protein binding sites. Bioinformatics 23: 3139 - 3146.
51. Bahadur R, Zacharias M (2008) The interface of protein-protein complexes: Analysis of contacts and prediction of interactions. Cellular and Molecular Life Sciences 65: 1059-1072.
52. Shulman-Peleg A, Mintz S, Nussinov R, Wolfson H (2004) Protein-protein interfaces: recognition of similar spatial and chemical organizations. In: Jonassen I, Kim J, editors. Algorithms in Bioinformatics: Springer Berlin / Heidelberg. pp. 194-205.

53. Binkowski TA, Joachimiak A (2008) Protein functional surfaces: global shape matching and local spatial alignments of ligand binding sites. BMC Struct Biol 8: 45.
54. Hofbauer C, Lohninger H, Aszodi A (2004) SURFCOMP: a novel graph-based approach to molecular surface comparison. J Chem Inf Comput Sci 44: 837-847.
55. Grindley HM, Artymiuk PJ, Rice DW, Willett P (1993) Identification of Tertiary Structure Resemblance in Proteins Using a Maximal Common Subgraph Isomorphism Algorithm. Journal of molecular biology 229: 707-721.
56. Konc J, Janezic D (2010) ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. Bioinformatics 26: 1160-1168.
57. Pulim V, Berger B, Bienkowska J (2008) Optimal contact map alignment of protein-protein interfaces. Bioinformatics 24: 2324-2328.
58. Pang B, Zhao N, Korkin D, Shyu C-R (2012) Fast protein binding site comparisons using visual words representation. Bioinformatics.
59. Sanner MF, Olson AJ, Spehner JC (1996) Reduced surface: an efficient way to compute molecular surfaces. Biopolymers 38: 305-320.
60. Osada R, Funkhouser T, Chazelle B, Dobkin D (2002) Shape distributions. ACM Transactions on Graphics 21: 807 - 832.
61. Belongie S, Malik J, Puzicha J (2002) Shape Matching and Object Recognition Using Shape Contexts. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24: 509-522.
62. Dijkstra EW (1959) A note on two problems in connexion with graphs. Numerische Mathematik 1: 269-271.
63. Lloyd S (1982) Least squares quantization in PCM. Information Theory, IEEE Transactions on 28: 129-137.
64. Winter C, Henschel A, Kim WK, Schroeder M (2006) SCOPPI: a structural classification of protein-protein interfaces. Nucleic Acids Res 34: D310-314.
65. Haralick R, Shanmugam K, Dinstein IH (1973) Textural Features for Image Classification. Systems, Man and Cybernetics, IEEE Transactions on 3: 610-621.
66. Bradley A (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30: 1145-1159.
67. Henrick K, Thornton JM (1998) PQS: a protein quaternary structure file server. Trends Biochem Sci 23: 358-361.
68. Kuang X, Han JG, Zhao N, Pang B, Shyu CR, et al. (2012) DOMMINO: a database of macromolecular interactions. Nucleic Acids Res 40: D501-506.
69. Mintz S, Shulman-Peleg A, Wolfson HJ, Nussinov R (2005) Generation and analysis of a protein-protein interface data set with similar chemical and spatial patterns of interactions. Proteins 61: 6-20.
70. Xu J-R, Zhang Y (2010) How significant is a protein structure similarity with TM-score=0.5? Bioinformatics.
71. Henschel A, Kim W, Schroeder M (2006) Equivalent binding sites reveal convergently evolved interaction motifs. Bioinformatics 22: 550-555.
72. Drickamer K (1993) Evolution of Ca(2+)-dependent animal lectins. Prog Nucleic Acid Res Mol Biol 45: 207-232.

73. Drickamer K, Fadden AJ (2002) Genomic analysis of C-type lectins. Biochem Soc Symp: 59-72.
74. Zelensky AN, Gready JE (2005) The C-type lectin-like domain superfamily. FEBS J 272: 6179-6217.
75. Korkin D, Davis FP, Sali A (2005) Localization of protein-binding sites within families of proteins. Protein Science 14: 2350-2360.
76. Dorai C, Jain AK (1997) COSMOS-A representation scheme for 3D free-form objects. Pattern Analysis and Machine Intelligence, IEEE Transactions on 19: 1115-1130.
77. Steinkellner G, Rader R, Thallinger GG, Kratky C, Gruber K (2009) VASCo: computation and visualization of annotated protein surface contacts. BMC Bioinformatics 10: 32.
78. Bron C, Kerbosch J (1973) Algorithm 457: finding all cliques of an undirected graph. Commun ACM 16: 575-577.
79. Besl PJ, McKay HD (1992) A method for registration of 3-D shapes. Pattern Analysis and Machine Intelligence, IEEE Transactions on 14: 239-256.
80. Konc J, Janezic D (2007) An improved branch and bound algorithm for the maximum clique problem. MATCH Communications in Mathematical and in Computer Chemistry.
81. Krissinel E, Henrick K (2004) Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. Acta Crystallogr D Biol Crystallogr 60: 2256-2268.
82. Tomita E, Tanaka A, Takahashi H (2006) The worst-case time complexity for generating all maximal cliques and computational experiments. Theor Comput Sci 363: 28-42.
83. Bentley JL (1975) Multidimensional binary search trees used for associative searching. Commun ACM 18: 509-517.
84. Gerstein M, Levitt M (1998) Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. Protein Sci 7: 445-456.
85. Menke M, Berger B, Cowen L (2008) Matt: local flexibility aids protein multiple structure alignment. PLoS Comput Biol 4: e10.
86. Murzin AG, Brenner SE, Hubbard T, Chothia C (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. Journal of molecular biology 247: 536-540.
87. Teyra J, Samsonov SA, Schreiber S, Pisabarro MT (2011) SCOWLP update: 3D classification of protein-protein, -peptide, -saccharide and -nucleic acid interactions, and structure-based binding inferences across folds. BMC Bioinformatics 12: 398.
88. Kolodny R, Koehl P, Levitt M (2005) Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. J Mol Biol 346: 1173-1188.
89. Pang B, Zhao N, Becchi M, Korkin D, Shyu CR (2012) Accelerating large-scale protein structure alignments with graphics processing units. BMC Res Notes 5: 116.
90. Lindholm E, Nickolls J, Oberman S, Montrym J (2008) NVIDIA Tesla: A Unified Graphics and Computing Architecture. Micro, IEEE 28: 39-55.

91. Chi PH, Pang B, Korkin D, Shyu CR (2009) Efficient SCOP-fold classification and retrieval using index-based protein substructure alignments. Bioinformatics 25: 2559-2565.
92. Kabsch W (1976) A solution for the best rotation to relate two sets of vectors. Acta Crystallographica Section A 32: 922-923.
93. Theobald DL (2005) Rapid calculation of RMSDs using a quaternion-based characteristic polynomial. Acta Crystallogr A 61: 478-480.
94. Siew N, Elofsson A, Rychlewski L, Fischer D (2000) MaxSub: an automated measure for the assessment of protein structure prediction quality. Bioinformatics 16: 776-785.
95. Carugo O, Pongor S (2001) A normalized root-mean-square distance for comparing protein three-dimensional structures. Protein Sci 10: 1470-1473.
96. Chandonia JM, Walker NS, Lo Conte L, Koehl P, Levitt M, et al. (2002) ASTRAL compendium enhancements. Nucleic Acids Res 30: 260-263.
97. Pang B, Kuang X, Zhao N, Korkin D, Shyu CR (2012) PBSword: a web server for searching similar protein-protein binding sites. Nucleic Acids Res 40: W428-434.
98. Andreeva A, Howorth D, Chandonia JM, Brenner SE, Hubbard TJ, et al. (2008) Data growth and its impact on the SCOP database: new developments. Nucleic Acids Res 36: D419-425.
99. Berman HM (2008) The Protein Data Bank: a historical perspective. Acta Crystallogr A 64: 88-95.
100. Li W, Jaroszewski L, Godzik A (2001) Clustering of highly homologous sequences to reduce the size of large protein databases. Bioinformatics 17: 282-283.
101. Jones S, Marin A, Thornton JM (2000) Protein domain interfaces: characterization and comparison with oligomeric protein interfaces. Protein Eng 13: 77-82.
102. Jones S, Thornton JM (1997) Analysis of protein-protein interaction sites using surface patches. J Mol Biol 272: 121-132.
103. Lee B, Richards FM (1971) The interpretation of protein structures: estimation of static accessibility. J Mol Biol 55: 379-400.
104. McDonald IK, Thornton JM (1994) Satisfying hydrogen bonding potential in proteins. J Mol Biol 238: 777-793.
105. Laskowski RA (1995) SURFNET: a program for visualizing molecular surfaces, cavities, and intermolecular interactions. J Mol Graph 13: 323-330, 307-328.
106. Laskowski RA (2009) PDBsum new things. Nucleic Acids Res 37: D355-359.
107. Kar G, Gursoy A, Keskin O (2009) Human cancer protein-protein interaction network: a structural perspective. PLoS Comput Biol 5: e1000601.
108. Lai YH, Li ZC, Chen LL, Dai Z, Zou XY (2012) Identification of potential host proteins for influenza A virus based on topological and biological characteristics by proteome-wide network approach. J Proteomics 75: 2500-2513.

VITA

Bin Pang

Bin Pang was born in Linyi, Shandong Providence, China, on June 12, 1971. He entered Shandong University in 1989 and received Bachelor of Engineering in Computer Science in 1993. After receiving Master of Engineering in Computer Science in 1996, he joined People's Bank of China, Jinan Branch. In 1999, he entered Institute of Computing Technology at Chinese Academy of Sciences and received Ph.D. in Computer Science in 2003. After that, he worked for NEC Labs China and Lucent Technologies China. In 2008, he joined Informatics Institute at University of Missouri-Columbia. He received Ph.D. in Bioinformatics in May, 2013.