Geometry-Based Flexible and Symmetric Protein Docking

Dina Schneidman-Duhovny,[1†] Yuval Inbar,[1†] Ruth Nussinov,[2,3‡] and Haim J. Wolfson[1*]
[1]*School of Computer Science, Beverly and Raymond Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel*
[2]*Basic Research Program, SAIC-Frederick, Inc., Laboratory of Experimental and Computational Biology, NCI–Frederick, Frederick, Maryland*
[3]*Department of Human Genetics and Molecular Medicine, Sackler Faculty of Medicine, Tel Aviv University, Tel Aviv, Israel*

ABSTRACT **We present a set of geometric docking algorithms for rigid, flexible, and cyclic symmetry docking. The algorithms are highly efficient and have demonstrated very good performance in CAPRI Rounds 3–5. The flexible docking algorithm, FlexDock, is unique in its ability to handle any number of hinges in the flexible molecule, without degradation in run-time performance, as compared to rigid docking. The algorithm for reconstruction of cyclically symmetric complexes successfully assembles multimolecular complexes satisfying C_n symmetry for any n in a matter of minutes on a desktop PC. Most of the algorithms presented here are available at the Tel Aviv University Structural Bioinformatics Web server (http://bioinfo3d.cs.tau.ac.il/). Proteins 2005;60:224–231.**

Key words: CAPRI; unbound docking; flexible docking; symmetric docking; PatchDock; FlexDock; SymmDock

INTRODUCTION

In this article we outline the principles of the docking algorithms used by the Tel Aviv University Structural Bioinformatics team in the CAPRI challenge Rounds 1–5.[1,2] Both the rigid docking algorithm, PatchDock,[3] and the flexible docking algorithm, FlexDock, significantly improved the performance of our previous rigid[4,5] and flexible[6] docking methods. The development of SymmDock, which performs rigid docking with cyclic symmetry constraints, was directly motivated by the CAPRI targets, such as Targets 9 and 10. The above mentioned targets also motivated a further development, namely, the combination of the flexible and symmetric docking algorithms, FlexSymmDock. Below we provide the algorithmic details of each method, followed by experimental results. The application of the algorithms to the targets of Rounds 3–5 and the analysis of their performance are outlined in a companion paper in this issue of *Proteins*.[2]

The detection of the docking transformations in these algorithms is based on integration of local shape complementarity evidence. This integration is done by the efficient geometric hashing technique[7] and its adaptations for the docking task. This local approach results in the high computational efficiency of the algorithms, which run in minutes on a "standard" Pentium IV PC for a typical CAPRI docking task. This enables potential docking of a protein against a database of targets. The popular Fast Fourier Transform (FFT)-based docking methods, which follow Katzir et al.,[8] perform a global transformation search that is time-consuming. The local approach also allows efficient integration of biologically derived binding-site information, which proved essential for the good performance of all the methodologies.[9] Incorporation of such information in our algorithms further reduces the run time and, especially, improves performance by reducing the number of false positives. A potential downside of a local approach is the possibility to lose the near-native solutions in early filtering stages. However, due to the intrinsically redundant nature of geometric hashing, where many different locally aligned features can "vote" for the same docking solution, near-native transformations are not eliminated unless severe conformational changes are involved. In CAPRI Rounds 3–5 we had at least an "acceptable" solution (according to the CAPRI assessors' definition[10] of acceptable, medium, and high quality of solutions) among the 10 top ones for all the targets except Target 13.

While most of the existing protein–protein docking algorithms are rigid, it is clear that for real-life docking tasks, one needs algorithms that can handle protein backbone flexibility, including significant hinge motions. Already, Target 1 of CAPRI involved such flexibility.[11] FlexDock handles docking with hinge-bending motions in one of the molecules without increasing the order of computational complexity compared to PatchDock. While it does require a priori knowledge of potential hinge

[†] These authors contributed equally to the work.

[‡] The publisher or recipient acknowledges the right of the U.S. Government to retain a nonexclusive, royalty-free license in and to any copyright covering the article.

Grant sponsor: Center of Excellence in Geometric Computing and Its Applications, funded by the Israel Science Foundation (administered by the Israel Academy of Sciences). Grant sponsor: Hermann Minkowski-Minerva Center for Geometry at Tel Aviv University (H. J. Wolfson). Grant sponsor: Federal funds from the National Cancer Institute, National Institutes of Health; Grant number: NO1-CO-12400. The content of this publication does not necessarily reflect the view or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organization imply endorsement by the U.S. Government.

*Correspondence to: Haim J. Wolfson, School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: wolfson@post.tau.ac.il

Received 16 January 2005; Accepted 7 March 2005
DOI: 10.1002/prot.20562

positions, its efficiency enables partial enumeration over such positions.

The development and implementation of SymmDock, which predicts the structure of a cyclically symmetric complex given its asymmetric unit, was directly motivated by the CAPRI challenge targets. We were not the first to develop a symmetric docking algorithm.[12,13] However, to the best of our knowledge, this is the only docking algorithm that works for any C_n complex by limiting the search space to symmetric transformations at its very beginning, thus spending no computational effort on generating candidate transformations that do not satisfy the symmetry constraints. The restriction to the C_n symmetry is achieved by clustering candidate symmetry axes, each of which is induced by a pair of matching interface points. Such a clustering approach is in the spirit of Generalized Hough Transform–type techniques.[14,15]

Since the above mentioned algorithms do not include a side-chain conformation refinement stage, they are expected to achieve only roughly accurate solutions, which do include minor steric clashes. Indeed, although we did solve almost all of the targets in Round 3–5, our solutions have been of medium or acceptable quality. Algorithms such as ICM[16] or RosettaDock[17] are able to achieve high accuracy due to the refinement of the side-chain conformations. We intend to introduce such a refinement stage for the high-scoring "soft-docking" solutions.

Both the PatchDock and the SymmDock algorithms are available as free Web services at the Tel Aviv University Structural Bioinformatics group site (http://bioinfo3d.cs.tau.ac.il/), with an average response time of several minutes.[18]

RIGID DOCKING

PatchDock[3] (http://bioinfo3d.cs.tau.ac.il/PatchDock/) is a molecular docking algorithm based on shape complementarity principles. Small-scale flexibility is taken into account implicitly by allowing some extent of steric clashes. The input is 2 molecules [given in the Protein Data Bank (PDB) format] of any type: proteins, DNA, peptides, and drugs. The output is a list of potential complexes sorted by their geometric shape complementarity score. Below we provide a short description of the algorithm, since it is a major ingredient in the novel flexible docking method and is also exploited in the symmetric docking.

The PatchDock algorithm exploits the fact that in order to compute a rigid motion between a pair of molecules, it is enough to detect local complementarity between the shapes (in fact, one corresponding triplet of points or a pair of complementary surface normals suffices). Thus, one can focus directly on aligning complementary features, without the need to perform an exhaustive search in the 6 dimensional (6D) space of rotations and translations. In practice, the surfaces of the molecules are first divided into patches according to rough surface shape (concave, convex, or flat). Complementary patches are identified and superimposed by shape-matching techniques. The algorithm has 3 major stages:

- Molecular Shape Representation. In this step we compute the sparse molecular surface of the molecule.[19] Next, we apply a surface segmentation algorithm that partitions the surface according to local shape curvature into concave, convex, and flat surface patches.
- Surface Patch Matching. We apply a hybrid of the geometric hashing and pose-clustering matching techniques to match "critical" surface points of the patches detected in the previous step. Concave patches are matched with convex ones and flat patches with any type of patches.
- Filtering and Scoring. The candidate complexes from the previous step are examined. We discard any complex with unacceptable steric clashes between the receptor and the ligand. Finally, the remaining candidates are ranked according to a geometric shape complementarity score, where surface contact is scored positively and "acceptable" steric clashes are penalized. Then, redundant solutions are removed by clustering based on a Root-Mean-Square Deviation (RMSD) between the C_α atoms clustering process.

For details of the PatchDock algorithm, the reader is referred to D. Duhovny et al.[3]

PatchDock enables incorporation of external information regarding potential binding sites. Given a list of "known" binding-site residues, the algorithm restricts the matching stage to patches that include these residues. In addition, the algorithm specifies for each candidate complex the minimal (user defined) percentage of binding-site residues area within the resolved interface. A classic example is antibody–antigen docking. There, the matching stage is typically restricted to the patches of the complementarity determining region (CDR) loops H3 and L3, and solutions with less than 30% H3 and L3 residues area in their interface are filtered out.

PatchDock was tested and verified on enzyme–inhibitor and antibody–antigen complexes from benchmark 0.0.[20] The algorithm is very efficient and finds the near-native solution in most of the cases, unless there are significant conformational changes upon binding. In March 2004, we launched a Web server[21] that enables users to run PatchDock (http://bioinfo3d.cs.tau.ac.il/PatchDock/). As of January 2005, over 1700 docking requests were handled. Notice that due to the efficient methodology of PatchDock, the average response time (delivered by e-mail) is in the order of minutes, though the server is installed on a standard PC. The efficiency of the algorithm is due to the a priori focusing on complementary shape characteristics, employment of geometric hashing, and an efficient scoring stage, which uses a multiresolution dot surface representation for an efficient detection of steric clashes and calculation of the geometric fit based on a Distance Transform Grid. Though the algorithm performs rigid docking, the filtering mechanism is tolerant to a small extent of steric clashes in order to handle cases that involve side-chain motions.

FLEXIBLE DOCKING

FlexDock is an algorithm for flexible hinge-bent protein–protein docking. The current version of the algorithm handles flexibility in one of the input molecules, assuming that the other molecule is rigid. While the number of hinges is not limited, we assume that the hinges impose a *chain-type topology* on the flexible molecule. In a chain-type topology, the rigid parts can be ordered so the hinges are placed between any 2 consecutive parts. Our algorithm is especially suitable for handling backbone flexibility, since hinges in the backbone of the molecule induce such a chain-type topology. Currently, FlexDock assumes that there is a significant contact between the rigid molecule and each part of the flexible molecule.

The algorithm docks all the rigid parts of the flexible molecule simultaneously. Then, it applies an efficient (polynomial time) graph-theoretic algorithm to detect the highest scoring consistent configurations of the candidate rigid part placements. In such a way the run-time performance of the algorithm remains of the same order of magnitude as the highly efficient PatchDock rigid docking algorithm.

The input of FlexDock consists of a rigid molecule and a flexible molecule, which is represented by an ordered set of its rigid parts and a list of connecting hinges. The output is a list of *multi-transformations*. Each multi-transformation represents 1 flexible docking solution. A multi-transformation is a list of transformations, one for each rigid part of the flexible molecule. The transformations place the rigid parts of the flexible molecule onto the rigid molecule, such that the obtained solution is consistent. A solution of flexible docking is consistent if it places the rigid parts such that (1) the hinge points common to neighboring rigid parts are almost overlapping, and (2) there is no (internal) steric clash between the transformed rigid parts of the flexible molecule.

The FlexDock algorithm has 2 major stages. The first stage performs rigid docking of all rigid parts. The second stage assembles the rigid docking results into consistent flexible solutions, producing a list of high-scoring multi-transformations. In the first stage, we apply the PatchDock algorithm described above for the docking of the rigid molecule with each rigid part of the flexible molecule. The input to the second assembly stage is a set of scored transformations for each rigid part. Each transformation set may include hundreds to thousands of candidates, in order to include a near-native solution even when it is not ranked high. We construct an *assembly graph*, where each transformation is represented by a node. The weight of the node is the (rigid docking) score of the transformation. Edges are added between nodes that represent consistent solutions of neighboring rigid parts. Source and target nodes serve as virtual start and end parts of the flexible molecule chain. The edges are directed from the source to the target node. The weight of an edge is the shape complementarity score between the transformed rigid parts of the flexible molecule. This weight accounts for additional self-contact between the rigid parts of the flexible molecule within the complex assembly. Such self-contact, though, is not mandatory. The assembly graph is directed and acyclic. An example of a graph for a flexible molecule with 3 rigid parts is shown in Figure 1(a). Each path between the source and target nodes represents a valid flexible docking solution. The multi-transformation that corresponds to the solution is the ordered list of transformations represented by each node in the path. The score of the path is the sum of the scores of its nodes and edges. The list of high-scoring, consistent multi-transformations is found by applying the dynamic programming algorithm to the constructed directed acyclic graph.

The presented algorithm performs fully flexible hinge-bent docking. There is no restriction on the amount of rotation at the hinge. The algorithm can handle multiple hinges in one of the molecules. It is important to mention that often a near-native solution is ranked high even when the rigid docking of the individual parts ranks the near-native solutions quite low on the rigid part docking list. This is due to the fact that the assembly stage favors hinge-consistent solutions. The algorithm is very efficient due to several factors: (1) Docking of each rigid part is performed only once during execution by a highly efficient rigid docking algorithm based on geometric hashing; (2) the assembly stage performs simultaneous construction of consistent multi-transformations from rigid docking solutions. The running times of the algorithm are comparable to the running times of the PatchDock rigid docking algorithm, namely, in the order of minutes on a desktop PC (see Experimental Results section).

CYCLIC SYMMETRY DOCKING

We have developed SymmDock, a docking-driven algorithm that predicts the structure of a cyclically symmetric complex given the structure of its asymmetric unit. A C_n symmetry type complex is a complex with a rotational symmetry of order n about a symmetry axis. The rotation angle α is of $360/n°$.

Let T be the symmetry transformation of a C_n symmetric complex C with an asymmetric unit U, then $T^0 = T^n = \mathrm{I}$ and $C = {}_{i=0}^{n-1}T^i(U)$. The complex contains n identical interfaces between the i and $i + 1$ units and the $n - 1$ unit and the original unit T^0 (U).

Our algorithm searches the symmetric transformation space for transformations that optimize the shape complementarity of the interface between neighboring units. Similar to our pairwise rigid docking algorithm PatchDock (see Rigid Docking section), the symmetric docking algorithm has 3 major stages: (1) surface representation; (2) surface matching; and (3) filtering and scoring.

SymmDock differs from PatchDock in 2 main modules: the matching module, where we limit the search to symmetric transformations only, and the clustering module, where we use the specific characteristics of symmetric transformations for the preliminary clustering. Other modules of the algorithm are similar to the ones in the PatchDock algorithm. In the sequel we present some geometric facts and focus on the 2 modules, which are specific to the SymmDock algorithm.

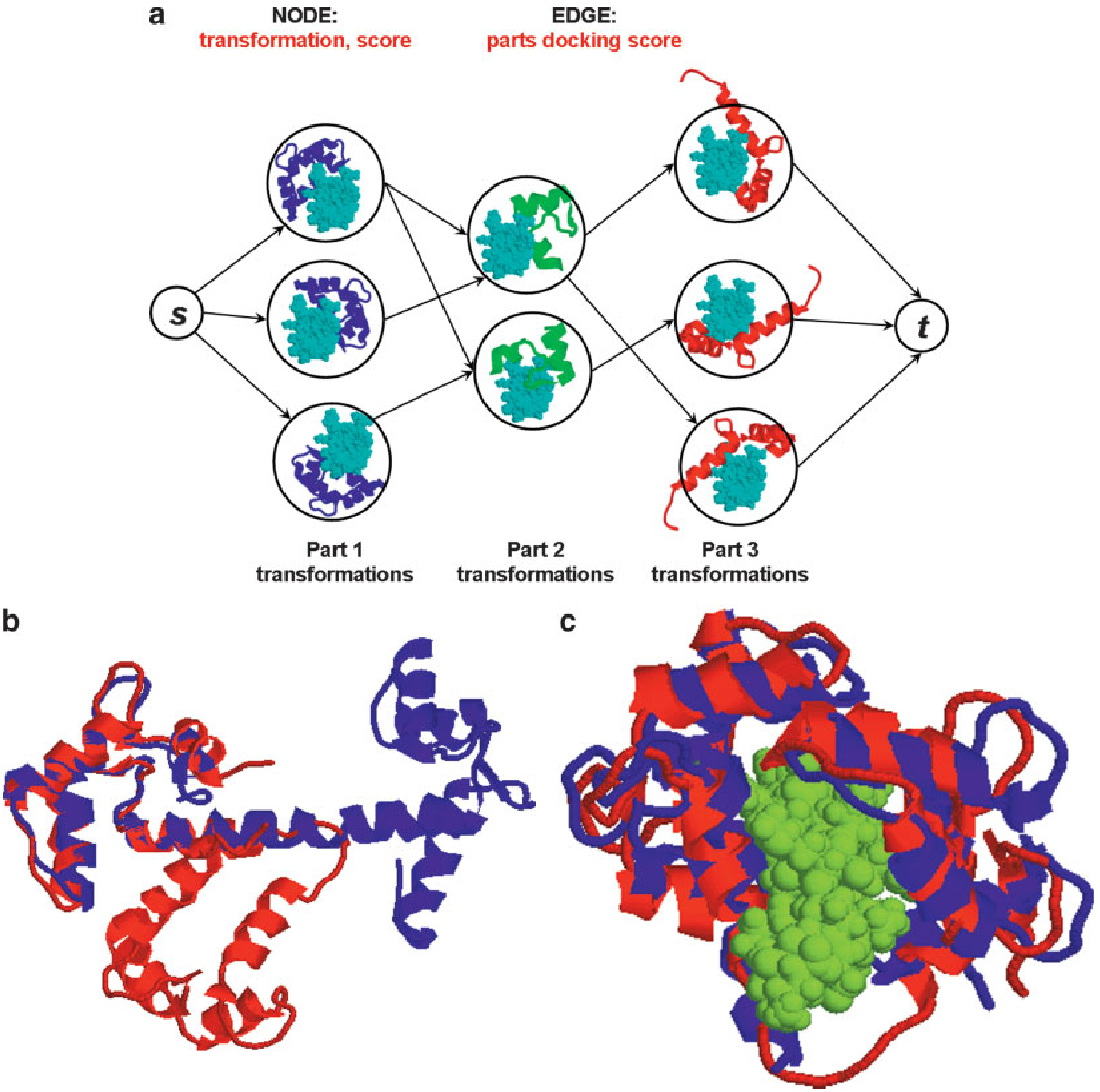

Fig. 1. (**a**) Illustration of FlexDock assembly graph for a target that consists of 3 rigid parts. (**b**) The alignment between the bound (PDB code: 2bbm, colored red) and the unbound (PDB code: 4cln, colored blue) structures of calmodulin. (**c**) The second best scoring FlexDock solution (RMSD = 2.12 Å) of the docking between the myosin kinase peptide and the unbound structure of calmodulin [as shown in (b) in blue]. The predicted solution of calmodulin is in blue. The conformation of the calmodulin in the bound complex is in red. The myosin kinase peptide is shown in green.

Geometric Preliminaries

Consider the interface between the consecutive units U and $T(U)$. We define the surface points A and B to be a *matching pair* under transformation T if $\|T(A) - B\| < \epsilon$, where ϵ is a surface density threshold. Simply put, (A,B) is a symmetric pair, if B in U and $T(A)$ in $T(U)$ are "almost touching" each other across the interface. Thus, any point on the symmetry axis of T is within the same distance from A and B. Moreover the distance of the axis from the middle of the AB segment is r, where r depends solely on the rotation angle α and the distance between A and B. Let $d = \|A - B\|$, then $r = \frac{d}{2}\cot\frac{\alpha}{2}$ [see Fig. 2(a)]. Notice that $n = 2 \Rightarrow r = 0$; therefore, the axis passes through the middle point between A and B. Let C be a circle of radius r that lies on the plane orthogonal to $\overrightarrow{AB}$ and centered at the middle of the AB segment. Any tangent l to the circle C (in its plane) is a potential symmetry axis of a transformation that rotates A to B.

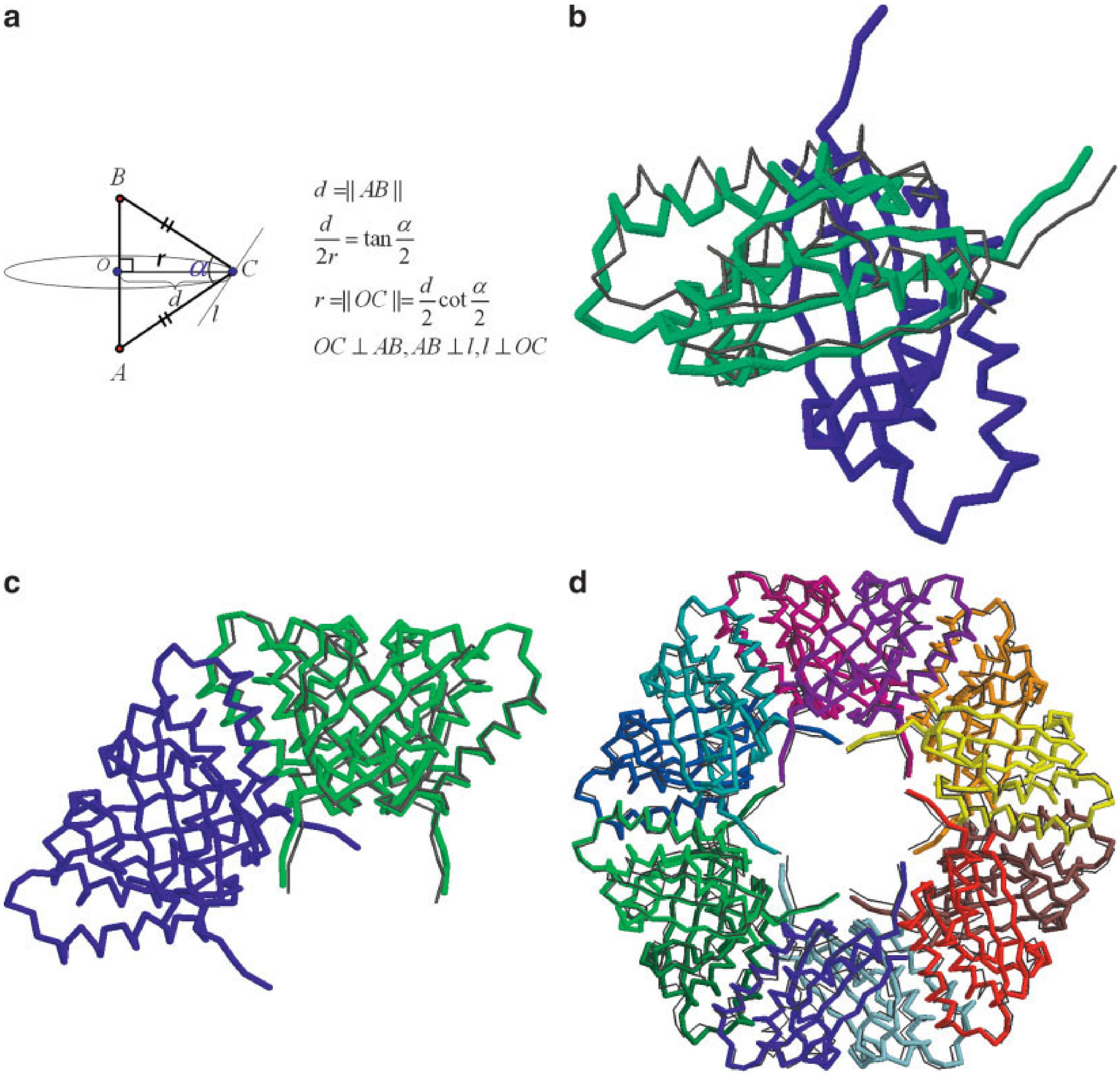
a
B
O
r
α
C
d
l
A
$d = \| AB \|$
$\frac{d}{2r} = \tan\frac{\alpha}{2}$
$r = \| OC \| = \frac{d}{2}\cot\frac{\alpha}{2}$
$OC \perp AB, AB \perp l, l \perp OC$
b
c
d

Fig. 2. Symmetric docking: the prediction of the dodecameric form of SP1. (**a**) Given two points *A* and *B* on the asymmetric unit surface. The rotation axis *l* of a cyclically symmetric transformation that transform *A* into *B* is a tangent to the circle that is centered at the middle point of *AB* and lies on its orthogonal plane. The radius of the circle depends solely on the distance between the points and the symmetry angle α (which is 360 divided by the symmetry order). The SP1-dodecamer prediction was generated by applying SymmDock twice. (**b**) The prediction of the dimeric form ($\alpha = 180$) of 2 contacting monomers. The top-ranked solution has an RMSD of 3.11 Å. (**c**) The top-ranked solution (1.4 Å RMSD) of the second run of SymmDock where the input unit was the predicted dimer shown in (b). Monomers of the same dimer are shown in the same color. Since we are looking for a hexamer of the dimer, α is set to 60. (**d**) The top-ranked prediction of the dodecamer superimposed on crystal structure. The RMSD between all the C_α atoms of the superimposed solution and of the crystal structure is 1.7 Å. In all subfigures the crystal structure is in gray.

Matching Module

We compute a sparse dot surface representation for the symmetric unit, which produces about 6 surface points for each surface atom.[19] This density is sufficient to receive a high number of matching pairs of surface points, since interfaces between consecutive units in such complexes are relatively large. Our matching procedure is as follows. For each pair of surface points *A*, *B* we sample their "candidate axes circle" for possible symmetry axes and compute a symmetric transformation for each possible axis. The sampling is done with a predefined sampling step (the default is 3°), meaning that 360/*step* transformations are created for each pair of surface points. An additional constraint is supplied by the molecular surface normals. Since, *B* and *T*(*A*) face each other across the interface, their normals should be (roughly) opposite: $T(\overrightarrow{n_A}) \cong -\overrightarrow{n_B}$. Thus, instead of sampling the whole circle, we sample only the 2 (countersided) arches that fulfill the opposing normal direction constraints. The algorithm also has a fast matching mode option that simply computes the tangent that optimally aligns $T(\overrightarrow{n_A})$ and $-\overrightarrow{n_B}$.

Candidate Transformation Clustering Module

Each matching pair votes for many different axes (depending on the sampling step and the opposing normal direction threshold). For the correct solution, different matching pairs, which belong to the original interface, should contribute consistent votes, namely, point to the same symmetry axis. A wide interface supplies many such pairs, so the votes for the "correct" axis should overcome the noise introduced by sampling the *C*-circle boundary [see Fig. 2(a)]. We represent a symmetry axis by 2 geometric characteristics: the axis direction and the projection of the center of mass of the asymmetric unit on the axis. The candidate axes from the matching stage are clustered according to these characteristics. We rank the clusters according to their number of votes. The representative symmetry axis for each cluster is simply the average axis, such that its direction is the average direction and it passes through the average projection point. To speed up the axes clustering procedure, we use the geometric hashing paradigm.

The SymmDock algorithm fully exploits the constraints imposed by cyclic symmetry. Moreover, by focusing on local features and employing an efficient voting mechanism, it is explicitly directed toward the selection of symmetric transformations that suggest wide interface areas, discarding candidate hypotheses with insignificant contacts. Therefore, the algorithm achieves the symmetric assembly within few minutes for an average-size asymmetric unit.

FLEXIBLE DOCKING WITH CYCLIC SYMMETRY

SymmDock performs symmetric docking of the rigid input molecule. Many monomers of homomultimers may consist of several domains. In such cases the different conformers of the monomer may present a different spatial arrangement of its domains. In the CAPRI experiment, there are 2 examples of such cases: Target 9,[22] where the 2 domains of the monomer move substantially between the activated and deactivated dimeric forms; and Target 10,[23] where domain movements occur when the monomer shifts from its dimeric to its trimeric form. We have developed FlexSymmDock for flexible docking with cyclic symmetry. The algorithm first invokes SymmDock on each domain separately. Then it performs the whole assembly by fitting the predicted symmetry axes of each rigid part solution and the corresponding hinge/s. Two symmetric transformations of neighboring rigid parts are consistent if (1) their symmetric axes are within (almost) the same distance from their common hinge point and (2) there is no (internal) steric clash between the parts once they are combined. Just like in the assembly stage of FlexDock, we also evaluate the self-contact between the domains of the same monomer. Since we could not find a data set to test the algorithm, it was only applied on the CAPRI Targets 9 and 10. The results are detailed in the companion article in this issue of *Proteins*.[2] In general, we had quite accurate results for Target 10. The solutions for Target 9 were not as accurate, since the linkers between the domains, which undergo conformational change between the activated and deactivated dimeric forms, play a major role in the interactions between the monomers of the deactivated form.

EXPERIMENTAL RESULTS

Flexible Docking

The FlexDock algorithm was tested and verified on a number of cases. Consider the interaction of myosin kinase peptide with calmodulin. Upon binding, there is large-scale movement of calmodulin, which involves splitting of 1 long helix [Fig. 1(b)]. The total rotation of one domain relative to the other is upwards of 150°. FlexDock successfully finds the near-native solution with RMSD 2.12 Å ranked second [Fig. 1(c)]. Note that the corresponding 2 rigid subpart docking solutions were ranked only 38 and 6; however, the requirement of hinge consistency "floated" the near-native solution to the second place. The running time for this case is 11 min on a PC with a Pentium IV processor.

Another interesting test case is the complex of adenosine triphosphate (ATP) and biotin carboxylase. One of the protein domains is twisted relative to the other domain by 80° upon ATP binding. Although, the rigid subpart docking stage of the algorithm finds near-native solutions with the (quite poor) ranks of 161 and 4880, the assembly stage ranks the near-native complex at place 17. This is due to the fact that upon binding of ATP, an intramolecular interface is created between the 2 domains of the flexible protein. The contribution of the shape complementarity score of this interface to the flexible solution combined with the hinge consistency requirement helps to rank the overall near-native solution significantly higher than the ranking of the rigid parts.

Symmetric Docking

The SymmDock algorithm was successfully tested on a number of complexes with cyclic symmetry. Generally, it is very difficult to find "unbound" test cases for symmetry docking, since most of the proteins crystallize already in the oligomeric state. Nevertheless, symmetry docking is relevant to the cases when there are conformational changes upon activation–deactivation of the oligomer, such as the case of CAPRI Target 9.[22] In some cases, such as CAPRI Target 10,[23] there are even changes in the number of monomeric units involved in the complex. This type of conformational changes occurs in many proteins in nature; however, in many cases, the PDB will contain only 1 most stable conformer. Therefore, our method was tested mostly on "bound" cases. We selected only complexes where the unit cell includes the whole multimer, such that the monomers are not related by crystallographic symmetry and there are some differences between them. The results are summarized in Table I. The near-native solu-

TABLE I. SymmDock Results

Target	PDB code	Unit size (C_α atoms)	Best rank of a near native solution[a]	CPU time MM:SS[b]
C_3 Symmetry				
Maltoporin sucrose	1af6	421	1 (2.03)	3:52
VP7, Viral coat	1bvp	349	1 (1.38)	3:27
GP41, transmembrane glycoprotein	1f23	74	1 (1.62)	0:40
PRD1 viral capsid	1hb7	370	1 (1.23)	3:23
Hemagglutinin	1hgg	503	1 (2.92)	7:01
Lymphokine	1tnf	152	1 (3.17)	1:19
GP2, envelope glycoprotein	2ebo	74	1 (1.42)	0:43
C_5 Symmetry				
Lumazine synthase	1c2y	155	1 (1.50)	3:02
Shiga-like toxin I	1c4q	69	1 (1.43)	0:51
Isomerase (6-epimerase)	1eq2	273	6 (1.21)	5:18
GTP cyclohydrolase I	1fb1	196	1 (1.24)	3:37
C_7 Symmetry				
GroEl, chain A	1aonA*	524	1 (3.77)	6:22
GroEl, chain H	1aonH*	524	1 (2.76)	3:24
11S regulator	1avo	200	1 (1.36)	3:36
snRNP SM-like protein	1h64	71	1 (0.63)	0:45
10 KDA chaperonin	1hx5	82	1 (0.95)	0:48
Transcriptional regulator	1ny6	247	1 (2.88)	3:18
Protective antigen (PA83)	1tzo	553	1 (3.44)	3:24
α-hemolysin	7ahl	293	1 (2.25)	4:27

For 18 out of the 19 targets, the SymmDock algorithm predicted as the top-ranking solution a near-native complex, performing its calculation within 7 minutes or less. All cases are bound docking targets; that is, for each case, the input monomer was extracted from its bound multimer complex.

[a] Highest rank of the solution with RMSD under 4 Å. The actual RMSD of the solution is in brackets.

[b] The running time of the algorithm on a Pentium IV PC.

*In the case of GroEL, there are 2 heptamers in the PDB file. The monomer unit is in different conformations in each heptamer. Therefore, we reconstructed both heptamers using 2 different monomer units.

tion was found for all the cases we tested. In all but 1 case, there is a solution with RMSD below 4 Å ranked 1. In the exceptional case, the rank is 6. The sizes of the monomer units vary between 69 and 553 amino acids. The running times are between 40 s to 7 min on a standard Pentium IV PC.

Another recently published, interesting example is the reconstruction of the SP1 dodecameric complex.[24] The dodecameric form is a hexamer in which the asymmetric unit is a homodimer. In order to predict the structure, we have applied the SymmDock algorithm twice. First, to predict the dimer from the monomer, then, using the predicted model of the dimer as the input unit, we have applied SymmDock for C_6 symmetry in order to predict the dodecameric form. All of the 5 top-scoring solutions in the dimer predictions were less than 5.0 Å RMSD from the native. The top-ranking solution, which had an RMSD of 3.11 Å from the native dimer structure [Fig. 2(b)], was the input "asymmetric unit" for the second run. The top-ranked solution from this stage placed the contacting neighboring dimer within 1.4 Å RMSD from its position within the crystal structure [Fig. 2(c)]. The best rigid superimposition of the corresponding C_α atoms between the top-ranking dodecamer and the native one gave a 1.7 Å RMSD. This solution is displayed in Figure 2(d).

CONCLUSIONS AND FUTURE WORK

We have presented a set of algorithms that has been developed and used by our group during the CAPRI experiment. The algorithms are very fast and have performed remarkably well in CAPRI Rounds 3–5. For all but 1 of the targets, medium quality or acceptable solutions have been achieved among the top 10. This is also due to the incorporation of binding site information based on available biological data for the targets or their homologs.

We do plan several extensions and improvements to the presented algorithms, which we outline below. The current version of FlexDock assumes significant contact between each rigid part of the flexible molecule and the rigid counterpart molecule. We intend to extend our algorithm to cases where such contact is small or does not exist at all. Obviously, such solutions are inferior when the score is mainly a geometric contact score. We also intend to model the flexible linker region between neighboring rigid parts to eliminate chemically unfeasible hinge movements. SymmDock will be adapted to handle "almost" symmetric arrangements of not-necessarily-congruent units (e.g., homologs).

A major improvement in docking accuracy is expected from the development of a side-chain placement refinement stage, which could be applied to a predefined number/percentage of the top "soft-docking" solutions.

ACKNOWLEDGMENTS

We thank the organizers of CAPRI and the experimentalists, who contributed structures for this exciting experiment.

REFERENCES

1. Schneidman-Duhovny D, Inbar Y, Polak V, Shatsky M, Halperin I, Benyamini H, Barzilai A, Dror O, Haspel N, Nussinov R, Wolfson HJ. Taking geometry to its edge: fast unbound rigid (and hinge-bent) docking. Proteins 2003;52:107–112.
2. Inbar Y, Schneidman-Duhovny D, Halperin I, Oron A, Nussinov R, Wolfson HJ. Approaching the CAPRI challenge with an efficient geometry-based docking. Proteins 2005;60:217–223.
3. Duhovny D, Nussinov R, Wolfson HJ. Efficient unbound docking of rigid molecules. In Guigo R, Gusfield D, editors. Workshop on algorithms in bioinformatics. Vol. 2452. Rome: Springer-Verlag; 2002. p 185–200.
4. Norel R, Lin SL, Wolfson HJ, Nussinov R. Molecular surface complementarity at protein–protein interfaces: the critical role played by surface normals at well placed, sparse points in docking. J Mol Biol 1995;252:263–273.
5. Fischer D, Lin SL, Wolfson HJ, Nussinov R. A geometry-based suite of molecular docking processes. J Mol Biol 1995;248:459–477.
6. Sandak B, Nussinov R, Wolfson HJ. A method for biomolecular structural recognition and docking allowing conformational flexibility. J Comp Biol 1998;5:631–654.
7. Lamdan Y, Wolfson HJ. Geometric hashing: a general and efficient model-based recognition scheme. In Proceedings of the IEEE International Conference on Comput Vision. Tampa, FL: IEEE Press; 1988. p 238–249.
8. Katchalski-Katzir E, Shariv I, Eisenstein M, Friesem AA, Aflalo C, Vakser IA. Molecular surface recognition: determination of geometric fit between protein and their ligands by correlation techniques. Proc Natl Acad Sci USA 1992;89:2195–2199.
9. Méndez R, Wodak SJ. Prediction of protein–protein interactions: the CAPRI experiment, its evaluation and implications. Curr Opin Struct Biol 2004;14:242–249.
10. Méndez R, Leplae R, De Maria L, Wodak SJ. Assessment of blind predictions of protein–protein interactions: current status of docking methods. Proteins 2003;52:51–67.
11. Nessler S, Fieulaine S, Poncet S, Galinier A, Deutscher J, Janin J. HPr kinase/phosphorylase,the sensor enzyme of catabolite repression in Gram-positive bacteria:structural aspects of the enzyme and the complex with its protein substrate. J Bacteriol 2003;185: 4003–4010.
12. Eisenstein M, Shariv I, Koren G, Friesem AA, Katchalski-Katzir E. Modeling supra-molecular helices: extension of the molecular surface recognition algorithm and application to the protein coat of the tobacco mosaic virus. J Mol Biol 1997;266:135–143.
13. Berchanski A, Eisenstein M. Construction of molecular assemblies via docking: modeling of tetramers with D2 symmetry. Proteins 2003;53:817–829.
14. Ballard DH. Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognit 1981;13:111–122.
15. Wolfson HJ. Generalizing the generalized Hough transform. Pattern Recognit Lett 1991;12:565–573.
16. Fernandez-Recio J, Totrov M, Abagyan R. Soft protein–protein docking in internal coordinates. Protein Sci 2002;11:280–291.
17. Gray JJ, Moughan S, Schueler-Furman O, Kuhlman B, Rohl CA, Baker D. Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. J Mol Biol 2003;331:281–299.
18. Schneidman-Duhovny D, Inbar Y, Nussinov R, Wolfson HJ. Patchdock and Symmdock: servers for rigid and symmetric docking. Nucleic Acids Res 2005;33.
19. Lin SL, Nussinov R, Fischer D, Wolfson HJ. Molecular surface representation by sparse critical points. Proteins 1994;18:94–101.
20. Chen R, Mintseris J, Janin J, Weng Z. A protein–protein docking benchmark. Proteins 2003;52:88–91.
21. Shatsky M, Dror O, Schneidman-Duhovny D, Nussinov R, Wolfson HJ. BioInfo3D: a suite of tools for structural bioinformatics. Nucleic Acids Res 2004;32:W503–W507.
22. Graille M, Zhou CZ, Receveur V, Collinet B, Declerck N, van Tilbeurgh H. Structure of the native and inactive LicT PRD from *B. subtilis*. J Biol Chem 2005;280:14780–14789.
23. Bressanelli S, Stiasny K, Allison SL, Stura EA, Duquerroy S, Lescar J, Heinz FX, Rey FA. Structure of a flavivirus envelope glycoprotein in its low-ph-induced membrane fusion conformation. EMBO J 2004;23:728–738.
24. Dgany O, Gonzalez A, Sofer O, Wang W, Zolotnitsky G, Wolf A, Shoham Y, Altman A, Wolf SG, Shoseyov O, Almog O. The structural basis of the thermostability of SP1, a novel plant (*Populus tremula*) boiling stable protein. J Biol Chem 2004;279: 51516–51523.