Bioinformatics, 33, 2017, i30–i36
doi: 10.1093/bioinformatics/btx233
ISMB/ECCB 2017

SnapDock—template-based docking by Geometric Hashing

Michael Estrin* and Haim J. Wolfson*

Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel

*To whom correspondence should be addressed.

Abstract

Motivation: A highly efficient template-based protein–protein docking algorithm, nicknamed SnapDock, is presented. It employs a Geometric Hashing-based structural alignment scheme to align the target proteins to the interfaces of non-redundant protein–protein interface libraries. Docking of a pair of proteins utilizing the 22 600 interface PIFACE library is performed in <2 min on the average. A flexible version of the algorithm allowing hinge motion in one of the proteins is presented as well.

Results: To evaluate the performance of the algorithm a blind re-modelling of 3547 PDB complexes, which have been uploaded after the PIFACE publication has been performed with success ratio of about 35%. Interestingly, a similar experiment with the template free PatchDock docking algorithm yielded a success rate of about 23% with roughly 1/3 of the solutions different from those of SnapDock. Consequently, the combination of the two methods gave a 42% success ratio.

Availability and implementation: A web server of the application is under development.

Contact: michaelestrin@gmail.com or wolfson@tau.ac.il

1 Introduction

Prediction of protein–protein interactions and the structures of the resulting complexes is a key task in Computational Structural Biology. Although experimentally determined structures of complexes are rapidly accumulating, they are far from being able to cover the complete interactome (Stumpf *et al.*, 2008). To bridge this gap a myriad of docking algorithms have been developed (Andrusier *et al.*, 2008; Bonvin, 2006; Halperin *et al.*, 2002; Huang, 2014; Smith and Sternberg, 2002). The CAPRI community wide experiment (Janin, 2005; Janin *et al.*, 2015) has significantly contributed to the development of docking algorithms and to an improved understanding of the computational challenges involved.

Over the years two major paradigms of protein–protein docking have emerged. The first one is the, so called, *ab initio* or template free docking, where the task is to model the structure of the protein–protein complex given the experimental (or modelled) structures of the individual docking partners without having prior knowledge of the interface structure. In the last decade with the improved coverage of experimentally determined protein–protein interfaces (Cukuroglu *et al.*, 2014; Gao and Skolnick, 2010; Kundrotas *et al.*, 2012) template-based docking (TBD) methods have emerged, where a database of structural interfaces is scanned and the candidate-docking proteins are aligned to both sides of the interface either by structural alignment or by threading the candidate chains on the interface structure (Muratcioglu *et al.*, 2015; Szilagyi and Zhang, 2014). The major advantages of TBD are higher computational speed and more reliable interface modelling, which is based on mimicking of experimentally derived interfaces.

One of the first TBD methods, which is based on an underlying interface database was PRISM (Ogmen *et al.*, 2005). In its recent version (Tuncbag *et al.*, 2011), PRISM is based on two algorithms, which have been developed in our Lab, MultiProt (Shatsky *et al.*, 2004) and FiberDock (Mashiach *et al.*, 2010). For each interface in the template database of PRISM MultiProt is used to structurally align the target proteins to both sides of the interface. The resulting modelled interfaces, which passed successfully structural and hot spot correspondence filtering, are submitted to flexible interface refinement by FiberDock, which allows both side chain and limited backbone flexibility of the resulting interface and computes a binding energy score for the interface. The candidate modelled interfaces are ranked by the FiberDock energy score. PRISM has shown significantly faster performance than *ab initio* docking methods, especially, in large-scale docking.

In this study, we introduce SnapDock, a new TBD algorithm, which adheres to the general PRISM scheme of structural alignment followed by flexible interface refinement and scoring of the candidate interfaces. The structural alignment step of SnapDock is performed by a Geometric Hashing type procedure (Lamdan and Wolfson, 1988; Nussinov and Wolfson, 1991). Its advantages are 2-fold. First, it is significantly faster than MultiProt and second, its structural alignment, which is based on pairs of separated C_α atoms, is truly sequence order independent, while for the MultiProt alignment to be reliable it requires a short consecutive matching fragment. We also present a Flexible Template-Based Docking scheme, which allows hinge-based flexibility in one of the molecules. SnapDock was validated on the ZLAB 4.0 Docking Benchmark (Hwang *et al.*, 2010) using the Dockground (Douguet *et al.*, 2006) and PIFACE (Cukuroglu *et al.*, 2014) template libraries. To mimic "real-life" large-scale docking, we applied SnapDock with the PIFACE library to model the interfaces of all the (non-redundant) complexes (some of them multimeric), which have been added to the PDB from February 26, 2014 and till September 19, 2016 (PIFACE was published in January 2014). SnapDock received results with at most 5 Å ligand RMSD deviation in one of the interfaces for 1284 out of 3547 complexes (34.83%), while running the *ab initio* docking PatchDock algorithm (Duhovny *et al.*, 2002) resulted in a success rate of 23.28%. Notably, about 1/3 of the PatchDock "hits" were different from the ones of SnapDock, resulting in a combined success ratio of 41.89%. These results agree with the findings of Vreven *et al.* (2014) regarding the partial complementarity of template-based and template-free docking methodologies.

2 Materials and methods

The SnapDock TBD algorithm is a multi-stage protocol for protein–protein docking, using available protein–protein interface template information in its modelling procedure. The stages of the algorithm include the processing of input template libraries, a novel structural alignment procedure [inspired by the Geometric Hashing (Lamdan and Wolfson, 1988) approach originated in computer vision applications] and the use of previously developed software for interface refinement and protein flexibility detection. The structural alignment procedure is similar in its mathematical principles to the geometric docking method that we previously developed for PatchDock (Duhovny *et al.*, 2002).

The SnapDock algorithm is divided into two major stages:

1. *Preprocessing stage*—In this stage, a template library of protein–protein interactions is preprocessed in order to extract critical structural features from the interacting interfaces and their surroundings. The extracted feature set is stored as a file on a persistent storage—named the table file, later to be used by the query stage.
2. *Query stage*—In this stage, given an input query of two protein molecules, the algorithm predicts their binding complex where the interface induced by their binding conformation is structurally similar to an interface in the template library. The algorithm uses the features extracted in the preprocessing stage to structurally align the interfaces and to create a superimposed binding complex model.

2.1 Template libraries

The quality of the results that the algorithm produces is highly dependent on the selection of the underlying template library. On the one hand, the library must be diverse enough to cover all the different interfaces currently represented in the Protein Data Bank. On the other hand, it must be restricted enough to filter out erroneous, low-quality and biologically irrelevant interfaces. We selected and tested two different state of the art libraries which tackled the task by two slightly different approaches.

1. The DOCKGROUND library (Anishchenko *et al.*, 2015) for template docking is a manually curated set of interfaces that was computationally refined and optimized for high quality. The library consists of 5936 non-redundant protein-protein interfaces.
2. The PIFACE library (Cukuroglu *et al.*, 2014) was created by clustering all the available protein–protein interfaces in the Protein Data Bank. The approach that was used is construction of a protein interface similarity network graph and finding communities (sub-graphs with dense inter-connectivity) in the network. The library (published in January 2014) consists of 22 604 non-redundant protein–protein interfaces.

Both libraries were tested and the results produced by the SnapDock algorithm were evaluated.

2.2 Feature extraction

A *base* is defined as a set of features extracted from the structure of a protein molecule, which is both sufficient for unambiguous definition of a 3D Euclidean reference frame and also enables the definition of a structural signature that is invariant to rigid 3D motion (rotation and translation). In SnapDock, we use both the coordinates of C_α/C_β atoms and the Secondary Structure Elements to define such bases. Specifically, for each two residues *A*, *B* on a given protein backbone we defined a base if those residues meet the following criteria:

1. The residues *A* and *B* are not located on the same Secondary Structure Element of the protein.
2. The Euclidean distance between the C_α coordinates of the residues is at least 4 Å and at most 13 Å.

The base is defined by the two $\overrightarrow{C_\alpha C_\beta}$ vectors of the residues, while the 3D motion invariant signature is defined by the 4-tuple (d, α, β, ω) as follows (see Fig. 1):

1. The Euclidean distance d between the C_α coordinates.
2. The angles α, β formed between the line segment connecting the C_α atoms and the line segments $\overrightarrow{C_\alpha C_\beta}$ for each of the residues.
3. The torsion angle ω between the plane induced by the C_α coordinates and the C_β of the first residue and the plane induced by the C_α coordinates and the C_β of the second residue.

The signature is the index/key used to insert the bases into a hash table in order to find matching bases. Two bases are considered matching, if their signature parameters are close enough, up to the following thresholds:

1. Euclidean distance difference: $\Delta(d) < 1.5\,Å$
2. Angles difference: $\Delta(\alpha) < 0.4^c\,\Delta(\beta) < 0.4^c$ and $\Delta(\omega) < 0.5^c$
3. Sum of angle differences: $\Delta(\alpha) + \Delta(\beta) + \Delta(\omega) < 0.9^c$

2.3 Preprocessing stage

During the preprocessing stage, the bases that represent the model information of a protein–protein interface template library are extracted. We filter the residues and only consider the ones that are located in the vicinity of the binding site (up to 12 Å). For each template, the bases are stored in a 4D-quantized hash table, where the 4D key is the structural signature of the base. The bin size, for each dimension, is the corresponding matching threshold for the

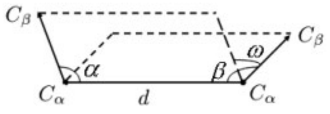

Fig. 1. A base formed by two residues and their C_α and C_β atoms

signature, as described in Section 2.2. The collection of hash tables is serialized to a file and stored in a persistent storage.

2.4 Docking protocol—query stage

2.4.1 Matching

We begin with extracting bases from the docked molecules and matching them to the pre-computed template bases loaded from the table file. The bases are matched quickly by performing a signature look-up in the hash table. Each match defines a rigid transformation (alignment) between one of the docked molecules and the corresponding template interface (Fig. 2).

2.4.2 Clustering and voting

Each matching pair of bases uniquely defines a rigid 3D transformation (pose), but provides little evidence about the alignment of the entire binding site. The entire collection of matched pairs is used to assess which poses represent a high-confidence structural alignment. This is done by using the Pose Clustering (Stockman, 1987) voting technique. The rational is that a good structural alignment will have a high number of locally matched base pairs that yield the same (or very close) pose.

The generated poses are stored in a “voting” hash table using the six-dimensional transformation parameters (for rotation and translation), such that each bin in the table represents a voxel in this 6D space. The number of poses for each bin is counted and only bins with a count exceeding the “above-expected” threshold (typically 10–15 matches) are passed to the next stage. Later, RMSD clustering of the poses is applied to eliminate the redundancy of almost-similar results and to reduce the number of possible solutions.

In order to determine the above-mentioned threshold we model the pose-voting process as an occupancy problem of placing m balls (poses) into n bins. Thus, in a random placement procedure the probability of having at least k balls in any given bin can be bounded by:

$$Pr[X \geq k] \leq \binom{m}{k}\left(\frac{1}{n}\right)^{k}$$

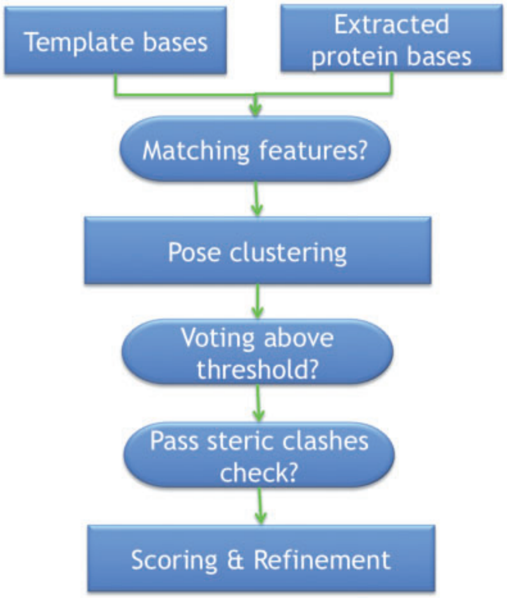

Fig. 2. Top level flow chart of the docking protocol of SnapDock

Consequently, for a given number of matching features we set the threshold k to be such that the probability of passing the threshold in a random voting procedure is relatively small. Specifically, in the described experiments we set the voting threshold to satisfy $Pr[X \geq k] \leq 2^{-10}$.

Each pose is an alignment of one of the docked proteins to one side of the template interface. The ligand–receptor complex pose is obtained by simultaneously superimposing both docked proteins on both sides of the interface. As each side might have few alignment candidates, an exhaustive all-to-all enumeration of all possible superimpositions is done (the number of poses passing the voting is usually small). The generated complexes are passed for filtering in the next step.

2.4.3 Filtering and refinement

Since the poses are computed based on local structural alignment in the vicinity of the template binding site, they may produce potential complexes that are physically impossible with unacceptable steric clashes. To filter out the undesired results, we test for clashes using a distance transform grid, similar to the one applied in the PatchDock algorithm (Duhovny *et al.*, 2002). Using the grid we can calculate for each transformed surface point of the ligand its distance from the surface of the receptor. If the distance is above a predefined penetration threshold (5 Å), the result is rejected. For the results that are not rejected the PatchDock geometric complementarity score is calculated. This score favours shape complementarity and penalizes the remaining steric clashes.

At this point, we have a set of roughly rigidly docked complexes. We use the FiberDock flexible refinement algorithm (Mashiach *et al.*, 2010) to considerably improve the docking accuracy of the associated proteins and bring them to a near-native orientation. The refinement algorithm accounts for both limited backbone and side chain flexibility. The results are reranked using the FiberDock calculated energy score.

2.5 Parallel computation

One way to boost the performance of the query phase is to parallelize the computation. Fortunately, the work to be done is *Embarrassingly Parallel*. The bases for each template can be matched with the docked molecules concurrently. The SnapDock implementation can spawn N worker threads (configured by the user) where the templates are distributed between the threads using a job queue. Once a thread is done working with a given template, it pulls the next template to process from the head of the queue.

2.6 Flexible TBD

The basic docking protocol can be extended to accommodate additional scenarios, including protein flexibility. Below we describe an automated docking protocol to account for large scale motions of the proteins backbone. It operates similar to the FlexDock algorithm (Schneidman-Duhovny *et al.*, 2005) (Figs. 3 and 4).

1. Partition into rigid parts—By applying the HingeProt method (Emekli *et al.*, 2008) on the given input protein, we detect the rigid parts and the hinge regions connecting them. The method employs Elastic Network Model to efficiently calculate the partition of the protein.
2. Independent docking of the rigid parts—The matching, clustering and voting procedures are executed for each of the parts, producing a list of viable partial docking solutions for each part independently.

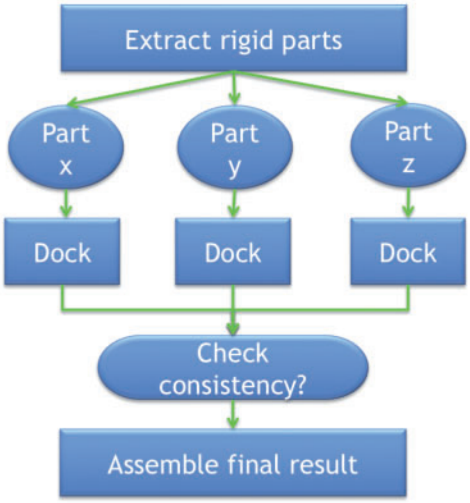

Fig. 3. Illustration of the layered solution graph. Nodes in each layer represent partial solutions of a given rigid part. The thick arrows represent assemblies of consistent solutions for all the rigid parts

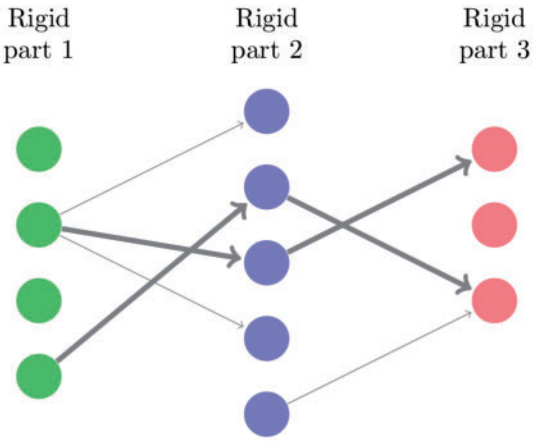

Fig. 4. Top level flow chart of the flexible docking protocol of SnapDock, using the building blocks of the rigid SnapDock protocol

3. Assembly of a consistent solution—Using a connected layered graph, where each layer contains nodes that represent all the possible solutions for one of the rigid parts. An edge is connecting two nodes if and only if the spatial distance between the N- and C-termini endpoints of the two consecutive rigid parts is below a given threshold (<5 Å). A path that travels through all the layers, connecting all the rigid parts, is a valid solution to the general docking problem. Afterwards, the solutions undergo filtering, scoring and refinement as previously described.

3 Results and discussion

3.1 Docking validation

To evaluate the performance of our algorithm we used the 154 test cases available in the ZLab benchmark 4.0 (Hwang *et al.*, 2010). The test cases are classified into three difficulty classes: rigid-body, medium and hard, based on the degree of the conformational changes the molecules undergo upon association. Each test case includes three structures. The structure of the two unbound molecules to be docked as well as the structure of their co-crystallized complex, so we can evaluate our predicted complex by a ligand–RMSD metric to the native bound complex counterpart.

Table 1. SnapDock success rate for finding a near native (ligand RSMD <5 Å) for the Zlab Benchmark 4.0 test cases, for each one of the used template libraries

	DOCKGROUND	PIFACE
X rigid-body	(48/109) 44.03%	(83/109) 73.39%
Medium	(5/25) 20%	(11/25) 44%
Hard	(4/20) 20%	(4/20) 20%

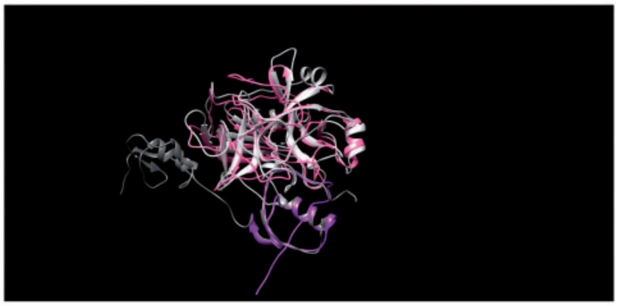

Fig. 5. The docking solution of 2CGA:B (pink) with 1HPT:A (purple) using the template 1TBQ H:R (gray)

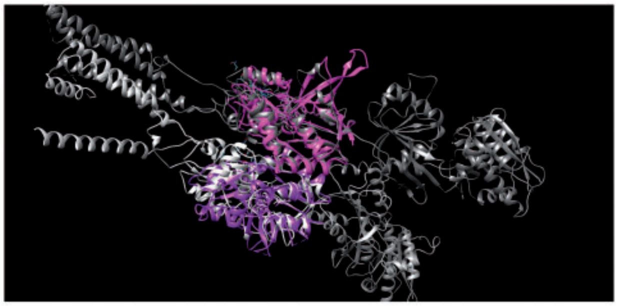

Fig. 6. The docking solution of 1Y20:A (pink) with 2A5S:A (purple) using the template 3KG2 A:D (gray)

Table 1 includes the success rates for the total dataset, for each one of the difficulty categories. We define a near-native (successful) result with C_α ligand RMSD cutoff of <5 Å that appears among the top 10 solutions that the algorithm produces.

We will discuss in detail few representative solutions we encountered while running the benchmark. The test case involving docking of Bovine Chymotrypsinogen (2CGA:B) (Wang *et al.*, 1985) and Bovine Plasma Retinol-binding Protein (1HPT:A) (Hecht *et al.*, 1992) was solved using the template interface of the double domain Kazal inhibitor rhodniin in complex with thrombin (1TBQ H:R) (van de Locht *et al.*, 1995), illustrated in Figure 5. The template interface includes a double domain binding to two different binding sites. SnapDock aligned the docked molecules on one of the domains. The sequence similarities between the docked molecules and the template are 47 and 52%, respectively.

The test case involving docking NR1 Ligand Binding Core (1Y20:A) (Inanobe et al., 2005) and NR2A Ligand Binding Core (2A5S:A) (Furukawa *et al.*, 2005) was solved using the template interface between two Glutamate receptor membrane proteins (3KG2 A:D) (Sobolevsky *et al.*, 2009), illustrated in Figure 6. The docking solution in this case is a partial structural alignment of the template interface. The sequence similarities between the docked molecules and the template are 59 and 20%, respectively.

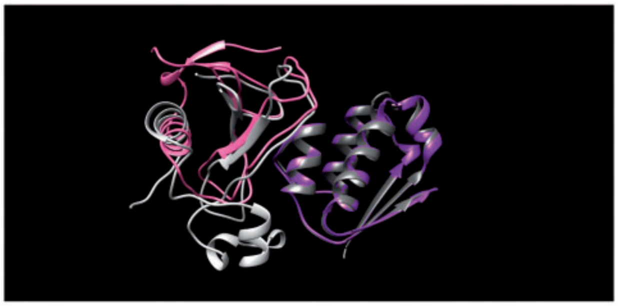

Fig. 7. The docking solution of 1RGH:B (pink) with 1A19:A (purple) using the template 1B27 A:D (gray)

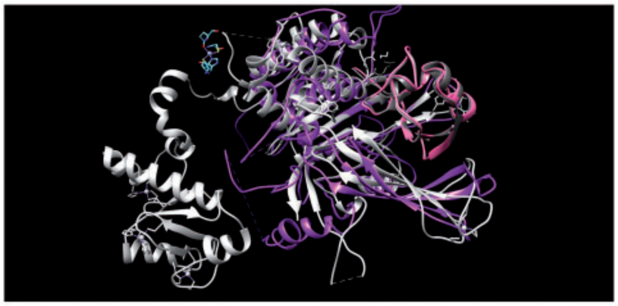

Fig. 8. The docking solution of 2AYN:A (purple) with 2FCN:A (pink) using the template 3MHS A:D (gray)

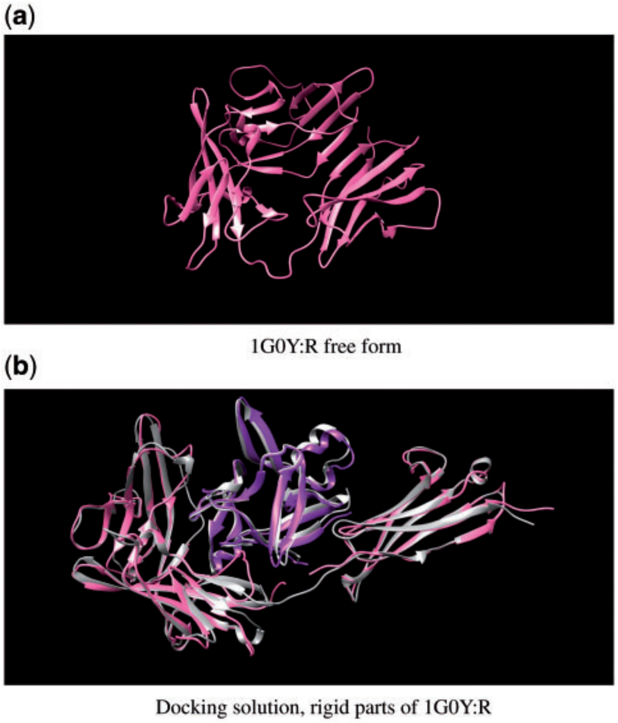

Fig. 9. The flexible docking solution of 1G0Y:R (pink) with 1ILR:1 (purple) using the template 1IRA X:Y (gray)

The test case involving docking of Ribonuclease Hydrolase (1RGH:B) (Sevcik *et al.*, 1996) and Barstar Ribonuclease Inhibitor (1A19:B) (Ratnaparkhi *et al.*, 1998) was solved using the template of a variation of the Ribonuclease–Barstar complex (1B27 A:D) (Vaughan *et al.*, 1999), illustrated in Figure 7. The Ribonuclease of the docked molecule is taken from the Bacillus amyloliquefaciens organism while in the template it was taken from the *Streptomyces aureofaciens* organism, having only 45% sequence similarity.

The test case involving docking of Ubiquitin carboxyl-terminal hydrolase 14 (2AYN:A) (Hu *et al.*, 2005) and Ubiquitin (2FCN:A) (Bang *et al.*, 2006) was solved using the template of the complex of Ubiquitin carboxyl-terminal hydrolase 8 and Ubiquitin (3MHS A:D) (Samara *et al.*, 2010), illustrated in Figure 8. The sequence similarity between the two different Ubiquitin carboxyl-terminal hydrolase (14 versus 8) is 44%.

The flexible SnapDock version was applied to the test case involving the docking of Interleukin-1 Receptor Type I (1G0Y:R) (Vigers *et al.*, 2000) with Interleukin-1 Receptor Antagonist Protein (1ILR:1) (Schreuder *et al.*, 1995). The test case was solved using the template of their crystallized complex (1IRA X:Y) (Schreuder *et al.*, 1997), illustrated in Figure 9. HingeProt detected that the Interleukin Receptor has two rigid parts, the rigid parts were docked independently and the partial solutions were assembled to a consistent overall solution.

3.2 Runtime performance

The major advantage of the SnapDock algorithm (and all methods using Geometric Hashing) is the rapid performance, enabling to test two potential docking candidates against the entire known collection of protein–protein interfaces in mere minutes. The algorithm was evaluated on a desktop workstation with two quad-core Intel i7-2600 3.4 GHz CPUs (8 cores total) and 16 GB DDR3 SDRAM memory. The algorithm was implemented as a multi-threaded programme fully utilizing all the available computational cores of the machine.

Table 2. Average running time and consumed physical disk space measured while running the benchmark in Section 3.1

	DOCKGROUND	PIFACE
Number of interfaces	5936	22604
Average run-time (s)	41.19	109.86
Table size (MB)	731.93	1630.32

Table 2 displays the average measured running time of the SnapDock algorithm working on the Zlab Benchmark 4.0, as well as the physical disk space consumed by the pre-computed feature tables.

To compare the speed of the algorithm to the PRISM method we have run the same docking experiment with MultiProt (Shatsky *et al.*, 2004) as the structural alignment method. We have used the same input molecules and the same extracted binding sites from the template library (as the alignment run time is dependent on the size of molecules). We measure the total run time of running the benchmark and divide it by the number of CPU cores used and by the number of template interfaces to get the average per-template running time. The results are shown in Table 3, where the structural alignment module of SnapDock is shown to be more than 10 times faster compared with MultiProt.

3.3 Interaction prediction

One major application of docking software is blind prediction of protein–protein interactions. It would be insightful to analyse how the SnapDock algorithm can benefit in “real life” scenarios. We tested the algorithm’s ability to predict the protein–protein interactions in a set of independent biological data. The collection of

Table 3. Average running time per-template measured while running the benchmark in Section 3.1

	SnapDock	MultiProt
Average run-time (s)	0.0388	0.4068

Table 4. Success rate of finding a successful result for a given interface

	Success rate
Under <10 Å	(2557/8964) 28.53%
Under <5 Å	(2315/8964) 25.83%

Table 5. Success rate of finding at least one successful interface in a PDB entry

	Success rate
Under <10 Å	(1412/3686) 38.31%
Under <5 Å	(1284/3686) 34.83%

Table 6. Comparison of the success rate per method of finding a successful result for a given interface

	PatchDock	SnapDock	Combined
Under <10 Å	17.75% (1591/8964)	28.53% (2557/8964)	33.58% (3010/8964)
Under <5 Å	12.73% (1141/8964)	25.83% (2315/8964)	29.12% (2610/8964)

Table 7. Comparison of the success rate per method of finding a successful result of at least one interface in a PDB entry

	PatchDock	SnapDock	Combined
Under <10 Å	31.15% (1148/3686)	38.31% (1412/3686)	48.02% (1770/3686)
Under <5 Å	23.28% (858/3686)	34.83% (1284/3686)	41.89% (1544/3686)

“future” experimentally determined structures from the PDB, which was not available at the creation time of the template library, was selected as a qualifying data set. A re-docking (bound docking) trial was run to roughly evaluate how the algorithm could a priori predict the experimental results.

The PIFACE library was chosen for this experiment. The template library was published in January 2014. Thus, we extracted all protein–protein interfaces in the PDB with a deposition date later February 26, 2014, up to September 19, 2016, when our experiment was conducted. Redundant interacting pairs were filtered out with a threshold of up to 50% sequence similarity. We remained with 8964 non-redundant protein–protein interfaces from 3686 different PDB entries. A docking experiment was run to evaluate whether our algorithm was able to predict those interactions. The success rate was measured as before—a solution with a given ligand-RMSD cutoff appearing in the top 10 ranked results of the algorithm. The success rates are shown in Table 4, for predicting protein-protein interfaces and in Table 5, for predicting at least a single protein-protein interface in each PDB entry.

Finally, we evaluate the performance of the algorithm by comparing it to the performance of template-free docking methods. The same dataset was used again for a docking experiment with the PatchDock *ab initio* method. The success rate was selected as before, considering the top 10 ranked solutions of the algorithm.

We also present the success rate by independently combining the two methods. The 10 top solutions here are simply the top 5 solutions of SnapDock and the top 5 solutions of PatchDock. The comparison between the success rates is shown in Table 6, for predicting protein-protein interfaces and in Table 7, for predicting at least a single protein-protein interface in each PDB entry.

4 Conclusions

We have presented a new TBD algorithm, SnapDock, which utilizes an interface template library to perform Geometric Hashing-based structural alignment of the putative docking partners on all the library templates, discards solutions with severe steric clashes and, finally, refines the surviving modelled interfaces by allowing side chain and limited backbone flexibility of the interacting proteins while ranking them by global energy. The method is highly efficient due to the preprocessing of the template library, which converts it into a Geometric Hashing table. Thus, docking with the Dockground template library, which includes almost 6000 templates took on the average 41 s per protein pair from the ZLAB version 4.0 benchmark, while docking with the 22 600 template PIFACE library took on the average 110 s per protein pair. We have also demonstrated a flexible docking version of the algorithm which performed well on several test cases. A large scale blind docking experiment, which aimed to model all the interfaces in the PDB, which have been uploaded after the completion of the PIFACE template library (structures uploaded in the period February 2014–September 2016), yielded a 35% success rate for SnapDock, 23% success rate for PatchDock with a combined (non-overlapping) success rate of 42%. Future work will include introduction of additional protein shape flexibility options in the docking scheme.

Funding

This research was supported by the Israel Science Foundation (Grant No. 1112/12), the I-CORE program of the Budgeting and Planning Committee and the Israel Science Foundation (Center No. 1775/12), by a grant from the Ministry of Science, Technology & Space, Israel & the Ministry of Foreign Affairs and International Cooperation General Directorate for Political Affairs & Security Italian Republic, and by the Hermann Minkowski Minerva Geometry Center.

Conflict of Interest: none declared.

References

Andrusier,N. *et al.* (2008) Principles of flexible protein–protein docking. *Proteins: Struct. Funct. Bioinform.*, **73**, 271–289.

Anishchenko,I. *et al.* (2015) Structural templates for comparative protein docking. *Proteins: Struct. Funct.Bioinform.*, **83**, 1563–1570.

Bang,D. *et al.* (2006) Dissecting the energetics of protein α-helix c-cap termination through chemical protein synthesis. *Nat. Chem. Biol.*, **2**, 139–143.

Bonvin,A.M. (2006) Flexible protein–protein docking. *Curr. Opin. Struct. Biol.*, **16**, 194–200.

Cukuroglu,E. *et al.* (2014) Non-redundant unique interface structures as templates for modeling protein interactions. *PLoS One*, **9**, e86738.

Douguet,D. *et al.* (2006) Dockground resource for studying protein–protein interfaces. *Bioinformatics*, **22**, 2612–2618.

Duhovny,D. *et al.* (2002). Efficient unbound docking of rigid molecules. In *International Workshop on Algorithms in Bioinformatics*, pp. 185–200. Springer.

Emekli,U. *et al*. (2008) Hingeprot: automated prediction of hinges in protein structures. *Proteins: Struct. Funct.Bioinform*., **70**, 1219–1227.

Furukawa,H. *et al*. (2005) Subunit arrangement and function in nmda receptors. *Nature*, **438**, 185–192.

Gao,M., and Skolnick,J. (2010) Structural space of protein–protein interfaces is degenerate, close to complete, and highly connected. *Proc. Natl Acad. Sci*., **107**, 22517–22522.

Halperin,I. *et al*. (2002) Principles of docking: an overview of search algorithms and a guide to scoring functions. *Proteins: Struct. Funct. Bioinform*., **47**, 409–443.

Hecht,H. *et al*. (1992) Three-dimensional structure of a recombinant variant of human pancreatic secretory trypsin inhibitor (kazal type). *J. Mol. Biol*., **225**, 1095–1103.

Hu,M. *et al*. (2005) Structure and mechanisms of the proteasome-associated deubiquitinating enzyme usp14. *EMBO J*., **24**, 3747–3756.

Huang,S.-Y. (2014) Search strategies and evaluation in protein–protein docking: principles, advances and challenges. *Drug Disc. Today*, **19**, 1081–1096.

Hwang,H. *et al*. (2010) Protein–protein docking benchmark version 4.0. *Proteins: Struct. Funct. Bioinform*., **78**, 3111–3114.

Inanobe,A. *et al*. (2005) Mechanism of partial agonist action at the nr1 subunit of NMDA receptors. *Neuron*, **47**, 71–84.

Janin,J. (2005) Assessing predictions of protein–protein interaction: the capri experiment. *Protein Sci*., **14**, 278–283.

Janin,J. *et al*. (2015) Assessing structural predictions of protein–protein recognition: the capri experiment. *Rev. Comput. Chem*., **28**, 137–173.

Kundrotas,P.J. *et al*. (2012) Templates are available to model nearly all complexes of structurally characterized proteins. *Proc. Natl Acad. Sci*., **109**, 9438–9441.

Lamdan,Y., and Wolfson,H.J. (1988). Geometric hashing: a general and efficient model-based recognition scheme. In *Proceedings of the 2nd International Conference on Computer Vision*, IEEE Computer Society, pp. 238–249.

Mashiach,E. *et al*. (2010) Fiberdock: flexible induced-fit backbone refinement in molecular docking. *Proteins: Struct. Funct. Bioinform*., **78**, 1503–1519.

Muratcioglu,S. *et al*. (2015) Advances in template-based protein docking by utilizing interfaces towards completing structural interactome. *Curr. Opin. Struct. Biol*., **35**, 87–92.

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

Ogmen,U. *et al*. (2005) Prism: protein interactions by structural matching. *Nucleic Acids Res*., **33**, W331–W336.

Ratnaparkhi,G.S. *et al*. (1998) Discrepancies between the NMR and x-ray structures of uncomplexed barstar: analysis suggests that packing densities of protein structures determined by NMR are unreliable. *Biochemistry*, **37**, 6958–6966.

Samara,N.L. *et al*. (2010) Structural insights into the assembly and function of the saga deubiquitinating module. *Science*, **328**, 1025–1029.

Schneidman-Duhovny,D. *et al*. (2005) Geometry-based flexible and symmetric protein docking. *Proteins: Struct. Funct. Bioinform*., **60**, 224–231.

Schreuder,H. *et al*. (1997) A new cytokine-receptor binding mode revealed by the crystal structure of the il-1 receptor with an antagonist. *Nature*, **386**, 194.

Schreuder,H.A. *et al*. (1995) Refined crystal structure of the interleukin-1 receptor antagonist. *Eur. J. Biochem*., **227**, 838–847.

Sevcik,J. *et al*. (1996) Ribonuclease from streptomyces aureofaciens at atomic resolution. *Acta Crystallographica Section D: Biol. Crystallogr*., **52**, 327–344.

Shatsky,M. *et al*. (2004) A method for simultaneous alignment of multiple protein structures. *Proteins: Struct. Funct. Bioinform*., **56**, 143–156.

Smith,G.R., and Sternberg,M.J. (2002) Prediction of protein–protein interactions by docking methods. *Curr. Opin. Struct. Biol*., **12**, 28–35.

Sobolevsky,A.I. *et al*. (2009) X-ray structure, symmetry and mechanism of an ampa-subtype glutamate receptor. *Nature*, **462**, 745–756.

Stockman,G. (1987) Object recognition and localization via pose clustering. *Comput. Vis. Graph. Image Process*., **40**, 361–387.

Stumpf,M.P. *et al*. (2008) Estimating the size of the human interactome. *Proc. Natl Acad. Sci*., **105**, 6959–6964.

Szilagyi,A., and Zhang,Y. (2014) Template-based structure modeling of protein–protein interactions. *Curr. Opin. Struct. Biol*., **24**, 10–23.

Tuncbag,N. *et al*. (2011) Predicting protein-protein interactions on a proteome scale by matching evolutionary and structural similarities at interfaces using prism. *Nat. Protoc*., **6**, 1341–1354.

van de Locht,A. *et al*. (1995) Two heads are better than one: crystal structure of the insect derived double domain kazal inhibitor rhodniin in complex with thrombin. *EMBO J*., **14**, 5149.

Vaughan,C.K. *et al*. (1999) Structural response to mutation at a protein-protein interface. *J. Mol. Biol*., **286**, 1487–1506.

Vigers,G.P. *et al*. (2000) X-ray crystal structure of a small antagonist peptide bound to interleukin-1 receptor type 1. *J. Biol. Chem*., **275**, 36927–36933.

Vreven,T. *et al*. (2014) Evaluating template-based and template-free protein–protein complex structure prediction. *Brief. Bioinform*., **15**, 169–176.

Wang,D. *et al*. (1985) Bovine chymotrypsinogen a: X-ray crystal structure analysis and refinement. Of a new crystal form at 1.8 Å resolution. *J. Mol. Biol*., **185**, 595–624.