PROTEINS
STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

Protein docking using case-based reasoning

Anisah W. Ghoorah,[1] Marie-Dominique Devignes,[2] Malika Smaïl-Tabbone,[1] and David W. Ritchie[3]*

[1] Université de Lorraine, LORIA, Campus Scientifique, 54506 Vandoeuvre-lès-Nancy, France

[2] CNRS, LORIA, Campus Scientifique, 54506 Vandoeuvre-lès-Nancy, France

[3] INRIA, LORIA, Campus Scientifique, 54506 Vandoeuvre-lès-Nancy, France

ABSTRACT

Protein docking algorithms aim to calculate the three-dimensional (3D) structure of a protein complex starting from its unbound components. Although *ab initio* docking algorithms are improving, there is a growing need to use homology modeling techniques to exploit the rapidly increasing volumes of structural information that now exist. However, most current homology modeling approaches involve finding a pair of complete single-chain structures in a homologous protein complex to use as a 3D template, despite the fact that protein complexes are often formed from one or more domain–domain interactions (DDIs). To model 3D protein complexes by domain–domain homology, we have developed a case-based reasoning approach called KBDOCK which systematically identifies and reuses domain family binding sites from our database of non-redundant DDIs. When tested on 54 protein complexes from the Protein Docking Benchmark, our approach provides a near-perfect way to model single-domain protein complexes when full-homology templates are available, and it extends our ability to model more difficult cases when only partial or incomplete templates exist. These promising early results highlight the need for a new and diverse docking benchmark set, specifically designed to assess homology docking approaches.

Proteins 2013; 81:2150–2158.

Key words: protein–protein interactions; domain–domain interactions; docking by homology; Pfam domains; domain binding sites; CAPRI.

INTRODUCTION

The protein docking problem is the problem of how to calculate the three-dimensional (3D) structure of a protein complex starting from the unbound components. This problem was first described some 30 years ago,[1] and since then many computational docking algorithms have been developed.[2] In the last 10 years, the Critical Assessment of Predicted Interactions (CAPRI) blind docking experiment has stimulated many improvements.[3,4] However, while good progress has been made,[5] it still remains challenging to produce a satisfactory 3D model of a protein complex using *ab initio* docking algorithms. On the other hand, several studies have shown that using experimental information to guide and constrain docking calculations can significantly improve the reliability of the predictions.[5,6] Given that the number of solved structures in the Protein Data Bank (PDB) appears to be growing exponentially,[7] the possibility to exploit existing structural knowledge of proteins and their interactions seems to be an increasingly promising way to model the 3D structures of unknown complexes.

Because protein domains may often be identified as structural and functional units, the 3D structures of protein complexes are often analyzed in terms of their component domain–domain interactions (DDIs). In recent years, several protein structure interaction databases have been described.[8] Some of these collect interactions between protein chains,[9–13] whereas others[14–19] annotate interactions using the Pfam,[20] SCOP,[21] or CDD[22] domain classifications. Since it has been shown that proteins with similar sequences often interact in similar ways,[23] and since it is well known that protein folds are often more conserved than their sequences,[24] it follows that homologous pairs of protein domains could be expected to interact in structurally similar ways. Indeed,

Grant sponsor: Agence Nationale de la Recherche; grant numbers: ANR-08-CEXC-017-01 and ANR-11-MONU-006-02.

*Correspondence to: David W. Ritchie, INRIA, LORIA, Campus Scientifique, BP 239, 54506 Vandoeuvre-lès-Nancy, France. E-mail: dave.ritchie@inria.fr

Received 1 June 2013; Revised 18 September 2013; Accepted 19 September 2013
Published online 7 October 2013 in Wiley Online Library (wileyonlinelibrary.com).
DOI: 10.1002/prot.24433

several studies have found that the locations of protein interaction sites are often conserved within domain families.[25,26] Therefore, it seems reasonable to suppose that known binding sites in domain families could be reused by different binding partners.

Several groups, including ourselves, have described homology-based protein-protein interaction (PPI) modeling approaches. For example, Lu *et al*.,[27] Grimm *et al*.,[28] Launay and Simonson,[29] and Mukherjee and Zhang[30] used threading techniques to predict the 3D structure of a complex starting from the sequences of the component proteins and a library of known complexes. Korkin *et al*.[31] used their comparative patch analysis technique together with knowledge of known binding sites to define docking restraints for modeling a selection of protein complexes. Günther *et al*.[32] model protein complexes by comparing the local surface similarity of a pair of query domains with a library of DDIs derived from the PDB. Kundrotas *et al*.[33] use PSI-BLAST to retrieve candidate sequence-based templates for the target complex which are then modeled in 3D using NEST.[34] Movshovitz-Attias *et al*.[35] recently carried out a detailed study on the utility of using structural templates when modeling conformational flexibility in a number of enzyme–inhibitor and antibody–antigen docking problems.

It should be noted that there does not yet exist a standard benchmark set for testing homology docking approaches. Each of the above studies worked with different datasets, and applied different sequence similarity criteria to exclude similar or trivially "redundant" instances. Furthermore, different groups place more or less emphasis on the quality of the templates retrieved compared to the level of sophistication in the subsequent modeling steps. Thus, it is difficult to compare quantitatively the performance of different homology modeling approaches.

Our own KBDOCK approach differs from earlier homology modeling approaches in that it works directly at the Pfam domain family level,[20] and it uses 3D superpositions of domain structures to identify domain family binding sites (DFBSs) which may then be reused as docking templates in a largely sequence-independent way.[36] At a conceptual level, our approach is inspired by a computational problem-solving approach known as case-based reasoning (CBR), which emerged during the early days of artificial intelligence research.[37] CBR is a very broadly defined method of problem-solving, and many types of CBR systems have been implemented in many different ways.[38] Nonetheless, most CBR systems typically maintain a case base (CB) of previous cases, and they solve problems (new cases) by applying four main steps,[39] namely, (i) retrieve the most similar case or cases from the CB, (ii) reuse or adapt those cases to better match the problem and to propose a solution, (iii) revise or refine the proposed solution if necessary, and (iv) retain the solved case in the CB for future use. In the context of modeling protein structures and PPIs, there is a clear parallel between homology modeling and CBR. Here, we omit the final step of storing the generated solutions in the CB to ensure that all predictions are derived only from experimentally solved and validated 3D structures. Nonetheless, as demonstrated below, explicitly adopting the CBR paradigm provides a formal way to retrieve, compare, and rank multiple candidate templates automatically.

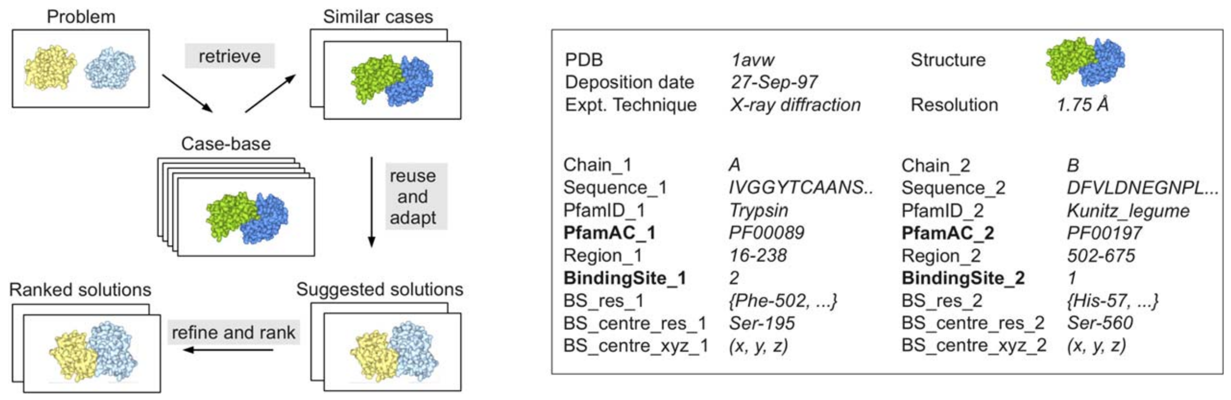
METHODS

The KBDOCK database

The KBDOCK database collects structural DDI information from the 3DID database,[14] and it superposes and clusters the members of each Pfam[20] domain family to define a nonredundant set of DFBSs.[36] The version of KBDOCK used here was built using a total of 140,612 DDIs involving 3755 Pfam domains gathered from 29,922 PDB structures. These DDIs were filtered, superposed, and spatially clustered by Pfam family to give a total of 1439 Pfam DFBSs which are involved in 1009 distinct domain family interactions.[36]

When considered at the Pfam level, we find that approximately 62% of Pfam families have just one hetero Pfam partner, and very few Pfam families have four or more DFBSs (excluding the immunoglobulins). In other words, if the domains of a given docking target exist in KBDOCK, there is a reasonably good chance that one or more DDI templates involving the target domains will be found. On the other hand, when considered at the family binding site level, a total of 82% of hetero DFBSs in KBDOCK have just one Pfam partner, leaving 18% which interact with more than one Pfam family. If we set aside all the families that bind only one Pfam and then compare the theoretical maximum number of DFBSs necessary to account for each remaining Pfam interaction (i.e., assuming a distinct DFBS per distinct family partner) with the actual number observed in KBDOCK, we obtain a ratio of 1201/810, or essentially three Pfam partners for every two hetero DFBSs. This estimate supports our hypothesis that targets lacking a family-level DDI template might still be modeled with a fair chance of success by "reusing" other DFBSs. Of course, it should be borne in mind that such an estimate may change in future updates of the database.

In the CBR paradigm, a case is a collection of attributes or features which describe a solved problem. In general, each case may be described by a number of *indexed* and *nonindexed* attributes. Indexed attributes are used for case retrieval, whereas nonindexed attributes provide useful contextual information. Here, the Pfam domain identifiers of experimentally determined structures of pairs of interacting domains serve as the main indexed attributes, whereas the nonindexed attributes include



PDB	1avw	Structure	
Deposition date	27-Sep-97		
Expt. Technique	X-ray diffraction	Resolution	1.75 Å
Chain_1	A	Chain_2	B
Sequence_1	IVGGYTCAANS..	Sequence_2	DFVLDNEGNPL...
PfamID_1	Trypsin	PfamID_2	Kunitz_legume
PfamAC_1	PF00089	**PfamAC_2**	PF00197
Region_1	16-238	Region_2	502-675
BindingSite_1	2	**BindingSite_2**	1
BS_res_1	{Phe-502, ...}	BS_res_2	{His-57, ...}
BS_centre_res_1	Ser-195	BS_centre_res_2	Ser-560
BS_centre_xyz_1	(x, y, z)	BS_centre_xyz_2	(x, y, z)

Figure 1

Left: an overview of the KBDOCK approach for modeling DDIs. Right: an example of a DDI case in the KBDOCK database. Each case consists of a collection of attributes or features. Those indexed attributes which may be used for case retrieval are shown in bold. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

PDB codes, PDB chain identifiers, amino acid sequences and atomic coordinates. If necessary, indexed attributes may be derived from the nonindexed attributes. For example, KBDOCK uses PfamScan[20] to determine the Pfam identifiers of the query domains automatically from their sequences.

As shown in Figure 1, the information associated with each case includes instance-specific information such as the lists of residues of each domain which participate in a specific DDI, along with other derived information such as the calculated geometric centre of the binding site, and the residue of each domain which KBDOCK assigns as the central residue of that particular binding site. By spatially clustering binding sites within Pfam families, KBDOCK also stores a family-level binding site identifier for each instance of a binding site. Thus, instances of DDIs in the CB may be grouped and retrieved according to both the Pfam families and the family-level binding sites involved. For example, the *Kunitz_legume* domain family has five nonredundant hetero DDI cases involving four distinct DFBSs. Because we define binding sites at the Pfam family level, KBDOCK identifies each binding site using a compound identifier of the form *PfamAC/BindingSite*. Thus, for example, "PF00197"/1 refers to the first DFBS of the *Kunitz_legume* family.

CBR docking by Prolog pattern matching

The CBR component of KBDOCK is written in the Prolog programming language, which is well suited for mixing database access, symbolic pattern matching, and numerical computation. By denoting a pair of Pfam DFBSs as $d1/b1$ and $d2/b2$, we use the notation $(d1/b1, d2/b2)$ to represent a DDI in the CB. For example, the family-level interface between *Trypsin* and *Kunitz_legume* (Fig. 1) is actually stored in Prolog as ("PF00089"/2, "PF00197"/1). We can then search for all DDIs involving *Trypsin* by pattern-matching against ("PF00089"/B1, D2/B2), or simply ("PF00089", D2), where unquoted upper case identifiers represent uninstantiated instances. This representation allows known binding sites to be given as part of the query if such knowledge is available, and it allows partial or incomplete CB matches to be represented. Naturally, to model a protein complex by CBR, the overall aim is to retrieve cases which match (or, more generally, which may be unified with) the given query specification. If both of the query domains can be unified with cases in the CB, we call this a full-homology (FH) problem and we denote the set of matching cases as $FH(d1, d2)$. It is worth noting that even for the most favorable problems in which FH templates exist, these may involve more than one pair of DFBSs. On the other hand, it is also possible for one or both of the given query domains to match individually one half of a known DDI. We call such problems semihomology (SH) problems, and we let $SH(d1, D2)$ and $SH(D1, d2)$ denote the two possible sets of SH cases for a given query, where $D1 \neq d1$ and $D2 \neq d2$. Furthermore, if matches are found for both query domains, we call it a SH-2 case, whereas if matches are found for only one of the query domains we call it a SH-1 case. This distinction becomes significant at the case refinement stage. Of course, if no matches are found then no homologues exist, and it would be necessary to search for other more distantly related cases or to use *ab initio* docking. On the other hand, if multiple matches are found, we need to consider and rank all of them. This is the task that we focus on here.

Modeling FH cases

Since we know from previous experience[36] that FH cases very often provide good 3D docking templates, we

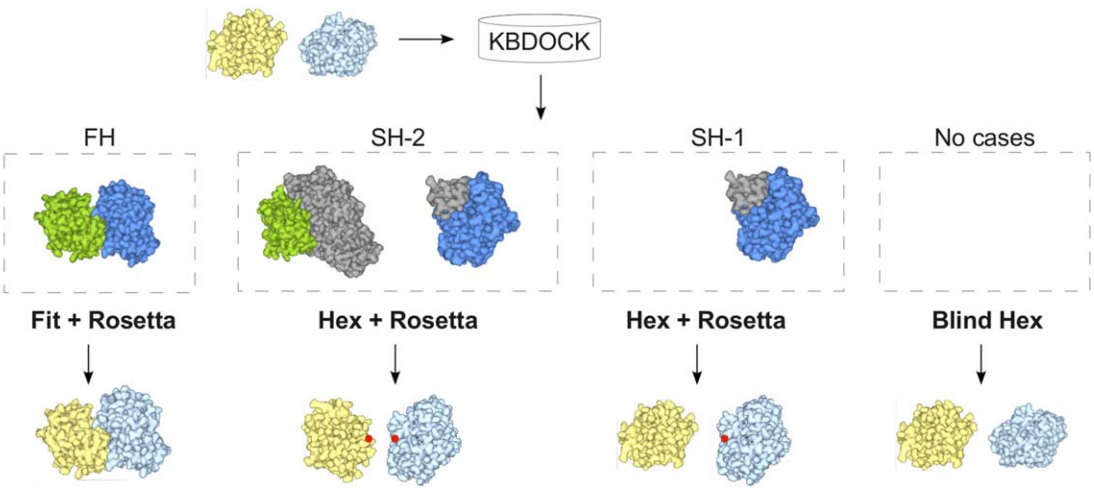


Figure 2

Schematic illustration of the KBDOCK docking protocol. For FH cases, the query domains are superposed onto the best template from the CB to form the initial solution which is then optimized by RosettaDock in "high-resolution" mode. Red spheres represent the central residue of each binding site, which is used to focus the *Hex* docking search. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

collect the instances in $FH(d1, d2)$ into groups having distinct pairs of DFBSs, and we rank the members of each group by their overall sequence similarity to the concatenated sequences of each pair. We then select the most similar member of each group, and superpose its domains onto the query using the ProFit least-squares fitting program (http://bioinf.org.uk) to give a small set of substitution-adapted solutions (candidate docking templates). Next, starting from the selected CBR solutions, we use version 3.4 of RosettaDock[40] in "high-resolution mode" with extended rotamer libraries (RosettaDock options -ex1 and -ex2aro) to perform a local docking search with side chain repacking to optimize each candidate docking template. We currently request 200 docking conformations per starting template. Finally, these conformations are sorted by RosettaDock interface energy and clustered by C_α root mean squared deviation (RMSD) using our own in-house greedy clustering algorithm, and the lowest energy member of each cluster is retained as a distinct docking prediction.

Modeling SH cases

For SH-2 cases, we assume that the target complex may be modeled using a pair of existing DFBSs. We therefore take the Cartesian product $P=SH(d1/B1)\times SH(d2/B2)$ to enumerate all possible pairs of candidate DFBSs. However, this does not form actual 3D interfaces between the DFBSs. It only gives a set of symbolic associations. Therefore, for each instance of P, we construct a putative DDI using the coordinates of the stored centre of gravity and central interface residues to locate the two domains on the global z axis with their central residues facing each other near the origin, and with $d1$ on the negative z axis and $d2$ on the positive z axis. Up to a small translation and an undetermined twist about the z axis, each pair of such configurations defines a putative pairwise interface, which is then rigidly refined using the *Hex* polar Fourier docking correlation program[41] with two angular constraints, β_1 and β_2.

For SH-1 problems, in which one or more DFBSs are known for just one of the query domains, the query domains are oriented on the z axis, as described above, using a random surface residue for the uninstantiated binding site centre residue. This starting orientation is then refined using *Hex* with just one angular constraint around the known binding site, and with the other domain being allowed to spin freely to search over its entire surface. If no DFBSs match the query, unconstrained blind docking is applied.

Here, each pairwise *Hex* docking run used 3D fast Fourier transformation shape-based correlation searches with range angles of $\beta_1=\beta_2=45^\circ$, as appropriate, and 40 translational steps of 0.5 Å along the z axis with respect to each given starting orientation. This generates approximately 6×10^7, 3.6×10^8, and 2×10^9 trial rigid body orientations for each pairwise SH-2, SH-1, and blind docking run, respectively, of which the top 2000 are rescored using our implementation of the DARS (Decoys as Reference State) potential.[42] This procedure is applied to each putative DDI. The top 100 solutions from each *Hex* run are then rigidly optimized and repacked using RosettaDock, as before. Thus, for each docking target, a total of 100 × 200 RosettaDock conformations are generated and scored for each CBR template retrieved from the CB. Figure 2 illustrates the above modeling steps schematically.

Table I
CBR Docking Results for the 24 FH Targets

Target PDB	Target type	Template PDB	SID 1	SID 2	DFBS pairs	DFBSs reused	KB–only		KB+ *Hex* +*Rosetta*		Blind *Hex*	
							Rank	RMSD	Rank	RMSD	Rank	RMSD
1ay7	E/R	1brs	29.4	94.1	1	Y + Y	1	11.1	6	7.40	–	–
1cgi	E/R	3sgb	20.6	34.7	1	Y + Y	1	5.8	1	6.45	2	9.07
1eaw	E/R	1tfx	41.5	39.6	3	Y + Y	1	1.8	1	4.58	7	8.06
1mah	E/R	1fss	58.9	96.7	1	Y + Y	1	1.2	1	1.99	2	4.63
1n8o	E/R	1ezu	41.3	95.3	1	Y + Y	1	8.5	2	9.23	–	–
1oph	E/R	1jmo	28.9	35.6	1	Y + Y	1	19.5	–	–	–	–
1yvb	E/R	1stf	33.8	19.8	1	Y + Y	1	9.5	–	–	–	–
2j0t	E/R	1bqq	46.7	31.2	1	Y + Y	1	8.8	4	7.67	5	9.07
2oul	E/R	2nqd	36.4	94.0	1	Y + Y	1	9.2	1	2.26	2	9.61
2sni	E/R	1cse	63.1	35.9	1	Y + Y	1	5.7	1	4.45	–	–
3sgq	E/R	1cgj	18.4	34.0	1	Y + Y	1	12.4	1	5.11	–	–
1ffw	O/R	1a0o	98.3	98.5	1	Y + Y	1	2.6	1	5.48	411	9.72
1fqj	O/R	1agr	63.6	35.3	1	Y + Y	1	5.6	1	6.16	–	–
1gpw	O/R	1ka9	58.5	37.6	1	Y + Y	1	3.8	1	5.57	5	8.78
1he1	O/R	1g4u	99.4	30.0	1	Y + Y	1	10.1	1	5.51	–	–
1xd3	O/R	1cmx	21.8	94.2	1	Y + Y	1	11.4	1	8.43	–	–
2a9k	O/R	2a78	98.8	33.2	1	Y + Y	1	4.2	6	8.21	–	–
2hle	O/R	1shw	44.2	26.7	1	Y + Y	1	13.8	1	7.48	–	–
2oob	O/R	2bwe	20.5	33.3	1	Y + N	*	*	*	*	335	9.90
1grn	O/M	1am4	93.8	96.0	1	Y + Y	1	6.4	2	2.02	–	–
1r6q	O/M	1r6o	98.0	98.8	1	Y + Y	1	12.4	–	–	–	–
2ayo	O/M	1nbf	22.1	97.1	1	Y + Y	1	18.9	1	4.00	–	–
2nz8	O/M	2dfk	24.0	71.3	1	Y + Y	1	9.4	9	4.48	–	–
3cph	O/M	1vg0	19.4	37.9	1	Y + Y	1	9.4	1	8.87	–	–

E, enzyme–inhibitor target; O, other target; R, rigid–body target; M, medium difficulty target; D, difficult target; Y, yes (binding site reused); N, no (binding site not reused); U, unknown binding site.

Here, KB denotes results taken directly from KBDOCK. An asterisk (*) denotes no solution was found by CBR because a binding site was not reused. A hyphen (–) denotes no solution found within the 200 Rosetta refined orientations or within the top 2000 blind *Hex* orientations. For one target (PDB 1eaw), multiple cases are retrieved and considered ("DFBS Pairs"). For this target, the table shows only a single template, being the one with the highest percentage sequence identity (SID) to the query structures.

RESULTS AND DISCUSSION

At the time of the present study, we estimated that 27 of the 58 CAPRI targets available either require or could benefit from a homology modeling step (i.e., at least one of the docking partners is given an amino acid sequence or a structural homologue exists in the PDB, respectively). However, only 12 of these involve single domain docking targets, and only 5 of those have Pfam-level homology templates in the KBDOCK database when date filtering is applied (see subsequently). Therefore, to work with a larger set of examples, we decided to test our homology modeling approach using a subset of the Protein Docking Benchmark 4.0 compiled by Weng's group.[43] This benchmark consists of 176 protein–protein complexes for which the bound structures and the unbound components of at least one of the docking partners are available. Weng's group divided the benchmark into 52 enzyme–inhibitor complexes, 25 antigen–antibody complexes, and 99 "other" complexes, and they classified each target as "rigid," "medium," and "difficult" according to the degree of conformational changes between the bound and unbound structures. Targets in the rigid class should be amenable to rigid-body docking algorithms, whereas difficult targets normally require a flexible docking algorithm to be used in conjunction with prior knowledge about the binding mode.

Here, we exclude the 25 antigen–antibody complexes because the antibody binding sites are known *a priori* and because the antigen binding sites generally do not entail homology. Furthermore, it is worth noting that when the structure of a complex is solved by X-ray crystallography, several closely related structures are often solved and deposited in the PDB by the same laboratory at the same time. Therefore, to increase the number of targets for which trivial solutions do not already exist, we filtered out any structures having a PDB deposition date equal to or later than that of the corresponding target. Additionally, in the current study we focus on single-domain complexes to understand the extent to which individual DFBSs are indeed reused during docking. Applying the above criteria gives a total of 54 single-domain complexes for which at least one of the docking partners has a nontrivial homologous domain in KBDOCK. We then applied the KBDOCK modeling protocol to the selected 54 target complexes using the above date filtering criteria for each target. This identified 24 targets which could be treated as FH cases, and 26 and 4

Table II
The CBR–Based Docking Results for the SH–2 Targets

Target PDB	Target type	Template PDBs	SID 1	SID 2	DFBS pairs	DFBSs reused	KB+*Hex* Rank	KB+*Hex* RMSD	KB+*Hex* +*Rosetta* Rank	KB+*Hex* +*Rosetta* RMSD	Blind *Hex* Rank	Blind *Hex* RMSD
1avx	E/R	1mct,1ava	100.0	–	5	Y + N	*	*	*	*	–	–
1r0r	E/R	1cse,1ppf	89.5	–	2	Y + Y	2	7.3	6	4.8	61	9.90
2o8v	E/R	1zun,2bto	–	99.0	2	N + Y	–	–	–	–	–	–
1acb	E/M	1cgj,2tec	100.0	–	2	Y + Y	1	8.6	8	7.6	–	–

E, enzyme–inhibitor target; R, rigid–body target; M, medium difficulty target; Y, yes (binding site reused); N, no (binding site not reused).
Here, KB denotes results taken directly from KBDOCK. An asterisk (*) denotes no solution was found by CBR because a binding site was not reused. A hyphen (–) denotes no solution found within the 200 Rosetta refined orientations or within the top 2000 blind *Hex* orientations. For one target (PDB 1eaw), multiple cases are retrieved and considered ("DFBS Pairs"). For this target, the table shows only a single template, being the one with the highest percentage sequence identity (SID) to the query structures.

targets which could be treated as SH-1 and SH-2 cases, respectively.

Table I summarizes the results obtained for the FH targets. As can be seen from the middle column ("DFBSs Reused"), this table shows that in 23 out of 24 FH cases the DFBSs of the selected template structures are indeed reused in the target complexes, according to our DFBS clustering criteria. The subsequent column ("KB-only") shows that in 15 of these cases, the ligand C_α RMSD between the template and the target (calculated after superposing corresponding pairs of template and target receptor C_α atoms) is less than 10Å, which broadly corresponds to an "acceptable" CAPRI docking prediction (i.e., rank $\leq$ 10 and RMSD $\leq$ 10Å). Compared to blind docking using *Hex* (final column) which gives only six acceptable predictions, these results are quite impressive. It is worth noting that despite the fact that the CBR protocol retrieves cases having the same Pfam family as the query domains, the retrieved domains show a large range of percent sequence identity ("SID" columns) with respect to the query domains, ranging from around 20 to 98%. Indeed, in only three cases (1ffw, 1grn, and 1r6q) do the two template domains together have greater than 90% sequence identity with respect to the query domains.

However, high sequence identity does not automatically imply low C_α RMSD with respect to the native complex, as exemplified by the case of 1r6q (12Å RMSD). On the other hand, using RosettaDock to refine the initial KBDOCK templates can improve the C_α RMSD of the KBDOCK templates. For example, for the 15 acceptable KBDOCK templates mentioned above, RosettaDock refinement improves the RMSD in six cases (2j0t, 2oul, 2sni, 1grn, 2nz8, and 3cph). Furthermore, RosettaDock converts six of eight "incorrect" CB templates into acceptable docking solutions (1ay7, 3sgq, 1he1, 1xd3, 2hle, and 2ayo), according to the CAPRI criteria. On the other hand, RosettaDock worsens the RMSD in eight cases (1cgi, 1eaw, 1mah, 1n8o, 1ffw, 1fqj, 1gpw, and 2a9k), and it completely loses the KBDOCK solution in a further three (1oph, 1yvb, and 16rq). We presume this is because in these cases RosettaDock's Monte Carlo sampling procedure is finding false local minima far from the native binding mode. Overall, the results in Table I demonstrate the near-perfect ability of the CBR approach to retrieve suitable FH templates from the CB when such cases exist. Even though side chain repacking in RosettaDock is considerably more computationally expensive than performing rigid-body *Hex* docking, this processing protocol is not especially expensive (taking approximately 50 CPU-minutes per FH target on a contemporary workstation), and it improves the average ligand C_α RMSD from 8.7 to 5.7 Å.

Tables II and III show the corresponding results for the targets which become more challenging SH-2 and SH-1 problems after date filtering. Table II shows that at least one DFBS is reused in all four SH-2 targets, but only two of the targets reuse both DFBSs together (1r0r and 1acb). Here, using focused *Hex* docking to locate the docking partners using the presumed binding sites produces two acceptable solutions whereas blind *Hex* produces none. Using RosettaDock to refine and repack the KBDOCK+*Hex* solutions improves the C_α RMSD, but slightly worsens the rank.

Table III shows that a known DFBS is reused in 15 of the 26 SH-1 targets. In more detail, KBDOCK+*Hex* produces acceptable solutions (CAPRI criteria) for six SH-1 targets. Using RosettaDock to repack these solutions improves the backbone RMSD in two cases (1fle and 2sic), but worsens it in two others (1zli and 1z0k). For the remaining two cases, it worsens the rank in one case (1gl1), and leaves the other (1ppe) essentially unchanged. In addition, using RosettaDock converts one incorrect KBDOCK+*Hex* solution (1hia) into an acceptable one. On the other hand, blind *Hex* docking also produces acceptable solutions for six SH-1 targets, of which five are common to the six mentioned above. Of these five (1gl1, 1ppe, 2sic, 1zli, and 1z0k), the difference in quality between the KBDOCK and blind *Hex* solutions is almost negligible. KBDOCK+*Hex* gives a slightly better rank and significantly lower RMSD in just one case (1z0k). Blind *Hex* gives a slightly better rank and similar

Table III
The CBR-Based Docking Results for the SH-1 Targets

							KB+*Hex*		KB+*Hex* +*Rosetta*		Blind *Hex*	
Target PDB	Target type	Template PDB	SID 1	SID 2	DFBSs	DFBSs reused	Rank	RMSD	Rank	RMSD	Rank	RMSD
1f34	E/R	1htr	51.0	–	1	N + U	*	*	*	*	5	8.64
1fle	E/R	1pyt	53.8	–	3	Y + U	1	7.00	1	5.55	–	–
1gl1	E/R	1acb	98.6	–	6	Y + U	8	8.01	82	7.9	6	7.32
1hia	E/R	2kai	64.3	–	3	Y + U	24	9.1	3	9.03	11	8.45
1oc0	E/R	1k9o	30.1	–	1	N + U	*	*	*	*	–	–
1ppe	E/R	1tab	100.0	–	2	Y + U	1	3.54	1	3.69	1	3.29
2sic	E/R	1cse	63.4	–	1	Y + U	1	9.69	1	3.40	1	7.33
1nw9	E/M	1i4e	34.9	–	1	N + U	*	*	*	*	23	6.49
1fq1	E/D	1buh	98.6	–	3	N + U	*	*	*	*	–	–
1zli	E/D	2bo9	7.5	–	1	Y + U	2	6.90	44	9.82	2	7.13
1buh	O/R	1fin	98.6	–	2	N + U	*	*	*	*	–	–
1e96	O/R	1foe	97.2	–	3	N + U	*	*	*	*	–	–
1gcq	O/R	1ycs	32.7	–	1	N + U	*	*	*	*	–	–
1i4d	O/R	1g4u	99.4	–	3	N + U	*	*	*	*	–	–
1kac	O/R	1akj	–	17.0	1	U + Y	–	–	–	–	265	8.76
1s1q	O/R	1otr	–	89.9	2	U + N	*	*	*	*	190	9.59
1z0k	O/R	1ukv	49.4	–	5	Y + U	1	3.20	309	9.8	2	7.06
2g77	O/R	1ukv	–	40.7	5	U + Y	–	–	–	–	55	8.58
1lfd	O/M	1wq1	99.4	–	2	Y + U	48	8.59	–	–	–	–
1mq8	O/M	1ijk	–	21.7	2	U + N	*	*	*	*	–	–
1wq1	O/M	1gua	57.7	–	2	Y + U	–	–	–	–	–	–
2h7v	O/M	1g4u	99.4	–	5	Y + U	–	–	–	–	–	–
1r8s	O/D	1r4a	55.6	–	3	Y + U	–	–	–	–	–	–
1y64	O/D	1nm1	89.3	–	4	Y + U	–	–	–	–	–	–
2ido	O/D	1zbu	17.8	–	1	N + U	*	*	*	*	–	–
2ot3	O/D	2hv8	38.3	–	5	Y + U	–	–	–	–	–	–

E, enzyme-inhibitor target; O, other target; R, rigid-body target; M, medium difficulty target; D, difficult target; Y, yes (binding site reused); N, no (binding site not reused); U, unknown binding site.
Here, KB denotes results taken directly from KBDOCK. An asterisk (*) denotes no solution was found by CBR because a binding site was not reused. A hyphen (-) denotes no solution found within the 200 Rosetta refined orientations or within the top 2000 blind *Hex* orientations. For one target (PDB 1eaw), multiple cases are retrieved and considered ("DFBS Pairs"). For this target, the table shows only a single template, being the one with the highest percentage sequence identity (SID) to the query structures.

RMSD in one other case (1gl1). For the sixth KBDOCK template (1fle), blind *Hex* fails to find any solution. On the other hand, blind *Hex* finds a solution for one further case (1f34) which KBDOCK missed because its binding site is not reused.

Figure 3 shows an example of the improvement in side chain and backbone coordinates that may be achieved using RosettaDock. However, applying RosettaDock to each of the top 100 solutions from *Hex* is computationally expensive, requiring on average around 175 CPU-hours per SH target. In comparison, both the KBDOCK+*Hex* and the blind *Hex* protocols can produce predictions with comparable backbone RMSDs in just a few seconds, as described above. However, it is worth noting that blind *Hex* docking is normally only successful for SH enzyme-inhibitor targets in the rigid category. Medium and difficult SH targets are hard to model successfully due to the difficulty of finding any near-native poses in the rigid-body refinement stage. Indeed, five of the seven other cases in which rigid-body refinement failed are either medium or difficult targets (1wq1, 2h7v, 1r8s, 1y64, and 2ot3). This suggests that a pair of proteins which undergo significant conformational changes during docking will be difficult to dock by homology unless there is a FH template in the CB. It is also interesting to note that in several of the SH-1 cases, a high sequence identity template (≥97%) is found for one of the structures (1fq1, 1buh, 1e96, and 1i4d), yet the DFBS of the corresponding solution structure is novel with respect to the data-filtered CB ("DFBS reused" = "N"). Hence, for SH cases, it does not necessarily follow that a high sequence identity between a query and a template implies that a binding site will be reused.

Overall, Tables II and III show that when DFBSs are indeed reused, around 50% of those cases can be successfully docked and repacked. However, the protocol described here assumes that DFBSs are reused in all targets. For targets which employ previously unseen binding sites, this could mean that no suitable templates are passed to the refinement stage, thus precluding the possibility of finding a solution (as denoted by an asterisk in Tables II and III). This suggests that a better protocol for modeling SH cases could be to apply both CBR and blind docking runs, and to upgrade the scores for blind

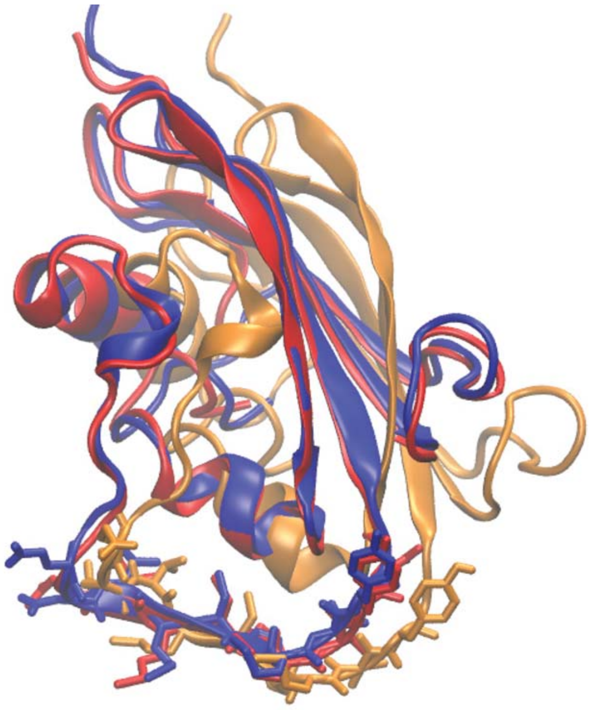


Figure 3

Ribbon cartoon representations of the best *Hex* and RosettaDock solutions for the SH-1 target 2sic, a complex between a peptidase (Pfam family: *Peptidase_S8*) and its inhibitor (*SSI*). This figure shows the conformations of the native crystal structure of SSI in red, the top *Hex* solution in orange, and the top RosettaDock solution in blue. The residues of the SSI inhibitory loop are drawn as sticks (the peptidase structure is not shown here). Compared to the *Hex* solution (ligand C_α RMSD of 9.7 Å), RosettaDock improves the side-chain and backbone coordinates considerably (3.40 Å RMSD).

docking solutions which involve known DFBSs. It also suggests that in cases where no FH templates are found, it could be fruitful to extend the case retrieval algorithm to consider DDIs between Pfam families which are structurally similar but not identical to those of the query domains. More generally, it would be interesting to study in greater depth the extent to which Pfam domains might reuse their binding sites to bind domains from other unrelated Pfam families. Indeed, it should be noted that several of Weng's benchmark targets involve protease/inhibitor complexes, and several of the other targets involve members of the *Ras* family. It would therefore be desirable to create a new homology docking benchmark with a diverse set of examples of varying difficulty with which to test more thoroughly template-based modeling approaches.

CONCLUSIONS

We have described a systematic CBR approach to model the 3D structures of protein complexes from structural DDIs, and we have tested it on a well known benchmark dataset. By working at the Pfam domain level, we were able to draw upon a large and nonredundant set of hetero DDIs, and by spatially clustering DDIs by domain family we were able to define family-level binding sites in a largely sequence-independent way. This allowed the problem of searching for 3D docking templates to be reduced essentially to a simple symbolic pattern matching exercise. The results in Table I show that for FH problems, our approach provides a near-perfect and automatic way to retrieve suitable 3D templates with which to build "acceptable" or better models of the target complexes, and that such models often rank within the first handful of solutions. Furthermore, even when FH templates do not exist in the CB, we find that known binding sites are reused in approximately 50% of the studied examples, and that our notion of SH templates provides a useful way to guide conventional rigid-body docking and refinement algorithms. Thus, our CBR method of reusing DDIs extends the reach of current homology modeling techniques.

With the rapid growth in the number of solved structures in the PDB, it seems clear that automatic homology modeling approaches will become increasingly important for modeling protein complexes. However, the rather limited range of different complexes in Weng's benchmark points to the need for more extensive testing on more diverse examples. We propose that assembling a new benchmark set for template-based protein docking, which contains several levels of difficulty like Weng's benchmark, would provide a useful resource for the docking community. The results presented here show that many FH targets can easily be modeled automatically by homology, although further refining a good database template using conventional protocols is difficult. On the other hand, our results for the SH cases suggest that the highly flexible targets in the difficult category will continue to be difficult to model by homology. Thus, there will remain many challenging docking targets for the CAPRI community to deal with in the coming years.

REFERENCES

1. Wodak SJ, Janin J. Computer analysis of protein–protein interaction. J Mol Biol 1978;124(2):323–342.
2. Halperin I, Ma B, Wolfson H, Nussinov R. Principles of docking: an overview of search algorithms and a guide to scoring functions. Proteins 2002;47:409–443.
3. Janin J, Henrick K, Moult J, Ten Eyck L, Sternberg MJE, Vajda S, Vakser I, Wodak SJ. CAPRI: a critical assessment of predicted interactions. Proteins 2003;52:2–9.
4. Wodak SJ, Mendez R. Prediction of protein–protein interactions: the CAPRI experiment, its evaluation and implications. Curr Opin Struct Biol 2004;14(2):242–249.
5. Lensink MF, Wodak SJ. Docking and scoring protein interactions: CAPRI 2009. Proteins 2010;78(15):3073–3084.
6. van Dijk AD, Boelens R, Bonvin AM. Data-driven docking for the study of biomolecular complexes. FEBS J 2005;272(2):293–312.

7. Berman HM. The Protein Data Bank: a historical perspective. Acta Crystallogr 2008;A38:88–95.
8. Tuncbag N, Kar G, Keskin O, Gursoy A, Nussinov R. A survey of available tools and web servers for analysis of protein–protein interactions and interfaces. Brief Bioinform 2009;10(3):217–232.
9. Levy ED, Pereira-Leal JB, Chothia C, Teichmann SA. 3D complex: a structural classification of protein complexes. PLoS Comput Biol 2006;2(11):1395–1406.
10. Douguet D, Chen HC, Tovchigrechko A, Vakser IA. Dockground resource for studying protein–protein interfaces. Bioinformatics 2006;22:2612–2618.
11. Higurashi M, Ishida T, Kinoshita K. PiSite: a database of protein interaction sites using multiple binding states in the PDB. Nucleic Acids Res 2009;37:D360–D364.
12. Kundrotas PJ, Zhu ZW, Vakser IA. GWIDD: genome-wide protein docking database. Nucleic Acids Res 2010;38:D513–D517.
13. Faure G, Andreani J, Guerois R. InterEvol database: exploring the structure and evolution of protein complex interfaces. Nucleic Acids Res 2012;40:D847–D856.
14. Stein A, Ceol A, Aloy P. 3did: identification and classification of domain-based interactions of known three-dimensional structure. Nucleic Acids Res 2010;39:D718–D723.
15. Xu Q, Dunbrack RL. The protein common interface database (ProtCID)—a comprehensive database of interactions of homologous proteins in multiple crystal forms. Nucleic Acids Res 2011;39:D761–770.
16. Davis FP, Sali A. PIBASE: a comprehensive database of structurally defined protein interfaces. Bioinformatics 2005;21:1901–1907.
17. Winter C, Henschel A, Kim WK, Schroeder M. SCOPPI: a structural classification of protein–protein interfaces. Nucleic Acids Res 2006;34:D310–D314.
18. Teyra J, Paszkowski-Rogacz M, Anders G, Pisabarro TM. SCOWLP classification: structural comparison and analysis of protein binding regions. BMC Bioinformatics 2008;9:9.
19. Shoemaker BA, Zhang D, Tyagi M, Thangudu RR, Fong JH, Marchler-Bauer A, Bryant SH, Madej T, Panchenko AR. IBIS (Inferred Biomolecular Interaction Server) reports, predicts and integrates multiple types of conserved interactions for proteins. Nucleic Acids Res 2012;40:D834–840.
20. Finn RD, Mistry J, Tate J, Coggill P, Heger A, Pollington JE, Gavin OL, Gunasekaran P, Ceric G, Forslund K, Holm L, Sonnhammer ELL, Eddy SR, Bateman A. The Pfam protein families database. Nucleic Acids Res 2010;38:D211–D222.
21. Murzin AG, Brenner SE, Hubbard T, Chothia C. SCOP—a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 1995;247(4):536–540.
22. Marchler-Bauer A, Anderson JB, Chitsaz F, Derbyshire MK, DeWeese-Scott C, Fong JH, Geer LY, Geer RC, Gonzales NR, Gwadz M, He S, Hurwitz DI, Jackson JD, Ke Z, Lanczycki CJ, Liebert CA, Liu C, Lu F, Lu S, Marchler GH, Mullokandov M, Song JS, Tasneem A, Thanki N, Yamashita RA, Zhang D, Zhang N, Bryant SH. CDD: specific functional annotation with the Conserved Domain Database. Nucleic Acids Res 2009;37:D205–D210.
23. Aloy P, Ceulemans H, Stark A, Russell RB. The relationship between sequence and interaction divergence in proteins. J Mol Biol 2003; 332(5):989–998.
24. Chothia C, Lesk AM. The relation between the divergence of sequence and structure in proteins. EMBO J 1986;5:823–826.
25. Korkin D, Davis FP, Sali A. Localization of protein-binding sites within families of proteins. Protein Sci 2005;14:2350–2360.
26. Shoemaker BA, Panchenko AR, Bryant SH. Finding biologically relevant protein domain interactions: conserved binding mode analysis. Protein Sci 2006;15(2):352–361.
27. Lu L, Lu H, Skolnick J. Multiprospector: an algorithm for the prediction of protein–protein interactions by multimeric threading. Proteins 2002;49(3):350–364.
28. Grimm V, Zhang Y, Skolnick J. Benchmarking of dimeric threading and structure refinement. Proteins 2006;63:457–464.
29. Launay G, Simonson T. Homology modelling of protein–protein complexes: a simple method and its possibilities and limitations. BMC Bioinformatics 2008;9:427.
30. Mukherjee S, Zhang Y. Protein–protein complex structure predictions by multimeric threading and template recombination. Structure 2011;19(7):955–966.
31. Korkin D, Davis FP, Alber F, Luong T, Shen M-Y, Lucic V, Kennedy MB, Sali A. Structural modeling of protein interactions by analogy: application to PSD-95. PLoS Comput Biol 2006;2(11):e153.
32. Günther S, May P, Hoppe A, Frommel C, Preissner R. Docking without docking: ISEARCH—prediction of interactions using known interfaces. Proteins 2007;69(4):839–844.
33. Kundrotas PJ, Lensink MF, Alexov E. Homology-based modeling of 3D structures of protein–protein complexes using alignments of modified sequence profiles. Int J Biol Macromol 2008;43(2):198–208.
34. Petrey D, Xiang Z, Tang CL, Xie L, Gimpelev M, Mitros T, Soto CS, Goldsmith-Fischman S, Kernytsky A, Schlessinger A, Koh IY, Alexov E, Honig B. Using multiple structure alignments, fast model building, and energetic analysis in fold recognition and homology modeling. Proteins 2003;53 Suppl 6:430–435.
35. Movshovitz-Attias D, London N, Schueler-Furman O. On the use of structural templates for high-resolution docking. Proteins 2010; 78(8):1939–1949.
36. Ghoorah AW, Devignes M-D, Smaïl-Tabbone M, Ritchie DW. Spatial clustering of protein binding sites for template based protein docking. Bioinformatics 2011;27(20):2820–2827.
37. Kolodner JL. An introduction to case-based reasoning. Artif Intell Rev 1992;6:3–34.
38. Aamodt A, Plaza E. Case-based reasoning; foundational issues, methodological variations, and system approaches. AI Commun 1994;7(1):39–59.
39. López de Mántaras R, McSherry D, Bridge DG, Leake DB, Smyth B, Craw S, Faltings B, Maher ML, Cox MT, Forbus KD, Keane MT, Aamodt A, Watson ID. Retrieval, reuse, revision and retention in case-based reasoning. Knowl Eng Rev 2005;20(3):215–240.
40. Gray JJ, Moughon S, Wang C, 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(1):281–299.
41. Ritchie DW, Kozakov D, Vajda S. Accelerating and focusing protein–protein docking correlations using multi-dimensional rotational FFT generating functions. Bioinformatics 2008;24(17):1865–1873.
42. Chuang G-Y, Kozakov D, Brenke R, Comeau SR, Vajda S. DARS (decoys as the reference state) potentials for protein–protein docking. Biophys J 2008;95(9):4217–4227.
43. Hwang H, Vreven T, Janin J, Weng Z. Protein–protein docking benchmark version 4.0. Proteins 2010;78(15):3111–3114.