Context Shapes: Efficient Complementary Shape Matching for Protein-Protein Docking

Zujun Shentu\textsuperscript{a}, Christopher Bystroff\textsuperscript{b}, and Mohammed J. Zaki\textsuperscript{a}

\textsuperscript{a}Department of Computer Science, \textsuperscript{b}Department of Biology, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Abstract

We describe an efficient method for partial complementary shape matching for use in rigid protein-protein docking. The local shape features of a protein are represented using novel boolean data structures called context shapes. The relative orientation of the receptor and ligand surfaces is searched using pre-calculated lookup tables. Energetic quantities are derived from shape complementarity and buried surface area computations using efficient boolean operations. Preliminary results indicate that our context shapes based approach outperforms state-of-the-art rigid docking algorithms like ZDOCK \cite{3, 4} and Geometric Hashing \cite{16, 24}. Moreover, our method was developed with fast database searching in mind.

1 Introduction

All biochemical processes involve some kind of molecular recognition, which can be defined as the formation of an energetically favorable docking between two molecules. Docking between small substrate molecules and enzymes defines the specificity of enzyme catalyzed reactions. Docking between proteins is involved in many cellular processes including cell signaling, regulation of enzyme function, intra-cellular trafficking, transcriptional regulation, the immune response, and many others.

A growing database of molecular interaction data \cite{25} combined with large amounts of structural data for proteins \cite{1}, sets the stage for the development of new computational methods that
address systems biology questions. For example, with a partial interaction map of proteins in the
| cell, we might want to predict the missing interactions. We may want to validate observed inter-
| actions, or predict the details of an observed docking at atomic resolution. Given proteomic data
| about the protein contents of the cell and their sub-cellular locations, a set of predicted binding
| interactions could enable biologists to test specific hypotheses about signal transduction pathways,
| which are comprised of shape-specific binding interactions. Computational predictions of protein-
| protein interactions could potentially be used for high throughput screening of a library of small
| ligands. Before we can contemplate such large-scale docking queries, we must overcome several
| computational hurdles, including the problem of efficiently encoding and searching molecular sur-
| faces. This paper addresses the surface encoding problem with an eye towards enabling large-scale
| computational docking experiments.

Interactions between proteins are governed by the kinetics and stability of the docked confor-
| mation, or “pose”. The kinetics of interactions are governed by the concentration of each molecular
| species in the cell, its sub-cellular location, and by its surface electrostatics. The stability of a pose
| is governed by the surface electrostatics and by the surface shape complementarity. Shapes that
| are more nearly complementary will fill space better when docked. The energetic stability of a pose
| can be approximated by the amount of surface area excluded from solvent, which is approximately
| the total area over which the two surfaces are complementary.

The new shape matching approach presented here successfully docks proteins in a rigid-body
| sense. Our preliminary results indicate that it finds the crystallographically determined pose within
| 2 Å Root Mean Squared Distance (RMSD) in a majority of the cases. More importantly, this method
| was developed with fast database searching in mind. Protein features are captured using the new
| notion of Context Shapes, which are boolean data structures describing the local solvent excluded
| surface. It is assumed that a point on one surface superimposes on a point on the other surface
| in the correctly docked pose, and the goal of our method is to find one such pair of points. A
| pair of context shapes contains sufficient information to calculate the position and energy of a
| two-body binding interaction, both accurately and rapidly. The relative orientation of the two
| surfaces is searched using pre-calculated lookup tables. Energetic quantities are derived from shape
complementarity and buried surface area computations using efficient boolean operations.

The geometric surface of a protein has been the basis of docking through shape matching in several previous studies summarized briefly in the following paragraphs. Please refer to the reviews by Via et al. [23], Halperin et al. [10], and Mendez et al. [15] for a more detailed analysis.

The Geometric Hashing [16, 24] technique, originally developed in the computer vision domain, was used in a protein docking algorithm [7]. A set of critical points are derived from the solvent excluded surface of a protein. Two critical points along with the mean of their surface normals define a local reference frame onto which the local critical points are projected. These transformation-invariant coordinates are extracted from the target proteins and saved in a hash table. In the recognition stage, a similar process is carried out on the query protein, and the hash table is used to find matching reference frames with maximally similar coordinates for their local critical points. This method is fast at the recognition stage but it requires a slow post-processing step in which the query protein is rotated into position to detect physical penetration.

A docking algorithm based on the fast Fourier transform (FFT) and Fourier correlation theory was first proposed in [12]. The algorithm begins by discretizing the surfaces of the two molecules using a 3D grid. A correlation function is defined to compute a match. In the process of matching, the receptor molecule is fixed and the ligand is rotated. For each rotation, each cell in the ligand is shifted to match with each cell in the receptor. This operation takes $O(n^6)$ steps in the translational/rotational space, where $n$ is the size of each dimension of the grid. The FFT reduces the complexity to $O(n^3 \ln n^3)$. This method has been used in FTDock [8], ZDOCK [3, 4], and GRAMM [22] to search the translational space, and in HEX [18] to search the rotational space.

Monte Carlo sampling was used in [9] to predict protein-protein complexes from the coordinates of the unbound monomer components in a two step process. In the first step, a low-resolution rigid-body Monte Carlo search is used to find a population of candidate configurations, translating and rotating one protein around the surface of the other through 500 Monte Carlo move attempts. Then in the second step, explicit side-chains are added to the protein backbones using a backbone-dependent rotamer packing algorithm. The optimal combination of rotamers is found using a simulated-annealing Monte Carlo search. A scoring function is used to rank or reject the candidate
configurations. This Monte Carlo sampling based approach is computationally slow and is not good for indexing a protein database for docking.

Methods to search for shape similarity have also been proposed. For example, in [19] they propose geometric comparison of molecular surfaces in searches for active sites and functional similarity. The molecular surface is represented using sparse critical points and the geometric hashing algorithm [14] is used for shape matching. [11] proposes using normal vectors with attributes of curvature, electrostatic potential and hydrophobicity, to represent the surface features. [21] presents a patch-based representation for molecular surfaces providing access to the patches via an edge algebra, which represents the topology of the molecular surface. [13] represents the molecular surface by a set of triangular meshes. Properties such as electrostatic potential are also considered and added at each vertex. The search method is based on the clique search algorithm on a graph, where a clique gives a partial match between the two surfaces. In [6] a grid based 3D profile approach is introduced. The 3D surface profile is derived from a 3D multiple alignment, and a small number (≥ 3) of cells, called heavy cells, representing the more conserved positions are identified. The heavy cells are used to form all the possible non-redundant triads that are used as search keys to screen the database of protein structures, to perform a complete structure/3D profile grid comparison. [17] uses a genetic algorithm to generate translations and rotations of the query protein surface relative to the target protein surface which is held static. A fitness function is used to determine the degree of overlap and similarity between the rotated query surface and the target surface.

2 Methods

2.1 Context Shapes Overview

The shape of a protein is defined by its Solvent Excluded Surface (SES) [2, 5], which is the boundary of the solvent excluded molecular volume, and it can be calculated by rolling a spherical probe of the size of the solvent molecule over the exposed contact surface of each atom. The SES is composed of faces having three types of curvature: (1) contact face: the solvent accessible atomic surface, (2) toroidal face: the saddle-shaped surface where the probe makes contact with two atoms and
(3) *re-entrant face:* the concave bowl-shaped surface where the probe makes contact with exactly three atoms.

If we extend SES by 2.8 Å (the diameter of a water molecule), we can get a new surface, called the *Solvent Included Surface (SIS).* The region between SES and SIS, called the *SIS layer,* includes a single layer of solvent molecules, a hydration layer. The stability of a pose can be approximated by the amount of surface area excluded from solvent, called the *Buried Surface Area (BSA).* The BSA of a pose is the sum of the BSAs of both proteins. Each BSA is the area of the local SES that intersects with the SIS of the other protein.

In principle, when two rigid shapes are positioned as close as possible without passing through each other, there are at least three points of actual contact, called *Physical Contact Points (PCP).* In practice, crystal structures of oligomeric proteins may have fewer than three PCPs, or they may cross each other slightly and thus have many overlapping points, but these are deviations due to errors in the structure (for example due to conformational changes), or errors in the way the surfaces were calculated. But if we can assume the deviations from this assumption will be small, then the task of finding the most stable pose reduces to the task of finding PCPs.

We define the notion of *Context Shape (CS)* as the shape of the protein that is inside of a sphere centered at a point on the surface. Context shapes are composed of *Context Rays* which emanate from the center of the sphere and are uniformly distributed on the sphere. Each context ray is composed of 32 bits, which are assigned a value of one or zero depending whether the location of the bit is inside or outside of the surface or a surface layer. Superimposing two CSs implies superimposing two surface points. We score superimposed CSs for complementarity in order to determine whether the pair of points might be a PCP. So the task of finding PCPs reduces to the problem of evaluating the shape complementarity of all CS pairs. The best shape complementarity is determined by trying all rotations of one CS versus the other. At each rotated position, the complementarity is evaluated using boolean operations on the aligned context rays. Looking only for PCPs obviates the need for a translational search.

Dense sampling of the SES was found to be essential to the success of finding the PCPs, but using too many points would slow the search. The SES can be efficiently represented by a sparse
set of critical surface points without significant loss of accuracy [7]. Critical surface points are the face centers of the contact and re-entrant portions of the SES, ignoring the toroidal faces. The face centers are computed by projecting the centroid of each face to a point on the face. Only the sparse critical surface points are considered to be possible PCPs, which dramatically reduces the search space.

The BSA, defined above, is the final score of a given pose. But before the relatively expensive calculation of the BSA, simpler filters were applied: the Solid Vector filter and the Overlap Volume (OV) filter.

The calculations of both OV and BSA require a one-to-one association of the context rays in the two CSs. Boolean operations are carried out on associated rays. Two rays are associated if, after rotating one CS relative to the other, the rays are nearest neighbors on the sphere. This is defined later in more detail. Since all CSs are based on the same template, a set of rotations and nearest neighbor ray associations were pre-calculated and stored as a Matching Table, greatly speeding up the rotational search. The first pruning step used a ray to the center of mass of the inside bits, called the Solid Vector. The solid vectors of two CSs should be approximately 180 degrees apart in true poses. Poses in which the angle between solid vectors was too small were assumed to be impossible, as illustrated in Figure 3, since this would lead to too much overlap. To speed the search, only entries in the matching table that had highly obtuse solid vector angles were used. The search for a PCP then reduces to evaluating the OV and BSA over rays associated by the selected entries in the matching table of pre-computed rotations.

Since docked proteins cannot pass through each other, we further pruned poses by considering the OV, which is defined as the extent to which the two surfaces penetrate each other in a pose. To calculate the OV, we created multiple surface layers projected inward and outward from the SES. Using multiple layers enabled us to permit shallow penetration of surfaces but disallow deep penetrations. Shallow overlaps in true poses can be the result of conformational changes in the proteins, but deep penetrations should never occur in true poses. Having multiple layers enabled the assignment of different weights to different depths of penetration.

For a given CS pair, all poses that were not pruned were scored using the BSA calculation. The
rotation with the highest BSA score was stored, to be ranked later with all other CS pairs.

In short, the method has three main steps: 1) surface sampling and local shape representation as CSs, 2) pruning and complementary shape matching on CS pairs, and 3) ranking of scores. Details of these steps are described next.

### 2.2 Local Shape Representation

The local shape features of a protein were captured using context shapes. A context shape was represented using a sphere of a given radius \( r \) centered on a surface point (a potential PCP). The local shape features were captured by sampling different parts of the protein body bounded by the sphere. The sampling of shape features within a context shape consisted of a set of \( \kappa \) vectors, called context rays (CR), originating from the center of the sphere and ending at points evenly distributed on the surface of the sphere. The sphere was sampled at a density high enough that there was one end point per \( \rho \AA^2 \), on a sphere of radius \( r \AA \). Thus there were approximately \( \kappa = \left\lfloor \frac{4\pi r^2}{\rho} \right\rfloor \) rays used to sample the context shapes. For example, choosing density \( \rho = 1 \), and with sphere radius \( r = 10\AA \), we get \( \kappa = 1256 \) rays.

For representational purposes a context ray was divided into \( \beta \) segments, where each segment had one of two possible states: inside (1) or outside (0) of the surface. Thus a context ray can be represented by a \( \beta \)-bit binary string. We used \( \beta = 32 \) segments (or bits) to represent a context ray, which is equivalent to one computer word on 32-bit machines and allowed for fast bit operations (64 bits can be used for a more dense sampling if required). We used the notation \( CR[i] \) to denote the \( i \)-th bit of context ray CR.

Context shapes were used to represent different types of shape features such as surfaces and layers. The context shape representing the local solid volume is illustrated in Figure 1. Starting with the SES we computed several layers both inwards and outwards with each layer having a thickness of 1 \( \AA \). For any given distance \( \delta\AA \), relative to the SES, we can compute the surface layer at that distance. Let \( S_\delta \) denote the surface at distance \( \delta \), where \( \delta \in [-r, r] \), and where \( \delta < 0 \) indicates inner layers, \( \delta > 0 \) indicates outer layers, \( \delta = 0 \) indicates the SES, and where \( \delta = -r \) and \( \delta = r \) denote the sphere boundaries inside and outside the SES, respectively. See Figure 2 (a) for an illustration.
Figure 1: Context Shape Representing the Local Volume at a Critical Surface Point. The protein, SES, and the sphere centered at critical point $O$, are shown in 2D for simplicity. The shaded area gives the local volume of the protein at $O$. The context rays used to sample the context shape are also shown. Each segment of the ray has one of two possible states: inside “1” (shown in bold) or outside “0” (shown in dashes) of the local protein volume. The binary string composed of $\beta$ bits for a given context ray is also shown.

Figure 2: (a) Layers Inside and Outside the SES. Different layers are used to generate local shape features. Each layer is at a given distance from the SES (inwards or outwards). (b) Four main types of context shapes are shown (as the shaded region): i) local volume, ii) local SES, iii) local inner layer volume, and iv) local outer layer volume.
A context shape is then the local volume (only within the sphere) of the protein bounded by two surface layers, and is given as $CS(S_l, S_u, r)$, where $S_l$ and $S_u$ denote the lower and upper surface boundaries, and $r$ is the radius of the sphere. Since we used a fixed $r$, we denote a context shape as $CS(S_l, S_u)$. Figure 2(b) shows the different types of context shapes representing different shape features. For example $CS(S_{-r}, S_0)$ represents the local volume. The context shape representing the local SES is simply $CS(S_0, S_0)$. The figure also shows the context shapes for an inner layer volume $CS(S_{-2}, S_{-1})$ and an outer layer volume $CS(S_1, S_2)$. We refer to the different context shapes by simpler mnemonics: $CS_{vol} = CS(S_{-r}, S_0); CS_{ses} = CS(S_0, S_0); CS_{inK} = CS(S_{-K}, S_{-K+1})$, for inner layers; and $CS_{outK} = CS(S_{K-1}, S_K)$, for outer layers (where $K$ is in units of Angstroms Å). In this work we used four in and four out layers, namely: $in4, in3, in2, in1, out1, out2, out3, out4$. See Figure 2(a) for an illustration of these layers. Finally, we refer to $CS(S_{-r}, S_{-4})$ as the $CS_{core}$ region inside the SES, and we refer to $CS(S_{0}, S_{2.8})$ as the $CS_{sis}$ region (for the solvent included surface), where $\delta = 2.8$ is the diameter of the water (solvent) molecule. Each context shape $CS(S_l, S_u)$ is given as a set of $\kappa$ context rays $\{CR_i | i \in [1, \kappa]\}$, where each $CR_i$ is a context ray, a binary string with a ‘1’ for segments within the bounded region, and ‘0’ outside the region. In the case of $CS_{ses}$, for a context ray, a single bit is set to ‘1’ for each segment that intersects the SES and all other bits are set to ‘0’. Also note that a different number of rays may be used for different context shapes.

**Generating Context Shapes**

The different surface layers were computed using a 3D grid with a resolution of 0.2Å $\times$ 0.2Å $\times$ 0.2Å (along the x, y, and z axes). Each cell has a type identifier $t \in \{ses, inner, outer, empty, core\}$, and is assigned cell coordinates $(c_x, c_y, c_z)$, with $c_i \in \mathbb{N}$ (for $i \in \{x, y, z\}$). The coordinates are used to denote the distance (in terms of the number of cells) to the closest ses cell along the three axes. A pre-computed table is used to expedite the actual (Euclidean) distance calculation.

To identify the layers, in the first step, the cells that intersect with the SES were marked ses; the cells that were inside of the protein were marked core and the cells that were outside of the protein were marked empty. Then for each ses cell, we checked all of its local cells, within a distance of 4Å, to see if the current ses cell was their closest ses cell. If such a cell was currently marked as
we changed its type to \textit{inner} and updated its coordinates with respect to the current \textit{ses} cell. Likewise, if a cell was marked as \textit{empty}, we updated its coordinates to the \textit{ses} cell and marked it as \textit{outer}. Finally, if a cell was already marked \textit{inner} or \textit{outer} we only updated its coordinates if the current \textit{ses} cell was closer. If any cell was updated, we recursively updated its local cells to see if the current \textit{ses} cell was also closer to them. Once the local cells for all the \textit{ses} cells were checked, we had two layers with a thickness of 4 Å from the SES, one inward and the other outward. A single pass was then made to generate all the context shapes ($CS_{\text{vol}}, CS_{\text{ses}}, CS_{\text{inK}}, CS_{\text{outK}}, CS_{\text{core}}, CS_{\text{sis}}$) based on the cell types and their Euclidean distance from the closest \textit{ses} cell.

\subsection{Complementary Shape Matching}

A pose, $\pi$, between the context shapes $CS_X^R$ from protein $P_R$ (the receptor) and $CS_Y^L$ from protein $P_L$ (the ligand), with $X, Y \in \{\text{vol, ses, sis, core, inK, outK}\}$ and $K \in [1, 4]$, was represented by a one-to-one mapping of the context rays between the two context shapes. The feasibility of the pose was assessed using the overlap volume, defined as the volume that is labeled as inside in both CSs.

For a given pose $\pi$, the overlap volume of protein $P_L$’s context shape $CS_{\text{vol}}^L$ with respect to the context shape $CS_X^R$ for layer $X$ of protein $P_R$, is given as:

$$
OV(CS_{\text{vol}}^L, CS_X^R, \pi) = \sum_{i=1}^{n} V(CR_{i}^L \land CR_{i}^R) \tag{1}
$$

where $CR_{i}^L \in CS_{\text{vol}}^L$, and $CR_{i}^R \in CS_X^R$ is a context ray mapped to $CR_{i}^L$ according to pose $\pi$. Here $CR_{i}^L \land CR_{i}^R$ denotes the bitwise AND operation between the two context rays. The overlap volume within a single thin cone-shaped segment of the sphere is given as $V(CR_{i}^L \land CR_{i}^R) = \sum_{j=1}^{\beta} \nu(j)V[j]$, where $\nu(j) = (CR_{i}^L[j] \land CR_{i}^R[j])$ and $V[j]$ is the actual volume corresponding to the $j$-th segment of the context ray. Depending on the choice of the layer $X$ above, we obtain different kinds of overlap volumes. The \textit{total overlap volume} between two context shapes is a symmetric quantity, and is given as $OV(CS_{\text{vol}}^L, CS_{\text{vol}}^R, \pi)$. It represents the overlap of one protein’s volume with the other, for a given pose. On the other hand the \textit{layered overlap volume} is asymmetric, and is given as $OV(CS_{\text{vol}}^L, CS_X^R, \pi)$ and $OV(CS_{\text{vol}}^R, CS_X^L, \pi)$, where $X$ is one of the \textit{inK} or \textit{outK} layers. It
represents the part of one protein’s volume that overlaps a given layer $X$ in the other protein, for a given pose.

### 2.3.1 Pruning Based on Overlap Volume

The optimal pose will have a low overlap volume. In the ideal case, the docked surfaces from the two proteins around the PCP should not have any overlap. However, since we use discrete steps to rotate the proteins to a candidate pose, we might not get the ideal pose. Data sampling also contributes some error, so a small amount of overlap must be allowed. Overlap volume is used as a filter to prune infeasible poses, especially overlap volume in deep layers. Poses that have a small but deep overlap volume (i.e., *sharp penetration*) are more likely to be false than poses that have shallow overlap volumes. In our method, any pose with a non-zero overlap in layers $in3$, $in4$ and $core$, more than $1\AA^3$ overlap in $in2$, or a total overlap volume of $100\AA^3$ or more was pruned.

The overlap filters can be formally stated as follows:

- **Prune Large Overlap**: if $OV(CS_{vol}^L, CS_{vol}^R, \pi) > 100\AA^3$, then reject pose $\pi$.

- **Prune Sharp Penetration**: For $A, B \in \{L, R\}$, reject pose $\pi$ if either: (a) $OV(CS_{vol}^A, CS_{X}^B, \pi) > 0$ for $X \in \{in3, in4, core\}$, or (b) $OV(CS_{vol}^A, CS_{in2}^B, \pi) > 1\AA^3$.

Note that L and R represent the ligand and receptor proteins, and the expression $A, B \in \{L, R\}$, means that the calculations were done twice, switching L and R superscripts.

### 2.3.2 Enumerating Rotations

Comparing two context shapes, $CS^L$ from the ligand protein and $CS^R$ from the receptor protein, begins by superimposing their centers. Then there are three rotational degrees of freedom that must be sampled in order to find all possible poses. We can sample the rotational space by generating a set of evenly distributed rotation axes on the surface of the sphere (e.g., we can choose one of the context rays, $CR_i$, of $CS^L$ and align it with each ray, $CR_i$, of $CS^R$), and then rotating in steps of $\gamma^\circ$ around each rotation axis, yielding $\left\lfloor \frac{360^\circ}{\gamma^\circ} \right\rfloor$ poses (e.g., for $\gamma = 5.8^\circ$, we get 62 poses). The first step assures that the $z$-axis is positioned uniformly on the sphere, and the second step assures
that the $x$ and $y$-axes positions are uniformly sampled. One problem with this naive approach is that it samples poses more densely around the polar regions and less densely around the equatorial regions (with respect to the rotation axis) on the sphere.

The problem of uniformly sampling rotation space is similar to the problem of uniformly sampling points on a sphere. To avoid over-sampling around the poles, we can scatter the rotation axes evenly onto the sphere. On $CS^R$, we uniformly sampled a dense set of points on the sphere, and randomly generated an arc of unit length for each point. We then generated an arc on $CS^L$, and superimposed it onto each arc from $CS^R$. In superimposing two arcs, there are only two possible poses, $180^\circ$ apart. This arc-by-arc superposition results in a uniform sampling of poses in all regions on the sphere. For example, consider a sphere with radius $r = 10\,\text{Å}$. In the naive approach we align one of the rays from $CS^L$ with each of the $\kappa = 1256$ rays of $CS^R$ to obtain the set of rotation axes. Since we try 62 poses per rotation axis (for an angular step size of $\gamma = 5.8^\circ$, corresponding to a maximum movement of $1.1\,\text{Å}$ of a surface point on a sphere with radius $10\,\text{Å}$), we get a total of $1256 \times 62 = 77872$ poses. For the uniform sampling approach, we can maintain the same number of poses by generating 38936 points on $CS^R$, each with a randomly generated arc. We can then take a fixed arc from $CS^L$, and pair it with each of the 38936 arcs on $CS^R$. Each such pair of arcs corresponds to two poses, $180^\circ$ apart, yielding a total of $38936 \times 2 = 77872$ uniform poses.

![Figure 3: Restricted Pose Search. For a given pose $\pi$ if the reversed solid vector of $CS_{vol}^R$ is not within an angle of $\phi$ around the solid vector of $CS_{vol}^L$, we reject the pose.](image)

To prune obviously impossible poses, we computed the center of mass of each context shape $CS_{vol}$. For two $CS_{vol}$s, the vectors to the center of mass, called the Solid Vectors, should be almost
in the opposite directions when correctly docked. We can safely restrict the search for good poses to those where the reversed solid vector of $CS_{vol}^R$ lies within an angle of $\phi^\circ$ (we used $\phi = 35^\circ$) around the solid vector of $CS_{vol}^L$, as illustrated in Figure 3. This reduces the number of poses to try by about 11-fold.

For each of the rotated positions, a one-to-one association of context rays between two context shapes was pre-computed and stored in a Matching Table, since table checking is much faster in general than rotation operations. In order to get an accurate one-to-one association, one of the CSs, $CS^R$, was sampled at a higher density, between $2\kappa$ and $3\kappa$ rays, versus $\kappa$ context rays for $CS^L$. Context rays in $CS^L$ were then associated one-to-one with a subset of the context rays in $CS^R$ by rotating and then finding the nearest neighbor.

Consider the example shown in Figure 4. Context shape $CS^L$ has $\kappa = 4$ context rays and $CS^R$ has 10. The context shapes are shown in 2D. There are 8 candidate poses using a step size of $45^\circ$. Once such pose corresponding to the second row $\pi_2$ in the matching table is shown on the left, i.e., $\pi_2 = \{a \rightarrow 2, b \rightarrow 5, c \rightarrow 7, d \rightarrow 10\}$.

Figure 4: Matching Table. $CS^L$ has 4 context rays, whereas $CS^R$ has 10. The context shapes are shown in 2D. There are 8 candidate poses using a step size of $45^\circ$. Once such pose corresponding to the second row $\pi_2$ in the matching table is shown on the left, i.e., $\pi_2 = \{a \rightarrow 2, b \rightarrow 5, c \rightarrow 7, d \rightarrow 10\}$.
has $2.5\kappa = 10$ rays. Both sets of rays are evenly distributed inside the sphere (shown here in 2D for simplicity), and are labeled consecutively in the clockwise direction. Assume we sample the poses in steps of $45^\circ$ (clockwise) starting from the alignment of ray $a \in CS_L$ with ray $1 \in CS_R$. Then there are eight possible poses, since $\frac{360^\circ}{45^\circ} = 8$. For each of these poses, we would have to rotate the rays of $CS_L$ and find the closest rays of $CS_R$. An example pose (after rotating $CS_L$ by $45^\circ$) is illustrated in the figure, given as $\pi_2 = \{a \rightarrow 2, b \rightarrow 5, c \rightarrow 7, d \rightarrow 10\}$.

2.4 Ranking Matches

![Diagram](image)

Figure 5: Buried Surface Area. $CS_R$'s SIS layer can include a single layer of water molecules. If any part of $CS_L$'s SES falls in the region, we can safely claim that to be buried. The buried part of $CS_R$'s SES can be calculated in the same way by using $CS_L$'s SIS layer.

To find the best docking orientation of two proteins, we estimated the amount of surface excluded from the solvent, called the Buried Surface Area (BSA). The BSA approximates the desolvation energy, which is the primary driving force of protein-protein interactions. The more surface excluded from the solvent, the better the binding energy. The BSA of one protein is the amount of its SES area that overlaps in the SIS layer of the other protein, as shown in Figure 5. The sum of the buried surface area from both context shapes is used to score the quality of the pose. In practice, to account for the approximate nature of PCPs, we used the weighted sum of the buried area in several external layers.

For a given pose $\pi$, the buried surface area of protein $P_L$'s context shape $CS^L_{ses}$ with respect to
the context shape $CS_X^R$ for layer $X$ of protein $P_R$, is given as:

$$BSA(CS_{ses}^L, CS_X^R, \pi) = \sum_{i=1}^{K} A(CR_i^L \land CR_i^R)$$

(2)

where $CR_i^L \in CS_{ses}^L$, and $CR_i^R \in CS_X^R$ is a context ray mapped to $CR_i^L$ according to pose $\pi$. The buried area is given as $A(CR_i^L \land CR_i^R) = \sum_{j=1}^{\beta} \alpha(j) A[j]$, where $\alpha(j) = (CR_i^L[j] \land CR_i^R[j])$ and $A[j]$ is the actual area corresponding to the surface point represented by bit $j$. Note that the total buried surface area for layer $X$ is the sum $BSA(CS^L, CS^R, X, \pi) = BSA(CS_{ses}^L, CS_X^R, \pi) + BSA(CS_{ses}^R, CS_X^L, \pi)$.

The scoring function used to rank different poses is the weighted sum of the buried area across several layers, given as follows:

$$Score(CS^L, CS^R, \pi) = w_1 \times BSA(CS^L, CS^R, in1, \pi) + \sum_{K=1}^{3} w_K \times BSA(CS^L, CS^R, outK, \pi)$$

(3)

The weights for each layer were chosen empirically to optimize the rankings; we used $w_1 = 4$, $w_2 = 1$, and $w_3 = 0.25$, indicating the relative importance of the buried area in each of the inner/outer layers. Finally, for the two context shapes $CS^L$ and $CS^R$ we find the best pose $\pi$, which has the highest score, given as: $Match(CS^L, CS^R) = \max_{\pi} Score(CS^L, CS^R, \pi))$. To match two proteins, we check all the possible context shape pairs and rank them in decreasing order of their scores.

2.5 Context Shapes: The Complete Method

The complete context shapes method is composed of two steps: off-line preprocessing and online shape matching.

The off-line preprocessing step is used to generate the context shapes and the matching table. Given a protein complex along with information about the chains to use as the ligand and receptor, we use the MSMS algorithm [20] to generate the SES. The SES is given as a triangular mesh consisting of a set of surface vertices and a set of surface triangles, also called the faces. Next we generate a 3D grid with cell size $(0.2\AA)^3$ to compute the surface layers, the context shapes, and the
context rays, as described previously. The matching table, a common device in matching of any pair of context shapes, is also pre-computed off-line.

1. $\text{CS}_R \leftarrow$ context shapes from receptor protein $P_R$;
2. $\text{CS}_L \leftarrow$ context shapes from ligand protein $P_L$;
3. Candidate-Pairs $\leftarrow \emptyset$;
4. $\textbf{foreach}$ Context Shape $\text{CS}_R$ in $\text{CS}_R$ $\textbf{do}$
    5. $\textbf{foreach}$ Context Shape $\text{CS}_L$ in $\text{CS}_L$ $\textbf{do}$
        6. Calculate angle $\phi$ between Solid Vector of $\text{CS}_L$ and Reversed Solid Vector of $\text{CS}_R$;
        7. if $\phi$ is greater than a user-specified angle threshold then
            8. Skip to next context shape in $\text{CS}_L$;
        9. $\textbf{foreach}$ Pose $\pi$ of context shapes $\text{CS}_R$ and $\text{CS}_L$ $\textbf{do}$
            10. Calculate the overlap volume $OV$, under the given pose $\pi$;
            11. if $OV$ is bigger than a user defined threshold value then
                12. Reject pose $\pi$;
            13. Calculate the buried surface area $BSA$, under the given pose $\pi$;
            14. Only keep the best pose $\pi$ with the largest buried surface area;
            15. Insert the tuple $(\text{CS}_R, \text{CS}_L, \pi, BSA)$ in Candidate-Pairs;
7. Sort Candidate-Pairs based on $BSA$ (decreasing order):

In the online matching step, given the matching table and the set of context shapes from the ligand and receptor proteins, our matching algorithm outputs a ranked list of possible poses. Note that we always treated the larger protein (the receptor) as fixed whereas the smaller protein (the ligand) was translated and rotated. The pseudo-code for our context shape matching algorithm is given in Figure 6. For each pair of context shapes, $\text{CS}_R$ in the receptor protein $P_R$ (line 4) and $\text{CS}_L$ in the ligand protein $P_L$ (line 5), we first apply the solid vector pruning (line 7). We next enumerate all the possible poses $\pi$ (line 9) using the matching table. Of the poses that pass the filters on the overlap volume (line 11), we retain only the pose that has the largest buried surface area (line 13). Finally the candidate pairs of context shapes, along with their pose, are sorted based on their buried surface area, and the top ranked ones are reported (line 15).

It is worth mentioning that we adopted a database oriented approach to the problem of setting the various parameter settings, and for better experiment management. Given two sets of context shapes, one from a receptor and the other from a ligand, the algorithm generated a set of SQL
(Structure Query Language) statements to insert the pose information, including partial scores, into a SQLite database (http://www.sqlite.org/). Then we wrote various SQL scripts to extract and analyze the data using different parameter settings. For example, we tried varying the weights for the different surface layers and varying pruning cutoffs while optimizing the ranks of the true poses.

3 Results and Discussion

To evaluate the effectiveness of our approach we compared it with two geometry-based docking algorithms found in literature: ZDOCK, a FFT based approach [4], and GH, a Geometric Hashing based approach [7]. In this study we used the test cases that were used by ZDOCK and GH, respectively, which are sets of protein complexes taken from the Protein Data Bank (PDB; www.pdb.org).

For these tests, MSMS [20] was used to calculate the SES of both the receptor and ligand proteins, with probe-radius 1.4Å and surface point density of two points per 1Å². The surface layers were computed in a 3D grid with a resolution of (0.2Å)³. The sparse critical surface points were calculated based on the analytical representation of the SES. On the receptor protein $P_R$, we used only the centers of re-entrant faces, whereas on the ligand protein, only the centers of contact faces were used. In both cases, if the area of a re-entrant or contact face was less than 1Å², it was ignored. If a contact face was bigger than 4Å², then we evenly re-sampled the face to obtain one point per 2Å². These sparse critical surface points were used as the centers of the context shapes. The radius of the template sphere was $r = 12Å$.

In the evaluation, a candidate context shape pair was marked as a “true pair” if the RMSD of the ligand protein was less than a certain threshold value. For ZDOCK the threshold used was 2.5Å and for GH it was 1.5Å or 2Å. The definitions of RMSD varied slightly between ZDOCK and GH. In ZDOCK, the RMSD was based on the ligand protein’s $C_\alpha$ interface atoms, which are atoms having at least one receptor atom within 10Å. In GH, the RMSD is based on all ligand atoms. We used the corresponding thresholds and RMSD definitions in our comparisons with each method.

To compare our results with those reported for ZDOCK [4], we ran experiments on the 25 bound test cases used by them. The results are shown in Table I. As shown in the table, our
Table I: Comparison with ZDOCK. **PDB** column gives the PDB id for the protein complex, as well as the chains (first element of the pair) which are taken together as the receptor and the chain (second element of the pair) used as the ligand; **Atoms** gives the sizes of the receptor and ligand; **CS-Pairs** gives the number of context shape pairs that survive the Overlap Volume filter; **Min-Dist** gives the distance of centers of the first “true pair” of context shapes in the original complex; **CS-RMSD** gives the RMSD value for the true pair, and **CS-Rank** gives the rank of the true pair according to our approach. A rank is in bold if it is better than the competing methods. The columns **PSC-Rank** and **ZD-Rank** give the rank of the first true pair according to ZDOCK’s PSC method (only shape based) and ZDOCK’s full method (shape + desolvation + electrostatics), respectively. Finally **ZD-RMSD** gives the RMSD according to ZDOCK’s full method.

<table>
<thead>
<tr>
<th>PDB</th>
<th>Atoms</th>
<th>CS Pairs</th>
<th>Min Dist</th>
<th>CS RMSD</th>
<th>CS Rank</th>
<th>PSC Rank</th>
<th>ZD Rank</th>
<th>ZD RMSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ACB(E,I)</td>
<td>(1733, 521)</td>
<td>135235</td>
<td>0.99</td>
<td>1.38</td>
<td>1</td>
<td>25</td>
<td>18</td>
<td>1.33</td>
</tr>
<tr>
<td>1AHW(DE,F)</td>
<td>(1737,1595)</td>
<td>345519</td>
<td>1.81</td>
<td>1.80</td>
<td>5</td>
<td>26</td>
<td>7</td>
<td>1.82</td>
</tr>
<tr>
<td>1AVW(A,B)</td>
<td>(1631,1268)</td>
<td>271215</td>
<td>0.87</td>
<td>0.94</td>
<td>1</td>
<td>45</td>
<td>3</td>
<td>2.07</td>
</tr>
<tr>
<td>1AVZ(B,C)</td>
<td>( 873, 461)</td>
<td>67116</td>
<td>0.35</td>
<td>1.62</td>
<td>424</td>
<td>-</td>
<td>53466</td>
<td>1.61</td>
</tr>
<tr>
<td>1BRC(E,I)</td>
<td>(1642, 410)</td>
<td>109755</td>
<td>0.59</td>
<td>0.43</td>
<td>9</td>
<td>173</td>
<td>24</td>
<td>2.32</td>
</tr>
<tr>
<td>1BR5(A,D)</td>
<td>( 863, 692)</td>
<td>90138</td>
<td>0.69</td>
<td>1.01</td>
<td>1</td>
<td>61</td>
<td>65</td>
<td>2.13</td>
</tr>
<tr>
<td>1BVK(DE,F)</td>
<td>(1742, 983)</td>
<td>212444</td>
<td>1.88</td>
<td>2.15</td>
<td>3842</td>
<td>974</td>
<td>821</td>
<td>2.34</td>
</tr>
<tr>
<td>1CGI(E,I)</td>
<td>(1798, 439)</td>
<td>142552</td>
<td>0.23</td>
<td>0.58</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2.41</td>
</tr>
<tr>
<td>1CHO(E,I)</td>
<td>(1748, 399)</td>
<td>114760</td>
<td>1.09</td>
<td>1.30</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1.57</td>
</tr>
<tr>
<td>1CSE(E,I)</td>
<td>(1919, 521)</td>
<td>140817</td>
<td>0.65</td>
<td>1.25</td>
<td>1</td>
<td>1537</td>
<td>198</td>
<td>2.20</td>
</tr>
<tr>
<td>1DFJ(I,E)</td>
<td>(3410, 950)</td>
<td>438252</td>
<td>1.88</td>
<td>2.41</td>
<td>4366</td>
<td>37</td>
<td>1</td>
<td>2.48</td>
</tr>
<tr>
<td>1DQJ(AB,C)</td>
<td>(1666,1000)</td>
<td>213616</td>
<td>0.66</td>
<td>1.24</td>
<td>1</td>
<td>1341</td>
<td>9249</td>
<td>2.37</td>
</tr>
<tr>
<td>1FSS(A,B)</td>
<td>(4225, 463)</td>
<td>278356</td>
<td>1.71</td>
<td>1.92</td>
<td>252</td>
<td>731</td>
<td>50</td>
<td>1.52</td>
</tr>
<tr>
<td>1MAH(A,F)</td>
<td>(4104, 459)</td>
<td>284832</td>
<td>1.84</td>
<td>1.72</td>
<td>5</td>
<td>849</td>
<td>24</td>
<td>1.29</td>
</tr>
<tr>
<td>1MDA(HL,A)</td>
<td>(3378, 790)</td>
<td>330162</td>
<td>1.98</td>
<td>1.46</td>
<td>2097</td>
<td>33988</td>
<td>18034</td>
<td>2.29</td>
</tr>
<tr>
<td>1MLC(AB,E)</td>
<td>(3288,1000)</td>
<td>395560</td>
<td>0.67</td>
<td>0.75</td>
<td>5</td>
<td>1106</td>
<td>128</td>
<td>1.65</td>
</tr>
<tr>
<td>1TGS(Z,I)</td>
<td>(1628, 415)</td>
<td>126075</td>
<td>0.77</td>
<td>1.34</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2.22</td>
</tr>
<tr>
<td>1UGH(E,I)</td>
<td>(1807, 646)</td>
<td>171213</td>
<td>0.73</td>
<td>1.32</td>
<td>1</td>
<td>305</td>
<td>8</td>
<td>2.25</td>
</tr>
<tr>
<td>1WEJ(LH,F)</td>
<td>(1702, 822)</td>
<td>184525</td>
<td>0.58</td>
<td>0.61</td>
<td>47</td>
<td>1396</td>
<td>183</td>
<td>1.04</td>
</tr>
<tr>
<td>1WQ1(G,R)</td>
<td>(2531,1321)</td>
<td>496188</td>
<td>0.72</td>
<td>1.65</td>
<td>1</td>
<td>5</td>
<td>15</td>
<td>1.31</td>
</tr>
<tr>
<td>2KAI(AB,I)</td>
<td>(1790, 438)</td>
<td>133263</td>
<td>1.81</td>
<td>1.49</td>
<td>2</td>
<td>1399</td>
<td>388</td>
<td>1.61</td>
</tr>
<tr>
<td>2PCC(A,B)</td>
<td>(2370, 846)</td>
<td>246960</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>22338</td>
</tr>
<tr>
<td>2PTC(E,I)</td>
<td>(1628, 445)</td>
<td>121746</td>
<td>0.48</td>
<td>1.01</td>
<td>1</td>
<td>1655</td>
<td>193</td>
<td>1.83</td>
</tr>
<tr>
<td>2SIC(E,I)</td>
<td>(1937, 763)</td>
<td>189924</td>
<td>0.63</td>
<td>1.26</td>
<td>1</td>
<td>241</td>
<td>11</td>
<td>2.37</td>
</tr>
<tr>
<td>2SNI(E,I)</td>
<td>(1937, 512)</td>
<td>145340</td>
<td>0.93</td>
<td>1.29</td>
<td>1</td>
<td>7434</td>
<td>1262</td>
<td>2.22</td>
</tr>
</tbody>
</table>
results (CS) compare favorably to ZDOCK’s PSC algorithm, which like ours is based on geometric information only. Out of 25 test cases, the best true pose ranked better for 21 cases, and tied in 2 cases. When compared with ZDOCK’s better method PSC+DE+ELEC (shape + desolvation + electrostatics), which is based on both geometric and chemical information, our results were still significantly better, suggesting that shape information is of dominating importance in binding interactions.

<table>
<thead>
<tr>
<th>PDB</th>
<th>Atoms</th>
<th>Min Dist</th>
<th>CS-RMSD</th>
<th>CS-Rank</th>
<th>GH-Rank</th>
<th>GH-RMSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1CPK(E, I)</td>
<td>(2782, 157)</td>
<td>0.96</td>
<td>0.98</td>
<td>1</td>
<td>1</td>
<td>1.20</td>
</tr>
<tr>
<td>3DFR(A, *1)</td>
<td>(1294, 81)</td>
<td>0.81</td>
<td>1.00</td>
<td>1</td>
<td>7</td>
<td>0.65</td>
</tr>
<tr>
<td>4MBN(A, *2)</td>
<td>(1217, 45)</td>
<td>1.37</td>
<td>1.38</td>
<td>1</td>
<td>1</td>
<td>0.48</td>
</tr>
<tr>
<td>4PHV(AB, I)</td>
<td>(1520, 92)</td>
<td>0.83</td>
<td>0.81</td>
<td>4</td>
<td>327</td>
<td>1.25</td>
</tr>
<tr>
<td>2IGF(LH, P)</td>
<td>(3364, 58)</td>
<td>1.62</td>
<td>1.56</td>
<td>3</td>
<td>1</td>
<td>1.03</td>
</tr>
<tr>
<td>4CPA(A, I)</td>
<td>(2437, 285)</td>
<td>0.84</td>
<td>1.77</td>
<td>18</td>
<td>147</td>
<td>1.51</td>
</tr>
<tr>
<td>1TGS(Z, I)</td>
<td>(1646, 416)</td>
<td>0.77</td>
<td>1.79</td>
<td>1</td>
<td>1</td>
<td>0.72</td>
</tr>
<tr>
<td>1CHO(E, I)</td>
<td>(1750, 400)</td>
<td>1.12</td>
<td>1.43</td>
<td>1</td>
<td>6</td>
<td>0.80</td>
</tr>
<tr>
<td>2PTC(E, I)</td>
<td>(1629, 454)</td>
<td>0.48</td>
<td>1.87</td>
<td>1</td>
<td>3</td>
<td>1.15</td>
</tr>
<tr>
<td>1TEC(E, I)</td>
<td>(2004, 522)</td>
<td>0.71</td>
<td>0.91</td>
<td>24</td>
<td>134</td>
<td>0.69</td>
</tr>
<tr>
<td>4SGB(E, I)</td>
<td>(1310, 380)</td>
<td>1.34</td>
<td>1.11</td>
<td>2</td>
<td>6</td>
<td>1.09</td>
</tr>
<tr>
<td>2SEC(E, I)</td>
<td>(1920, 530)</td>
<td>1.10</td>
<td>1.41</td>
<td>1</td>
<td>8</td>
<td>1.06</td>
</tr>
<tr>
<td>4TPI(Z, I)</td>
<td>(1629, 456)</td>
<td>0.94</td>
<td>0.93</td>
<td>2</td>
<td>2</td>
<td>0.91</td>
</tr>
<tr>
<td>2MHB(B, A)</td>
<td>(1134,1069)</td>
<td>1.03</td>
<td>1.53</td>
<td>1</td>
<td>1</td>
<td>0.79</td>
</tr>
<tr>
<td>4HVP(B, A)</td>
<td>( 758, 758)</td>
<td>0.86</td>
<td>1.08</td>
<td>1</td>
<td>1</td>
<td>0.92</td>
</tr>
<tr>
<td>5HVP(B, A)</td>
<td>( 760, 760)</td>
<td>0.71</td>
<td>1.72</td>
<td>1</td>
<td>1</td>
<td>0.75</td>
</tr>
<tr>
<td>2HFL^2(LH, Y)</td>
<td>(3227,1000)</td>
<td>0.54</td>
<td>1.41</td>
<td>1</td>
<td>13</td>
<td>2.01</td>
</tr>
</tbody>
</table>

Table II: Comparison with Geometric Hashing. PDB column gives the PDB id for the protein complex, as well as the chains (first element of the pair) used as the receptor and the chain (second element of the pair) used as the ligand; Atoms gives the sizes of the receptor and ligand; Min-Dist gives the distance of centers of the first “true pair” of context shapes in the original complex; CS-RMSD gives the RMSD value for the true pair, and CS-Rank gives the rank of the true pair according to our approach. A rank is in bold if it is better than the competing method. The column GH-Rank gives the rank of the first true pair and GH-RMSD gives the RMSD according to GH. *1, *2: the Methotrexate and Heme segments of these complexes are the ligands, respectively. ^3For 2HFL, GH used the active site of the receptor, which is a much easier scenario; our result is only based on the full geometric shape of the receptor and ligand.

To compare our approach with the Geometric Hashing method, we ran experiments on the “bound” test cases (the protein in the docked conformation is used) reported in [7]. Like our method, the GH method used only the geometric shape of the two components. The comparison
with GH is listed in Table II. Out of 16 test cases, CS ranks were better in 9, and tied in 6.

![Complex of Alpha-Chymotrypsin with its Inhibitor](image)

**Figure 7:** Complex of Alpha-Chymotrypsin with its Inhibitor. The PDB code is 1CHO; Chain E (Alpha-Chymotrypsin) is used as the Receptor, and chain I (Turkey Ovomucoid 3rd Domain) is used as the Ligand. The receptor’s surface is shown, and for the ligand only the backbone is shown. The dark backbone corresponds to the actual binding pose from the PDB, whereas the grey backbone corresponds to a predicted pose (the “true pair”).

An example of a predicted true pair for one of the test cases (1CHO) from Table I is shown in Figure 7. The predicted pose matches very well with the crystallographic bound conformation. Note also, that for one of the ZDOCK’s test cases, 2PCC, we did not find a match at all. Even ZDOCK’s shape based method (PSC) could not find the actual binding site. Only the full ZDOCK approach was able to find a true pair, albeit with a very poor ranking. Upon closer inspection, we found that 2PCC has a big cavity in the original interface. The shape of the surface at the interface is so irregular that the solid vector filter may have pruned the true poses. The interface is also irregular for test case 1DFJ. In some cases, such as 1MDA and 1BVK, the actual interface was relatively small. Several false poses had larger buried surface areas and thus ranked better than the true pose. If we include chemical properties such as electrostatic potential map in the scoring function, the prediction accuracy would be improved. This is planned for future work.

**Computational Time:** The principle goal of this work was to improve the efficiency of docking calculations, and in that goal as well as in overall accuracy, it has succeeded when compared to
previous methods. In its current state, CS took 3 hours on a desktop with an AMD 2.0Ghz CPU to dock the test case 1TEC. ZDOCK took 9 hours on the same machine\(^1\). Improvements in speed are anticipated by implementing various new ideas. For example, by converting the SES into a set of triangles with approximately equal area, which can simply count bits to get the BSA. Also, by hierarchical clustering of similar surfaces, many poses can potentially be pruned at once.

4 Conclusions

Our results show that geometric complementarity of the interface surface plays a key role in protein docking. The success of this method in correctly identifying docked poses shows that the assumptions that underly the method are correct for the most part. Specifically, we assumed that poses with superimposed surface points (PCPs) were the only important poses, and we implicitly eliminated numerous poses that did not have PCPs. We also assumed that the energy of a docked pose was best estimated as proportional to the buried surface area, but only if the two shapes had little or no overlap. Success in the absence of any electrostatic considerations showed that shape complementarity plays a dominant role in determining binding specificity.

Several improvements are still possible. The results presented here have addressed only the docking of rigid shapes, therefore it is a goal to find ways to handle flexible shape matching, which we hope will improve performance on the more realistic, “unbound” test cases (results not shown). Electrostatics will become an important factor when docking highly charged molecules such as DNA and certain small molecules, therefore future studies will address ways to incorporate electrostatics. Finally, we have already touched upon the need to improve the computational time and to refine the method to handle some of the harder cases.

References


\(^1\)We used the source code provided by ZDOCK’s authors for the comparison. We were not able to compare the time to GH since its source code is not available.


