WILEY InterScience® DISCOVER SOMETHING GREAT

PROTEINS STRUCTURE ■ FUNCTION ■ BIOINFORMATICS

FireDock: Fast interaction refinement in molecular docking

Nelly Andrusier,[1] Ruth Nussinov,[2,3] and Haim J. Wolfson[1]*

[1]School of Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel

[2]Basic Research Program, SAIC-Frederick, Inc., Center for Cancer Research Nanobiology Program, NCI - Frederick, Frederick, Maryland 21702

[3]Department of Human Genetics and Molecular Medicine, Sackler Faculty of Medicine, Tel Aviv University, Tel Aviv 69978, Israel

ABSTRACT

Here, we present FireDock, an efficient method for the refinement and rescoring of rigid-body docking solutions. The refinement process consists of two main steps: (1) rearrangement of the interface side-chains and (2) adjustment of the relative orientation of the molecules. Our method accounts for the observation that most interface residues that are important in recognition and binding do not change their conformation significantly upon complexation. Allowing full side-chain flexibility, a common procedure in refinement methods, often causes excessive conformational changes. These changes may distort preformed structural signatures, which have been shown to be important for binding recognition. Here, we restrict side-chain movements, and thus manage to reduce the false-positive rate noticeably. In the later stages of our procedure (orientation adjustments and scoring), we smooth the atomic radii. This allows for the minor backbone and side-chain movements and increases the sensitivity of our algorithm. FireDock succeeds in ranking a near-native structure within the top 15 predictions for 83% of the 30 enzyme-inhibitor test cases, and for 78% of the 18 semi-unbound antibody-antigen complexes. Our refinement procedure significantly improves the ranking of the rigid-body PatchDock algorithm for these cases. The FireDock program is fully automated. In particular, to our knowledge, FireDock's prediction results are comparable to current state-of-the-art refinement methods while its running time is significantly lower. The method is available at http://bioinfo3d.cs.tau.ac.il/FireDock/.

Proteins 2007; 69:139–159.

Key words: *docking refinement; side-chain flexibility; linear programming; rigid-body minimization; binding free energy; protein-protein docking.*

INTRODUCTION

The goal of molecular docking is to predict the three-dimensional structure of two or more individual molecules upon complexation. Experimentally, it is still a hard and time-consuming problem. Thus, computational methods, which perform well on existing docking benchmarks, are required. The difficulty in computational docking derives from the structural changes that partners undergo during the binding. These structural changes can include only side-chain movements or be more significant such as backbone deformations.

The docking procedure usually consists of two main steps: docking candidate generation and their refinement. The first step is performed by efficient global search of relative receptor-ligand orientations. It should be comprehensive enough to include the near-native candidates and at the same time selective enough to generate a feasible number of candidates. The majority of the docking methods use simplified rigid models of the molecular structures and apply various techniques for candidate generation. The most computationally efficient techniques are FFT,[1] geometric hashing,[2–4] and the polar Fourier Transform.[5] The refinement stage is aimed at singling out the correct solutions from the large set of candidates. Since molecules are not rigid bodies, the docking simulation requires an accurate model of the binding molecules. This model should consider both backbone and side-chain flexibility, while optimizing the molecules' relative orientation with respect to each other. The refinement stage attempts to meet this challenge by "improving" the "rigid" candidate hypotheses.

The Supplementary Material referred to in this article can be found at http://www.interscience.wiley.com/jpages/0887–3585/suppmat/

Grant sponsor: NIAID; Grant sponsor: NIH; Grant number: 1UC1AI067231; Grant sponsor: Binational US-Israel Science Foundation; Grant sponsor: Israel Science Foundation; Grant number: 281/05; Grant sponsor: Hermann Minkowski-Minerva Center for Geometry; Grant sponsor: NCI, NIH; Grant number: NO1-CO-12400; Grant sponsor: Intramural Research Program, NCI, NIH

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

Received 28 August 2006; Revised 11 January 2007; Accepted 5 March 2007
Published online 27 June 2007 in Wiley InterScience (www.interscience.wiley.com).
DOI: 10.1002/prot.21495

Existing methods for docking candidate generation are extremely fast; on the other hand, the refinement step remains a time-consuming task. Even though the majority of the existing refinement methods allow only side-chain flexibility, the computational complexity is still high. The general flexible refinement problem, with possible backbone movements, still remains a challenge. A first step toward its solution is the fast side-chain flexible refinement, which can be used as a building block for the backbone flexible refinement.

Here, we describe the FireDock (Fast Interaction REfinement in molecular DOCKing) method for the refinement and rescoring of rigid-body docking candidates. The starting candidates are generated by the PatchDock method,[4] developed in our group. It is a highly efficient rigid-docking algorithm, based on shape complementarity of soft molecular surfaces. The method obtained very good results in rounds 3–5 of the CAPRI* challenge, and can be further improved by incorporating explicit flexibility and an appropriate scoring function. Recently, a new category of "scorers" was opened in CAPRI. This category aims to test the ability of different docking groups to refine and rerank the given candidates. Each participating "predictor" group submits 100 models to the common basket of models, which assemble the candidates set. To succeed in this category, a refinement stage is required.

FireDock optimizes the binding of each candidate by allowing flexibility in the side-chains and adjustments of the relative orientation of the molecules. Most of the interface residues, which are important to the binding recognition, remain in the near-unbound conformations upon complexation. This concept has been proposed by several studies[6–8] and is validated in our analysis. Here, we use this observation and restrict the side-chain flexibility only to the clashing interface residues. In addition, the atomic radii of the partners are smoothed in the rigid-body optimization and scoring stages. This "coarse refinement" approach is the key to the efficiency of the proposed method. Scoring of the refined candidates is based on softened van der Waals interactions, Atomic Contact Energy (ACE),[9] electrostatic, and additional binding free energy estimations.

Here, we present the results of a large-scale test on enzyme-inhibitor and antibody-antigen cases, derived from the docking benchmarks.[10,11] The refinement stage performed by FireDock clearly improves the ranking of PatchDock for these cases. FireDock runs $\approx$ 4 seconds per candidate solution on a Pentium©IV CPU 3.2 GHz with 1 GB RAM.

PREVIOUS METHODS

Most of the refinement algorithms comprise of the three following procedures: side-chain prediction, rigid-body optimization, and, finally, ranking of the candidates.

*Community wide experiment on the comparative evaluation of protein-protein docking for structure prediction.

The side-chain prediction problem has been widely studied in the recent years as a step of side-chain assignment on a fixed backbone, mostly in the context of protein design and homology modeling. Existing methods are highly efficient and obtain accurate predictions if the given backbone conformation is close to the native (with an RMSD below 1 Å). To overcome the computational difficulty, the residue conformations are usually discretized by the statistically favored conformations called *rotamers*. Most of the methods refer to the problem as a *combinatorial optimization* problem with a set of possible rotamers for each residue and a pair-wise energy function between them. The aim is to find the rotamers combination with the global minimal energy, i.e., a global minimal energy conformation (GMEC). The problem has been proved to be NP-hard[12] and inapproximable.[13] In practice, residues topological restraints can be used in many cases to find efficiently the GMEC.

The dead-end elimination (DEE) method pioneered by Desmet,[14] is based on pruning the rotamers, which will not be a part of GMEC, since there are better alternatives. Bahadur *et al.*[15] define a weighted graph of noncolliding rotamers and search for the *maximum edge-weight clique*. Other methods use *residue interaction graph* formulation. The assumption that distant residues do not interact leads to the resulting sparse graph. The SCWRL[16] algorithm takes advantage of this graph topology by generating *biconnected components* and applying the *dynamic programming* paradigm to find a GMEC. For graphs that cannot be conveniently decomposed into small biconnected components, SCATD[17] defines a more sophisticated tree decomposition of the residue interaction graph. In addition to the *rotamer reduction* by DEE, many methods use *residue reduction* too. The R3 method[18] repeatedly performs residue and rotamer reduction. Recent methods find the GMEC using the *integer linear programming* algorithm.[19–21] The majority of the global optimization algorithms use a simplified pair-wise energy function, which includes only the repulsive van der Waals interaction. Therefore, the residue interaction graph is sparse enough and can be solved efficiently. When the function is extended by additional terms, like the attractive van der Waals, solvation, and electrostatics, the global optimization methods are less effective.

To overcome the efficiency problem with a more realistic energy function, one can use heuristic algorithms, which however do not guarantee finding the GMEC. Monte Carlo (MC) methods[22] randomly choose some residue and exchange a current rotamer by another. The new energy is accepted or rejected by the Metropolis criterion.[23] The self-consistent mean-field method[24,25] iteratively refines a conformational matrix, which includes the probability of certain rotamers to be optimal. Other approaches like genetic algorithms[26] and neural networks[27] are also used to solve the side-chains prediction problem.

DOI 10.1002/prot

Several methods explore the side-chains conformational space without rotamers restriction.[28,29] Abagyan *et al.*[29] use the biased probability MC method for the minimization in side-chain torsion angles space. Some researchers treat side-chain flexibility by performing short MD simulations.[6,8,30]

The rigid-body optimization stage aims to bring the molecules to optimal fit by minimizing the interaction energy in the six-dimensional translation and rotation space of rigid movements. Most of the methods use randomized multidimensional minimization techniques. These techniques usually work iteratively and in each iteration find the best direction for the minimum search and apply a line search in this direction.

The energy function is very important for accurate refinement and correct solution discrimination. Currently, it differs between methods and is coupled with the search procedure and the molecule representation. However, a common tendency can be outlined. Most of the methods express the van der Waals interactions by the modified Lennard-Jones 6–12 potentials with a smoothed repulsive part to avoid extreme sensitivity to clashes caused by small backbone movements and the rotamer discretization. The electrostatic effect is commonly estimated by the Coulomb model and knowledge-based potentials. The desolvation free energy is mostly represented by implicit solvation models. Good results are obtained by a very simple and fast ACE.[9] Many methods use additional knowledge-based terms such as the rotamer probabilities, the residue pair potentials, etc.

Here, we briefly review a number of previously developed methods for the refinement of rigid-body docking solutions. The RosettaDock refinement[31,32] performs repeated cycles of side-chain optimization and adjustments of the relative orientation of the molecules. The side-chain optimization is based on the Monte Carlo sampling of the rotamers with minimization in the torsion space. The rigid-body optimization of RosettaDock is based on the MC gradient-based minimization of the binding score. In addition, as a validation and quality test, each candidate is examined for the binding funnel around its location. ICM-DISCO[33] uses internal coordinates for the molecular complex representation. The internal variables of the interface ligand residues and the six rigid-body positional variables are optimized by the biased probability MC method. 3D-DOCK[34] also performs repeated cycles of side-chains optimization and rigid-body energy minimization. The side-chains conformations are optimized by the mean-field approach, which takes into account the surrounding solvent molecules by the "soft" sphere Langevin dipole model. HEX refinement[5] uses a softened energy function to overcome the side-chain flexibility in the rigid-body minimization. RDOCK[35] refines the candidates by repeated energy minimizations with CHARMM.[36] SmoothDock[37] uses short MD simulations to predict conformations of anchor side-chains[6] at the predocking phase. They optimize the binding of each candidate by the simplex rigid-body minimization and by sliding the ligand closer to the receptor and repeated CHARMM minimizations. HADDOCK[30] refines the candidates by repeated cycles of simulated annealing with different degrees of freedom. In the final stage, restrained MD simulations with explicit solvent are performed. ATTRACT[38] performs rigid-body docking with multicopies of residues. BiGGER[39] refers to side-chain flexibility by molecular surface softening for the statistically most flexible residues.

METHOD

In this section, we describe the FireDock method. First, we explain the general scheme of the method. Next, we describe the binding model and score used here and detail every stage of the method.

Outline of the algorithm

The input to the FireDock method are "rigid-docking" candidates. FireDock optimizes them by allowing side-chain flexibility and small rigid-body movements. Figure 1 depicts the flowchart of the method. The candidates of the complex structure are generated by the PatchDock algorithm.[4] Each candidate is subsequently refined by restricted interface side-chain rearrangement and by soft rigid-body optimization. The interface side-chain optimization restricts the side-chain flexibility to the interface clashing residues. The flexibility is modeled by rotamers and the obtained combinatorial optimization problem is solved by integer linear programming.[19–21] Following the rearrangement of the side-chains, the relative position of the docking partners is refined by Monte Carlo minimization of the binding score function. The refined candidates are ranked by the binding score. This score is an ACE-based free energy function, which also includes softened van der Waals interactions, partial electrostatics, and additional estimations of the binding free energy.

Finally, the 25 top-ranked candidates are refined again with full interface side-chain optimization and soft rigid-body minimization. In this final "full refinement" stage, the atomic radii are less smoothed in the rigid-body optimization and scoring stages.

Binding model

The side-chains flexibility is modeled by rotamers. We use the popular Dunbrack *backbone-dependent rotamer library*.[40] The rotamer set for each residue is created by collecting all high-probability rotamers from the library until their accumulated probabilities reach a predefined threshold. Here, we define small and extended rotamer sets with thresholds equal to 0.9 and 0.99, respectively. In

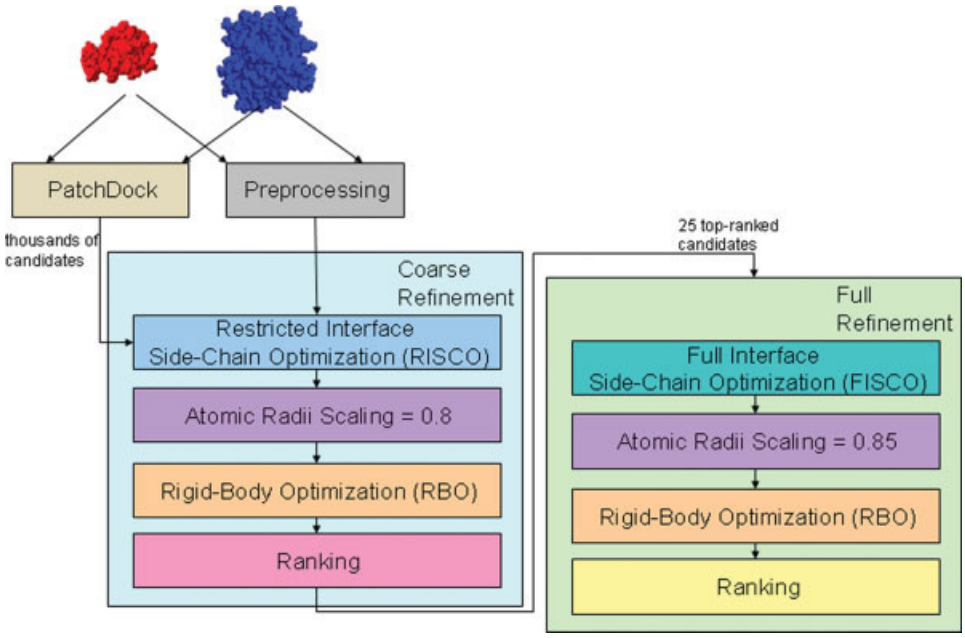

Figure 1
The FireDock flow.

addition, in the extended rotamer set, the three standard χ_1 angles are tripled by adding and subtracting one standard deviation.[31] This results in nine rotamers for χ_1 instead of three. The unbound residue conformation is also included in the residue rotamers set.

Binding score

The *binding free energy* is the change in the free energy of the system, which occurs upon complex formation,

$$\Delta G = G^C - (G^R + G^L) \quad (1)$$

where G^C is the free energy of the receptor-ligand complex. G^R and G^L are the free energies of the uncomplexed receptor and ligand, respectively. The binding score for the candidates ranking is an approximation of the binding-free energy function. Many docking methods calculate only G^C, which is enough for ranking, because G^R and G^L are constant for all the candidates.

We define an *interface residue* as a residue with at least one atom within a 6 Å of any atom of the docking partner. All the interface residues' atoms are called *interface atoms*. In our binding model, we optimize only the interface region of a given complex, and only the interface side-chains can change their conformations. Therefore, we split the complex to the interface (intrf) and noninterface (nintrf) regions and assume that $G^C_{\text{nintrf}} = G^R_{\text{nintrf}} + G^L_{\text{nintrf}}$. In addition, the complex energy can be split to the *intramolecular energy* inside each molecule (intra) and the *intermolecular energy* (inter). The final ΔG [Eq. (2)] is described by the intermolecular complex energy and the change in the interface intramolecular energies of each molecule caused by binding.

$$\begin{aligned}\Delta G = \Delta G_{\text{intr f}} &= G^C_{\text{intr f}} - (G^R_{\text{intr f}} + G^L_{\text{intr f}}) \\ &= G^C_{\text{intr f\_inter}} + \Delta G^L_{\text{intr f\_intra}} + \Delta G^R_{\text{intr f\_intra}} \end{aligned} \quad (2)$$

Solvation

The desolvation free energy is estimated by the ACE potential[9]:

$$\Delta G_{\text{ACE}} = \sum_i \sum_j e_{ij} \quad (3)$$

The parameter e_{ij} is defined as the effective free energy change when a bond between two atoms of type i and j is replaced by solute-solvent bonds.

Electrostatics

The electrostatic contribution is approximated by the atomic pair-wise partial Coulomb energy:

$$E_{\text{elec}} = \sum_{i,j} 332.0637 \frac{q_i q_j}{\varepsilon \hat{d}_{ij}} = \sum_{i,j} 332.0637 \frac{q_i q_j}{\hat{d}_{ij}^2} \quad (4)$$

where q_i, q_j are the charges of the atoms i, j, derived from the CHARMM19 force field.[41] $\hat{d}_{ij}$ is the modified

DOI 10.1002/prot

distance between atoms i, j and equals to $\max(d_{ij}, 3\ \text{Å})$ to avoid singularity as $d_{ij} \to 0$.[31] The dielectric coefficient is approximated by distance ($\varepsilon = d_{ij}$). The charges of only the most polar groups are considered (ARG, LYS, ASP, GLU, C-terminus, and N-terminus). Other groups have a weaker contribution to the electrostatic energy and are partially covered in the hydrogen bonds and π-stacking calculation. In addition, E_{elec} is separated to attractive/repulsive and short/long-range categories as in Ref. 31: E_{attEl} ($E_{\text{elec}} < 0$ and $d_{ij} < 5$ Å), E_{repEl} ($E_{\text{elec}} > 0$ and $d_{ij} < 5$ Å), $E_{\text{l\_attEl}}$ ($E_{\text{elec}} < 0$ and $d_{ij} \geq 5$ Å), and $E_{\text{l\_repEl}}$ ($E_{\text{elec}} > 0$ and $d_{ij} \geq 5$ Å).

Internal energy

The source of the internal energy change is in the molecular deformations upon binding. It includes bond stretching, angle bending, and torsional energy terms. Here, we do not change bond lengths and angles, but we do change the torsion angles of residues by choosing rotamers. The torsion energy term is reflected by a backbone-depended rotamer frequency derived from the Dunbrack library[40]:

$$E_{\text{rot}}(i_r) = -\log \frac{S(i, r, \phi_i, \psi_i)}{S(i, r_{\text{unb}}, \phi_i, \psi_i)} \tag{5}$$

$S(i,r,\phi_i,\psi_i) = P(i,r,\phi_i,\psi_i)$, which is the rotamer r probability of residue i with the ϕ_i,ψ_i backbone angles. The unbound residue conformation is also added to the residue rotamers set with the score $S(i,r_{\text{unb}},\phi_i,\psi_i) = P(i,r_{\max},\phi_i,\psi_i) + 0.3$, where $r_{\max}$ is the most common rotamer for the current residue i. Note that the rotamer probabilities were replaced by scores, since they were modified, and their sum is not one.

Hydrogen and disulfide bonds

The contribution of the hydrogen bonds is calculated by the 12-10 hydrogen potential[42]:

$$E_{\text{HB}} = 5\left(\frac{r_0}{d_{ij}}\right)^{12} - 6\left(\frac{r_0}{d_{ij}}\right)^{10}, \quad 2.74\ \text{Å} < d_{ij} < 3.5\ \text{Å} \tag{6}$$

Here, d_{ij} is the distance between the interface acceptor and donor atoms and $r_0 = 2.9$ Å is the optimal distance for hydrogen bonding.

The disulfide bond contribution is -1 energy units and is summed for all interface S–S pairs with distance between 1.9 and 2.1 Å.

van der Waals interactions

We use the same definition of van der Waals energy as in Ref. 31. Specifically, the van der Waals energy between the two atoms a_i and a_j is defined as the modified Lennard-Jones 6–12 potential with linear short-range repulsive score, i.e.,

$$E_{\text{VdW}}(a_i, a_j) = \begin{cases} \varepsilon_{ij}\left(\dfrac{\sigma_{ij}^{12}}{r_{ij}^{12}} - 2\dfrac{\sigma_{ij}^{6}}{r_{ij}^{6}}\right), & \text{for } r_{ij} > 0.6\sigma_{ij} \\ \varepsilon_{ij}(A + (r_{ij} - 0.6\sigma_{ij})B), & \text{otherwise} \end{cases} \tag{7}$$

$$\text{where } A = \frac{\sigma_{ij}^{12}}{0.6\sigma_{ij}^{12}} - 2\frac{\sigma_{ij}^{6}}{0.6\sigma_{ij}^{6}}, B = -12\frac{\sigma_{ij}^{12}}{0.6\sigma_{ij}^{13}} + 12\frac{\sigma_{ij}^{6}}{0.6\sigma_{ij}^{7}} \tag{8}$$

The parameter σ_{ij} is the atomic radii sum and the parameter ε_{ij} is the energy well depth, derived from the CHARMM19 force field.[41] The hydrogen atomic radii are reduced by 40% because of their uncertain positions. All the energy calculations are performed within 6 Å interatomic distance. If $E_{\text{VdW}}(a_i,a_j)$ is a positive number, then it is added to the repulsive van der Waals term E_{repVdW}, otherwise to the attractive van der Waals term E_{attrVdW}. The repulsion penalty for hydrogen donor–acceptor pairs is neglected (if the distance between hydrogen and acceptor atoms is less than 2.5 Å).

To overcome the possible backbone fluctuations during binding, we use softened van der Waals interactions in particular stages of the algorithm. $E_{\text{s\_attrVdW}}$ and $E_{\text{s\_repVdW}}$ are the softened van der Waals interactions, which are calculated as E_{attrVdW} and E_{repVdW}, but with reduced atomic radii.

π-Stacking and aliphatic interactions

π–π and hydrophobic interactions are known to be important for the stability of the protein–protein interactions. Recently, Crowley *et al.*[43] showed that the cation-π interactions are also important contributors to protein–protein interfaces. These are the interactions of the positive amines of Lys or Arg with the π-electron cloud of the aromatic side-chains (Phe, Tyr, Trp). π–π interactions occur between the aromatic residues: Phe, Tyr, His, Trp, and Pro. Actually, these interactions should be reflected by the electrostatic energy. However, here, we use the partial point-charge-based Coulomb energy, which is not sufficient for the full expression of these interactions. Therefore, we add three additional components to the total binding score: E_{pipi} for the π – π interactions, E_{catpi} for the cation-π interactions, and E_{aliph} for the aliphatic interactions. The aliphatic pairs consist of Leu, Ile, and Val.

Misura *et al.*[44] compared the arrangement of the cation-π, π–π and aliphatic pairs in the native and non-native structures. They found that particular orientations and distances characterize the native structures. Here, we neglect the orientation and use only the distance between the planar group centers, which are defined in Ref. 44. If the distance is between 5.5 and 6.5 Å, then the contribution is -1.5 energy units. For the larger distance, with the bound of 7.5 Å, the contribution is weaker and

equals −0.5 energy units. These scores are accumulated in E_{pipi}, E_{catpi} and E_{aliph} depends on the particular pair.

Correlation between binding score and experimental binding free energy

The binding free energy can be approximated by a sum of the previously described components. This calculated binding score is correlated with the experimental binding free energy. The correlation was tested for nine protease-inhibitor complexes as in Ref. 9. The obtained correlation coefficient is 0.83.

"Insideness" measure

The active site of enzymes is usually located in a cavity. The inhibitor enters inside the active site to neutralize its activity. The measure proposed here is a very rough measure for how much the ligand is inside the receptor ($E_{insideness}$). It is defined as the square distance between the receptor interface center of mass (ICM) and the ligand weighted interface center of mass (WICM). The WICM is calculated over the ligand interface atoms with weights equal to the number of its interacting pairs (atoms within 6 Å) in the receptor. The formal definition of $E_{insideness}$ appears in the "Supplementary Materials". The $E_{insideness}$ is expected to be smaller for concave interfaces. Figure 2 shows the difference between ICM and WICM. It schematically illustrates flat and concave interfaces of equal size, with each face colored differently. Despite the similar van der Waals interactions in both cases, the concave interface is more likely and we should distinguish it from the flat one. It is easy to see that the emphasized atom from the blue molecule is in the vicinity of more pink atoms if it is positioned in the cavity. If we assume that the atoms are distributed uniformally in the interface, the number of contacts for the blue atom is proportional to the pink volume of the sphere presented in the figure. In general, the variance of atomic contacts is small for flat interfaces and increases with the interface complexity. This observation is reflected geometrically in the WICM. For simplicity, Figure 2(c,d) demonstrate an interface with only one layer of atoms on each side, showing that the WICM of the blue face is shifted to the ICM of the pink one in the concave interface. In the case of enzyme-inhibitor interaction, we calculate $E_{insideness}(E,I)$. The measure can also be applied to the other interaction types, which are known to be concave. The overall contribution of $E_{insideness}$ to the enzyme-inhibitors prediction performance is analyzed in the Results section.

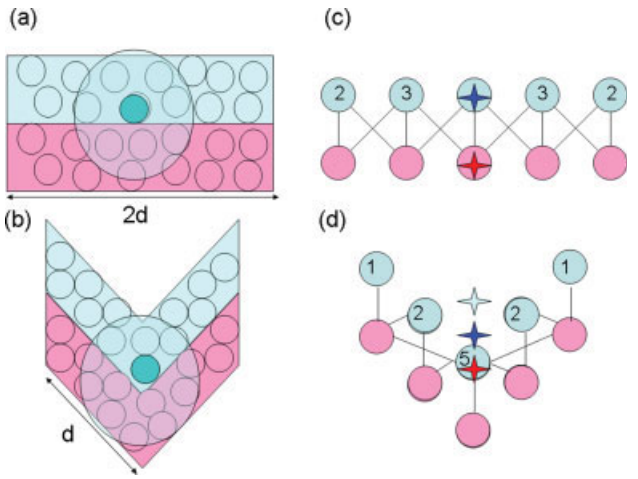

Figure 2

*The schematic representation of flat (**a,c**) and concave (**b,d**) interfaces. Each molecule side is colored differently. The emphasized blue atom is in the vicinity of more pink atoms in the concave case. (c) and (d) present an interface with one layer of atoms on each side. The edges and the numbers in circles represent the contacts with the opposite side. The blue and dark blue stars represent the ICM and the WICM of the blue atoms respectively. The red star indicates the ICM of the pink atoms. The distance between the red and the dark blue stars, i.e., the "insideness" measure, is smaller for the concave interface.*

Preprocessing

The two input molecules are preprocessed to speed up the refinement procedure. The preprocessing includes the following stages:

1. Addition of missing atoms. PDB files usually do not include hydrogen atoms, which can contribute to the binding score calculation. Therefore, we use the program REDUCE[45] to add and optimize them. In addition, other residue atoms can also be missing. We add them using the BALL library,[46] without any optimization. These residues should be defined as flexible and will be optimized during the side-chain optimization stage.
2. Rotamer building for all the surface residues of the nonrigid input proteins. A surface residue is determined by having at least one atom exposed to the solvent. If the given molecule is rigid (for example, the bound state), there is no need to build its rotamers. This prebuilding is helpful because of the high number of candidate binding sites, which usually cover all the molecular surface. The small and the extended rotamer sets were defined in Binding model section. The average numbers of rotamers per residue are 6 and 20 for the small and the extended sets, respectively.
3. Rotamer intramolecular energy calculation. The calculation is performed in each molecule separately and is described in the following section. These energies are used in the side-chains optimization algorithm and in the intramolecular binding score calculation.

Interface side-chain optimization

In this section, we describe the interface side-chain optimization (ISCO) procedure. We assume that each

DOI 10.1002/prot

unbound native structure is in its GMEC. The conformational changes upon complex formation should be modeled in such a way that the complex energy will be minimal. For a given rigid docking candidate, this problem is approximated by side-chain optimization in the interface. The FireDock ISCO consists of the following steps:

1. Selection of the movable residues.
2. Energy calculation for the rotamers.
3. Formulation of the side-chain optimization problem as an integer linear programming problem.
4. Solution of a relaxed linear programming problem.
5. If the linear program solution is not integral, then the integer linear program is solved.

Selection of movable residues

In general, all the side-chains can change their conformation upon binding. However, experimentally, several studies point out that most of the interface residues, which are important to the binding recognition, remain in the near-unbound conformations upon complexation.

Rajamani *et al.*[6] observed that the recognition of binding partners is based on ready-made *anchor residues* and *grooves*. The anchor residues were identified as the residues with the largest change in SASA (solvent-accessible-surface-area) after binding. They performed MD simulations of these residues free of the binding partners and found that the preferred MD conformer is close to the bound. However, it is not always close to the X-ray unbound structure. The binding grooves do not change significantly after MD and their bound and unbound forms are close.

Li *et al.*[7] examined the characteristics of the complemented pockets and the voids, which compose the protein interfaces. The *complemented pockets* are the concave surface regions in the unbound state that become complemented by the *protruding residues* in the bound state. They found that the complemented pockets and the protruding residues are enriched in structurally conserved residues and energetic hot spots. They also showed that the majority of the complemented pockets, which correspond to the binding grooves,[6] do not change significantly their conformation upon complex formation. The protruding residues correspond to the anchor residues.[6] Both studies suggest that the protruding residues and the complemented pockets are highly important in the binding process, and that the complemented pockets preserve their conformation during binding.

The protruding residues are mostly long and flexible residues with unbound conformations, which can disturb the correct docking prediction. These residues are likely to move during the docking process, and the others should be predominantly fixed. While only a few protruding residues exist in the interface, there are many residues, which are found in the pockets. Smith *et al.*[8] remarked that about 1/3 of the residues change rotameric state between the bound and unbound conformations. In the core of the interface, this fraction is slightly less, 1/4. Our results, presented in the Results section agree with these observations. They show that ~73% of the interface residues remain in the near-unbound state upon complex formation. In addition, the existing energy functions are not sufficiently accurate and lead to too much flexibility in side-chain optimization. An additional important aspect is that starting with a non-native position increases the possibility of choosing the wrong rotamer during the optimization.

These observations suggest that the docking side-chain optimization should use the unbound information. RosettaDock algorithm was improved by including the unbound conformers in the rotamer set.[31,32] We have decided to go further and restrict the flexibility only to the high-energy residues. Currently, only the clashing residues are movable, because ISCO scores include only the repulsive van der Waals. In the future, the flexibility can be extended, for example, to the buried polar residues. The restricted version of ISCO is called restricted interface side-chain optimization (RISCO). The unrestricted version is called full interface side-chain optimization (FISCO). The residue flexibility restriction comes close to not influencing the ISCO performance as seen in Figure 6 and Table IX, but decreases significantly the false-positives in the docking predictions (Optimal refinement resolution section). In addition, the general large-scale ISCO problem turns into few smaller independent subtasks for optimization of the clashing regions, which can be solved more efficiently.

Energy calculation for rotamers

The interface total energy is approximated by the pairwise and self-energies of the interface side-chains. The pairwise energy of two side-chains is the sum of their interatomic energies. The self side-chain energy accumulates its atomic energies with backbone and other non-side-chain atoms. Figure 3(a) illustrates the graph representation of the side-chain optimization problem. The nodes represent the rotamers and two rotamers are connected by an edge if they interact. Each edge is assigned a weight equal to the interaction energy between the corresponding rotamers. The weights of the self-edges are the self energies of the rotamers. The GMEC is achieved by choosing one node per residue to minimize the weight of the induced subgraph.

We use the following self and pair-wise energies:

$$E_{\mathrm{ISCO}}(i_r) = E_{\mathrm{repVdW}}(i_r) + KE_{\mathrm{rot}}(i_r) \tag{9}$$

$$E_{\mathrm{ISCO}}(i_r, j_s) = E_{\mathrm{repVdW}}(i_r, j_s) \tag{10}$$

where $E_{\mathrm{ISCO}}(i_r)$ is the self-energy of rotamer r built for the residue i, which includes the repulsive part of the van

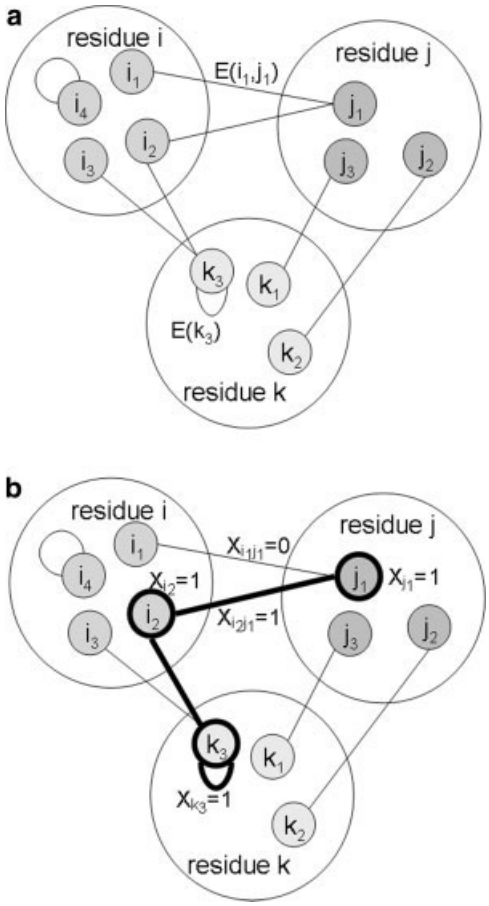

Figure 3

*(**a**) The graph representation of the side-chain optimization problem; (**b**) A possible assignment of the decision variables.*

der Waals interaction energy of the rotamer atoms with the static environment (backbone atoms and other fixed atoms). Note that interactions of rotamer atoms with the backbone atoms of the adjacent residues are excluded to avoid redundancy with E_{rot}. $E_{\text{ISCO}}(i_r,j_s)$ is the pair-wise energy between rotamer r of residue i and rotamer s of residue j, which consists of only the repulsive van der Waals energy between these rotamers' atoms. K is the weight of the E_{rot} term and is optimized to six. E_{rot} reflects not only the statistical preference of each rotameric state for the given backbone angles, but also the preference of the unbound conformation.

Other energy terms were found to be insufficiently accurate and caused more undesirable changes of side-chains in already near-bound configurations. SCWRL[16] and other successors use a similar approach for the simple energy function, with a linear repulsive energy term and log-probability term from the backbone-dependent rotamer library.

The rotamer building and energy calculation inside each molecule are performed in the preprocessing stage. During the optimization, only the intermolecular energies are calculated for each complex candidate.

ISCO as an integer linear programming problem

We aim to find the combination of rotamers with the lowest total energy. Actually, it is an optimization problem, which can be solved by the linear programming approach.[19–21] We use the following integer linear programming formulation defined in Ref. 21:

$$\text{Minimize} \sum_{(i,r)} E_{\text{ISCO}}(i_r) y_{i_r} + \sum_{(i,r,j,s):i<j,j\in N(i)} E_{\text{ISCO}}(i_r, j_s) x_{i_r j_s} \tag{11}$$

$$\text{subject to :} \quad \sum\nolimits_r y_{i_r} = 1, \quad i \in V \tag{12}$$

$$\sum\nolimits_r x_{i_r j_s} = y_{j_s}, \quad i \in V, j \in N(i) \tag{13}$$

$$x_{i_r j_s} \in \{0,1\}, \quad i,j \in V \tag{14}$$

$$y_{i_r} \in \{0,1\}, \quad i \in V \tag{15}$$

where V is the set of movable residues. The decision variable y_{i_r} equals one if and only if rotamer r of residue i is in GMEC. The decision variable $x_{i_r j_s}$ equals one if both r and s rotamers are in GMEC and their pair-wise energy is included in the global minimal energy. Distant residue pairs do not affect the optimization, and so the corresponding variables can be removed. Two residues are interacting if at least one of their $E_{\text{ISCO}}(i_r,j_s)$ is nonzero. Let $N(i)$ be the set of i's interacting residues. Constraints 12 guarantee that exactly one rotamer for each residue is selected. Constraints 13 ensure that only pair-wise energies between the selected rotamers are included in the global minimal energy. Figure 3(b) illustrates a possible assignment of the decision variables.

Kingsford *et al.*[21] proposed additional constraints for rotamer pairs with nonpositive energies, which are irrelevant here, since all the energies are positive.

Solving linear programming

The ISCO problem has been proven to be NP-hard[12] with an exponential time approximation.[13] The integer linear programming formulation given above is also NP-hard. By relaxation of constraints 14 and 15 to the form of $0 \le x \le 1$, the ISCO problem can be solved by a polynomial linear programming (LP) algorithm.[47,48] We use the *simplex* algorithm, developed by Dantzig,[49] which is exponential in the worst case, but highly effi-

DOI 10.1002/prot

cient for most practical purposes. It scans over the vertices of the feasible region and stops at an optimum vertex that does not have better neighbors.

If the LP algorithm finds an integral solution, then we achieve the optimal solution efficiently. Otherwise, the problem requires the use of an integer linear programming algorithm, which can be significantly slower. In practice, the ILP runs sufficiently fast because of the relatively small number of movable residues.

In 99.9% of the tested cases, the LP solution is integral. This is true for both FISCO and RISCO with the simple energy function described in "Energy calculation for rotamers section". However, incorporating additional energy terms involves more interrotamer interactions and hence hardens the problem. For example, incorporating an electrostatic component leads to an increased percentage of noninteger solutions as observed by Kingsford *et al.*[21]

We use the CPLEX 7.1 package[50] with AMPL[51] to solve the LP and ILP problems. The AMPL model is defined by Kingsford *et al.*[21]

Rigid-body optimization

The starting candidates for our algorithm are produced by a "soft surface" rigid docking algorithm. It treats the side-chain flexibility implicitly by the partners' surface softening, which allows small surface penetrations. Therefore, the binding partners' relative position is approximated and should be optimized simultaneously with the side-chains. Simultaneous optimization is computationally expensive and is usually replaced by two repeated successive stages of side-chain rearrangement and rigid-body optimization. Moreover, the starting position can include interface clashing regions, which cannot be resolved by side-chain optimization. In such a case, we can further optimize the binding by minimizing the interaction energy in the six-dimensional space of rigid movements (translation and rotation). This stage is performed by Monte Carlo sampling with local minimizations. A similar optimization technique is used in RosettaDock.[31]

The rigid-body optimization procedure consists of the following repeated steps:

1. Random perturbation of the ligand.
2. Local energy minimization.
3. An obtained position is accepted or rejected by the Metropolis criterion.[23] If the new position results in a lower energy, the move is unconditionally accepted. Otherwise, it is accepted with some probability. If the position is rejected, the ligand is returned to the previous position.
4. Goto 1.

This procedure is repeated 50 times and the position with the best energy is chosen.

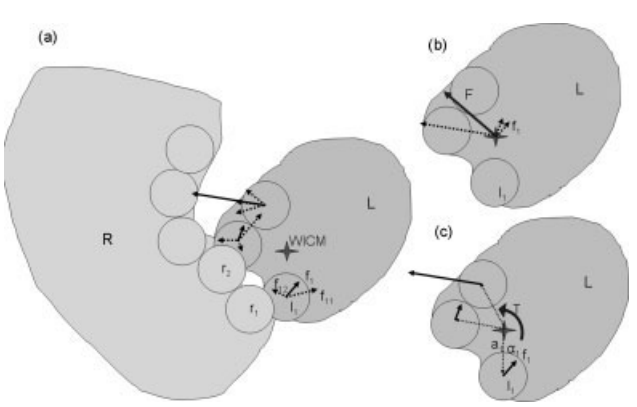

Figure 4

*Calculation of the ligand atomic forces (**a**), the total ligand force (**b**), and the total ligand torque (**c**).*

Local energy minimization

The binding energy function for the rigid-body minimization is approximated by

$$E_{\mathrm{RBM}} = KE_{\mathrm{s\_attrVdW}} + E_{\mathrm{s\_repVdW}} \tag{16}$$

where $E_{\mathrm{s\_attrVdW}}$ and $E_{\mathrm{s\_repVdW}}$ are softened van der Waals interactions, which are described in Binding score section. K is the relative weighting of the attractive van der Waals score in comparison to the repulsive van der Waals score. The choice of $K = 5$ yields the best performance results.

We use the BFGS quasi-Newton algorithm[52,53] to locate the nearest local energy minimum. The algorithm uses the energy gradient, which is actually the opposite to the force that the receptor applies to the ligand. The force applied to a ligand atom is calculated by the vector summation of the pair-wise forces between the atom and its neighbor atoms from the receptor. Figure 4(a) depicts schematically the atomic force vectors (the calculation appears in the "Supplementary Materials"). The total force, acting on the ligand, is the vector summation of its atomic forces. In addition, we need to calculate the "rotational force," called the torque. We define the center of the ligand rotation as its *weighted interface center of mass* as described in Binding score section. Figure 4(b,c) depicts schematically the force vector $\vec{F}$ and the torque τ, which are acting on the ligand (the calculation appears in the "Supplementary Materials").

Ranking

After optimizing the candidates by allowing side-chain flexibility and small rigid-body movements, we aim to separate the near-native candidates from the others by a ranking score.

Table I
The Average Times of FireDock

Size (No. of interface residues)	60.00
Prebuilding (s)	0.05
RISCO (s)	0.08
RBO (s)	4.28
All (s)	4.4

Pre-building is the time of intermolecular energy calculation between the interface rotamers plus the time of the LP formulation of the SCO problem.

First, we have tried to use the sum of the energy components described in the "Binding score" section. However, this score gave a poor ranking of the unbound refined candidates. The optimized near-native models are still less accurate than the bound structures and some energy components are more sensitive to these inaccuracies. For example, the van der Waals energy is softened to account for the inaccuracies, in contrast to the sensitive electrostatic energy function. Therefore, to achieve a good ranking score for unbound refined candidates, the scoring function should be optimized. Complexes of different types (e.g., enzyme-inhibitor, antibody-antigen) differ in their interface characteristics. We customize our scoring functions to express these differences.

Since the side-chain optimization algorithm involves only the van der Waals repulsion energy and favors the unbound state, $\Delta G^{L}_{\text{intr f\_intra}}$ and $\Delta G^{R}_{\text{intr f\_intra}}$ are approximated by only the repulsive van der Waals energy change. In addition, we can assume that the original intramolecular conformation does not include clashes. Therefore, only the intramolecular steric clashes in the complex interface and the internal energy term are considered in $\Delta G^{L}_{\text{iner f\_intra}}$ and $\Delta G^{R}_{\text{iner f\_intra}}$.

To optimize and test the scoring functions, we have divided the benchmark set into training and test sets. The cases used for the training are marked in Tables II and III. Actually, two training data sets were assembled,

Table II
The Docking Results of PatchDock and PatchDock + FireDock for the Enzyme-Inhibitor Cases Defined in Docking Benchmark Section

		PatchDock medium accuracy			PatchDock acceptable accuracy			FireDock medium accuracy			FireDock acceptable accuracy		
Complex	No. of candidates	Rank	Lrms	Irms	Rank	Lrms	Irms	Rank	Lrms	Irms	Rank	Lrms	Irms
Benchmark 1.0													
1ACB*	3592	58	2.76	0.68	58	2.76	0.68	1	3.86	0.96	1	3.86	0.96
1BRC	1194	202	6.55	1.53	17	15.85	3.87	1	6.74	1.6	1	6.74	1.6
1CHO*	1811	87	2.44	1.13	87	2.44	1.13	5	3.58	1.25	2	6.23	2.47
1CSE*	2481	304	6.91	1.91	304	6.91	1.91	2	5.77	1.71	2	5.77	1.71
1FSS*	2194	204	2.66	1.3	204	2.66	1.3	9	1.76	1	4	11.25	3.9
1TGS	1842	21	2.64	0.96	9	6.95	3.02	7	1.92	0.6	4	7.1	2.99
1UGH*	1447	93	4.01	2.31	42	5.53	2.52	1	4.74	2.41	1	4.74	2.41
2KAI	2364	91	4.66	1.5	14	10.82	3.94	2	4.23	1.4	2	4.23	1.4
2PTC*	1356	44	9.79	1.96	44	9.79	1.96	1	6.07	1.21	1	6.07	1.21
Semiunbound													
1STF	2318	5	1.33	0.35	5	1.33	0.35	1	1.61	0.42	1	1.61	0.42
1TAB	1013	71	3.31	1.31	71	3.31	1.31	3	3.84	1.28	3	3.84	1.28
2TEC	2729	26	1.32	0.4	26	1.32	0.4	1	0.61	0.26	1	0.61	0.26
4HTC	568	2	4.79	2.09	2	4.79	2.09	4	4.15	1.87	4	4.15	1.87
Benchmark 2.0													
1AVX	3397	278	9.26	1.99	104	8.7	4.68	2	3.9	0.8	1	9.21	2.9
1AY7	1926	658	4.97	1.47	183	8.16	3.36				21	5.7	2.41
1BVN*	2669				74	8.89	3.29	2641	1.86	0.76	6	7.66	3.46
1CGI	1511	204	4.45	2.13	177	9.73	3.72	10	4.13	2.01	10	4.13	2.01
1D6R	1399				101	9.56	3.41				8	13.9	3.36
1EAW	2589	67	4.66	1.87	67	4.66	1.87	2	3.95	1.56	2	3.95	1.56
1EZU*	2269				213	6.93	2.14	15	6.56	1.92	15	6.56	1.92
1F34	2507	45	5.09	1.92	14	5.67	2.42	9	5.38	1.89	8	8.79	5.08
1HIA	795				644	10.93	3.82				13	9.21	3.94
1MAH	2598	292	2.8	0.98	292	2.8	0.98	9	1.45	0.47	2	10.73	3.84
1PPE	674	222	5.2	1.91	3	9.99	4.4	1	3.64	1.21	1	3.64	1.21
1TMQ*	4642	3	4.65	1.84	3	4.65	1.84	14	1.76	0.71	14	1.76	0.71
1UDI	1921	8	4.21	2.06	8	4.21	2.06	5	3.58	1.62	5	3.58	1.62
2MTA*	3925	194	3.32	0.95	15	7.41	3.7	2	5.25	1.7	2	5.25	1.7
2SIC	3632	162	3.33	0.79	32	19.16	3.48	1	2.94	0.58	1	2.94	0.58
2SNI	1735				229	9.39	3.96				13	11.21	3.1
7CEI*	1152				31	14.4	3.09	11	4.19	1.73	11	4.19	1.73

We list the *rank* of the first medium/acceptable accuracy candidate and its *Lrms* and *Irms* (defined in Assessment Criteria Section). The cells are empty for the cases which do not include any medium/acceptable accuracy candidate. The cases, which were used in optimizing the scoring function, are marked with *.

DOI 10.1002/prot

Table III
The Docking Results of PatchDock and PatchDock + FireDock for the Antibody-Antigen Cases

Complex	No. of candidates	PatchDock medium accuracy			PatchDock acceptable accuracy			FireDock medium accuracy			FireDock acceptable accuracy		
		Rank	Lrms	Irms	Rank	Lrms	Irms	Rank	Lrms	Irms	Rank	Lrms	Irms
Benchmark 1.0 semiunbound													
1BQL*	1697	10	3.73	0.86	10	3.73	0.86	2	2.3	0.59	2	2.3	0.59
1EO8*	2718	1484	6.73	1.9	339	6.48	2.47	12	4.93	1.2	5	7.82	3.21
1FBI*	3575	123	5.24	1.72	15	7.05	4.27	11	3.97	0.9	11	3.97	0.9
1IAI	2123	1521	5.48	1.87	53	9.71	7.09				6	9.94	3.23
1JHL*	4145	27	6.01	1.76	27	6.01	1.76	10	1.22	0.53	10	1.22	0.53
1MEL	1059	2	1.63	0.5	2	1.63	0.5	3	4.89	1.76	1	11.92	3.83
1NMB	5941	2577	2.85	1.13	317	10.84	3.41	15	1.47	0.51	15	1.47	0.51
1QFU	2409	26	3.43	1.34	26	3.43	1.34	1	2.91	0.86	1	2.91	0.86
2VIR	3024	349	5.65	1.31	330	9.81	3.9	8	1.86	0.63	8	1.86	0.63
Benchmark 2.0 semiunbound													
1BJ1*	1682	228	13.98	1.78	228	13.98	1.78	11	8.39	1.12	11	8.39	1.12
1FSK*	1637	30	9.9	1.09	9	10.76	2.4	1	4.13	1.02	1	4.13	1.02
1IQD*	914	79	14.6	1.8	9	5.18	2.43				14	6.84	2.61
1K4C	1096	305	7.51	1.27	305	7.51	1.27	11	8.48	0.95	11	8.48	0.95
1NCA*	2616	70	6.08	1.7	70	6.08	1.7	10	3.81	0.96	10	3.81	0.96
1NSN	501				16	9.27	5.6				105	9.7	5.93
1QFW	1244	165	3.56	1.14	165	3.56	1.14	46	3.34	0.81	46	3.34	0.81
2JEL	707	15	16.31	1.9	11	28.71	3.02	2	17.75	1.98	2	17.75	1.98
2QFW*	942	22	3.29	0.91	22	3.29	0.91	15	4.73	1.08	15	4.73	1.08
Benchmark 1.0 + 2.0 unbound													
1AHW	1509	14	5.39	1.22	13	10.61	2.6	63	3.7	1.33	3	24.82	2.93
1BVK	3162	670	8.75	1.53	63	17.22	3.63	479	3.65	1.14	36	13.2	3.28
1DQJ	1201				51	8.56	4.74				24	21.54	3.98
1E6J	1061	127	9.95	1.15	127	9.95	1.15	73	12.75	1.95	11	16.84	3.45
1JPS	1658	24	9.12	1.19	24	9.12	1.19	583	6.8	1.02	28	18.75	2.26
1MLC	1832	52	11.71	1.19	52	11.71	1.19	12	9.32	1.03	12	9.32	1.03
1VFB	3325	335	6.87	1.76	38	16.22	3.5	1166	8.36	1.59	36	16.66	2.6
1WEJ	2220	275	6.78	1.16	228	25.53	2.68	10	1.56	0.67	10	1.56	0.67
2VIS	4626				506	10.09	2.67				360	20.22	3.85

enzyme-inhibitor and antibody-antigen sets. The weights of the scoring terms were optimized to achieve the best ranking for the training sets. We used support vector machine (SVM) training[54,55] and heuristic optimization to calibrate the weights. The final scoring functions for ranking are as follows:

$$E_{EI} = E_{s\_attrVdW} + 0.95E_{s\_repVdW} + 1.6E_{ACE} + 0.07E_{attEl} + 0.12E_{repEl} + 0.3E_{l\_repEl} + 1.32E_{HB+SS} + E_{pipi} + 0.8E_{catpi} + 0.5E_{aliph} + 1.55E_{insideness} \quad (17)$$

$$E_{AA} = 1.5E_{s\_attrVdW} + 0.6E_{s\_repVdW} + 1.6E_{ACE} + 0.21E_{attEl} + 0.21E_{repEl} + 0.46E_{l\_attrEl} + 0.69E_{l\_repEl} + 1.2E_{HB+SS} + E_{pipi} + 0.7E_{catpi} + 2.5E_{aliph} \quad (18)$$

where E_{EI} is the ranking function for the enzyme-inhibitor complexes and E_{AA} is for the antibody-antigen cases. The terms with zero or negative weights were excluded from the functions. E_{EI} and E_{AA} were optimized on the cases from the training set and then applied to all the benchmark cases. These weighted scoring functions do not correlate with the experimental binding free energy.

TIME EFFICIENCY

Table I summarizes the average times of the FireDock stages performed on one candidate. As can be seen, the

Table IV
Summary of the Prediction Success of PatchDock and PatchDock + FireDock

Type	PatchDock medium accuracy	PatchDock acceptable accuracy	FireDock medium accuracy	FireDock acceptable accuracy
Unbound Enz/Inh	2/26	7/26	21/26	25/26
Semiunbound Enz/Inh	2/4	2/4	4/4	4/4
All Enz/Inh	4/30	9/30	25/30	29/30
Unbound Ant/Ant	1/9	1/9	2/9	4/9
Semi-unbound Ant/Ant	3/18	6/18	14/18	16/18
All Ant/Ant	4/27	7/27	16/27	20/27

The prediction is successful if it includes at least one medium/acceptable accuracy structure within the 15 top-scoring candidates.

Table V
Results Following the Coarse and the Full Refinement Stages (Detailed in the Text) for the Enzyme-Inhibitor Cases

	PatchDock candidate		Following FireDock coarse refinement			Following full FireDock		
Complex	Lrms	Irms	Rank	Lrms	Irms	Rank	Lrms	Irms
Benchmark 1.0								
1ACB	2.76	0.68	1	3.47	0.9	1	2.67	1.09
1BRC	6.55	1.53	1	8.04	1.75	1	6.74	1.6
1CHO	2.44	1.13	5	2.78	1.2	5	1.97	0.88
1CSE	6.91	1.91	4	5.63	1.62	2	5.77	1.71
1FSS	2.66	1.3	1	2.22	1.23	9	1.76	1
1TGS	2.64	0.96	1	1.79	0.71	7	1.92	0.6
1UGH	4.01	2.31	20	4.75	1.73	1	4.74	2.41
2KAI	4.66	1.5	6	4.09	1.46	2	4.23	1.4
2PTC	4.09	1.55	1	6.01	1.32	1	2.86	0.58
Semiunbound								
1STF	1.33	0.35	1	3.44	0.78	1	1.61	0.42
1TAB	3.31	1.31	2	1.16	0.53	3	2.52	1
2TEC	1.32	0.4	1	0.93	0.36	1	0.61	0.26
4HTC	2.74	1.31	4	1.88	0.76	4	3.08	1.08
Benchmark 2.0								
1AVX	3.35	1.07	2	5.77	1.12	2	3.9	0.8
1AY7	8.16	3.36	21	4.13	1.84		5.7	2.41
1BVN	7.89	3.41		7.79	3.54		7.66	3.46
1CGI	4.45	2.13	15	4.12	1.99	10	4.13	2.01
1D6R	9.56	3.41		9.47	3.4		9.63	3.49
1EAW	3.59	1.66	3	3.48	1.57	2	3.32	1.58
1EZU	6.93	2.14		7.46	2	15	6.56	1.92
1F34	5.09	1.92	24	4.36	2.5	9	4.35	2.23
1HIA	10.19	4.17		9.73	4.09		9.21	3.94
1MAH	2.8	0.98	11	1.92	0.67	9	1.45	0.47
1PPE	3.34	1.36	2	3.12	1.17	1	3.64	1.21
1TMQ	1.49	0.82	2	2.24	0.82	14	1.76	0.71
1UDI	4.21	2.06	9	3.67	1.91	5	3.58	1.62
2MTA	3.33	1.79	5	5.11	1.64	2	5.25	1.7
2SIC	3.33	0.79	1	2.8	0.6	1	2.94	0.58
2SNI	10.71	2.95		9.51	2.63		11.21	3.1
7CEI	7.46	2.51	8	4.64	1.33	11	4.19	1.73
Average	*4.71*	*1.76*		*4.52*	*1.57*		*4.30*	*1.57*

Lrms and *Irms* show the best RMSD values within the top 25 candidates ranked by FireDock. The *PatchDock candidate* column shows the RMSD values of the initial candidates. The *rank* columns show the best rank of the medium accuracy candidate within the 25 first models (empty cells mean that no medium accuracy models exist in top 25).

rigid-minimization stage is the bottle-neck of the FireDock program. The PatchDock algorithm generates thousands of the docking candidates in a few minutes. The average running time of FireDock is ~4.4 seconds per candidate and around 2 hours per case (1600 candidates).

RESULTS

In this section, we show docking prediction results of the PatchDock+FireDock framework for benchmark cases and compare these with the performance of RosettaDock. In addition, we test the FireDock performance on decoys generated by another rigid-docking protocol.

Docking benchmark

The protein complexes and their unbound subunits from docking benchmarks 1.0 and 2.0[10,11] are used here to test the docking performance. The benchmarks divide the available test cases into different categories by their difficulty: "rigid-body" cases, medium, and high-difficulty cases. In the "rigid-body" cases, the backbone movement is very small, while in the difficult cases the backbone deformation during binding is significant. Since our method is applicable to cases without significant backbone movements, we use only the "rigid-body" cases. Benchmark 2.0[11] is the updated version of benchmark 1.0.[10] Many cases from benchmark 1.0 were updated and transferred from the "rigid-body" category to the more difficult categories. Therefore, we use the nonredundant set of "rigid-body" cases from both benchmarks.

In the current study, we have decided to focus on the enzyme-inhibitor and antibody-antigen cases. First, the PatchDock algorithm was applied to all the cases.

Table VI
Results Following the Coarse and the Full Refinement Stages for the Antibody-Antigen Cases

	PatchDock candidate		Following FireDock coarse refinement			Following full FireDock		
Complex	Lrms	Irms	Rank	Lrms	Irms	Rank	Lrms	Irms
Benchmark 1.0 semi-unbound								
1BQL	3.14	1.19	3	1.37	0.5	2	2.3	0.59
1EO8	6.48	2.47	18	4.75	0.98	12	4.93	1.2
1FBI	4.62	1.08	24	4.91	1.04	11	3.97	0.9
1IAI	8.48	2.34	8	6.59	1.88		9.94	3.23
1JHL	1.69	0.53	7	1.3	0.6	10	1.22	0.53
1MEL	1.63	0.5	19	3.9	1.12	3	4.17	1.06
1NMB	4.28	1.51	25	1.69	0.65	15	1.47	0.51
1QFU	3.43	1.34	1	2.28	0.9	1	2.91	0.86
2VIR	7.74	1.57	18	2.12	0.75	8	1.86	0.63
Benchmark 2.0 semi-unbound								
1BJ1	2.35	0.6	2	8.86	1.3	11	8.39	1.12
1FSK	5.42	2.11	1	2.86	0.97	1	4.13	1.02
1IQD	10.21	2.81		8.93	3.1		6.84	2.61
1K4C	7.51	1.27	5	4.22	0.67	11	7.29	1
1NCA	6.08	1.7	24	5.62	1.71	10	3.81	0.96
1NSN								
1QFW								
2JEL	4.14	1.38	10	12.93	1.58	2	5.24	1.1
2QFW	3.29	0.91	20	5.33	1.13	15	4.73	1.08
Benchmark 1.0 + 2.0 unbound								
1AHW	24.72	2.86		23.5	2.88		24.82	2.93
1BVK								
1DQJ	19.57	4		21.54	3.98		21.89	4.01
1E6J	17.11	3.61		16.94	3.37		16.84	3.45
1JPS								
1MLC	11.71	1.19	3	8.96	1.66	12	9.32	1.03
1VFB								
1WEJ	6.78	1.16	8	4.95	0.87	10	1.56	0.67
2VIS								
Average	*7.64*	*1.72*		*7.31*	*1.51*		*7.03*	*1.45*

The empty lines mean that no acceptable accuracy models exist in top 25.

DOI 10.1002/prot

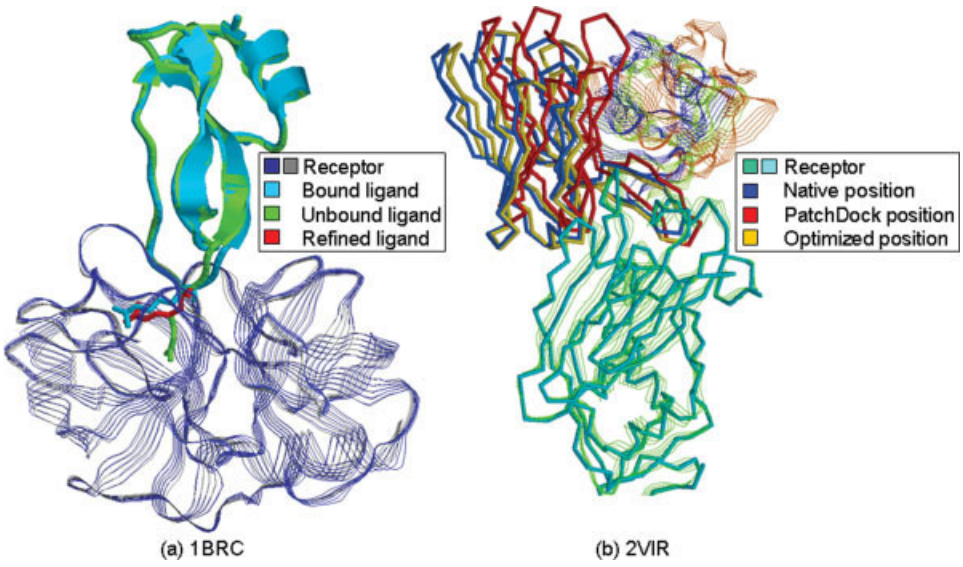

Figure 5

*(**a**) Example for correct prediction of a hot-spot residue (Arg15) by the FireDock ISCO. (**b**) Example for approaching a near-native structure. The position of the ligand is optimized from 7.7 Å to 1.8 Å RMSD from the "native" orientation.*

The cases for which the PatchDock candidate list does not include any structure of acceptable accuracy are excluded from the analysis. Thus, the remaining list includes 30 enzyme-inhibitor cases (26 unbound, 4 semi-unbound) and 27 antibody-antigen cases (9 unbound, 18 semi-unbound).

Assessment criteria

To assess the solution accuracy, we use two measures: the ligand RMSD (Lrms) and the interface RMSD (Irms). The ligand RMSD is calculated by the superposition of the bound and the unbound receptors and calculating the pair-wise root-mean-squared distance of the corresponding ligand C_α atoms. The interface RMSD is calculated over C_α atoms of the interface residues after finding the best superposition of the bound and the unbound interfaces. For the Irms calculation, an interface residue is defined as a residue with at least one atom within 10 Å of any atom of the docking partner.

We define here two metrics of accuracy: medium accuracy (with Lrms $\leq$ 5 Å or Irms $\leq$ 2 Å) and acceptable accuracy (with Lrms $\leq$ 10 Å or Irms $\leq$ 4 Å). These RMSD metrics with additional contact fraction measures are used in CAPRI for the assessment of submitted models.[56]

Docking prediction results

The docking prediction ability is measured by its ranking of near-native structures, which are defined as structures with medium or acceptable accuracy. Tables II and III show docking prediction results of the PatchDock + FireDock framework for the enzyme-inhibitor and antibody-antigen cases. Table IV summarizes the docking prediction success of PatchDock and PatchDock + FireDock. FireDock succeeds in ranking a medium accuracy structure within the top 15 predictions for 25 of the 30 enzyme-inhibitor test cases, and for 14 of the 18 semi-unbound antibody-antigen complexes. The table shows the significant improvement achieved by the FireDock refinement and re-ranking over the rigid PatchDock ranking.

Tables V and VI show the gradual RMSD improvement achieved by the coarse and full refinement when compared with PatchDock. The average ligand RMSD improvement is 0.4 Å for the enzyme-inhibitor cases, and 0.6 Å for the antibody-antigen complexes. The RMSD values decrease after the refinement stages for 20 of the 30 enzyme-inhibitor cases, and for 14 of the 21 antibody-antigen complexes.

Figure 5 illustrates the productive changes following the ISCO and RBO stages.

FireDock is a stand-alone refinement method

To assess whether FireDock can be applied to a rigid-docking protocol other than PatchDock, i.e., its independence, we tested its performance on ZDOCK 2.3. Table VII shows the performance of the FireDock method when applied to the ZDOCK 2.3 decoys.[57] The results

Table VII
FireDock Performance on the ZDOCK 2.3 Decoys

	ZDOCK 2.3[a]		ZDOCK 2.3[b]		FireDock[c]	
Complex	Rank	Irms	Rank	Irms	Rank	Irms
Enzyme-inhibitor						
1ACB	18	1.33	18	1.19	1	0.69
1AVW	3	2.07	3	2.08	1	2.1
1BRC	24	2.32	13	2.5	1	0.8
1BRS	65	2.13	65	2.02	102	2.23
1CGI	4	2.41	4	1.89	1	2.01
1CHO	3	1.57	3	1.62	2	1.53
1CSE	198	2.2	198	2.26	1	0.97
1DFJ	1	2.48	1	2.18	17	1.34
1FSS	50	1.52	50	1.41	2	1.47
1MAH	24	1.29	24	1.23	3	0.89
1TGS	3	2.22	3	1.58	1	0.64
1UGH	8	2.25	8	2.16	3	1.21
2KAI	388	1.61	388	1.5	1	0.9
2PTC	193	1.83	193	1.87	3	0.58
2SIC	11	2.37	11	2.33	1	0.71
2SNI	1262	2.22	1262	2.05	2	2.01
1PPE*	1	0.9	1	0.82	1	0.38
1STF*	1	0.88	1	0.85	1	0.56
1TAB*	79	1.21	79	1.17	18	0.97
1UDI*	5	1.19	5	1.13	7	2.16
2TEC*	1	0.76	1	0.74	1	0.4
4HTC*	3	2.46	3	2.38	1	1.24
Statistics	13/22	1.78	14/22	1.68	21/22	1.17
Antibody-antigen						
1AHW	7	1.82	7	1.72	17	2.16
1BVK	821	2.34	821	1.94	651	1.61
1DQJ	9249	2.37				
1MLC	128	1.65	128	1.97	26	2.43
1WEJ	183	1.04	183	1.01	26	0.65
1BQL*	13	1.07	13	0.85	1	0.48
1EO8*	1497	0.96	1497	0.91	475	1.27
1FBI*	642	2.03	642	1.91	225	2.17
1IAI*	997	1.7	340	2.45	157	1.04
1JHL*	333	1.37	333	1.34	46	0.64
1MEL*	3	1.19	3	0.99	1	1.36
1NCA*	1	1.06	1	1.04	30	0.56
1NMB*	135	0.98	135	0.97	163	0.62
1QFU*	388	1.14	388	1.06	48	0.96
2JEL*	233	1.46	233	1.42	9	1.97
2VIR*	1101	1.03	1101	0.89	204	0.65
Statistics	4/16	1.45	4/16	1.36	4/16	1.24

We list the rank of the first hit (a model with *Irms* $\leq$ 2.5 Å) and its *Irms*. "Statistics" rows show the success ratio (at least one hit within the 20 top-ranked models) and the average *Irms*.
[a]ZDOCK 2.3 performance with 6° rotational sampling interval as published in.[57]
[b]Rank and Irms of ZDOCK 2.3 decoys according to our calculations.
[c]Rank and Irms of ZDOCK 2.3 decoys refined and re-ranked by FireDock.

show that for the enzyme-inhibitor cases, the ranking is improved significantly. However, no improvement is observed for the antibody-antigen cases. In contrast to the ZDOCK 2.3 algorithm, the PatchDock method focuses on the complementarity-determining regions (CDRs) of the antibodies. This leads to a decreased number of false-positive candidates. In most of the cases, the top-ranked near-native structures have similar scoring values for the candidates generated by PatchDock and ZDOCK 2.3.

Comparison of RosettaDock and PatchDock+FireDock performance

Table VIII compares the docking prediction results of RosettaDock and PatchDock+FireDock. The RosettaDock results presented here were obtained by the improved version of RosettaDock.[32] The top 200 lowest-energy

Table VIII
Comparison of the RosettaDock and PatchDock + FireDock Performance on the Benchmark 1.0 Cases

	RosettaDock		PatchDock + FireDock	
Complex	Rank	Lrms	Rank	Lrms
1A0O	>25	6.3	>25	
1ACB	6	3.58	1	2.67
1AHW	23	7.65	>25	
1ATN	1	1.27	X	
1AVW	>25	6.48	4	4.15
1BQL	2	1.33	2	2.3
1BRC	5	4.3	>25	
1BRS	6	3.25	15	
1CGI	>25		2	3.42
1CHO	1	2.37	5	1.97
1CSE	>25	8.12	>25	5.77
1DFJ	>25		>25	
1EO8	>25		12	7.82
1FBI	>25	5.02	11	
1FSS	>25		9	1.76
1GLA	X		5	3.39
1IGC	>25		X	
1IAI	X		>25	
1IGC	X		>25	1.22
1JHL	X		10	4.02
1MAH	>25		1	4.17
1MEL	1	3.74	3	
1NCA	1	0.54	10	3.81
1NMB	6	1.64	15	
1PPE	1	2.99	1	1.62
1QFU	2	3.29	1	2.91
1SPB	1	0.92	1	1.46
1STF	1	2.3	1	1.61
1TAB	3	2.76	3	3.84
1TGS	>25	5.61	7	1.92
1UDI	2	2.52	>25	
1UGH	1	1.76	1	4.74
1WEJ	X		10	1.56
1WQ1	>25		1	3.41
2BTF	9	2.72	3	0.96
2JEL	2	3.66	>25	5.24
2KAI	7	3.58	2	4.23
2PTC	1	3.19	3	2.86
2SIC	1	3.11	>25	6.18
2SNI	24	8.89	>25	8.13
2TEC	5	1.84	1	0.61
2VIR	14		8	1.86
4HTC	>25		4	4.15
Statistics	25/43	3.61	30/43	3.35

We list the rank of the first hit (a model with *Lrms* $\leq$ 5 Å) and the lowest *Lrms* value among the top 10 models (only *Lrms* $\leq$ 10 Å are presented). "X" appears for those cases where RosettaDock global search was not attempted because of unsuccessful local search near the solution. "X" in PatchDock + FireDock means that PatchDock candidates are too far from the native solution. The "Statistics" row gives the success ratio (at least one hit within 25 top-ranked models) and the average *Lrms* for each method.

DOI 10.1002/prot

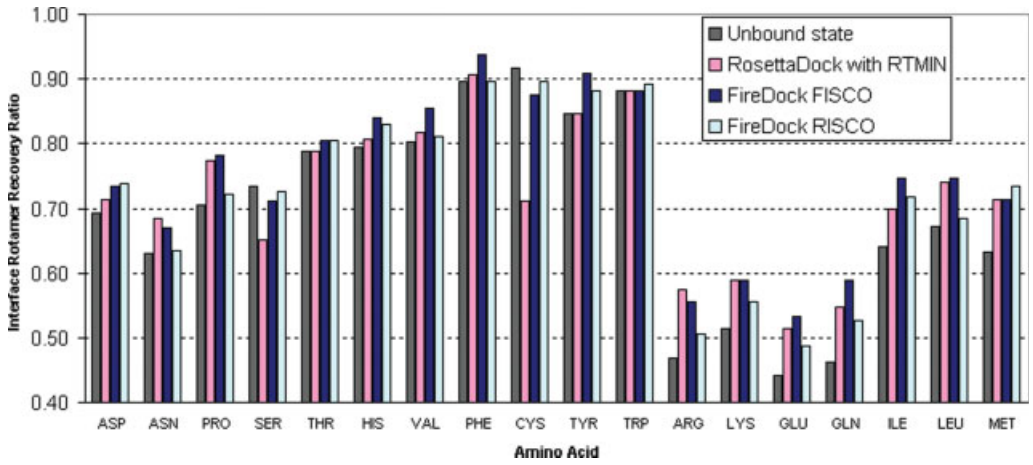

Figure 6

SCO performance of RosettaDock and FireDock relative to unbound structures for 57 complex interfaces. The performance is measured by the number of the near-native residues, which have χ_1 and χ_2 within 40 degrees from the native angles. For each residue, the first bar from the left is for the unbound structures, the second is for the RosettaDock refined structures. The third and the fourth bars are for FireDock FISCO and RISCO, respectively.

decoys were clustered, and the clusters were sorted according to their sizes. Representatives of the clusters are considered in the evaluation. The results were obtained from Ora Schueler-Furman and Chu Wang (personal communication, January 2006). The Table shows the rank of the first model with Lrms $\leq$ 5 Å and the lowest Lrms value among the top 10 models. RosettaDock succeeds in ranking a near-native structure within the top 25 predictions for 25 of the 43 cases, while FireDock succeeds in 30 cases of 43. The table also indicates the average Lrms for each method.

Blind predictions

CAPRI (http://capri.ebi.ac.uk) is the first community wide experiment on the comparative evaluation of protein–protein docking methods by blind prediction. Recently, a new category of "scorers" (http://www.scmbb.ulb.ac.be/capri/) was opened in CAPRI. This category aims to assess the ability of different docking groups to refine and rerank the given candidates. Each participating group submits 100 models to the common basket of models, which assemble the candidates set. We used the given candidates for determining the starting positions of the ligand near the receptor and the unbound subunits were refined from these starting positions. For the two last targets 25 and 26, FireDock achieved medium-accuracy solutions (measured by CAPRI criteria) among the 10 submitted models.

ANALYSIS

In this section, we attempt to evaluate the performance and the contribution of each FireDock stage. In addition, we estimate the optimal refinement resolution for successful ranking. Finally, we analyze the unsuccessful cases and assess the contribution of scoring terms to prediction.

Side-chain optimization

Most of the side-chain prediction methods have originated in folding, where no unbound structures are available. Therefore, the commonly used performance measurement is based on recovering near-native side-chain conformations for a given backbone without any information about the unbound side-chains. In docking, since unbound residues conformers are available, they can be used as a reference for the performance analysis.

To evaluate the ISCO performance without accounting for backbone flexibility, we start with the native complex

Table IX

The Percentage of the Unbound-Near-Bound Residues and the Near-Natives for the Refined Structures by RosettaDock and FireDock

	% of near-native ($\chi_{1+2} < 40$)		% of near-native (distance $<$ 1.5 Å)	
Structure	With Cys	Without Cys	With Cys	Without Cys
Unbound state	73.44	72.82	71.52	70.70
RosettaDock with RTMIN	75.25	75.39	72.06	71.98
FireDock FISCO	77.61	77.27	74.88	74.42
FireDock RISCO	75.59	75.11	73.47	72.78

We list the results for the two definitions of the "near-native residue", which are described in the text. *Without Cys* columns show the percentage of near-native residues without counting Cys residues because of its high-preference to be fixed.

structures and replace the bound residue conformers by their appropriate unbound conformers. For this purpose, we use the mapping between bound and unbound residues. The mapping was obtained by a sequence alignment of the bound and the unbound structures. Some positions in the alignment were unmatched, i.e., these are different residue types in the bound and the unbound structures (substitutions or deletions). These unmatched residues were replaced by their highest probability rotamers. Therefore, the starting structures do not include any side-chain in its bound conformation. In the semi-unbound cases, the bound proteins are not considered. The test set includes ~3000 residues, which were derived from 57 native complexes from all categories of benchmark 1.0[10] (including difficult test cases). It is important to emphasize that this special test set was used only for the ISCO performance evaluation. All other results were evaluated on unbound structures.

To assess the method performance, we define a residue conformation to be *near-native* if its χ_1 and χ_2 angles are within 40° of their native dihedral angles.[16,32] The performance was measured by *residue recovery* for each residue type, which is defined as the ratio between the number of near-native residues and all residues of a certain type. Symmetric terminal groups of Asp, Phe, Tyr, and possibly flipped Asn and His were counted as near-native for the 180°-flipped conformations too. The near-native unbound conformations are called the *unbound-near-bound* residues.

Figure 6 and Table IX show the performance of the side-chain optimization by FireDock and RosettaDock[32] relative to the unbound structures. Both FISCO and RISCO by FireDock are presented. Recall that FISCO is a full interface side-chain optimization, while RISCO rearranges only the clashing residues. Both FireDock versions use a simplified energy function, which includes only the repulsive van der Waals and rotamer probability scores. RosettaDock uses a more sophisticated energy function with solvation, electrostatics, and hydrogen bonding terms. However, FireDock yields a better recovery rate for most of the amino acids. Only Asn and Arg benefit from the more complex energy function. These amino acids are usually solvent-exposed and flexible during binding.

Table IX shows the percentage of the unbound-near-bound residues and the near-natives for the refined structures measured by two metrics. The second metric defines the residue as near-native, if the distances between its heavy atoms to the corresponding native atoms are under 1.5 Å. The results show that ~73% of the interface residues remain in the near-unbound state for the 57 analyzed complexes. However, we should note that most of the analyzed complexes are "rigid-body" cases (Docking benchmark section), which do not undergo substantial backbone changes during binding. Therefore, this observation is relevant for side-chain refinement methods, but should be further investigated for general cases. The results also show that the percentage of the near-native residues does not increase significantly after side-chain optimization. RosettaDock SCO and FireDock FISCO add only 2 and 4% to the recovery rate, respectively. It is important to note that this is an upper-bound since we use optimal unrealistic conditions: bound backbones and native receptor-ligand orientations. In this situation, FISCO yields better recovery rate than RISCO. In the Optimal refinement resolution section, we will analyze the influence of the flexibility restriction on the blind docking predictions.

We have attempted to use the RosettaDock energy function in the framework of integer linear programming and obtained worse results. We have observed that in many cases, SCO changes the conformations of unbound-near-bound residues. The number of such undesirable changes increases when the more sophisticated function is used. The reason is probably that the existing energy functions are not sufficiently accurate and lead to over-flexibility in the side-chain optimization. Starting with non-native positions and unbound backbones will lead to even more undesirable changes in the residues' conformations and probably to the lower recovery rate. Nevertheless, side-chain optimization is essential for successful docking predictions, as we will demonstrate in the following section.

Contribution of ISCO and RBO

In this section, we analyze the contribution of the rigid-body optimization (RBO) and the ISCO stages to

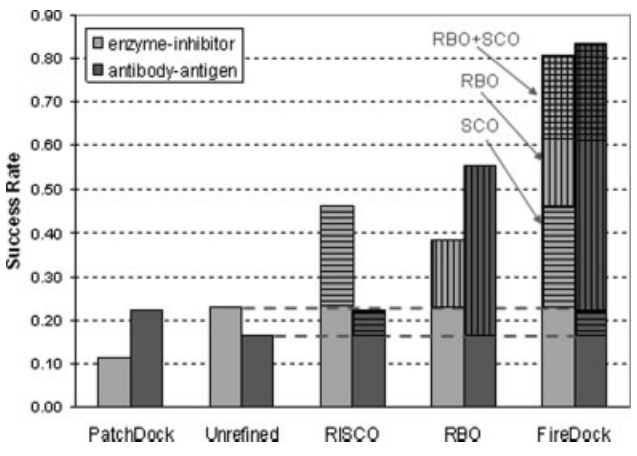

Figure 7

Prediction success rate of PatchDock and each FireDock stage. Unrefined *refers to the FireDock ranking without refinement. The areas covered by horizontal and vertical lines represent the SCO and RBO contributions respectively. The added value of their combination is illustrated by a grid pattern. The test set includes the 26 unbound enzyme-inhibitor cases and the 18 semi-unbound antibody-antigen cases. The prediction is successful if it includes at least one medium accuracy structure within the 25 top-ranked candidates.*

DOI 10.1002/prot

the docking prediction success. Figure 7 shows the prediction success rate of PatchDock and each FireDock stage. The presented variations of FireDock are determined by the parameters described in Table XII from the "Supplementary Materials." First, the figure gives the success rate obtained by PatchDock ranking and FireDock ranking without refinement. Next, it presents the prediction results for the SCO and RBO stages performed separately. The figure also assists in estimating the relative contribution of each stage to the complete protocol. The areas covered by horizontal and vertical lines represent the SCO and RBO contributions, respectively. The added value of their combination is illustrated by a grid pattern.

It is interesting to notice that the SCO stage is more productive for the enzyme-inhibitor cases. However, the improvement achieved by the RBO is significantly larger for the antibody-antigen cases. The reason is the different interface structure. For cases with more concave interfaces, as in the enzyme-inhibitor complexes, the rigid docking stage succeeds in generating more accurate candidates. Therefore, the SCO yields better results for these cases. In contrast, the antibody-antigen interfaces are relatively flat, and in these cases, SCO is less successful and RBO contributes considerably more. The added value of SCO and RBO combination is very significant in all the cases. This means that with the improved side-chains the RBO stage works better.

The FireDock "coarse refinement" consists of one iteration of RISCO and RBO. It is possible to perform more such iterations. However, we did not see apparent improvement when additional iterations were performed.

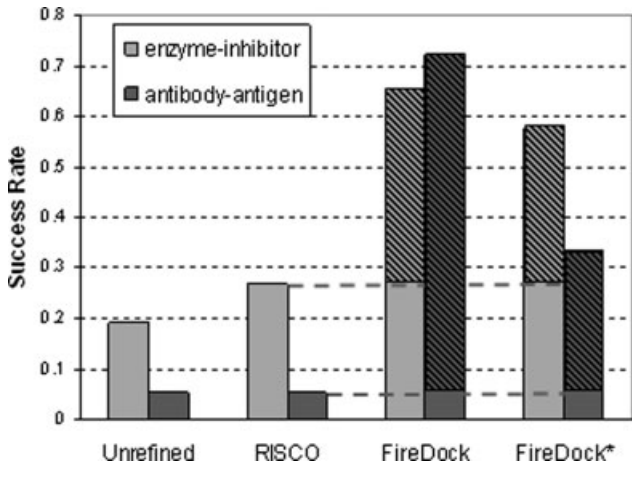

Figure 8

Prediction success rate of FireDock by ranking without the repulsive van der Waals term. The version, marked by asterisk, also excludes the attractive van der Waals term. The test set and the criterion for success are as defined in Figure 7.

Optimal refinement resolution

The improvement achieved by ISCO, compared to the unbound structure, is not significant as described in Side-chain optimization section. Moreover, starting with a non-native position increases the possibility of choosing a wrong rotamer in ISCO, which can lead to a wrong direction in RBO. Allowing full side-chain flexibility causes excessive conformational changes. These changes may distort preformed structural signatures, which are important for binding recognition. To reduce this effect, we restrict the side-chain flexibility. This restriction is also expected to decrease the number of false-positive solutions. Based on these observations, we have decided to test the docking prediction success for different levels of side-chain flexibility. Two factors control the side-chain flexibility are selection of the movable residues (described in Selection of movable residues section) and the size of the rotamer set (Binding model section).

The atomic radii smoothing in the rigid-body optimization and scoring is important because of the possible small backbone movements during binding and the inaccuracies in the side-chain conformations. Clearly, when we allow more flexibility in side-chains, atomic radii should be smoothed less. Side-chain flexibility and atomic radii scaling are gradually increased and prediction results are analyzed. Figure 9 from the "Supplementary Materials" depicts prediction success rate of FireDock with increasing resolution. The best prediction performance is achieved with minimal flexibility in side-chains and radii scaling = 0.8 during RBO and scoring. This analysis gives rise to the "coarse refinement" stage in FireDock.

Rigid-body optimization

The elimination of steric clashes after the refinement stage leads to the ranking improvement. Figure 8 illustrates that the ranking after the rigid-body optimization is drastically improved even without accounting for the repulsive van der Waals term. The RBO stage minimizes the energy function, which consists of the repulsive and the attractive van der Waals terms only. The figure shows significant improvement without both these terms. This observation suggests that the rigid-body minimization of the van der Waals function is coupled with other energy terms. In the future, we intend to explore this phenomenon and try to understand whether the reason is computational or biological dependence.

Analysis of failures

The obvious prerequisite to succeed in FireDock reranking is to include at least one near-native structure in the initial set of input candidates. The second prerequisite is that the input molecules do not undergo significant backbone conformational changes during the bind-

ing. However, in some cases, FireDock does not succeed in singling out the correct solutions even when both conditions are satisfied. Here, we attempt to analyze these cases.

The results for the enzyme-inhibitor cases are successful and show that all the cases are predicted with acceptable-accuracy within the top 21 solutions. However, the medium-accuracy solutions are not always top-ranked. In the antibody-antigen complexes, the docking performance is good for the semi-unbound cases, but is dissatisfying for the unbound cases. We examine the ranking of "medium accuracy" solutions, described in Tables II and III.

Four enzyme-inhibitor cases (1AY7, 1D6R, 1HIA, 2SNI) and five antibody-antigen cases (1IAI, 1IQD, 1NSN, 1DQJ, 2VIS) do not include any medium-accuracy solution. In most of these cases, the starting PatchDock candidates also do not include medium-accuracy solutions. Only in the 1AY7, 1IAI, and 1IQD cases, PatchDock candidates include medium-accuracy ones; however, FireDock chooses other lower-energy solutions, which are less accurate. In the 1BVN case, the medium-accuracy candidate is poorly ranked because the RBO stage does not succeed in resolving the clashes. If we exclude the repulsive van der Waals term from the scoring function, the candidate is ranked 16.

In general, medium-accuracy solutions in the unbound antibody-antigen targets are poorly ranked. The antibody-antigen pairs have flat interfaces, compared with the concave enzyme-inhibitor interfaces. The flat interfaces are harder for prediction because of weaker backbone constraints, which lead to a large number of false-positive structures. The results for the semi-unbound antibody-antigen cases are successful because the conformation of one of the molecules is fixed and defines constraints, which are induced by its side-chains.

Table X
Contribution of the Scoring Terms to the Prediction Success

Reduction in success rate (%)	Scoring terms
Enzyme-inhibitor	
36	s_attrVdW
32	ACE
16	s_repVdW
8	Insideness, attrEl, repEl
4	HB + SS, catpi, aliph, l_repEl
Antibody-antigen	
35	s_attrVdW
29	ACE
24	attrEl, repEl, pipi
18	l_attrEl, l_repEl, aliph
12	s_repVdW, HB + SS

The table shows a reduction in the success rate when the listed terms are excluded from the total score. Each term is excluded alone (not in combination). For example, the *attrEl* exclusion results in 8% reduction, the *repEl* term has a similar effect.

Table XI
Contribution of the Scoring Terms to the Prediction Success

Success rate (%)	Scoring function
Enzyme-inhibitor	
100	E_{El}
25	s_attrVdW
64	ACE
64	1.6ACE + s_attrVdW
64	1.6ACE + s_attrVdW + 0.95s_repVdW
64	1.6ACE + 0.07attrEl + 0.12repEl
72	1.6ACE + s_attrVdW + 0.07attrEl + 0.12repEl
76	1.6ACE + 1.55insideness
80	1.6ACE + s_attrVdW + 0.95s_repVdW + 0.07attrEl + 0.12repEl

Each entry shows the success rate when using the listed scoring function.

Tables XIII and XIV from the "Supplementary Materials" show the ranking of the artificially inserted "native" orientations. To assess the scoring function ability in discriminating the correct solutions, we have checked the ranking of the bound structures among the other candidates. For most enzyme-inhibitor pairs, the bound structures are top-ranked. However, the ranking of the refined *unbound* subunits in the "native" orientation is less successful. The major reason for that are the changes in the backbone configuration. In the antibody-antigen pairs, the ranking of the bound structures is low for a larger number of cases: 3 of 18 semi-unbound targets (1JHL, 1NMB, 2VIR) and 5 of 9 unbound targets (1BVK, 1JPS, 1VFB, 1WEJ, 2VIS). In these cases, the scoring function is not specific enough to eliminate the wrong candidates. It is interesting to analyze the data from the tables for the unsuccessful cases described earlier. The reasons for the prediction failure can be roughly divided into two categories: insufficient sampling and a nonadequate energy function. The first problem appears only for few cases (1BVN, 1IAI, and 1IQD), where the starting bound positions are top-ranked, despite the blind low-ranking. For the rest of the unsuccessful cases, the ranking of the bound configurations is also low. This shows that the energy function for these cases is not adequate.

Contribution of scoring terms to prediction

Tables X and XI summarize the contribution of the scoring terms to the prediction success. Table X shows the reduction in the success rate when the listed scoring terms are excluded from the total score. Omitting the softened attractive van der Waals and ACE terms causes the most significant reduction in the success rate. The repulsive softened van der Waals is important for the elimination of the wrong candidates for the enzyme-inhibitor cases. However, it is less important for the antibody-antigen cases, which are characterized by a flat interface with weaker backbone constraints.

DOI 10.1002/prot

Table XI shows the success rate by using partial scoring functions for the enzyme-inhibitor cases. The softened attractive van der Waals alone yields poor discrimination, while the ACE term alone demonstrates high success rate. Moreover, addition of the van der Waals terms to the ACE does not improve the performance. The improvement appears only when the electrostatic terms are added. It is interesting to notice that the "insideness" measure causes a significant increase in the success rate, however its exclusion has less influence. This means that this term serves as a backup for other scoring terms in the enzyme-inhibitor cases and can be used for the fast simplified evaluation.

CONCLUSIONS AND FUTURE WORK

We have presented an efficient method FireDock for refinement and rescoring of rigid-body docking predictions. The heart of the method is the "coarse refinement," which consists of the restricted side-chain optimization followed by soft rigid-body minimization. The following observations lead to the "coarse refinement" effectiveness:

- Most of the interface residues, which are important to the binding recognition, do not change significantly their conformation upon complexation.[6–8] Our results agree with this observation and show that for the 57 analyzed complexes, ~73% of the interface residues remain in the near-unbound state.
- If full flexibility is permitted in the interface side-chains, almost every candidate can be well-packed. This results in low discrimination of correct solutions by the van der Waals score terms.
- The existing energy functions are not accurate enough and lead to excessive flexibility in the interface side-chain optimization. The number of the undesirable changes is reduced by restricting side-chain flexibility and by using a simplified energy function with repulsive van der Waals and rotamer frequency terms. The restriction helps in preserving more near-native conformations and to achieve higher accuracy in interface contacts, which are expressed in the binding score.
- The soft rigid-body minimization compensates for the possible backbone fluctuations and the rotamer discretization.

The experimental results justify these observations and show that "coarse refinement" helps in eliminating the false-positive candidates and promotes the correct solutions.

The method was tested on 30 enzyme-inhibitor and 27 antibody-antigen cases, derived from the docking benchmarks 1 and 2.[10,11] FireDock succeeds in ranking a near-native structure within the top 15 predictions for 83% of the 30 enzyme-inhibitor test cases, and for 78% of the 18 semi-unbound antibody-antigen complexes. The prediction is fully automated, without using any biological information specific to the predicted complexes. The refinement improves the rigid docking of PatchDock as expected. In addition, we used FireDock to refine and rerank the candidates from the "scorers" category of CAPRI. Medium-accuracy solutions were ranked in the top 10 submitted models for targets 25 and 26. The FireDock method was also tested on decoys generated by ZDOCK 2.3 and was effective for the enzyme-inhibitor cases.

The prediction results described here show poor results for unbound antibody-antigen complexes. We have checked the ranking of the artificially inserted bound structures among the other candidates. In the antibody-antigen pairs, the ranking of the bound structures is low for many cases. This means that the scoring function is not always adequate in eliminating the wrong candidates. In addition, we have analyzed the contribution of the scoring terms to the prediction success. The most powerful predictor is ACE. However, additional energy components, such as attractive van der Waals and electrostatic terms, are necessary to achieve better performance.

One of our goals was to achieve fast side-chain flexible refinement in order to incorporate it as a building block into future general refinement algorithms. To our knowledge, FireDock is significantly less time consuming than other refinement programs. Its average running time is 4.4 s/candidate and around 2 h/case (1600 candidates). The time was measured on a Pentium© 4 CPU 3.2 GHz with 1 GB RAM. The results show that this time efficiency does not affect the method performance. For example, the cascade of programs PatchDock and FireDock is significantly faster and performs slightly better than RosettaDock.

Recent works point out that the correct solution is usually located in a spatial cluster of low-energy configurations.[31,33,35,58] This happens because the ligand reaches the binding site of the receptor from the broad neighborhood region. The ranking we perform here is based only on the energy score. A check of the binding funnels in the ranking stage may improve the method performance. In addition, a potential performance improvement will be explored through repeating the optimization stages with gradually increasing resolution.

Our results show that rigid-body refinement is essential for docking prediction and is also a bottle-neck phase in the FireDock running time. The rigid-body refinement method described here can be improved in its performance and efficiency. For example, the transformations derived from the atomic force vectors can be clustered. The high-populated clusters define the preferable directions for the rigid-body minimization. Inconsistent transformations are filtered out in this way and do not divert the ligand move from the right direction.

The linear programming approach can be used more generally in optimization of flexible parts of the interface. Protein backbone and small molecules can be also refined by similar approach by using their different conformers. Symmetric refinement can be implemented by additional constraints in linear programming formulation.

ACKNOWLEDGMENTS

We thank Dina Schneidman-Duhovny and Dr. Ora Furman-Schueler for fruitful conversations and data. We also thank the members of our Structural Bioinformatics Group (BioInfo3D) for helpful discussions. The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organization imply endorsement by the US Government.

REFERENCES

1. Katchalski-Katzin 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.
2. Lamdan Y, Wolfson HJ. Geometric hashing: A general and efficient model-based recognition scheme. In: Proceedings of the IEEE international conference on computer vision, Tampa, Florida, December 1998. pp 238–249.
3. Fischer D, Lin SL, Wolfson HJ, Nussinov R. A geometry-based suite of molecular docking processes. J Mol Biol 1995;248:459–477.
4. 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. Berlin: Springer-Verlag; 2002. pp 185–200.
5. Ritchie DW, Kemp GJL. Protein docking using spherical polar fourier correlations. Proteins 2000;39:178–194.
6. Rajamani D, Thiel S, Vajda S, Camacho CJ. Anchor residues in protein-protein interactions. Proc Natl Acad Sci USA 2004;101:11287–11292.
7. Li X, Keskin O, Ma B, Nussinov R, Liang J. Protein-protein interactions: hot spots and structurally conserved residues often locate in complemented pockets that preorganized in the unbound states: implications for docking. J Mol Biol 2004;344:781–795.
8. Smith GR, Sternberg MJE, Bates PA. The relationship between the flexibility of proteins and their conformational states on forming protein-protein complexes with application to protein-protein docking. J Mol Biol 2005;347:1077–1101.
9. Zhang C, Vasmatzis G, Cornette JL, DeLisi C. Determination of atomic desolvation energies from the structures of crystallized proteins. J Mol Biol 1997;267:707–726.
10. Chen R, Mintseris J, Janin J, Weng Z. A protein-protein docking benchmark. Proteins 2003;52:88–91.
11. Mintseris J, Wiehe K, Pierce B, Anderson R, Chen R, Janin J, Weng Z. Protein-protein docking benchmark 2.0: an update. Proteins 2005;60:214–216.
12. Pierce NA, Winfree E. Protein design is np-hard. Protein Eng 2002; 15:779–782.
13. Chazelle B, Singh M, Kingsford CL. A semidefinite programming approach to side-chain positioning with new founding strategies. INFORMS J Comput 2003;16:308–392.
14. Desmet J, De Maeyer M, Hazes B, Lasters I. The dead-end elimination theorem and its use in protein side-chain positioning. Nature 1992;356:539–542.
15. Bahadur KCD, Tomita E, Suzuki J, Akutsu T. Protein side-chain packing problem: A maximum edge-weight clique algorithmic approach. J Bioinf Comput Biol 2005;3:103–126.
16. Canutescu AA, Shelenkov AA, Dunbrack RL, Jr. A graph-theory algorithm for rapid protein side-chain prediction. Protein Sci 2003;12:2001–2014.
17. Xu J. Rapid protein side-chain packing via tree decomposition. Lect Notes Bioinf (Subser Lect Notes Comp Sci) 2005;3500:423–439.
18. Xie W, Sahinidis NV. Residue-rotamer-reduction algorithm for the protein side-chain conformation problem. Bioinformatics 2006;22: 188–194.
19. Eriksson O. Side chain-positioning as an integer programming problem. LNCS 2001;2149:128–141.
20. Althaus E, Kohlbacher O, Lenhof HP, Muller P. A combinatorial approach to protein docking with flexible side chains. J Comput Biol 2002;9:597–612.
21. Kingsford CL, Chazelle B, Singh M. Solving and analyzing side-chain positioning problems using linear and integer programming. Bioinformatics 2005;21:1028–1036.
22. Holm L, Sander C. Fast and simple monte carlo algorithm for side chain optimization in proteins: Application to model building by homology. Proteins: Struct Funct Genet 1992;14:213–223.
23. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculations by fast computing machines. J Chem Phys 1953;21:1087–1092.
24. Koehl P, Delarue M. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. J Mol Biol 1994;239:249–275.
25. Mendes J, Soares CM, Carrondo MA. Improvement of side-chain modeling in proteins with the self-consistent mean field theory method based on an analysis of the factors influencing prediction. Biopolymers 1999;50:111–131.
26. Tuffery P, Etchebest C, Hazout S, Lavery R. A new approach to the rapid determination of protein side chain conformations. J Biomol Struct Dyn 1991;8:1267–1289.
27. Hwang JK, Liao WF. Side-chain prediction by neural networks and simulated annealing optimization. Protein Eng 1995;8:363–370.
28. Lee C, Subbiah S. Prediction of protein side-chain conformation by packing optimization. J Mol Biol 1991;217:373–388.
29. Abagyan R, Totrov M. Biased probability monte carlo conformational searches and electrostatic calculations for peptides and proteins. J Mol Biol 1994;235:983–1002.
30. Dominguez C, Boelens R, Bonvin AM. Haddock: a protein-protein docking approach based on biochemical or biophysical information. J Am Chem Soc 2003;125:1731–1737.
31. 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;331:281–299.
32. Wang C, Schueler-Furman O, Baker D. Improved side-chain modeling for protein-protein docking. Protein Sci 2005;14:1328–1339.
33. Fernandez-Recio J, Totrov M, Abagyan R. Identification of protein-protein interaction sites from docking energy landscapes. J Mol Biol 2004;335:843–865.
34. Jackson RM, Gabb HA, Sternberg MJ. Rapid refinement of protein interfaces incorporating solvation: application to the docking problem. J Mol Biol 1998;276:265–285.
35. Li L, Chen R, Weng Z. Rdock: refinement of rigid-body protein docking predictions. Proteins 2003;53:693–707.
36. Brooks BR, Bruccoleri RE, Olafson BD, States DJ, Swaminathan S, Karplus K. CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations. J Comput Chem 1983;4: 187–217.
37. Camacho CJ, Gatchell DW. Successful discrimination of protein interactions. Proteins 2003;52:92–97.
38. Zacharias M. Protein-protein docking with a reduced protein model accounting for side-chain flexibility. Protein Sci 2003;12:1271–1282.

 DOI 10.1002/prot

39. Palma PN, Krippahl L, Wampler JE, Moura JG. Bigger: A new (soft)docking algorithm for predicting protein interactions. Proteins 2000;39:372–384.
40. Dunbrack RL, Jr. Cohen FE. Bayesian statistical analysis of protein side-chain rotamer preferences. Protein Sci 1997;6:1661–1681.
41. Neria E, Fischer S, Karplus M. Simulation of activation free energies in molecular systems. J Chem Phys 1996;105:1902–1921.
42. Kortemme T, Baker D. A simple physical model for binding energy hot spots in proteinprotein complexes. Proc Natl Acad Sci USA 2002;99:14116–14121.
43. Crowley PB, Golovin A. Cation-pi interactions in protein-protein interfaces. Proteins 2005;59:231–239.
44. Misura KM, Morozov AV, Baker D. Analysis of anisotropic side-chain packing in proteins and application to high-resolution structure prediction. J Mol Biol 2004;342:651–664.
45. Word JM, Lovell SC, Richardson JS, Richardson DC. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. J Mol Biol 1999;285:1735–1747.
46. Kohlbacher O, Lenhof HP. Ball-rapid software prototyping in computational molecular biology. Bioinformatics 2000;16:815–824.
47. Khachiyan LG. A polynomial algorithm in linear programming. Soviet Math Doklady 1979;20:191–194.
48. Karmarkar N. A new polynomial-time algorithm for linear programming. Combinatorica 1984;4:375–395.
49. Dantzig GB. Linear programming and extensions. Princeton, NJ: Princeton University Press; 1963.
50. ILOG CPLEX. ILOG CPLEX 7.1, 2000. http://www.ilog.com/products/cplex/.
51. Fourer R, Gay DM, Kernighan BW. AMPL: A modeling language for mathematical programming. Pacific Grove, CA: Brooks/Cole; 2002.
52. Broyden CG. The convergence of a class of double-rank minimization algorithms. J Inst Math Appl 1970;6:76–90.
53. Fletcher R. A new approach to variable metric algorithms. Comp J 1970;13:317–322.
54. Vapnik VN. Statistical learning theory. New York: Wiley; 1998.
55. Ruping S. mySVM-Manual. University of Dortmund, Lehrstuhl Informatik VIII; 2000.
56. Mendez R, Leplae R, Lensink MF, Wodak SJ. Assessment of capri predictions in rounds 3–5 shows progress in docking procedures. Proteins 2005;60:150–169.
57. Chen R, Li L, Weng Z. Zdock: An initial-stage protein-docking algorithm. Proteins 2003;52:80–87.
58. Comeau SR, Gatchell DW, Vajda S, Camacho CJ. Cluspro: an automated docking and discrimination method for the prediction of protein complexes. Bioinformatics 2004;20:45–50.