Protein-Protein Docking with F$^2$Dock 2.0 and GB-rerank

Rezaul Chowdhury$^1$, Muhibur Rasheed$^1$, Donald Keidel$^2$, Maysam Moussalem$^1$, Arthur Olson$^2$, Michel Sanner$^2$, Chandrajit Bajaj$^2$\*

$^1$Department of Computer Science, Institute of Computational Engineering and Sciences, University of Texas at Austin, 1 University Station C0200, Austin, Texas 78712, USA
$^2$The Scripps Research Institute, 10550 North Torrey Pines Road, La Jolla, CA 92037, USA
\* E-mail: bajaj@cs.utexas.edu

Abstract

**Motivation:** Computational simulation of protein-protein docking can expedite the process of molecular modeling and drug discovery. This paper reports on our new F$^2$Dock protocol which improves the state of the art in initial stage rigid body exhaustive docking search, scoring and ranking by introducing improvements in the shape-complementarity and electrostatics affinity functions, a new knowledge-based interface propensity term with FFT formulation, a set of novel knowledge-based filters and finally a solvation energy (GBSA) based reranking technique. Our algorithms are based on highly efficient data structures including the dynamic packing grids and octrees which significantly speed up the computations and also provide guaranteed bounds on approximation error.

**Results:** The improved affinity functions shows superior performance compared to their traditional counterparts in finding correct docking poses at higher ranks. We found that the new filters and the GBSA based reranking individually and in combination significantly improve the accuracy of docking predictions with only minor increase in computation time. We compared F$^2$Dock 2.0 with ZDock 3.0.2 and found improvements over it, specifically among 176 complexes in ZLab Benchmark 4.0, F$^2$Dock 2.0 finds a near-native solution as the top prediction for 22 complexes; where ZDock 3.0.2 does so for 13 complexes. F$^2$Dock 2.0 finds a near-native solution within the top 1000 predictions for 106 complexes as opposed to 104 complexes for ZDock 3.0.2. However, there are 17 and 15 complexes where F$^2$Dock 2.0 finds a solution but ZDock 3.0.2 does not and vice versa; which indicates that the two docking protocols can also complement each other.

**Availability:** The docking protocol has been implemented as a server with a graphical client (TexMol) which allows the user to manage multiple docking jobs, and visualize the docked poses and interfaces. Both the server and client are available for download.


Introduction

The study of protein-protein interactions plays an important role in understanding the processes of life [1]. Though advancements in X-ray crystallography and other imaging techniques have led to the extraction of near-atomic resolution information for numerous individual proteins; the creation, crystallization and imaging of macromolecular complexes, as extensively required for drug design, still remains a difficult task. Among the atomic structures of proteins deposited in the Protein Data Bank [2], only a very small percentage are complexes. Hence, the need for fast and robust computational approaches to reliably predict the structures of protein-protein complexes is growing. An important step towards understanding protein-protein interactions is *protein-protein docking* which can be defined as computationally finding the
relative transformation and conformation of two proteins that results in a stable (energetically favorable) complex if one exists.

Given two rigid proteins and some characteristic (e.g., electron density) function(s) of the molecules, one can construct an appropriate representation of them and also define a correlation function based on cumulative overlap of the characteristic functions. Then it is possible to conduct a combinatorial search in a 6D parameter space of all possible relative translations and orientations of the two proteins to find the optimal. Hence in computational perspective, docking is a search over the space of possible orientations of two proteins to find the (set of) optimum(s) of a scoring function designed to mimic physico-chemical interaction of proteins.

The combinatorics of the search can be reduced by using coarse grids and rotational angles [3], and by using a-priori knowledge of suitable binding sites [4]. For docking without prior knowledge about possible binding sites, exhaustive sampling is required to improve the probability of finding the global minimum energy configuration. In such cases, Fast Fourier Transforms has been used to speed up the cumulative scoring function computations [3–5]. Spherical Fourier correlation based approaches were presented in multiple studies [5–9]. However, if binding sites are known, or inferred based on some initial stage docking, then a finer resolution search involving local flexibility can be applied to improve the accuracy of the fit [10–12].

Accuracy of docking predictions is dependent on the scoring model’s ability to distinguish between native and non-native poses. Docking based on structural (shape) complementarity alone has shown to be adequate for a range of proteins [4, 13, 14]. To represent shape complementarity, a grid based double skin layer approach became the base of many variations and software, e.g., DOT [15], ZDOCK [16], PIPER [17], MolFit [18, 19] and RDOCK [20]. However, energy and bioinformatics based scoring terms have been shown to improve the accuracy of predictions and a combination of multiple scoring terms have become the norm in current docking software. For example, DOT 2.0 [15] is based on van der Waals energy and Poisson-Boltzmann electrostatics, ZDock 3.0.2 [21] uses pairwise shape complementarity, electrostatics, and pairwise potentials known as Interface Atomic Contact Energies (IFACE). PIPER [17] is based on shape complementarity and electrostatics using a Generalized Born (GB) type formulation, and uses a new class of structure-based potentials referred to as DARS (Decoys As the Reference State) where the potentials are derived from a large set of docking conformations as decoys. FRODOCK [22] is a recent spherical harmonics based docking tool that uses van der Waals, electrostatics and desolvation potential terms in its correlation function. Some docking or reranking techniques solely use coarse-grained potentials trained on large benchmark of decoys [23, 24]. We leave the reader to consult the reviews [25–29] for further information.

In [30] we described a non-equispaced Fast Fourier Transform (NFFT) based rigid-body protein–protein docking algorithm for efficiently performing the initial docking search (based on shape and electrostatics complementarity). Compared to traditional grid based Fourier docking algorithms, the algorithm was shown to have lower computational complexity and memory requirement. The algorithm was extended in [31] to F²Dock ($F^2 = \text{Fast Fourier}$), which included an adaptive search phase (both translational and rotational) for faster execution.

In this paper we describe a new version (F²Dock 2.0) which includes improved shape-complementarity and electrostatics functions as well as a new on-the-fly affinity function based on interface propensity and hydrophobicity. The current version uses uniform FFT, but exploits the sparsity of FFT grids for faster execution and restricts its search within a narrow band around the larger molecule. A clustering phase penalizes docking poses that are structurally similar to poses with better scores and a set of efficient on-the-fly filters penalize potential false positives based on Lennard-Jones potential, steric clashes, interface propensity, interface area, residue-residue contact preferences, antibody active sites, and glycine richness at the interface for enzymes. The filters are implemented using fast multipole type recursive spatial decomposition techniques [32, 33]. A solvation energy based reranking program GB-rerank [32, 34] has also been implemented using an approximation scheme which can be tuned for speed-accuracy trade-off.
Both F²Dock 2.0 and GB-rerank have been implemented as multithreaded programs for faster execution on multicore machines. Our molecular visualization software TeχMol serves as a front-end to F²Dock 2.0 in a client-server mode of execution [35]. F²Dock has been calibrated based on an extensive experimental study of the rigid-body complexes from Zlab benchmark 2.0 [36] and tested on Zlab benchmark 4.0 [37] (which includes the complexes from benchmark 2.0).

The paper is organized as follows. In the next section, we describe the latest version of F²Dock 2.0. Results are presented and discussed in the next section, followed by conclusions and plans for future research.

**Methods**

Let A and B be two proteins with $M_A$ and $M_B$ atoms respectively. Without loss of generality, we assume that $M_A \geq M_B$, i.e., A is the larger of the two proteins. We refer to A and B as "receptor" and "ligand", respectively.

Figure 1 gives a high level overview of the algorithm. The rest of this section details the various aspects of the algorithm.

2.1 Overall Strategy

First, F²Dock 2.0 performs exhaustive search in 6D space of relative configuration of B with respect to A. We use a discrete and uniform sampling of 3D rotational space and then use FFT to score a discrete 3D translational space. Given $N_R$ rotational samples and $N^3$ translational grid, F²Dock 2.0 computes $N_R N^3$ scores. However, only a constant multiple of $N_R$ scores and their corresponding poses are retained for the next step. Let us denote this set as $Q$. A particular pose is expressed as a tuple $< t, r, s >$ where $t$ is the translation, $r$ is the rotation and $s$ is the corresponding score.

We apply a very simple clustering scheme based on proximity of the poses in $Q$ to reshuffle the order such that the top few poses are dissimilar to each other. Though this step does not affect the overall ratio of true and false positives, it increases the chance of finding at least one near-native solution at the top of the order. It is important because the next stage of filtering is only performed on the top few (2000 by default) poses. Let this reduced list be called $Q'$.

The filters are designed based on knowledge-based scoring potentials, which are described in Section 2.5, to update the scores of the poses of $Q'$, reorder them and output them as final predictions from F²Dock 2.0. Some filters are defined only for specific types of proteins like Antibodies or Enzymes.

The results from F²Dock 2.0, or a subset of it, can optionally be reranked using a solvation energy (generalized Boltzman model) based reranker called GB-rerank which generally improves ranks of near native solutions.

2.2 Exhaustive Search Using FFT

2.2.1 Rotational Sampling

The rotation space is sampled using uniformly distributed Euler angles as in [13, 15, 38]. For each sampling interval $\Delta$ the sample set is equivalent to a set of points uniformly distributed on a projective sphere such that the angular distance [39] between any two points in the set is at most $\Delta$. This approach provides a much better distribution of samples than sampling each angle ($\theta, \phi, \chi$) separately and requires fewer samples for the same $\Delta$. 
2.2.2 Translational Sampling and Scoring.

FFT-based scoring of the translational grid (see, e.g., [3, 4]) involves two forward (one each for molecules A and B) and one inverse FFT computations. Since the forward FFT of the stationary molecule A can be precomputed, in practice, only one forward (involving molecule B) and one inverse FFT must be computed for each rotation. Current version of F^2Dock 2.0 uses uniform FFT but exploits the sparsity of the input and the output grids for faster computation as follows.

- **Forward FFT (sparsity of the input grid)**: The input grids are large enough so that they can at least contain the following two spheres side-by-side: the smallest sphere enclosing molecule A, and another sphere having the same radius \( r_B \) as the largest distance from the geometric center of molecule B to an atom of B. Hence, when discretized to such a grid molecule B will occupy only a fraction of the grid points around the grid center. Thus many grid planes will remain completely empty (i.e., have zero values only). 3D FFT is sped up by ignoring recursive calls that compute 2D FFT’s of such empty planes.

- **Inverse FFT (sparsity of the output grid)**: In the output translational grid we need to score only the grid points that are within a band around the stationary molecule A such that if the geometric center of molecule B lies outside the band the two molecules can never touch. Note that the number of gridpoints in the band is \( O(N^2) \) as opposed to \( N^3 \). This band can be approximated during the initial precomputation phase by running a sphere of radius \( r_B \) (defined in previous paragraph) over the surface of A which can be done using a single call to FFT. This sparseness of the output grid is exploited to speed up inverse FFT.

2.2.3 Cost of FFT-based Affinity Function Computations

For any rotation \( \mathbf{r} \) the FFT-based scoring takes \( O(M_A + M_B + N^3 \log N) \) time, where \( N^3 \) is the size of the FFT grid. Hence, for \( N_R \) rotations the total running time is \( O(N_R(M_A + M_B + N^3 \log N)) \). We use the publicly available FFTW package [40] for our FFT-based scoring.

2.3 FFT-based Affinity Functions

F^2Dock 2.0 uses three affinity functions during exhaustive search over the rotational and translational space. The score is defined as a weighted sum of the shape complementarity, electrostatics and interface propensity scores. Using a new approach to skin-core definition and weighting, the shape complementarity term has been greatly improved from the original version (F^2Dock), and the interface propensity is a novel addition defined using statistical residue potentials.

2.3.1 Shape Complementarity (SC)

The original version of F^2Dock [31] used the traditional double skin layer approach for shape complementarity [41]. Two skin regions are defined (Figure 2): a grown skin region around A, and the surface skin of B, which consists of the surface atoms of B. The atoms of A and the inner atoms of B form core regions.

A good docking pose of A and B will have large skin-skin overlap and small core-core overlap, and in order to identify such poses constant positive imaginary weights are assigned to the core atoms and constant positive real weights to skin atoms/pseudo-atoms. An integral of the superposition of the molecules has two real contributions: the core-core overlaps contribute negatively and the skin-skin overlaps contribute positively. Hence the real part can be used to rank docking poses based on shape complementarity. The magnitude of the imaginary part of the integral due to skin-core clashes (caused by pseudo-atom vs atom overlaps) is not desirable and assigned a smaller negative weight in the accumulated score.
The current version ($F^2$Dock 2.0) uses an *improved double skin layer approach* which differs from the traditional approach in four ways. First, the receptor skin layer does not touch the receptor van der Waals surface and the radius of skin atoms are different. This is based on our observation of the gap between the VDW surfaces of the receptors and ligands in Zlab benchmark 2.0 [36]. Second, the weights assigned to receptor skin atoms are computed based on the curvature of the skin around that atom. Such weighting encourages convex-concave and concave-convex interfaces as opposed to large flat interfaces. Third, the core atoms of molecule $A$ are assigned weights using an increasing function of depth (distance of the atom center from the surface of $A$). Such weightings discourage deeper core-core overlaps more strongly. And fourth, since in the traditional approach the ligand skin is defined using its surface atoms, the skin thickness varies and can be too thin in some areas. Therefore, we use a double layer of ligand atoms as its skin. Refer to supplemental materials for in depth discussion on the skin-core definition as well as FFT based correlations.

2.3.2 Electrostatics (E)

In the previous version ($F^2$Dock) [31], we defined the electrostatics affinity function similar to the simplified model for electrostatics described by Gabb et. al. [4], which allows efficient FFT-based computation. The first protein's electric potential is computed and matched against the charges in the other. In this version ($F^2$Dock 2.0), we replace point charges with a Gaussian to reduce discretization errors on the grid (See supplement for details).

2.3.3 Interface Propensity (IP) and Hydrophobicity (HP)

$F^2$Dock 2.0 scores the interfaces between molecules $A$ and $B$ using the per-residue interface propensity values computed in [42] which are based on relative frequencies of residues in the interfaces of a set of 63 protein-protein complexes [43]. Let $IP(R)$ denote the natural logarithm of the interface propensity value of a residue $R$. The $IP$ values for the 20 amino acid residues lie between -0.38 (ASP) and 0.83 (TRP). A residue with higher $IP$ value is likely to occur more frequently in a protein-protein interface than one with a lower value.

Let $iAtom^+_t,B(r)$ and $iAtom^-_t,B(r)$ denote the set of atoms in the interface of $P \in \{A, B\}$ in this docking pose that have positive and negative $IP$ values, respectively. Also let $iAtom^+_{t,B}(A) = iAtom^+_t,B(A)$ and $iAtom^-_{t,B}(B)$, for $s \in \{+, -\}$. We assign an interface propensity score to the pose:

$$IP\text{-score}_{t,B}(A, B) = -\frac{\sum_{a \in iAtom^+_t,B(A,B)} IP(a)}{\min\left(IP_c, \sum_{a \in iAtom^-_{t,B}(A,B)} IP(a)\right)},$$

where $IP_c = \max_{P(R) < 0} IP(R)$. We also penalize very small interfaces by setting $IP$-score to a negative value when $\lvert iAtom^+_t,B(A, B) \rvert$ is below a user-defined threshold.

See supplemental material for details on the FFT based formulation and parameter selections.

2.4 Proximity Clustering

We consider all docking poses in $Q$ sorted by score, and penalize a pose if there are similar poses with higher score. Suppose $A + B_{t,r}$ is a docking pose with score $s$ that is currently being considered. For a given $\delta \geq 0$, let $K(B_{t,r}, s, \delta)$ be the number of docking poses with score at least $s$ and with geometric center within distance $\delta$ from that of $B_{t,r}$. Then we penalize $A + B_{t,r}$ as follows.

1. If $K(B_{t,r}, s, \delta) > 0$, then the score of the pose $A + B_{t,r}$ is reduced by 80%,

2. otherwise, if $K(B_{t,r}, s, 2\delta) > 3$, then the score of the pose $A + B_{t,r}$ is reduced by 50%,
3. otherwise, if $K(B_t, s, 3\delta) > 6$, then the score of the pose $A + B_t$ is reduced by 10%.

The objective of this proximity based penalty is to increase diversity at the top of the order which increases the possibility of getting at least one near native solution at the top of the order. But it also penalizes all true positives (except one). As a result, F$^2$Dock 2.0 tends to get a near native solution at high ranks for many complexes, but the total number of near native solutions for any particular complex is not high.

### 2.4.1 Cost of Clustering

We use our dynamic packing grid data structure [33] to speed-up this computation, and the overall time required for this step is $O(N_Q \log N_Q)$ (with high probability), where $N_Q$ is the number of poses originally in $Q$.

### 2.5 Filters

To penalize potential false positives and thus improve the ranks of correct solutions, F$^2$Dock 2.0 uses several filters based on the Lennard-Jones potential, the number of steric clashes, interface area, interface propensity, residue-residue contact preferences, antibody active sites, and the frequency of glycine residues at the interface for enzymes.

Only the interface regions of the two molecules at a given pose contribute to the terms used in the filters. We have developed an octree-based hierarchical spatial decomposition technique [32] and the Dynamic Packing Grids data structure [33] for efficiently locating the interface regions and for computing local interactions. Since the overall algorithm for computing each term is similar and only varies in the exact type of local interactions we compute, we present the algorithm only once (in our discussion of the interface propensity filter).

#### 2.5.1 Interface Propensity Filter

This filter computes the interface propensity of the interfaces of the molecules at a given pose and penalizes or rewards the pose based on empirically determined thresholds.

We sample and weigh quadrature/integration points from the surface of each molecule as described in [32, 34]. The sampling can be considered a decomposition of the surface into small patches, where each quadrature point is representative of a patch $\pi$. The weight of a quadrature point is the same as the area $a_\pi$ of the corresponding patch. Each quadrature point is also labeled with the average of the interface propensity values of the atoms near the point. Let, $h_\pi$ denote the interface propensity label of a quadrature point. The interface propensity contribution of a quadrature point is defined as $a_\pi h_\pi$ if $\pi$ is on the interface, and 0 otherwise.

We approximate the interface propensity as $IP^{\text{score}}_{t,r}(A, B) \approx -(v_2 + v_4)/\min(IP'_t, v_1 + v_3)$, where, $IP'_t = (\max_{h_k \leq 0} h_k)(\min_{\pi \in A \cup B} a_\pi)$ (defined in Section 2.3.3). Here $v_1$ and $v_3$ are the sum of the negative interface propensity contributions of the quadrature points of two molecules; and $v_2$ and $v_4$ are the sum of the positive contributions. We reward docking poses with large $IP^{\text{score}}_{t,r}(A, B) \times (v_2 + v_4)$ values, and penalize a pose if its $IP^{\text{score}}$ is below a threshold.

The crucial step in the approximation is identifying quadrature points that are on the interface. We store the quadrature points into a DPG data structure, and we also store them in an octree. The octree is a hierarchical and adaptive subdivision of space such that a node of the tree represents a regular cube in 3-space, A node is split if it contains more than a user-defined number of quadrature points in it. Given a specific pose, we trace the two octrees starting from the roots to identify the leaves that are close to each other. Then for each pair of neighboring octree-cells, we use DPG to identify quadrature points in one leaf which have a neighbor in the other leaf. The overall cost of the algorithm is $O((M_A + M_B) \log(M_A + M_B))$
with high probability, where $M_A$ and $M_B$ are the number of atoms in $A$ and $B$. However, in practice it runs even faster and approaches $O(n_{int})$, where $n_{int}$ is the number of quadrature patches on the interface.

### 2.5.2 Residue-Residue Contact Filter

Contact preferences derived from a non-redundant set of 621 protein-protein interfaces of known high resolution structures [44] are used to penalize potential false positives. Two residues are considered to be in contact if the distance between their $C_\beta$ atoms ($C_\alpha$ for Gly) is less than 6 Å. In [44], log-normalized contact preferences $G_{ij}$ for each pair of amino acid types are reported. Positive values of $G_{ij}$ indicate that residues $i$ and $j$ prefer to form contacts, negative values indicate the opposite.

Given a docking pose $A + B_{t,r}$, we identify all residue-residue contacts at the interface of the two molecules using a fast algorithm similar to the one used in Section 2.5.1, and compute the sum of all positive and negative $G_{ij}$ values denoted by $G^+$ and $G^-$, respectively. Then we penalize the pose if the ratio of $G^+$ and $G^-$ is outside a user-specified range.

### 2.5.3 Lennard-Jones Filter

We approximate the Lennard-Jones ($LJ$) potential between molecules $A$ and $B_{t,r}$ as follows: $LJ(A, B_{t,r}) = \sum_{i \in A, j \in B_{t,r}} (a_{ij}/r_{ij}^{12} - b_{ij}/r_{ij}^{6})$, where $r_{ij}$ is the distance between atoms $i \in A$ and $j \in B_{t,r}$, and constants $a_{ij}$ and $b_{ij}$ depend on the atom types. The well depths $\mu$ and equivalence contact distances of homogeneous pairs $r_{eqm}$ are taken from the Amber force field [45, 46]. Poses with positive $LJ$ potential are penalized. However, we allow soft clashes in the cases of unbound-(un)bound docking by reducing the $r_{eqm}$ values by a constant factor which effectively reduces the inter-atomic clash distances ($r_{ij}$ values).

### 2.5.4 Clash Filter

Two atoms $a \in A$ and $b \in B$ with van der Waals radii $r_a$ and $r_b$, respectively, are considered to be clashing provided the distance between their centers is smaller than a threshold. F$^2$Dock 2.0 counts the total number of clashes $n_C$ between molecules $A$ and $B(t,r)$ and penalizes if $n_C > \mu_C$, where $\mu_C$ is a user-defined constant.

### 2.5.5 Interface Area Filter

This filter penalizes a docking pose if the interface area is outside the range of areas derived empirically from known native interfaces. We define the interface area as the sum of the weights of the quadrature points on the interface, where the weights and the interface is defined the same way as in the interface propensity filter.

### 2.5.6 Glycine Filter

Enzyme active sites are rich in Glycines, particularly $G-X-Y$ and $Y-X-G$ oligopeptides, where $X$ and $Y$ are polar and non-polar residues, respectively, and $G$ is glycine [47]. The $X$ and $Y$ residues are typically small in size and low in polarity, and the frequency of those two types of oligopeptides is significantly higher in enzyme active regions than in other parts of the enzyme molecule. Therefore, enzyme surface oligopeptides with these properties are marked and for a given docking pose, the number of these motifs occurring at the interface are counted. If this count is below a user-specified threshold $\mu_G$, the pose is penalized. Conversely, poses with higher $G-X-Y/Y-X-G$ frequency at the interface are rewarded by adding this (weighted) count to the total score.
2.5.7 Antibody-Antigen Contact Filter

As reported in [48] (available at http://www.bioinf.org.uk/abs/allContacts.html) based on a set of 26 known antibody-antigen complexes, each of the following three regions in an antibody will make at least one antigen contact (burial by at least 1 Å² change in solvent accessibility): (1) CDR-L1 or CDR-H1, (2) CDR-L3 and (3) CDR-H3.

Given a potential antibody-antigen docking pose, F²Dock 2.0 computes \( N_{L1,H1} \), \( N_{L3} \) and \( N_{H3} \), the number of antigen atoms that are in the close neighborhood of any atom in the antibody regions CDR-L1/CDR-H1, CDR-L3 and CDR-H3, respectively. The CDR (Complementarity Determining Region) loops are identified using the method described in [49]. F²Dock 2.0 penalizes poses if the computed values are outside the ranges observed in the native antibody-antigen interfaces in our training set.

2.5.8 Cost of Filtering

Using our algorithm described in [32] based on octrees [50] and our Dynamic Packing Grid (DPG) data structure [33], the scores for each filter can be evaluated in \( O \left( \frac{1}{\epsilon} (M_A + M_B) \log (M_A + M_B) \right) \) w.h.p. (for an input of size \( n \), an event \( E \) occurs w.h.p. (with high probability) if, for any \( \alpha \geq 1 \) and \( c \) independent of \( n \), \( Pr(E) \leq 1 - \frac{c}{n^\alpha} \) ) time and \( O(M_A + M_B) \) space. Assuming that each filter is applied on at most \( N_F \) configurations, the total time taken by all filters is \( O \left( \frac{1}{\epsilon} N_F (M_A + M_B) \log (M_A + M_B) \right) \) (w.h.p.). Assuming \( N_F = O(N_k) \), where \( N_k \) is the number of samples in the rotations space, the running time reduces to \( O \left( \frac{1}{\epsilon} N_R (M_A + M_B) \log (M_A + M_B) \right) \) (w.h.p.).

2.6 Solvation Energy Based Reranking with GB-rerank

GB-rerank approximates the change in solvation energy of a complex and reranks the list of top docking poses produced by F²Dock 2.0 based on the resulting \( \Delta E_{sol} \) values. In order to approximate \( \Delta E_{sol} \), GB-rerank precomputes the \( E_{sol} \) values for molecules \( A \) and \( B \), and then computes \( E_{sol} \) for each docking pose.

The solvation energy \( E_{sol} \) consists of the energy to form cavity in the solvent \( E_{cav} \), the solute-solvent van der Waals interaction energy \( E_{vdw(s-s)} \), and the electrostatic potential energy change due to the solvation (also known as the polarization energy, \( E_{pol} \)) [51–55].

\[
E_{sol} = E_{cav} + E_{vdw(s-s)} + E_{pol}
\]

The first two terms are often modeled as \( E_{cav} = pV + \sum \gamma_i A_i \) and \( E_{vdw(s-s)} = \rho_0 \sum_i \int_{ex} u_i^{att}(x_i, r) d^3 r \), where \( x_i \) and \( r \) are the position and center of atom \( i \), \( p \) is the solvent pressure, \( V \) is the molecular volume, \( A_i \) is the solvent accessible surface area of atom \( i \) and \( \gamma_i \) is its solvation parameter, \( \rho_0 \) is the bulk density, and \( u_i^{att} \) is the van der Waals dispersive component of the interaction between atom \( i \) and the solvent [51, 56]. The last term, \( E_{pol} \), can be approximated using the Generalized Born theory [57], whereby:

\[
E_{pol} = -\frac{\tau}{2} \sum_{i,j} q_i q_j / \sqrt{r_{ij}^2 + R_i R_j e^{-\frac{r_{ij}^2}{\tau}}},
\]

where \( q_i, q_j \) are the atomic charges and \( r_{ij} \) is the distance between atoms \( i \) and \( j \), \( R_i \) is the effective Born radius of atom \( i \) and \( \tau = 1 - \frac{1}{\epsilon} \).

The algorithms for rapidly approximating these terms have been presented in [32].
2.6.1 Overall Cost of Reranking

Assuming that GB-rerank is applied on \( N_G \) docking poses, its total running time is
\[
\mathcal{O} \left( \frac{1}{\epsilon^3} N_G (M_A + M_B) \log (M_A + M_B) \right) \text{ (w.h.p.).}
\]

Typically \( N_G = \mathcal{O}(N_R) \), where \( N_R \) is the number of samples in the rotations space, and so the running time reduces to
\[
\mathcal{O} \left( \frac{1}{\epsilon^3} N_R (M_A + M_B) \log (M_A + M_B) \right) \text{ (w.h.p.).}
\]

2.7 Dataset Preparation

F2Dock takes two PDB files as inputs. First the PDB files are processed by PDB2PQR [58] where missing atoms such as Hydrogens are added, the protein is optimized for hydrogen bonding, and charge and radius parameters are assigned using the AMBER force-field available in PDB2PQR. If the given PDB has missing residues or too many residues with missing backbone atoms, then our curation process fails and F2Dock cannot be used without manually curating the PDB or using other curation software.

Then pseudo-atoms are added above the surface of the receptor (i.e., stationary molecule), and surface atoms of the ligand (i.e., moving molecule) are detected. These atoms are marked as skin atoms, and the rest as core atoms.

2.8 Parameter Selection Based on Complex Type

F2Dock 2.0 has several free parameters in its pipeline. We can broadly classify the parameters into several groups. For parameters like the charge and radii of atoms, or the hydrophobicity and interface propensity of residues etc., we either use well-established parameters (for example, from the AMBER [59] force field) or derive from previously published results (for example, interface propensity values from [42]). Some parameters are internal to a scoring function for example the distance dependent dielectric for electrostatics, or the thickness of the skin used in shape complementarity. These parameters are trained using manual parameter sweeps based on a small number (4-5 per complex type) of complexes. However, we produced multiple configurations for each complex and chose the set of parameters which maximizes the corresponding individual scoring term for the near native poses. A similar strategy was used for selecting the thresholds used to penalize poses during filtering. Finally, there are the parameters that govern the weights assigned to different scoring terms when they are combined as well as the weights (or percentages) by which poses are penalized. These parameters are the most difficult to train as the scoring terms are not independent and the relative influence of a term might vary for different complexes. These parameters were trained based on the 60 complexes from Zlab’s protein-protein benchmark 2.0 [36] as follows.

The complexes in the benchmark are categorized into four main types: Antibody-Antigen (A) and Antibody-bound Antigen (AB), Enzyme-Inhibitor/Enzyme-Substrate (E), and other (O) types. We identify that the classification is not only functional, but it also has significant effect on scoring function design since different scoring terms bear different level of significance for different categories of complexes. For example, it is known that binding interfaces of Enzymes are rich in Glycines, which lead us to design a filter based on Glycine richness and it is applied only for Enzyme type of complexes. For each class of complexes (9 Antibody-Antigen, 9 Antibody-bound Antigen, 21 Enzyme-Inhibitor/Enzyme-Substrate and 21 Others), we train the weight parameters separately. The objective for the training is to improve the ranks of near-native solutions for as many complexes as possible. We performed parameter sweeps for each of the weights that combines the FFT based scores based on the above objective for each of the categories. Then we examined the effect of applying each of the filter, one at a time, and controlled its penalty to improve the results.

We do realize that our manual scheme has its drawbacks, specially since it does not sufficiently cover the entire space of possible values for the parameters. We are actively trying to use machine learning schemes to train the parameters in a more robust way. However, so far our attempt of using quadratic
programming and random forest learning based on thousands of negative and positive examples based
on this benchmark have failed to produce a set of parameters which outperform the manually calibrated
set of parameters.

Default values of all the parameters for different types of complexes can be found in the user manual
for F^{2}Dock 2.0 downloadable from our website (link given in the abstract).

2.8.1 Automated Detection of Complex Types

Since F^{2}Dock 2.0’s parameters are optimized separately for antibody-antigens and enzyme-inhibitors/
enzyme-substrates, and a general set of parameters are used for all other types of complexes, the user
only needs to specify the complex type to ensure the set of optimized parameters are applied. If the
type is unknown, F^{2}Dock 2.0 tries to determine which set of parameters to use as follows. If F^{2}Dock
2.0 locates the six CDR loops (L1, L2, L3, H1, H2 and H3) in the protein sequence using the algorithm
in [49], it identifies it as an antibody and uses the corresponding parameter set. Otherwise, if neither
molecule is identified as an antibody and at least one of the molecules has at least 200 residues and at
least 8% of its surface residues are Glycines then F^{2}Dock 2.0 uses the enzyme complex parameter set.
Finally, if both tests fail, a set of parameters for the general case is used. Among the complexes in the
Zlab benchmark 2.0, F^{2}Dock 2.0 fails to identify only one antibody (1KXQ) and three enzymes (1AY7,
1UDI and 2MTA). See supplement for details.

Results

We present the results of our experiments to explore the contribution of the new scoring terms and
filters available in F^{2}Dock 2.0 as well as the solvation energy based re-ranker GB-rerank on prediction
accuracy. These experiments are carried out on the set of complexes in Zlab’s benchmark 2.0 [36] which
contains 60 complexes. Then we run F^{2}Dock 2.0 with the best set of parameters on the complexes in
the Zlab benchmark 4.0 [37], and compare the performance with ZDock 3.0.2 [21]. The complexes in
both the benchmarks are categorized into rigid-body (easy), medium and difficult (flexible) based on the
RMSD between the bound and unbound states of the proteins. They are also categorized into four main
types: Antibody-Antigen (A) and Antibody-bound Antigen (AB), Enzyme-Inhibitor/Enzyme-Substrate
(E), and other (O) types. As mentioned before, F^{2}Dock 2.0 uses different set of parameters for the
different categories and we have also compared our results for each category separately.

3.9 Evaluation Criteria

F^{2}Dock 2.0’s search leaves the receptor stationary and searches over the orientations of the ligand. Hence,
to evaluate the accuracy of a predicted pose, we compute the deviation between the predicted position
of the ligand and its correct position as the root mean squared distance (RMSD) of the interface atoms. Note
that correct position of the ligand for unbound test cases can be approximated by aligning the unbound
components to their bound counterparts. The unbound ligands in the ZLab benchmarks are provided
after alignment with bound counterparts and hence can be used as the approximate truth without further
manipulations. We assume that an atom is on the interface if the distance between its center and the
center of any atom on the other molecule is less than 10Å. We define $L_I$ as the set of all backbone atoms
of the ligand which are on the interface when the ligand is placed in its native pose w.r.t the receptor (to
find the native pose for an unbound case, we simply align the unbound receptor and unbound ligand to
their bound counterparts). If the position of ligand atom $a_i$ is $x_i^*$ in the native pose and $x_i^P$ in a predicted
pose $P$, then the interface RMSD is computed as $IRMSD = \sqrt{\frac{1}{|L_I|} \sum_{a_i \in L_I} (|x_i^* - x_i^P|^2)}$. A predicted
solution is considered a hit provided its IRMSD value is at most 5Å.
In the remaining text and supplement, we refer to the hit with the lowest RMSD as the ‘best’ hit and the hit with the highest rank as the ‘top’ hit. In most of our results, we compare protocols based on the rank of the ‘top’ hit. Given a set of complexes $C$, and a protocol $S$, we define $C_S(x)$ as the set of complexes such that for each complex $c \in C_S(x)$, the top hit lies within the range $[1, x]$. Clearly, for a given $x$ a higher $C_S(x)$ is better. Hence, to compare the accuracy of two protocols $S_1$ and $S_2$, we can simply compare $C_{S_1}(x)$ and $C_{S_2}(x)$ for different $x$. In general we use a few specific values for $x$ ([$1,1$], [$1,5$], [$1,10$], [$1,50$], [$1,100$], [$1,500$] and [$1,1000$]). We are specially interested in the first few ranges which shows off the accuracy of the scoring model, and the last range which shows off the applicability of the model over a broad range of complexes.

Two residues $R_i \in A$ and $R_j \in B$ are considered to be in contact if the distance between the centers of any atom $a_{ii} \in R_i$ and any atom $a_{jj} \in R_j$ is less than a threshold. The set of residue-residue contacts for the native pose of the receptor and ligand are defined as the native contacts $N$. For a given predicted pose, we compute the set of residue-residue contacts for that pose as $C$. The set of native contacts for that pose is hence defined as $N' = N \cap C$. Now, we define another metric based on native contacts as $F_{nat} = |N'|/|N|$. We follow the well known CAPRI criteria that uses a combination of $F_{nat}$ and $IRMSD$ to classify predictions as high, medium, acceptable and incorrect.

3.10 Analyzing the Improvements due to New Affinity Functions and Filters

3.10.1 Effectiveness of the New Skin-Core Definition

We have compared the new improved double skin approach to the traditional approach (used in $F^2$Dock [31]) in terms of their prediction accuracy on the rigid-body complexes of the Zlab Benchmark 4.0. In these tests only the shape complementarity term was used, and hence the results are not as accurate as the default combination of scoring and filtering terms can produce.

In Figure 3(a), we clearly notice the improvement offered by the floating skin approach over the traditional which validates our idea that a softer definition of skin is better for unbound docking. However, the traditional skin approach performs slightly better for the bound-bound (re-docking) test cases (Figure 3(b)). Figure 3(c) shows that as a result of the improved skin definition, $F^2$Dock 2.0’s shape complementarity function outperforms DOT and ZDock on the rigid complexes from Zlab benchmark 2.0 (bound-bound).

3.10.2 Effects of Various Filters on Quality of Solutions

Figure 4(top) shows how the number of test cases (rigid-body test cases from Zlab benchmark 2.0 [36]) with at least one hit in top 1, top 10, top 50, top 100, top 500 and top 1000 changes as various affinity functions and filters in $F^2$Dock 2.0 are applied. The filters are applied to the top 2000 predictions after using the FFT based affinity terms and clustering. In this experiment, we have specified the complex type (A/AB, E and O) for each test case. Clearly, each of the filters (except interface area filter) individually improves the ranks of the top solution, and the best outcome is generated when the default combination of filters are used. For example, after the FFT based scoring, we get a hit at rank 1 for 10 complexes, but after filtering it improves to 17. Since the antibody and enzyme filters do not apply to all types of complexes, we compare their effect only on the particular type of complexes. For example, Figure 4(bottom) shows the effectiveness of the enzyme filter.

The series of plots in Figure 5 shows a detailed breakdown of the effect of different scores/filters for each complex separately. On the X-axis, we list the complexes and the Y-axis shows the change of the rank of the top hit. In the figures, an improvement is defined as producing the top hit at a better rank. We use the results of using just shape complementarity as the base case and analyze the relative improvements as more and more terms are added.
When we activate Lennard-Jones filter, clash filter and proximity clustering after shape complementarity we get hits for 4 new test cases, and the rank of the top hit improves for 15 more (see Figure 5(a)). However, we also lose hits in top 1000 for 3 test cases, and the rank of the top hit degrades for one test case. Overall, the application of these filters and clustering seem largely beneficial. The best results are obtained for enzyme-inhibitor/enzyme-substrate complexes, as for more than 50% of these complexes rank of the top hit improves.

When electrostatics is turned on we get hits in top 1000 for 9 test cases for which we did not have a single hit before, and for 14 other cases rank of the top hit improve (see Figure 5(b)). However, we lose hits 1 test case, and for 4 others rank of the top hit degrades.

The FFT-based interface propensity scoring is activated next which improves the rank of the top hit for 30 test cases (i.e., for around 50% of all cases) among which 7 cases did not have a single hit before (see Figure 5(c)). Among these 7 cases with new first hits 5 are antibody-antigen or antigen-bound antibody complexes, and none are enzyme-inhibitor or enzyme-substrate.

The interface propensity filter is turned on next. It improves the rank of the top hit for 25 complexes, and degrades for 5 (see Figure 5(d)). For 3 test cases we did not have a single hit in top 1000 before among which 2 are antibody-antigens.

The residue-residue contact filter which is activated next improves the rank of the top hit for 27 test cases, and degrades for none (see Figure 5(e)). The enzyme-inhibitor and enzyme-substrate complexes seem to have benefited the least from this filter.

Next we apply the antibody contact filter and the Glycine filter. The antibody contact filter improves the rank of the top hit for 9 antibody-antigen and antigen-bound antibody test cases, and degrades for 3, while the Glycine filter slightly improves the same for 4 enzyme-inhibitor/enzyme-substrate complexes (see Figure 5(f)).

More comparisons with respect to the RMSD of the best hit, the total number of hits, and the lowest RMSD are provided in the supplement.

3.10.3 Effects of Post-processing with GB-rerank

Figure 6 shows the impact of applying GB-rerank (after the initial docking phase) on the rigid-body test cases from Zlab benchmark 2.0 [36]. GB-rerank improves the ranks of the top hit for 9 antibody-antigen and antigen-bound antibody complexes, and 10 complexes of type ”other” (see Figure 6).

The post-processor is least effective on enzyme-inhibitor/enzyme-substrate complexes since the enzyme filter has already improved the ranks quite well. On the other hand, for the ‘other’ complexes, GB-rerank produces the most significant improvements, since specific filters cannot be applied in these cases. Hence if the complex is known to be Enzyme, then GB-rerank should not be applied.

3.10.4 Performance of F²Dock 2.0 with and without User-specified Complex Type

Figure 7 compares the performance of F²Dock 2.0 with and without user-specified complex types on Zlab’s protein-protein docking benchmark 2.0. When no complex type is specified F²Dock 2.0 tries to identify antibody-antigen complexes by locating the CDR loop regions of the antibody. Among the 17 such complexes in our experiments 16 are correctly identified by F²Dock 2.0. It fails to identify 1KXQ which is an antibody-antigen complex from a Camelid (camels, llamas, etc.) [60]. Camelids produce functional antibodies that do not have light chains and CH1 domains, and so F²Dock 2.0’s antibody detection system fails to identify such antibodies. Hence for 1KXQ the set of parameter values optimized for complexes of ”other” type is applied, and the result is only slightly worse than what is obtained with the parameter set optimized for antibody-antigen complexes. F²Dock 2.0 fails to select the correct parameter set for the following three enzyme-inhibitor/enzyme-substrate complexes among the 21 included in the experiments: 1AY7, 1UDI and 2MTA. While for 1UDI and 1AY7 F²Dock 2.0 is still able to get a hit in the top 100 and top 500, respectively, it fails to get any hit in the top 1000 for 2MTA. For all other complexes the
results remain the same except for 1WEJ for which we get slightly different results in the two set of experiments due to the non-determinism (arising from multiple concurrent threads) that exists in the proximity clustering phase.

3.11 Comparison with ZDock

In this section we compare the performance of F$^2$Dock 2.0 and ZDock 3.0.2 [21, 61] on the complexes from Zlab benchmark 4.0 [37]. We acquired the executable for ZDock 3.0.2 from their website and ran it following the steps specified in the accompanying instructions and used the PDB files downloaded from ZLab’s website without any modification. F$^2$Dock 2.0 used the same set of PDBs after performing the preprocessing we mentioned in Section 2.7. Note that ZDock 3.0.2 also applies their own preprocessing which is part of the mark.sur script provided with the executable. Both programs used 15° rotational sampling. F$^2$Dock 2.0 used user-specified complex types.

In Figure 8, we show a summary of the performances in terms of the number of complexes where each protocol found at least one hit in different ranges (see the X-axis). Note that having a higher Y-axis value for any instance shows that the corresponding protocol is successful on complexes than the other. In Figure 8(a) we compare the performances over the entire Zlab benchmark 4.0 containing 176 complexes. We find that for each of the ranges except one, F$^2$Dock 2.0 performs better than ZDock 3.0.2. F$^2$Dock 2.0 is specially impressive since it gets a hit at rank 1 for 22 of the complexes (which is 1/8th of the dataset) as opposed to 13 found by ZDock 3.0.2. Overall both ZDock 3.0.2 and F$^2$Dock 2.0 finds at least one solution for about the same number of complexes, 104 and 106 respectively.

Figures 8(b)-(d) compares F$^2$Dock 2.0 and ZDock 3.0.2 using the same metrics but considers each type of complex separately. For antibodies there is not much to choose between the two protocols. For other types F$^2$Dock 2.0 is successful for a lower number of complexes, and is comparable only at relatively high ranks. However, for Enzymes, F$^2$Dock 2.0 completely outperforms ZDock 3.0.2 across the board.

Based on these results, we can clearly see that F$^2$Dock 2.0 produces much more reliable predictions for Enzymes, but there is not much difference for antibodies and other type of complexes. But Tables 1 to 4 shows that even for antibodies and other types F$^2$Dock 2.0 provides significant contributions since the two protocols are often successful for different complexes and hence compliment each other. For example, among the antibodies, F$^2$Dock 2.0 finds a solution for 1QFW and 1I9R for which ZDock 3.0.2 does not find any solutions, on the other hand ZDock 3.0.2 finds a solution for 1NSN where F$^2$Dock 2.0 fails. Similarly among the other complexes, only F$^2$Dock 2.0 is successful for 1J2J, 2A5T, 2A9K, 2HQS, 3BP8, 1K5D, 1R6Q, 2Z0E, 3CPH and 1ATN. Hence, it is advisable to use both of these protocols specially for other type of complexes to increase the possibility of finding a correct solution.

Next, we compare the rate of success of the two protocols. Let us assume that the total number of hits (counting multiple hits found for a complex) found within a range [0, x] across all the complexes be $H(x)$. Now we define the rate of success as $y(x) = H(x) \times 100 / H(1000)$ which measures how quickly a protocol finds its hits. A protocol with a higher ratio has higher true positive rate near the top of the list. If we plot this function, we expect to see a curve which rises sharply and then gradually flattens and converges to $y(x) = 1.0$. In Figure 9, we see that F$^2$Dock 2.0 has consistently better success rate than ZDock 3.0.2.

A closer look at Tables 1 to 4 shows that the RMSDs of the predictions by F$^2$Dock 2.0 is poorer than ZDock 3.0.2 in more occasions than it is better. This is due to our softer skin approach which rewards docking poses which have slightly larger gap between them, and our stringent clash and VDW filters which discard ligand poses which comes too close. This is beneficial for unbound complexes with larger conformational change, but prevents ligands of rigid (easy cases in the benchmark) from getting as close as they could be placed. The result clearly shows that ZDock 3.0.2 gets better RMSDs for rigid cases, and F$^2$Dock 2.0 is better for non-rigid cases. At this point, it should be mentioned that F$^2$Dock 2.0 is designed solely as a initial stage docking tool, which can quickly perform exhaustive search and return good leads at high ranks. Hence the poses it finds are generally acceptable or medium quality as defined
in the criteria used in the CAPRI [62] challenge (tables summarizing F^2Dock 2.0’s performance using the CAPRI criteria can be found in the supplement). Local refinements (rigid body or flexible) can then be performed on a small number of top solutions to further improve their RMSDs and minimize the energies. There are a host of such tools available including ROSETTA [10], Amber [59], FireDock [63] etc. We conclude this section with the observation that F^2Dock 2.0 shows better overall performance, with significant improvement for Enzymes. For other type of complexes the performance is comparable and sometimes complementary.

3.12 Running Times

To evaluate the average running times and the relative consumption by each scoring term/filter we performed a set of experiments run on a 3 GHz 2×dual-core (i.e., 4 cores) AMD Opteron 2222 processor with 4 GB RAM. On average, the FFT phase took around 23 minutes or 35% of the total running time, the interface propensity filter took 20%, GB-rerank accounted for around 42%, and the remaining 3% is spent on the other filters. GB-rerank and interface propensity filter take longer to compute than other filters, since the computation is based on surface quadrature points, whose number is a constant multiple of the number of atoms. Figure 10 shows how the different components of F^2Dock 2.0 and GB-rerank contribute to the total running time of the docking and reranking process on the rigid-body test cases from Zlab benchmark 2.0 [36]. Overall, about 30% time is taken up by the FFT based affinity functions, 30% is taken up the the filters (mostly the interface propensity filter), and around 40% by the GB-rerank.

F^2Dock 2.0 leverages from the embarrassingly parallel nature of the computation using multithreaded computations on multi-core machines. Note that each of the N_R FFT computations are independent of each other and can be run in parallel. Scores for each of filter terms for each of the poses in Q can also be computed in parallel. Specifically, given q cores and T tasks the simplest strategy is to distribute T/q tasks to each cores. But this approach often leads to unbalanced exploitation of the cores if the tasks given to different cores take different amount of time to complete. For example, the running times of the filters are proportional to the size of the interface which varies between different poses. So our technique initially sends only one task to each core and maintains a queue of remaining tasks, and then whenever a core is done with its task, it gets another one from the queue. This scheduling ensures that every core is exploited equally and hence the overall completion time is quicker.

Conclusion

We have developed an enhanced version (F^2Dock 2.0) of our protein-protein docking program F^2Dock 2.0 with improved scoring functions, complete with dynamic clustering and filtering and generalized Born based solvation energetic reranking. The on-the-fly FFT-based scoring function is a weighted combination of shape-complementarity, Coulombic electrostatics complementarity, and interface propensity terms. The on-the-fly docking also includes filters based on Lennard-Jones potential, steric clashes, residue-residue contact statistics and an extremely fast approximation of solvation energy using a newly developed fast multipole type implementation with octree data structures. Our implementation results and numerous tests show that each of these terms and filters significantly improves the accuracy of docking predictions. Our use of highly efficient data structures including the dynamic packing grids for near constant time neighborhood search and near-far distance clustering using octrees, significantly speed up the computations for each of the 'on-the-fly' scoring and filtering terms. GB-rerank's solvation energy based post-processing suite is also optimized using these efficient data structures with the best tradeoffs of docking accuracy vs. speed. The entire software is highly parallel and can be run efficiently on multicores and clusters of multicores (e.g., many modern supercomputers). We have also developed a GUI based interface (TexMol) for easily preparing and running a docking process and interactively visualize,
compare different solutions along with several relevant statistics including interface area, residue contacts, binding energy etc.

4.13 Future work
Currently F²Dock 2.0 addresses flexibility by simply performing a ‘soft’ complementarity with the goal of identifying a near-native solution at a higher rank. The solutions can be optimized further by side chain refinement near the binding site as well as applying small rigid-body perturbations which moves all atoms of the ligand. There are several software including AMBER [59], ROSETTA [10], SCRWL [64] etc. which can be used to achieve this objective. We are currently working on an algorithm for dead-end elimination with better complexities compared to SCRWL. The automatic assignment of all the docking parameters remains a very active area of research for us; we are currently pursuing a semi-supervised computational learning algorithm over the space of different protein families which will further improve the performance of F²Dock 2.0 and GB-rerank. We are exploring ways to further improve the speed of F²Dock 2.0 specially using GPU level parallelism. We are also actively working on extending F²Dock 2.0 to Protein-RNA and RNA-RNA docking. Finally, a web service supporting similar features as TexMol is under construction.

5 Acknowledgments
The molecular visualizations shown in this paper were all produced via the latest version of our in house program TexMol [35].

References


### Table 1.

Comparison of the performance of F$^2$Dock 2.0 and ZDock 3.0.2 for each of the 25 antibody-antigen and antigen-bound antibody complexes from ZLab’s benchmark 4.0 in terms of the rank and RMSD of the top hit and the best hit. Boldfaced entries indicate better performance on the particular metric for the complex. Empty entries indicate that no hits were found for that complex by the corresponding protocol.

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Complex</th>
<th>Rank of First Hit</th>
<th>RMSD of First Hit</th>
<th>Rank of Lowest RMSD</th>
<th>Lowest RMSD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ZDock</td>
<td>F2Dock</td>
<td>ZDock</td>
<td>F2Dock</td>
</tr>
<tr>
<td>Easy</td>
<td>1AHW</td>
<td>354</td>
<td>8</td>
<td>4.5</td>
<td>4.4</td>
</tr>
<tr>
<td></td>
<td>1BJ1</td>
<td>1</td>
<td>64</td>
<td>1.9</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td>1BVK</td>
<td>184</td>
<td>205</td>
<td>3.6</td>
<td>4.9</td>
</tr>
<tr>
<td></td>
<td>1DQJ</td>
<td>374</td>
<td>74</td>
<td>4.0</td>
<td>4.9</td>
</tr>
<tr>
<td></td>
<td>1E6J</td>
<td>3</td>
<td>126</td>
<td>4.1</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>1FSK</td>
<td>1</td>
<td>1</td>
<td>2.9</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>1I9R</td>
<td>-</td>
<td>9</td>
<td>-</td>
<td>3.9</td>
</tr>
<tr>
<td></td>
<td>1IQD</td>
<td>18</td>
<td>4</td>
<td>4.3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>1JPS</td>
<td>1266</td>
<td>186</td>
<td>2.1</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td>1K4C</td>
<td>583</td>
<td>105</td>
<td>2.9</td>
<td>4.4</td>
</tr>
<tr>
<td></td>
<td>1KXQ</td>
<td>2</td>
<td>1</td>
<td>1.2</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td>1MLC</td>
<td>57</td>
<td>11</td>
<td>2.0</td>
<td>3.8</td>
</tr>
<tr>
<td></td>
<td>1NCA</td>
<td>11</td>
<td>168</td>
<td>1.7</td>
<td>3.7</td>
</tr>
<tr>
<td></td>
<td>1NSN</td>
<td>1267</td>
<td>-</td>
<td>1.6</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1QFW</td>
<td>-</td>
<td>120</td>
<td>-</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>1VFB</td>
<td>250</td>
<td>191</td>
<td>3.1</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>1WEJ</td>
<td>9</td>
<td>5</td>
<td>1.5</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>2FD6</td>
<td>3</td>
<td>62</td>
<td>5.0</td>
<td>4.4</td>
</tr>
<tr>
<td></td>
<td>2I25</td>
<td>2</td>
<td>122</td>
<td>3.0</td>
<td>3.9</td>
</tr>
<tr>
<td></td>
<td>2JEL</td>
<td>4</td>
<td>1</td>
<td>3.5</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>2VIS</td>
<td>4</td>
<td>1</td>
<td>3.5</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>9QFW</td>
<td>2</td>
<td>1</td>
<td>4.0</td>
<td>3.9</td>
</tr>
<tr>
<td>Medium</td>
<td>1BGX</td>
<td>4</td>
<td>126</td>
<td>1.7</td>
<td>3.7</td>
</tr>
<tr>
<td>Hard</td>
<td>1E4K</td>
<td>1</td>
<td>63</td>
<td>1.9</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td>2HMI</td>
<td>1</td>
<td>63</td>
<td>1.9</td>
<td>2.3</td>
</tr>
</tbody>
</table>

Table 1. Comparison of the performance of F$^2$Dock 2.0 and ZDock 3.0.2 for each of the 25 antibody-antigen and antigen-bound antibody complexes from ZLab’s benchmark 4.0 in terms of the rank and RMSD of the top hit and the best hit. Boldfaced entries indicate better performance on the particular metric for the complex. Empty entries indicate that no hits were found for that complex by the corresponding protocol.

Tables
<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Complex</th>
<th>Rank of First Hit</th>
<th>RMSD of First Hit</th>
<th>Rank of Lowest RMSD</th>
<th>Lowest RMSD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ZDock</td>
<td>F2Dock</td>
<td>ZDock</td>
<td>F2Dock</td>
</tr>
<tr>
<td>Easy</td>
<td>1AVX</td>
<td>25</td>
<td>1</td>
<td>3.5</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>1AY7</td>
<td>577</td>
<td>2</td>
<td>2.5</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1BVN</td>
<td>3</td>
<td>1</td>
<td>1.2</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>1CGI</td>
<td>10</td>
<td>76</td>
<td>4.0</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>1CLV</td>
<td>3</td>
<td>1</td>
<td>2.3</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td>1D6R</td>
<td>-2</td>
<td>59</td>
<td>4.6</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>1DFJ</td>
<td>9</td>
<td>1</td>
<td>4.1</td>
<td>4.4</td>
</tr>
<tr>
<td></td>
<td>1E6E</td>
<td>5</td>
<td>20</td>
<td>3.2</td>
<td>4.7</td>
</tr>
<tr>
<td></td>
<td>1EAW</td>
<td>68</td>
<td>1</td>
<td>3.4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1EWY</td>
<td>53</td>
<td>14</td>
<td>4.2</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>1EZU</td>
<td>841</td>
<td>170</td>
<td>4.9</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>1F34</td>
<td>62</td>
<td>2</td>
<td>3.4</td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td>1FLE</td>
<td>31</td>
<td>3</td>
<td>5.0</td>
<td>3.7</td>
</tr>
<tr>
<td></td>
<td>1GL1</td>
<td>73</td>
<td>326</td>
<td>2.6</td>
<td>3.8</td>
</tr>
<tr>
<td></td>
<td>1GX3</td>
<td>1173</td>
<td>4.9</td>
<td>1173</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1HIA</td>
<td>1</td>
<td>18</td>
<td>3.4</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>1JTG</td>
<td>1</td>
<td>7</td>
<td>2.6</td>
<td>4.6</td>
</tr>
<tr>
<td></td>
<td>1MAH</td>
<td>1</td>
<td>1</td>
<td>3.1</td>
<td>2.7</td>
</tr>
<tr>
<td></td>
<td>1NS0</td>
<td>7</td>
<td>11</td>
<td>3.4</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>1OC0</td>
<td>1590</td>
<td>-</td>
<td>4.8</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1OPH</td>
<td>1694</td>
<td>-</td>
<td>3.9</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1OYV</td>
<td>15</td>
<td>7</td>
<td>4.9</td>
<td>3.6</td>
</tr>
<tr>
<td></td>
<td>1PEP</td>
<td>1</td>
<td>2.9</td>
<td>2.3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>1ROI</td>
<td>138</td>
<td>39</td>
<td>2.2</td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td>1TMQ</td>
<td>16</td>
<td>1</td>
<td>3.6</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>1UDI</td>
<td>24</td>
<td>1</td>
<td>3.5</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>1YV8</td>
<td>1</td>
<td>2.4</td>
<td>2.8</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2ABZ</td>
<td>-5</td>
<td>-</td>
<td>2.8</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2B42</td>
<td>3</td>
<td>1</td>
<td>4.2</td>
<td>3.9</td>
</tr>
<tr>
<td></td>
<td>2J0T</td>
<td>-</td>
<td>19</td>
<td>2.8</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2MTA</td>
<td>76</td>
<td>90</td>
<td>4.4</td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td>2OSQ</td>
<td>29</td>
<td>654</td>
<td>5.0</td>
<td>3.7</td>
</tr>
<tr>
<td></td>
<td>2OUL</td>
<td>1</td>
<td>1</td>
<td>1.7</td>
<td>4.9</td>
</tr>
<tr>
<td></td>
<td>2PCC</td>
<td>496</td>
<td>10</td>
<td>2.6</td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td>2SIC</td>
<td>5</td>
<td>1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td></td>
<td>2SN1</td>
<td>177</td>
<td>1</td>
<td>3.8</td>
<td>4.7</td>
</tr>
<tr>
<td></td>
<td>2UU5</td>
<td>693</td>
<td>7</td>
<td>4.4</td>
<td>4.1</td>
</tr>
<tr>
<td></td>
<td>3SGQ</td>
<td>428</td>
<td>110</td>
<td>4.0</td>
<td>2.6</td>
</tr>
<tr>
<td></td>
<td>4CPA</td>
<td>1</td>
<td>1</td>
<td>4.4</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>7CEI</td>
<td>1</td>
<td>1</td>
<td>4.4</td>
<td>4.1</td>
</tr>
<tr>
<td></td>
<td>BOYV</td>
<td>-</td>
<td>220</td>
<td>3.6</td>
<td>-</td>
</tr>
<tr>
<td>Medium</td>
<td>1ACB</td>
<td>126</td>
<td>22</td>
<td>4.4</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>1IJK</td>
<td>81</td>
<td>88</td>
<td>3.0</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>1IIW</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1IKL</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1M10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1NW9</td>
<td>-</td>
<td>321</td>
<td>4.9</td>
<td>-</td>
</tr>
<tr>
<td>Hard</td>
<td>1F6M</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1FQ1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1PXV</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1ZLI</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2O3B</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2: Comparison of the performance of F²Dock 2.0 and ZDock 3.0.2 for each of the 52 enzyme-inhibitor and enzyme-substrate complexes from ZLab's benchmark 4.0 in terms of the rank and RMSD of the predicted structures.
<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Complex</th>
<th>Rank of First Hit</th>
<th>RMSD of First Hit</th>
<th>Rank of Lowest RMSD</th>
<th>Lowest RMSD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ZDock F2Dock ZDock F2Dock</td>
<td>ZDock F2Dock ZDock F2Dock</td>
<td>ZDock F2Dock ZDock F2Dock</td>
<td></td>
</tr>
<tr>
<td>Easy</td>
<td>1A2K</td>
<td>1348 44</td>
<td>4.3 2.4</td>
<td>1894 44</td>
<td>3.2 2.4</td>
</tr>
<tr>
<td></td>
<td>1AK4</td>
<td>1090 964</td>
<td>3.5 4.3</td>
<td>1090 964</td>
<td>3.5 4.3</td>
</tr>
<tr>
<td></td>
<td>1AKJ</td>
<td>546 39</td>
<td>2.9 3.4</td>
<td>1632 39</td>
<td>1.7 3.4</td>
</tr>
<tr>
<td></td>
<td>1AZS</td>
<td>42 -</td>
<td>2.9 -</td>
<td>61 -</td>
<td>2.0 -</td>
</tr>
<tr>
<td></td>
<td>1B6C</td>
<td>1 3</td>
<td>2.9 4</td>
<td>1 3</td>
<td>2.9 4</td>
</tr>
<tr>
<td></td>
<td>1BUH</td>
<td>30 431</td>
<td>3.6 4.8</td>
<td>1961 431</td>
<td>3.0 4.8</td>
</tr>
<tr>
<td></td>
<td>1E96</td>
<td>1171 278</td>
<td>3.8 5</td>
<td>1171 314</td>
<td>3.8 4.2</td>
</tr>
<tr>
<td></td>
<td>1EFN</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1F51</td>
<td>589 -</td>
<td>4.6 -</td>
<td>589 -</td>
<td>4.6 -</td>
</tr>
<tr>
<td></td>
<td>1FC2</td>
<td>- 1190</td>
<td>- 5</td>
<td>- 1570</td>
<td>- 4.1</td>
</tr>
<tr>
<td></td>
<td>1FCC</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1FFW</td>
<td>73 325</td>
<td>4.5 4.7</td>
<td>1349 1291</td>
<td>4.0 3</td>
</tr>
<tr>
<td></td>
<td>1FQJ</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1GCQ</td>
<td>1105 -</td>
<td>1.4 -</td>
<td>1105 -</td>
<td>1.4 -</td>
</tr>
<tr>
<td></td>
<td>1GHQ</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1GLA</td>
<td>1708 -</td>
<td>3.9 -</td>
<td>1708 -</td>
<td>3.9 -</td>
</tr>
<tr>
<td></td>
<td>1GPW</td>
<td>3 1</td>
<td>3.6 3.7</td>
<td>134 5</td>
<td>2.1 2.6</td>
</tr>
<tr>
<td></td>
<td>1H9D</td>
<td>1006 -</td>
<td>4.5 -</td>
<td>1006 -</td>
<td>4.5 -</td>
</tr>
<tr>
<td></td>
<td>1HCF</td>
<td>175 1225</td>
<td>4.0 4.7</td>
<td>225 1225</td>
<td>1.9 4.7</td>
</tr>
<tr>
<td></td>
<td>1HE1</td>
<td>1141 574</td>
<td>4.7 4.9</td>
<td>1141 574</td>
<td>4.7 4.9</td>
</tr>
<tr>
<td></td>
<td>1I4D</td>
<td>571 -</td>
<td>4.2 -</td>
<td>571 -</td>
<td>4.2 -</td>
</tr>
<tr>
<td></td>
<td>1J2J</td>
<td>- 182</td>
<td>- 4.6</td>
<td>- 182</td>
<td>- 4.6</td>
</tr>
<tr>
<td></td>
<td>1JWH</td>
<td>7 -</td>
<td>3.6 -</td>
<td>78 -</td>
<td>1.9 -</td>
</tr>
<tr>
<td></td>
<td>1K74</td>
<td>2 3</td>
<td>1.2 3.8</td>
<td>2 7</td>
<td>1.2 2.6</td>
</tr>
<tr>
<td></td>
<td>1KAC</td>
<td>502 8</td>
<td>4.5 4.4</td>
<td>1527 99</td>
<td>1.9 4.1</td>
</tr>
<tr>
<td></td>
<td>1KLU</td>
<td>1957 -</td>
<td>3.4 -</td>
<td>1957 -</td>
<td>3.4 -</td>
</tr>
<tr>
<td></td>
<td>1KTZ</td>
<td>535 98</td>
<td>2.8 3.9</td>
<td>535 166</td>
<td>2.8 2.9</td>
</tr>
<tr>
<td></td>
<td>1KXP</td>
<td>1 7</td>
<td>1.6 3.5</td>
<td>1 260</td>
<td>1.6 2.6</td>
</tr>
<tr>
<td></td>
<td>1ML0</td>
<td>4 2</td>
<td>3.1 4.3</td>
<td>8 123</td>
<td>3.1 3.2</td>
</tr>
<tr>
<td></td>
<td>1OFU</td>
<td>84 -</td>
<td>4.5 -</td>
<td>347 -</td>
<td>3.1 -</td>
</tr>
<tr>
<td></td>
<td>1PVH</td>
<td>748 -</td>
<td>4.5 -</td>
<td>1192 -</td>
<td>1.5 -</td>
</tr>
<tr>
<td></td>
<td>1QA9</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1RLB</td>
<td>3 555</td>
<td>4.6 5</td>
<td>232 555</td>
<td>3.4 5</td>
</tr>
<tr>
<td></td>
<td>1RV6</td>
<td>2 2</td>
<td>1.3 4</td>
<td>2 694</td>
<td>1.3 2.2</td>
</tr>
<tr>
<td></td>
<td>1S1Q</td>
<td>756 -</td>
<td>1.9 -</td>
<td>1243 -</td>
<td>1.4 -</td>
</tr>
<tr>
<td></td>
<td>1SBB</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1T6B</td>
<td>58 525</td>
<td>3.6 4</td>
<td>1510 752</td>
<td>2.8 2.7</td>
</tr>
<tr>
<td></td>
<td>1US7</td>
<td>74 -</td>
<td>1.1 -</td>
<td>74 -</td>
<td>1.1 -</td>
</tr>
<tr>
<td></td>
<td>1WDW</td>
<td>2 1</td>
<td>1.2 2.5</td>
<td>2 1</td>
<td>1.2 2.5</td>
</tr>
<tr>
<td></td>
<td>1XD3</td>
<td>8 1</td>
<td>4.0 4.2</td>
<td>86 1298</td>
<td>2.6 3.9</td>
</tr>
<tr>
<td></td>
<td>1XU1</td>
<td>912 -</td>
<td>5.0 -</td>
<td>912 -</td>
<td>5.0 -</td>
</tr>
<tr>
<td></td>
<td>1Z0K</td>
<td>8 307</td>
<td>3.3 3.3</td>
<td>8 307</td>
<td>3.3 3.3</td>
</tr>
<tr>
<td></td>
<td>1Z5Y</td>
<td>20 -</td>
<td>3.4 -</td>
<td>423 -</td>
<td>2.5 -</td>
</tr>
<tr>
<td></td>
<td>1ZHH</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1ZHI</td>
<td>65 202</td>
<td>4.4 4</td>
<td>324 202</td>
<td>2.1 4</td>
</tr>
<tr>
<td></td>
<td>2A5T</td>
<td>- 268</td>
<td>- 3.6</td>
<td>- 618</td>
<td>- 2.9</td>
</tr>
<tr>
<td></td>
<td>2A9K</td>
<td>- 558</td>
<td>- 3.4</td>
<td>- 558</td>
<td>- 3.4</td>
</tr>
<tr>
<td></td>
<td>2AJF</td>
<td>475 -</td>
<td>3.6 -</td>
<td>475 -</td>
<td>3.6 -</td>
</tr>
<tr>
<td></td>
<td>2AYO</td>
<td>37 1108</td>
<td>3.3 2</td>
<td>138 1108</td>
<td>2.5 2</td>
</tr>
<tr>
<td></td>
<td>2B4J</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>2BTF</td>
<td>53 95</td>
<td>4.7 4.5</td>
<td>148 377</td>
<td>3.8 3.4</td>
</tr>
<tr>
<td></td>
<td>2FJU</td>
<td>261 228</td>
<td>3.2 4.2</td>
<td>261 333</td>
<td>3.2 3.5</td>
</tr>
</tbody>
</table>

Table 3. Comparison of the performance of F2Dock 2.0 and ZDock 3.0.2 for each of the 99 other type of complexes from ZLab's benchmark 4.0 in terms of the rank and RMS D of the top hit and the best.
<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Complex</th>
<th>Rank of First Hit</th>
<th>RMSD of First Hit</th>
<th>Rank of Lowest RMSD</th>
<th>Lowest RMSD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ZDock F2Dock</td>
<td>ZDock F2Dock</td>
<td>ZDock F2Dock</td>
<td>ZDock F2Dock</td>
</tr>
<tr>
<td>Easy</td>
<td>2G77</td>
<td>15 8</td>
<td>1.5 3.7</td>
<td>15 917</td>
<td>1.5 1.3</td>
</tr>
<tr>
<td></td>
<td>2HLE</td>
<td>31 4</td>
<td>4.1 3.8</td>
<td>31 4</td>
<td>4.1 3.8</td>
</tr>
<tr>
<td></td>
<td>2HQS</td>
<td>- 27</td>
<td>- 4.1</td>
<td>- 125</td>
<td>- 2.7</td>
</tr>
<tr>
<td></td>
<td>2O0B</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>2O0R</td>
<td>766 16</td>
<td>4.4 4</td>
<td>766 63</td>
<td>4.4 2</td>
</tr>
<tr>
<td></td>
<td>2VDB</td>
<td>5 -</td>
<td>1.2 -</td>
<td>5 -</td>
<td>1.2 -</td>
</tr>
<tr>
<td></td>
<td>3BP8</td>
<td>- 474</td>
<td>- 5</td>
<td>- 699</td>
<td>- 3.3</td>
</tr>
<tr>
<td></td>
<td>3D5S</td>
<td>71 1</td>
<td>3.1 3.2</td>
<td>609 4</td>
<td>2.5 2.7</td>
</tr>
<tr>
<td></td>
<td>1GP2</td>
<td>61 193</td>
<td>4.4 4</td>
<td>107 193</td>
<td>2.8 4</td>
</tr>
<tr>
<td></td>
<td>1GRN</td>
<td>1299 401</td>
<td>4.3 4.8</td>
<td>1299 401</td>
<td>4.3 4.8</td>
</tr>
<tr>
<td></td>
<td>1HE8</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1I2M</td>
<td>267 545</td>
<td>2.2 2.6</td>
<td>267 545</td>
<td>2.2 2.6</td>
</tr>
<tr>
<td></td>
<td>1IB1</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1K5D</td>
<td>- 521</td>
<td>- 4.3</td>
<td>- 521</td>
<td>- 4.3</td>
</tr>
<tr>
<td></td>
<td>1LFD</td>
<td>85 990</td>
<td>4.6 4.6</td>
<td>466 1235</td>
<td>4.5 4.1</td>
</tr>
<tr>
<td></td>
<td>1MQ8</td>
<td>1455 -</td>
<td>3.2 -</td>
<td>1455 -</td>
<td>3.2 -</td>
</tr>
<tr>
<td></td>
<td>1N2C</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1R6Q</td>
<td>- 180</td>
<td>- 3.7</td>
<td>- 311</td>
<td>- 3.5</td>
</tr>
<tr>
<td></td>
<td>1SYX</td>
<td>211 2</td>
<td>4.8 4.7</td>
<td>211 11</td>
<td>4.8 3</td>
</tr>
<tr>
<td></td>
<td>1WQ1</td>
<td>81 -</td>
<td>4.0 -</td>
<td>81 -</td>
<td>4.0 -</td>
</tr>
<tr>
<td></td>
<td>1XQS</td>
<td>19 61</td>
<td>3.8 4.2</td>
<td>45 833</td>
<td>2.6 3.7</td>
</tr>
<tr>
<td></td>
<td>1ZM4</td>
<td>6 -</td>
<td>4.1 -</td>
<td>631 -</td>
<td>2.7 -</td>
</tr>
<tr>
<td></td>
<td>2CFH</td>
<td>1 119</td>
<td>3.8 2.6</td>
<td>2 119</td>
<td>1.7 2.6</td>
</tr>
<tr>
<td></td>
<td>2H7V</td>
<td>1112 -</td>
<td>4.6 -</td>
<td>1112 -</td>
<td>4.6 -</td>
</tr>
<tr>
<td></td>
<td>2HRK</td>
<td>3 -</td>
<td>3.7 -</td>
<td>3 -</td>
<td>3.7 -</td>
</tr>
<tr>
<td></td>
<td>2J7P</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td>Hard</td>
<td>2NZ8</td>
<td>64 -</td>
<td>4.5 -</td>
<td>64 -</td>
<td>4.5 -</td>
</tr>
<tr>
<td></td>
<td>2OZA</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>2Z0E</td>
<td>- 169</td>
<td>- 3.9</td>
<td>- 169</td>
<td>- 3.9</td>
</tr>
<tr>
<td></td>
<td>3CPH</td>
<td>- 250</td>
<td>- 4.3</td>
<td>- 250</td>
<td>- 4.3</td>
</tr>
<tr>
<td></td>
<td>1ATN</td>
<td>- 1307</td>
<td>- 2.7</td>
<td>- 1307</td>
<td>- 2.7</td>
</tr>
<tr>
<td></td>
<td>1BKD</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1DE4</td>
<td>84 -</td>
<td>4.7 -</td>
<td>84 -</td>
<td>4.7 -</td>
</tr>
<tr>
<td></td>
<td>1EER</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1FAK</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1H1V</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1IBR</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1IRA</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1JK9</td>
<td>510 422</td>
<td>4.2 2.5</td>
<td>790 422</td>
<td>4.1 2.5</td>
</tr>
<tr>
<td></td>
<td>1JK9</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1LZD</td>
<td>44 144</td>
<td>4.6 4</td>
<td>44 144</td>
<td>4.6 4</td>
</tr>
<tr>
<td></td>
<td>1R8S</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>1Y64</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>2C0L</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>2IDo</td>
<td>130 156</td>
<td>3.6 4.5</td>
<td>154 156</td>
<td>3.5 4.5</td>
</tr>
<tr>
<td></td>
<td>2OT3</td>
<td>121 -</td>
<td>4.6 -</td>
<td>327 -</td>
<td>4.5 -</td>
</tr>
</tbody>
</table>

Table 4. Comparison of the performance of F2Dock 2.0 and ZDock 3.0.2 for each of the 99 other type of complexes from ZLab's benchmark 4.0 in terms of the rank and RMSD of the top hit and the best hit.
Figures

Figure 1. High-level overview of rigid-body protein-protein docking using F$^2$Dock 2.0 and GB-rerank. F$^2$Dock 2.0 performs exhaustive 6D search in discretized rotational and translational space where it computes a score for each sampled orientation of the ligand with respect to a stationary receptor. The scoring function is a weighted combination of shape complementarity, electrostatics and interface propensity based affinity terms (refer to Section 2.3 for details). The top few orientations (poses) of the ligand are kept in a priority queue. Then top several thousand poses from the queue are clustered based on the distance between the geometric centers of different poses of $B$. All but the best scoring pose of a cluster is penalized by reducing the score. The resulting reordered list is then passed through several soft filters in order to further penalize potential false positives. Finally, as a separate post-processing step, the ranked docking poses are re-scored and reranked based on the change in solvation energy caused by each pose.

Figure 2. Definition of skin and core for shape complementarity. (Left) Traditional double skin-layer approach for shape complementarity, (Right) Improved approach with curvature-based weighting of skin atoms and depth dependent weighting of core atoms of molecule $A$, and depth dependent weighting of the atoms of $B$.

Figure 3. Effectiveness of the New Skin-Core Definition. (a-b) Comparison of the performance of F$^2$Dock 2.0’s shape complementarity function with traditional skin and the new floating skin approach, in terms of the number of complexes for which the top hit is within the ranges mentioned in the X-axis. (a) On the rigid-body unbound-(un)bound complexes from Zlab Benchmark 4.0. (b) On the rigid-body bound-bound complexes from Zlab Benchmark 4.0. (c) Comparison of the shape complementarity functions of DOT, ZDock 2.1 and F$^2$Dock 2.0 on the rigid-body bound-bound complexes from Zlab benchmark 2.0.

Figure 4. Analysis of the efficacy of the different filters and affinity terms used in F$^2$Dock 2.0. (top) Improvements in the rank of the top hit (of rigid-body test cases from Zlab benchmark 4.0) as various affinity functions and filters in F$^2$Dock 2.0 are activated one after another. (bottom) Improvements in the rank of the top hit for the Enzyme type of complexes from Zlab benchmark 4.0.
Figure 5. Changes in the rank of top hit as various options in F$^2$Dock 2.0 are activated one after another (on the rigid-body test cases from Zlab benchmark 2.0 [36]). (a) Lennard-Jones Filter (LJ), clash filter (CL) and proximity clustering (PC) are activated after shape complementarity (SC). (b) Electrostatics & charge complementarity (EL) after SC+LJ+CL+PC, (c) interface propensity (IP) after SC+LJ+CL+PC+EL. (d) Interface propensitly filter (PF) after SC+LJ+CL+PC+EL+IP, (e) Residue-residue contact filter (RC) after SC+LJ+CL+PC+EL+IP+PF, and (f) Antibody contact filter (AF) or glycine filter (GF) after SC+LJ+CL+PC+EL+IP+PF+RC.

Figure 6. Effect of performing GBSA based reranking. The plot shows the change of the rank for the first hit. A positive change indicates that the reranker improves the result. For most complexes, specially for complexes where a knowledge-based based filter (Antibody or Enzyme) could not be applied, GB-rerank improves the rank of top hit compared to the results produced by F$^2$Dock 2.0 (for the rigid-body test cases from Zlab benchmark 2.0 [36]).

Figure 7. Performance of F$^2$Dock 2.0 with and without user-specified complex type. When complex type is not specified in the input, F$^2$Dock 2.0's performance does not change significantly. In most cases, it can automatically detect the complex-type and apply the correct set of parameters. Tests are based on rigid body cases from Zlab's Protein-protein docking Benchmark 2.0.

Figure 8. Comparison of ZDock 3.0.2 [21] and F$^2$Dock 2.0. (a) On all 176 complexes from Zlab Benchmark 4.0 [37], (b) On 25 antibody-antigen and antigen-bound antibody complexes, (c) On 52 enzyme-inhibitor and enzyme-substrate complexes, and (d) on the 99 other type of complexes.

Figure 9. Comparison of the rate of success of F$^2$Dock 2.0 and ZDock 3.0.2. On the 176 complexes from ZLab's benchmark 4.0. Rate of success is defined as the percentage of the hits found within the top x ranks, where x is the corresponding value of the X-axis. Clearly F$^2$Dock 2.0 has a better ratio.

Figure 10. Running time of F$^2$Dock 2.0 and its components. (a) Average running time of each affinity function and filter of F$^2$Dock 2.0. GB-rerank consumes a major portion of the time (42%), the FFT phase takes about 30% time and the rest is taken by the filters and clustering. The labels in the figure are actual time in minutes. (b) Running times of F$^2$Dock 2.0 on the rigid-body test cases from Zlab benchmark 2.0 [36] showing percentage of running time due to each affinity function and filter of F$^2$Dock 2.0 for each complex.
Protein-Protein Docking with F²Dock 2.0 and GB-rerank

Supplement

Rezaul Chowdhury¹, Muhibur Rasheed¹, Donald Keidel², Maysam Moussalem¹, Arthur Olson², Michel Sanner², Chandrajit Bajaj²,*

1 Department of Computer Science, University of Texas at Austin, 1 University Station C0500, Austin, Texas 78712, USA
2 The SCRIPPS Research Institute, Torrey Pines Rd N, La Jolla, CA 92037, USA
* E-mail: bajaj@cs.utexas.edu

Contents

1 Details on Affinity Function Computations ........................................... 2
  1.1 Shape Complementarity ................................................................. 2
  1.1.1 Skin-core Definition and Weighting ........................................... 2
  1.1.2 FFT based formulation ............................................................. 4
  1.2 Electrostatics (E) ................................................................. 5
  1.2.1 FFT based formulation ............................................................. 5
  1.3 Interface Propensity (IP) and Hydrophobicity (HP) ................................. 6
  1.3.1 FFT based formulation ............................................................. 6

2 Details on Filters ........................................................... 8
  2.1 Lennard-Jones Filter ................................................................. 8
  2.2 Interface Area Filter ................................................................. 9
  2.3 Interface Propensity Filter ................................................................ 9
  2.4 Clash Filter ................................................................................. 10
  2.5 Residue-Residue Contact Filter ..................................................... 10
  2.6 Glycine Filter .............................................................................. 11
  2.7 Antibody-Antigen Contact Filter ..................................................... 11
  2.8 Overall Cost of Filtering ............................................................... 11

3 Solvation Energy Based Reranking ..................................................... 12
  3.1 Approximating $E_{pol}$ (Polarization Energy) ...................................... 12
  3.2 Approximating $E_{vdw(s-s)}$ (Dispersion Energy) ................................ 13
  3.3 Approximating $E_{cav}$ (Cavity Forming Energy) .................................. 13
  3.4 Overall Cost of Reranking .................................................................. 14

4 Additional Results on the Effects of Various Affinity Functions and Filters ....... 15

5 Additional Results Comparing ZDock 3.0.2 and F²Dock 2.0 .......................... 25
1 Details on Affinity Function Computations

This section adds detail on the FFT based implementation and parameter selection of three affinity functions: shape complementarity, electrostatics and interface propensity (or hydrophobicity).

1.1 Shape Complementarity

First we present the new skin-core definitions in detail.

1.1.1 Skin-core Definition and Weighting

- **Floating Receptor (i.e., Stationary Molecule A) Skin:** Unlike the traditional approach [1] the receptor skin layer does not touch the receptor van der Waals surface (i.e., there is a gap between the skin and the core regions), and also the radius of those skin atoms differ from that in the traditional approach. The plots in Figure S2 justify this skin-core gap which, for each of 60 rigid-body test cases chosen from Zlab Benchmark Suite 2.0 [2], plots average distance and standard deviation of ligand atom centers from the receptor van der Waals surface when the ligand is bound to the receptor. We have considered only those ligand atoms that lie within 3Å from the receptor surface in bound condition. For the bound-bound case (when both molecules are crystallized together in the complex form) the average distance lies between 1.6Å and 2.0Å, while for unbound-unbound case (when the two molecules are crystallized separately) it lies between 1.4Å and 2.0Å most of the time. The standard deviation lies between 0.55Å and 0.75Å for the bound-bound case, and between 0.6Å and 1Å for the unbound-unbound case. In our experiments pseudo skin atoms of radius 1.1Å with their centers placed at a distance of 1.7Å from the receptor vdW surface (so there is a gap of 0.6Å) performed much better than the traditional approach where pseudo atoms of radius 1.4Å are placed touching the receptor vdW surface.

Figure S1: **Traditional vs improved double-skin definitions.** (Left) Traditional double skin-layer approach for shape complementarity, (Right) Curvature-based weighting of skin atoms and depth dependent weighting of core atoms of molecule A, and depth dependent weighting of the atoms of B

1.1.1 Skin-core Definition and Weighting

- **Floating Receptor (i.e., Stationary Molecule A) Skin:** Unlike the traditional approach [1] the receptor skin layer does not touch the receptor van der Waals surface (i.e., there is a gap between the skin and the core regions), and also the radius of those skin atoms differ from that in the traditional approach. The plots in Figure S2 justify this skin-core gap which, for each of 60 rigid-body test cases chosen from Zlab Benchmark Suite 2.0 [2], plots average distance and standard deviation of ligand atom centers from the receptor van der Waals surface when the ligand is bound to the receptor. We have considered only those ligand atoms that lie within 3Å from the receptor surface in bound condition. For the bound-bound case (when both molecules are crystallized together in the complex form) the average distance lies between 1.6Å and 2.0Å, while for unbound-unbound case (when the two molecules are crystallized separately) it lies between 1.4Å and 2.0Å most of the time. The standard deviation lies between 0.55Å and 0.75Å for the bound-bound case, and between 0.6Å and 1Å for the unbound-unbound case. In our experiments pseudo skin atoms of radius 1.1Å with their centers placed at a distance of 1.7Å from the receptor vdW surface (so there is a gap of 0.6Å) performed much better than the traditional approach where pseudo atoms of radius 1.4Å are placed touching the receptor vdW surface.
Figure S2: **Interface gap for bound and unbound complexes in benchmark 4.0.** Plots show average distance (in red) and standard deviation of ligand atom centers from the receptor van der Waals surface when the ligand is bound to the receptor. Only the ligand atoms that lie within 3Å from the receptor surface in bound condition are considered. The plots are for the rigid-body test cases from Zlab Benchmark Suite 2.0 [2]. Top: Bound-bound. Bottom: Unbound-unbound

- **Curvature dependent Weighting of Receptor Skin:** The weights assigned to receptor skin atoms are computed based on the curvature of the skin around that atom (as opposed to
the constant weights assigned to skin atoms in the traditional approach [1]). Such weighting encourages convex-concave and concave-convex interfaces as opposed to large flat interfaces.

We know that the centers of molecule A’s skin atoms are placed at a constant distance \(d\) from the receptor vdW surface \(S\). Let us enlarge \(S\) outward by distance \(d\) so that it passes through the skin atom centers, and let \(V\) be the volume enclosed by the enlarged surface. Let \(B_k\) be a ball of radius \(\delta\) at the center of skin atom \(k\), and \(B_{in}^k = B \cap V\) and \(B_{out}^k = V \setminus B_{in}^k\). We define:

\[
\rho(k) = \max\left\{ \frac{|B_{in}^k|}{|B_{in}^k|}, \frac{|B_{out}^k|}{|B_{out}^k|} \right\}.
\]

Then the weight assigned to skin atom \(k\) of molecule A is \(c_{SC}_{A,k} = \rho(k)\).

We approximate \(\rho(k)\) from the 3D grid embedding of molecule A along with its skin region, and \(\delta = 3.6\) produced good results in our experiments.

- **Depth dependent Weighting of Receptor Core:** The core atoms of molecule A are assigned weights using an increasing function of depth (distance of the atom center from the surface of A). Such weightings discourage deeper core-core overlaps more compared to shallower ones by making it difficult to compensate for deep core-core overlaps with wide skin-skin overlaps.

The core atom weights are computed based on atomic layers. Any atom which contributes to the solvent excluded surface (SES) is considered an exposed atom. The exposed atoms of a molecule lie in layer 1. When atoms in layer 1 are removed, the new exposed atoms make layer 2. In general, when atoms in layers 1 to \(l \geq 1\) are removed, the exposed atoms among the remaining ones make layer \(l + 1\). The atomic layer of core atom \(k\) of A is denoted by \(L_A(k)\), and the weight assigned to that atom is \(c_{SC}_{A,k} = i \cdot \alpha^{L_A(k)}\), where \(\alpha \geq 1\) is a constant. A value of \(\alpha = 1.1\) worked quite well in our experiments.

Hence overall, the weight assigned to atom/pseudo-atom \(k\) of molecule A:

\[
c_{SC}_{A,k} = \begin{cases} 
\rho(k) & \text{if atom } k \in \text{skin}(A), \\
i \cdot \alpha^{L_A(k)} & \text{otherwise}.
\end{cases}
\]

- **Thicker Skin of the Ligand (i.e., Moving Molecule B) with Skin-Core Overlap:** Since in the traditional approach [1] the ligand skin is defined using its surface atoms, the skin thickness varies and can be too thin in some areas. It turns out that a double layer ligand skin with skin-core overlap performs better than a single skin layer disjoint from the core. The weights assigned to the atoms of B are as follows.

\[
c_{SC}_{B,k} = \begin{cases} 
1 & \text{if } L_B(k) = 1, \\
1 + i & \text{if } L_B(k) = 2, \\
i & \text{otherwise}.
\end{cases}
\]

### 1.1.2 FFT based formulation

Now suppose the user-specified relative weights for various types of overlaps are as follows:

\[
w_{ss} = \text{reward for (unit) skin-skin overlap},
\]

\[
w_{sc} = \text{reward for (unit) skin-core overlap},
\]

\[
w_{cc} = \text{reward for (unit) core-core overlap}.
\]
\[ w_{cc} = \text{penalty for (unit) core-core overlap}, \text{ and} \]
\[ w_{sc} = \text{reward/penalty for (unit) skin-core overlap}. \]

Let \( A' \) denote molecule \( A \) with its grown skin layer. If the weight assigned to an atom/pseudo-atom \( k \) (of molecule \( P \in \{ A', B \} \) ) is \( c_k^{SC} = c_k^{SC, Re} + i \cdot c_k^{SC, Im} \), then the affinity function for shape complementarity is:

\[
f_{P}^{SC}(x) = \sum_{k \in P} \left( \sqrt{w_{ss}} \cdot c_k^{SC, Re} + i \cdot \sqrt{w_{cc}} \cdot c_k^{SC, Im} \right) \cdot g_{P,k}^{SC}(x),
\]

where, \( g_{P,k}^{SC}(x) = e^{-\beta \left( (x-c_k)^2 \gamma^2 - 1 \right)} \) with blobbiness \( \beta = 2.3. \)

The overall shape complementarity score for translation \( t \) and rotation \( r \) is:

\[
H_{A,B}^{SC}(t, r) = \text{Re} \left( F_{A,B}^{SC}(t, r) \right) + \frac{w_{sc}}{w_{ss} \cdot w_{cc}} \cdot \text{Im} \left( F_{A,B}^{SC}(t, r) \right),
\]

where, \( F_{A,B}^{SC}(t, r) = \int_x f_{A}^{SC}(x) T_t (\Delta_r (f_{B}^{SC}(x))) \) with \( T \) and \( \Delta \) being the translation and the rotation operator, respectively.

### 1.2 Electrostatics (E).

In [3] Gabb et. al. described a simplified model for electrostatics which allows efficient FFT-based computation of the term during docking search. We used this simple electrostatics model in the original version of F^2Dock 2.0 [1]. The first protein’s electric potential is computed and matched against the charges in the other. Charge assignments are made using PDB2PQR [4]).

Two affinity functions \( f_{A}^{E} \) and \( f_{B}^{E} \) are defined for molecule \( A \) and \( B \), respectively.

\[
f_{A}^{E}(x) = \sum_{k \in A} q_k \frac{E(x-c_k)(x-c_k)}{E(x-c_k)(x-c_k)} \cdot g_{A,k}^{E}(x)
\]

and

\[
f_{B}^{E}(x) = \sum_{k \in B} q_k \delta(x-c_k) \cdot g_{B,k}^{E}(x),
\]

where, \( q_k \) is the Coulombic charge on atom \( k \), \( \delta(x) \) is the Kronecker delta function with value 1 at \( ||x|| = 0 \), and 0 everywhere else, \( g_{A,k}^{E}(x) = g_{B,k}^{E}(x) = 1 \) and \( E(x) \) is the distance dependent dielectric constant [3] as given below.

\[
E(x) = \begin{cases} 
4 & \text{if } ||x|| \leq 6 \text{Å}, \\
80 & \text{if } ||x|| > 8 \text{Å}, \\
38 \cdot ||x|| - 224 & \text{otherwise}.
\end{cases}
\]

#### 1.2.1 FFT based formulation

In the current version of F^2Dock 2.0, we define for \( P \in A, B \),

\[
g_{P,k}^{E}(x) = e^{-\beta \left( (x-c_k)^2 \gamma^2 - 1 \right)},
\]
where, $\gamma > 0$ is a constant, and blobbiness $\beta = 2.3$. In our experiments the new definition of $g^E_{P,k}(x)$ with $\gamma = 3.4$ performed much better than the original Gabb et al. formulation. Smoothing with the Gaussian function as above has the effect of reduced discretization error on the grid.

The overall electrostatics score for translation $t$ and rotation $r$ is:

$$H^E_{A,B}(t, r) = w_E \cdot Re\left(F^E_{A,B}(t, r)\right),$$

where, $F^E_{A,B}(t, r) = \int_x f^E_A(x)T_t \left(\Delta_r\left(f^E_B(x)\right)\right)$, and $w_E$ is the user-specified weight given to electrostatics interaction.

### 1.3 Interface Propensity (IP) and Hydrophobicity (HP).

F$^2$Dock 2.0 scores the interfaces between molecules $A$ and $B$ using the per-residue interface propensity values computed in [5] based on the relative frequencies of different residues in the interfaces of a set of 63 protein-protein complexes from [6]. Let $IP(R)$ denote the natural logarithm of the interface propensity value of a residue $R$. Alternatively, the user can choose to use per-residue Hydrophobicity values from [7]. Note that, Hydrophobic residues are more likely to be found on the interface, and hence Hydrophobicity and Interface propensity model the same chemical property and hence are used as substitutes in F$^2$Dock 2.0, as opposed to using both at the same time.

The $IP$ values for the 20 amino acid residues lie between -0.38 (for ASP) and 0.83 (for TRP). A residue with a higher $IP$ value is likely to occur more frequently in a protein-protein interface than one with a lower $IP$ value. Hydrophobic residues Phe, Met, Ile, Leu and Val, polar aromatic residues Trp, Tyr and His, and the charged residue Arg favor interface locations, and they have positive $IP$ values. Additionally, Cys and Asn also have positive $IP$. The $IP$ value of an atom is set to the $IP$ value of the residue it belongs to.

Suppose $A + B_{t,r}$ is a docking pose obtained by rotating molecule $B$ by $r$ and translating by $t$. Let $iAtom^+_{t,r}(P)$ and $iAtom^-_{t,r}(P)$ denote the set of atoms in the interface of $P \in \{A, B\}$ in this docking pose that have positive and negative $IP$ values, respectively. Also let $iAtom^s_{t,r}(A, B) = iAtom^+_{t,r}(A) + iAtom^-_{t,r}(B)$, for $s \in \{+, -\}$. Then we assign the following interface propensity score to the docking pose:

$$IP-score_{t,r}(A, B) = \frac{-\sum_{a \in iAtom^+_{t,r}(A, B)} IP(a)}{\min(IP_A, \sum_{a \in iAtom^-_{t,r}(A, B)} IP(a))},$$

where $IP_t = \max_{IP(R) < 0} IP(R)$.

#### 1.3.1 FFT based formulation

We show below how we approximate $IP-score_{t,r}(A, B)$. For this approximation, all atoms of $A$ and $B$ are treated as solid spheres.

For $Q \in \{A, B\}$, we define two affinity functions $f^{P,Q}_A$ and $f^{P,Q}_B$ for $A$ and $B$, respectively, and evaluate $F^{IP,Q}_{A,B}(t, r) = \int_x f^{P,Q}_A(x)T_t \left(\Delta_r\left(f^{P,Q}_B(x)\right)\right)$. Let $Q' \in \{A, B\} \setminus \{Q\}$.

We set the radius of each atom of $Q$ to its van der Waals radius, while the van der Waals radius of each atom of $Q'$ is extended by a non-negative constant $r_{ext}^1$.

We assign weight $c^{IP,Q}_{P,k}$ to each atom $k \in P = \{Q, Q'\} = \{A, B\}$ as follows:

$^1r_{ext} = 1.5\text{Å}$ produced good results in our experiments.
\[ c_{Q,k}^{IP} = \begin{cases} IP(k) & \text{if } k \text{ surface atom, } IP(k) > 0, \\ -i \cdot IP(k) & \text{if } k \text{ surface atom, } IP(k) < 0, \\ 0 & \text{otherwise.} \end{cases} \]

\[ c_{Q',k}^{IP} = \begin{cases} 1 & \text{if } k \text{ is a surface atom,} \\ 0 & \text{otherwise.} \end{cases} \]

Then we define:

\[ f_{IP,Q}(x) = \sum_{k \in Q} c_{Q,k}^{IP} \cdot g_{Q,k}^{IP}(x) \]

where \( g_{Q,k}^{IP}(x) = 1 \) if \( |x - c_k| \leq r_k \) and 0 otherwise; and

\[ f_{IP,Q'}(x) = \bigvee_{k \in Q'} c_{Q',k}^{IP} \cdot g_{Q',k}^{IP}(x) \]

where \( g_{Q',k}^{IP}(x) = 1 \) if \( |x - c_k| \leq r_k + r_{ext} \) and 0 otherwise.

The overall interface propensity score for translation \( t \) and rotation \( r \) (i.e., approximated IP-score \( t,r(A,B) \)) is:

\[ H_{A,B}^{IP}(t,r) = w_{IP} \cdot \frac{\text{Re} \left( F_{A,B}^{IP,A}(t,r) + F_{A,B}^{IP,B}(t,r) \right)}{IP + \text{Im} \left( F_{A,B}^{IP,A}(t,r) + F_{A,B}^{IP,B}(t,r) \right)}, \]

where, \( w_{IP} \) is the user-specified weight given to interface propensity.
2 Details on Filters

In this section we describe the implementations of various filters available in F²Dock 2.0. Each filter has a preprocessing step (e.g., construction of octrees and initialization of other relevant data structures) which is executed only once before F²Dock 2.0 enters the rotations loop in phase I, and so not included in the running times of the filters which are executed in each iteration of the loop. Also octrees constructed for the initial configuration of molecule B can be easily transformed on-the-fly so that it correctly matches any rotated configuration of B by applying the corresponding rotation matrix on the node coordinates (e.g., node center) and atoms/points stored in them.

2.1 Lennard-Jones Filter.

We approximate the Lennard-Jones ($\text{LJ}$) potential between molecules $A$ and $B_{t,r}$ given by the following expression.

$$LJ(A, B_{t,r}) = \sum_{i \in A, j \in B_{t,r}} \left( \frac{a_{ij}}{r_{ij}^{12}} - \frac{b_{ij}}{r_{ij}^{6}} \right),$$

where $r_{ij}$ is the distance between atoms $i \in A$ and $j \in B_{t,r}$, constants $a_{ij}$ and $b_{ij}$ depend on the type (e.g., C, H, O, etc.) of the two atoms involved. For any fixed pair of atom types $a_{ij}$ and $b_{ij}$ are fixed, and are calculated from the Amber force field using well depths $\mu$ and equivalence contact distances of homogeneous pairs $r_{eqm}$ as follows (assuming $X = \text{atomType}(i \in A)$ and $Y = \text{atomType}(j \in B_{t,r})$) [8, 9].

$$a_{ij} = \mu_{XY} r_{eqm,XY}^{12}, \quad b_{ij} = 2\mu_{XY} r_{eqm,XY}^{6},$$

$$\mu_{XY} = \sqrt{\mu_{XX} + \mu_{YY}}, \quad r_{eqm,XY} = \frac{1}{2}(r_{eqm,XX} + r_{eqm,YY}).$$

We assume that $X, Y \in \{C, H, N, O, P, S\}$.

Our approach is to identify potential docking solutions based on a cutoff potential. Assuming that the docking solution can not have steric clashes at the interface, we can filter (or penalize) all poses with positive $LJ$ potential. However, though this assumption is not unreasonable for bound-bound docking, in the case of bound-unbound and unbound-unbound docking we need to allow some steric clashes. Hence we soften the potential by reducing the $r_{eqm,XX}$ values by a constant factor $\delta_{LJ}$ which has the effect of reducing the inter-atomic clash distances\(^2\) by the same factor. As a result docking poses with negative $LJ$ values can now have some clashes, and the maximum number of allowed clashes will depend on the value of $\delta_{LJ}$. In our experiments $\delta_{LJ} = 0.3$ produced good results.

Observe that direct computation of $LJ(A, B_{t,r})$ requires $O(M_A M_B)$ time, where $M_A$ (resp. $M_B$) is the number of atoms in molecule $A$ (resp. $B$). Instead we use our octree-based fast multipole-type algorithm described in [10] to obtain a $1 + \epsilon$ approximation of the potential in $O \left( \frac{1}{\epsilon} (M_A + M_B) \log (M_A + M_B) \right)$ time and $O (M_A + M_B)$ space, where $\epsilon \in (0, 1]$ is a user-defined approximation parameter.

\(^2\)i.e., $r_{ij}$ values at which two atoms are considered to be in a clash
2.2 Interface Area Filter.

We sample weighted quadrature/integration points from the surface of each molecule as described in [11, 10]. Sum of the weights of all such integration points is equal to the molecular surface area up to quadrature approximation error, and each integration point can be treated as the center of a small surface patch with its weight being the patch area. Each such patch \( \pi \) with center \( c_\pi \) and area \( a_\pi \) is represented by the tuple \(< c_\pi, a_\pi \>\).

We then evaluate the following expression as an approximation of the interface area between molecules \( A \) and \( B_{t,r} \).

\[
\text{InterfaceArea}(A, B_{t,r}) = \sum_{\pi \in IP_A(B_{t,r}) \cup IP_{B_{t,r}}(A)} a_\pi,
\]

where,

\[
IP_{P_1}(P_2) = \left\{ \pi_1 | (\pi_1 \in S_{P_1}) \wedge \exists \pi_2 \in S_{P_2} (\text{dist}(c_{\pi_1}, c_{\pi_2}) < \mu_{IA}) \right\}
\]

and \( \mu_{IA} \) is a user-defined constant.

Using our algorithm described in [10] based on octrees [12] and our Dynamic Packing Grid (DPG) data structure [13], the expression above can be evaluated in \( O((M_A + M_B) \log (M_A + M_B)) \) (w.h.p.) time and \( O(M_A + M_B) \) space. However, this algorithm runs much faster in practice, i.e., in \( O(N_{int}) \) time (w.h.p.), where \( N_{int} \) is the total number of patches in the interface.

2.3 Interface Propensity Filter.

As in Section 2.2 we assume that we have a patch decomposition of the surface of each molecule \( P \in \{A, B\} \). However, each patch \( \pi \) is now represented as a triple \(< c_\pi, a_\pi, h_\pi \>\) where the additional value \( h_\pi \) is a measure of interface propensity of \( \pi \) in the following sense. We assume that each atom \( k \in P \) is assigned an interface propensity value \( h_k \) as in Section 1.3, and \( h_\pi = \frac{1}{|S_{P,\pi}|} \sum_{k \in S_{P,\pi}} h_k \), where \( S_{P,\pi} = \{k | (k \in P) \wedge (\text{dist}(c_k, c_\pi) < \nu r_k)\} \), \( c_k \) is the center of atom \( k \), \( r_k \) is its van der Waals radius, and \( \nu \) is a user-defined constant.

This filter computes \( IPVec(A, B_{t,r}) \) which is defined as the following vector:

\[
\begin{bmatrix}
\sum_{\pi \in IP_A(B_{t,r}) \cap IP_{B_{t,r}}(A)} h_\pi a_\pi, & \sum_{\pi \in IP_A(B_{t,r}) \cup IP_{B_{t,r}}(A)} h_\pi a_\pi, \\
\sum_{\pi \in IP_{B_{t,r}}(A)} h_\pi a_\pi, & \sum_{\pi \in IP_{B_{t,r}}(A)} h_\pi a_\pi
\end{bmatrix}
\]

where \( IP_{P_1}(P_2) \) is as defined in Section 2.2.
This vector can be computed simultaneously with the approximate interface area in Section 2.2 incurring negligible overhead. The algorithm is given in [10].

Once we have \( IPVec(A, B_{t,r}) = (v_1, v_2, v_3, v_4) \), we approximate \( IP-score_{t,r}(A, B) \) defined in Section 1.3 as follows:

\[
IP-score_{t,r}(A, B) \approx -\frac{v_2 + v_4}{\min(IP'_c, v_1 + v_3)},
\]

where, \( IP'_c = (\max_{h_k < 0} h_k) (\min_{\pi \in A \cup B} a_\pi) \).

We also observed that rewarding docking poses with larger positive interface propensity weighted area \( v_2 + v_4 \) generally improves the ranks of near native poses. However, rewarding based on the product of \( IP-score_{t,r}(A, B) \) and \( v_2 + v_4 \) seems to work even better in practice. This composite term which we define as \( IP-product_{t,r}(A, B) \) below, rewards a docking pose with higher \( v_2 + v_4 \) value more than one with lower such value when both have the same \( IP-score \).

\[
IP-product_{t,r}(A, B) = IP-score_{t,r}(A, B) \times (v_2 + v_4)
\]

\[
\approx -\frac{(v_2 + v_4)^2}{\min(IP'_c, v_1 + v_3)}.
\]

The interface propensity filter in F^2Dock 2.0 penalizes a docking pose provided its \( IP-score \) is below a user-defined threshold \( \mu_{IP} \). The \( IP-product \) value, after weighting it by a user-specified weight \( w'_{IP} \), is added to the overall score of the pose.

2.4 Clash Filter.

Two atoms \( a \in A \) and \( b \in B \) with van der Waals radii \( r_a \) and \( r_b \), respectively, are said to be in a clash provided the distance between their centers is smaller than \( \alpha(r_a + r_b) \), where \( \alpha \) is a user-defined positive constant. F^2Dock 2.0 counts the total number of atomic clashes between molecules \( A \) and \( B_{t,r} \) as follows.

\[
Clash_{t,r}(A, B) = |\{b | (b \in B_{t,r}) \land \exists a \in A(r_{a,b} < \alpha(r_a + r_b))\}|
\]

Direct computation of this number requires \( O(M_AM_B) \) time. So we use an octree-based spatial subdivision approach that is similar to the interface area approximation algorithm used in Section 2.2 and has similar running time.

F^2Dock 2.0 penalizes a docking pose \( A + B_{t,r} \) provided \( Clash_{t,r}(A, B) > \mu_C \), where \( \mu_C \) is a user-defined constant. A value of around 10 for \( \mu_C \) seems to work well in practice.

2.5 Residue-Residue Contact Filter.

F^2Dock 2.0 uses the pairwise contact preferences listed in either Table III (without volume normalization) or Table IV (normalized w.r.t. residue volumes) of [14] based on user preference with Table III being the default. The contact preferences were derived as follows. Two residues are considered to be in contact if the distance between their \( C_\beta \) atoms (\( C_\alpha \) for Gly) is less than 6 Å. The normalized number of contacts between residue types \( i \) and \( j \) is defined as \( Q_{ij} = C_{ij}/\sum_{k<l} C_{kl} \), where \( C_{ij} \) is the actual number of contacts observed between residue types \( i \) and \( j \). The expected number of contacts \( W_{ij} \) is obtained by assuming that there are no contact preferences between residues of
different types. Then the likelihood of contact between residues types $i$ and $j$ is defined as (Table III of [14]): $G_{ij} = 10 \log \frac{Q_{ij}}{\overline{Q}_{ij}}$.

The volume-normalized likelihood $G_{ij}(v)$ (Table IV of [14]) is obtained by replacing $Q_{ij}$ with:

$$Q_{ij}(v) = C_{ij} \times V_i \times V_j / \sum_{k<l} (C_{kl} \times V_k \times V_l),$$

where $V_i$ is the volume of residue $i$.

Given a docking pose $A + B$, we identify all residue-residue contacts at the interface of the two molecules using a fast algorithm similar to the one used in Section 2.3, and compute the sum of all positive and negative $G_{ij}$ values (or $G_{ij}(v)$ values if chosen by the user) denoted by $G^+$ and $G^-$, respectively. Then we compute the following ratio:

$$RC\text{-score}_{A,B}(A,B) = -\frac{G^+}{\min(G^-, RC_e)}.$$

where $RC_e = \max_{G_{ij} < 0} G_{ij}$ (or $RC_e = \max_{G_{ij}(v) < 0} G_{ij}(v)$ if the user chooses volume-normalized contact preferences).

$F^2$Dock 2.0 penalizes a docking pose if its $RC\text{-score}$ is below a user-specified constant $\mu_R C$. Without volume normalization $\mu_R C = 3.0$ seems to work well in practice. Docking poses can also be penalized based on user-specified lower and upper bounds for $G^+$ and $G^-$, respectively.

### 2.6 Glycine Filter.

We mark the oligopeptides with the properties described above on the enzyme surface, and for any given docking pose count the number of these motifs occurring at the interface using a fast algorithm similar to the one used for counting atomic clashes in Section 2.4.

### 2.7 Antibody-Antigen Contact Filter.

We say that an atom $a$ is in close neighborhood of another atom $b$ provided the distance between the centers of $a$ and $b$ is at most $2(r_a + r_b)$. Given a potential antibody-antigen docking pose, $F^2$Dock 2.0 computes three quantities: $N_{L1 \cup H1}$, $N_{L3}$ and $N_{H3}$, denoting the number of antigen atoms that are in the close neighborhood of any atom in the antibody regions CDR-L1/CDR-H1, CDR-L3 and CDR-H3, respectively. These counts are obtained using a fast algorithm similar to the one used in Section 2.4. $F^2$Dock 2.0 penalizes a docking pose provided $N_{L1 \cup H1} < \mu_{L1 \cup H1}$ or $N_{L3} < \mu_{L3}$ or $N_{H3} < \mu_{H3}$, otherwise it adds $w_{AC} (N_{L1 \cup H1} + N_{L3} + N_{H3})$ to the total score, where $\mu_{L1 \cup H1}$, $\mu_{L3}$, $\mu_{H3}$ and $w_{AC}$ are user-specified quantities. $F^2$Dock 2.0 uses $\mu_{L1 \cup H1} = 100$, $\mu_{L3} = \mu_{H3} = 150$ and $w_{AC} = 10$ by default. The CDR (Complementarity Determining Region) loops are identified using the method described in [15].

### 2.8 Overall Cost of Filtering.

Summing up the cost of all filters described above for a single run, we obtain

$$O \left( \frac{1}{c^3} (M_A + M_B) \log (M_A + M_B) \right) \text{ (w.h.p.).}$$

Assuming that each filter is applied on at most $N_F$ configurations, the total time taken by all filters is

$$O \left( \frac{1}{c^3} N_F (M_A + M_B) \log (M_A + M_B) \right) \text{ (w.h.p.).}$$

Assuming $N_F = O(N_R)$, where $N_R$ is the number of samples in the rotations space, the running time reduces to

$$O \left( \frac{1}{c^3} N_R (M_A + M_B) \log (M_A + M_B) \right) \text{ (w.h.p.).}$$
3 Solvation Energy Based Reranking

The solvation energy $E_{\text{sol}}$ consists of the energy to form cavity in the solvent ($E_{\text{cav}}$), the solute-solvent van der Waals interaction energy ($E_{\text{vdw(s-s)}}$), and the electrostatic potential energy change due to the solvation (also known as the polarization energy, $E_{\text{pol}}$) [16, 17, 18, 19].

$$E_{\text{sol}} = E_{\text{cav}} + E_{\text{vdw(s-s)}} + E_{\text{pol}}$$

The first two terms are often modeled as [16, 20]

$$E_{\text{cav}} = pV + \sum_i \gamma_i A_i$$

and

$$E_{\text{vdw(s-s)}} = \rho_0 \sum_i \int_{\text{ex}} u_{i}^{\text{(att)}}(x_i, r)d^3r$$

where $p$ is the solvent pressure, $V$ is the molecular volume, $A_i$ is the solvent accessible surface area of atom $i$ and $\gamma_i$ is its solvation parameter, $\rho_0$ is the bulk density, and $u_{i}^{\text{(att)}}$ is the van der Waals dispersive component of the interaction between atom $i$ and the solvent.

The last term, $E_{\text{pol}}$, can be approximated using Generalized Born (GB) theory [21].

$$E_{\text{pol}} = -\tau \sum_{i,j} \frac{q_i q_j}{\sqrt{r_{ij}^2 + R_i R_j e^{-\frac{r_{ij}^2}{\tau^2 R_i R_j}}}}, \quad (1)$$

where $\tau = 1 - \frac{1}{\epsilon}$, and $R_i$ is the effective Born radius of atom $i$.

GB-rerank approximates each of these terms as described in the following sections, and reranks the list of top docking poses produced by F$^2$Dock 2.0 based on the resulting $\Delta E_{\text{sol}}$ values. In order to approximate $\Delta E_{\text{sol}}$, GB-rerank precomputes the $E_{\text{sol}}$ values for molecules $A$ and $B$, and then computes $E_{\text{sol}}$ for each docking pose.

3.1 Approximating $E_{\text{pol}}$ (Polarization Energy).

We first estimate all Born radii of the molecule/complex using our fast approximation scheme described in [10] followed by another of our algorithms from [10] for estimating $E_{\text{pol}}$ from the approximated Born radii. Given a molecule/complex consisting of $M$ atoms, both algorithms run in $\mathcal{O}(\frac{1}{\epsilon} M \log M)$ time using $\mathcal{O}(M)$ space, where $\epsilon > 0$ is an approximation parameter that provides a speed-accuracy tradeoff. The smaller the value of $\epsilon$ the more accurate the algorithms are, and the larger the $\epsilon$ value the faster they run.

GB-rerank uses either the $r^4$- or the $r^6$-approximation of $R_i$ (see [10]) based on user preference. By default, it uses the $r^6$-approximation [22] which shows better accuracy for spherical solutes as well as for proteins [23].

$$\frac{1}{R_i^3} = \frac{3}{4\pi} \int_{\text{ex}} \frac{1}{|r - x_i|^6} d^3r, \quad i \in [1, M]. \quad (2)$$

where $\Gamma$ is the molecular surface, $\mathbf{n}(r)$ is the unit outward normal on the molecular surface at $r$, and $x_i$ is the center of atom $i$. 

We obtain the following discrete surface formulation of Equation (2) by applying the divergence theorem and Gaussian quadrature.

\[
\frac{1}{R^3_i} \approx \frac{1}{4\pi} \sum_{k=1}^{N} w_k \frac{(r_k - x_i) \cdot \mathbf{n}_k}{|r_k - x_i|^6}.
\]

(3)

where, the \( r_k \)'s are \( N = \mathcal{O}(M) \) Gauss quadrature/integration points [11] on the molecular surface, and \( w_k \) is a weight assigned to \( r_k \) in order to achieve higher order of accuracy for small \( N \).

In order to avoid the slowdown due to the repeated computation of quadrature points for each docking pose \( A + B_{t,r} \), F^2Dock 2.0 precomputes the set of quadrature points \( Q_A \) and \( Q_B \) for \( A \) and \( B \), respectively, and chooses all quadrature points from \( Q_A \cup Q_{B_{t,r}} \) that are not on the interface of \( A \) and \( B_{t,r} \) for use during the Born radii estimation of \( A + B_{t,r} \).

3.2 Approximating \( E_{\text{vdw}(s-s)} \) (Dispersion Energy).

The solute-solvent van der Waals interaction energy (also known as dispersion energy) is modeled as [16, 20]:

\[
E_{\text{vdw}(s-s)} = \rho_0 \sum_{i=1}^{M} \int_{\text{ex}} u^{(\text{att})}_i(x_i, \mathbf{r}) d^3 \mathbf{r}.
\]

where \( \rho_0 \) is the bulk density, and \( u^{(\text{att})}_i \) is the van der Waals dispersive component of the interaction between atom \( i \in [1, M] \) and the solvent which is given as follows.

\[
u^{(\text{att})}_i(x_i, \mathbf{r}) = \frac{1}{|\mathbf{r} - x_i|^6}
\]

Thus

\[
E_{\text{vdw}(s-s)} = \rho_0 \sum_{i=1}^{M} \int_{\text{ex}} \frac{1}{|\mathbf{r} - x_i|^6} d^3 \mathbf{r}
\]

(4)

If \( R_i \) is the Born radius of atom \( i \) calculated using the \( r^6 \)-approximation (i.e., Equation 2/3), then Equation 4 can be rewritten as:

\[
E_{\text{vdw}(s-s)} \approx \rho_0 \sum_{i=1}^{M} \frac{1}{R_i^3}
\]

(5)

Therefore, \( E_{\text{vdw}(s-s)} \) can be approximated in \( \mathcal{O}(M) \) time once the Born radii of all atoms are available.

3.3 Approximating \( E_{\text{cav}} \) (Cavity Forming Energy).

Instead of computing \( E_{\text{cav}} \) for \( A \), \( B \) and each \( A + B_{t,r} \) separately, we approximate \( \Delta E_{\text{cav}} \) with the buried surface area of \( A + B_{t,r} \) which is approximated using the same algorithm used for interface area filter in Section 2.2. The algorithm runs in \( \mathcal{O}((M_A + M_B) \log (M_A + M_B)) \) (w.h.p.) time (much faster in practice; see Section 2.2) using \( \mathcal{O}(M_A + M_B) \) space.
3.4 Overall Cost of Reranking.

Assuming that GB-rerank is applied on $N_G$ docking poses, its total running time is

$$O\left(\frac{1}{\epsilon^3}N_G(M_A + M_B)\log(M_A + M_B)\right) \text{ (w.h.p.)}.$$  

Typically $N_G = O(N_R)$, where $N_R$ is the number of samples in the rotations space, and so the running time reduces to

$$O\left(\frac{1}{\epsilon^3}N_R(M_A + M_B)\log(M_A + M_B)\right) \text{ (w.h.p.)}.$$
4 Additional Results on the Effects of Various Affinity Functions and Filters

Figures S3 to S10 track the changes in the rank of the top hit, RMSD of the top hit, number of hits in top 1000, and the minimum RMSD in the top 1000 predictions, respectively, as the various F²Dock 2.0 options are activated one after another on the rigid-body test cases from Zlab benchmark 2.0 [2].

When we activate Lennard-Jones filter, clash filter and proximity clustering after shape complementarity we get hits for 4 new test cases, and the rank of the top hit improves for 15 more (see Figure S3(top)). However, we also lose hits in top 1000 for 3 test cases, and the rank of the top hit degrades for one test case. Overall, the application of these filters and clustering seem largely beneficial. The best results are obtained for enzyme-inhibitor/enzyme-substrate complexes, as for more than 50% of these complexes rank of the top hit improve.

The impact on the RMSD of the top hit is not clear though it has improved for 5 test cases in the enzyme-inhibitor/enzyme-substrate group (see Figure S5(top)). Similar is the impact on the minimum RMSD in the top 1000 predictions (see Figure S9(top)).

The number of hits in the top 1000 predictions is reduced in majority of the cases (see Figure S7(top)) as the two filters penalize predictions that are close to the target complex but have steric clashes. The proximity clustering step also penalizes poses that are very similar to poses with higher scores.

When electrostatics is turned on we get hits in top 1000 for 9 test cases for which we did not have a single hit before, and for 14 other cases rank of the top hit improve (see Figure S3(middle)). However, we lose all hits for 1 test case, and for 4 others rank of the top hit degrades.

Though there does not seem to be any particular trend in the change of the RMSD of the top hit except for the 9 cases with new first hits as mentioned above (see Figure S5(middle)), for antibody-antigen and antigen-bound antibody complexes the minimum RMSD in the top 1000 seems to have generally improved (see Figure S9(middle)). For 22 test cases the number of hits in the top 1000 predictions has improved while for 5 cases this number has dropped (see Figure S7(middle)).

The FFT-based interface propensity scoring is activated next which improves the rank of the top hit for 30 test cases (i.e., for around 50% of all cases) among which 7 cases did not have a single hit before (see Figure S3(bottom)). Among these 7 cases with new first hits 5 are antibody-antigen or antigen-bound antibody complexes, and none are enzyme-inhibitor or enzyme-substrate. Ranks have degraded for 5 test cases. Minimum RMSD in top 1000 has improved for 20 test cases degrading for only 1 (see Figure S9(bottom)), and RMSD of top hit has also degraded for only 1 test case (see Figure S5(bottom)). The number of hits has increased for 28 test cases and decreased for 7 (see Figure S7(bottom)).

The interface propensity filter is turned on next. It improves the rank of the top hit for 25 complexes, and degrades for 5 (see Figure S4(top)). For 3 test cases we did not have a single hit in top 1000 before among which 2 are antibody-antigens. The number of hits has increased for 16 cases, and decreased for 5 (see Figure S8(top)). The impacts on the RMSD of the top hit and the minimum RMSD in top 1000 are not significant in general (see Figures S6(top) and S10(top)).

The residue-residue contact filter which is activated next improves the rank of the top hit for 27 test cases, and degrades for none (see Figure S4(middle)). The enzyme-inhibitor and enzyme-substrate complexes seem to have benefited the least from this filter. The impact on the number of hits in top 1000 is not significant and is mixed (see Figure S8(middle)). The RMSD of the top hit

15
and the minimum RMSD in top 1000 do not change except for the 3 cases for which we get new first hits (see Figures S6(middle) and S10(middle)).

Next we apply the antibody contact filter and the Glycine filter. The antibody contact filter improves the rank of the top hit for 9 antibody-antigen and antigen-bound antibody test cases, and degrades for 3, while the Glycine filter slightly improves the same for 4 enzyme-inhibitor/enzyme-substrate complexes (see Figures S4(bottom)). The antibody contact filter improves the number of hits in the top 1000 predictions for 5 complexes (see Figure S8(bottom)).

All other changes are insignificant.
Figure S3: Changes in the rank of top hit as various options in F^2Dock 2.0 are activated one after another (on the rigid-body test cases from Zlab benchmark 2.0 [2]). **Top:** Lennard-Jones Filter (LJ), clash filter (CL) and proximity clustering (PC) are activated after shape complementarity (SC), **Middle:** electrostatics & charge complementarity (EL) after SC+LJ+CL+PC, **Bottom:** interface propensity (IP) after SC+LJ+CL+PC+EL.
Figure S4: Changes in the rank of top hit as various options in F$^2$Dock 2.0 are activated one after another (on the rigid-body test cases from Zlab benchmark 2.0 [2]). Continued from S3. Top: interface propensity filter (PF) after SC+LJ+CL+PC+EL+IP, Middle: residue-residue contact filter (RC) after SC+LJ+CL+PC+EL+IP+PF, and Bottom: antibody contact filter (AF) or glycine filter (GF) after SC+LJ+CL+PC+EL+IP+PF+RC.
Figure S5: Changes in the RMSD of the top hit as various options in F^2Dock 2.0 are activated one after another (on the rigid-body test cases from Zlab benchmark 2.0 [2]).

Top: Lennard-Jones Filter (LJ), clash filter (CL) and proximity clustering (PC) are activated after shape complementarity (SC), Middle: electrostatics & charge complementarity (EL) after SC+LJ+CL+PC, Bottom: interface propensity (IP) after SC+LJ+CL+PC+EL.
Figure S6: Changes in the RMSD of the top hit as various options in F$^2$Dock 2.0 are activated one after another (on the rigid-body test cases from Zlab benchmark 2.0 [2]). Continued from S5. **Top:** interface propensity filter (PF) after SC+LJ+CL+PC+EL+IP, **Middle:** residue-residue contact filter (RC) after SC+LJ+CL+PC+EL+IP+PF, and **Bottom:** antibody contact filter (AF) or glycine filter (GF) after SC+LJ+CL+PC+EL+IP+PF+RC.
Figure S7: Changes in the number of hits in the top 1000 docking poses generated by F^2Dock 2.0 as various options in F^2Dock 2.0 are activated one after another (on the rigid-body test cases from Zlab benchmark 2.0 [2]). **Top:** Lennard-Jones Filter (LJ), clash filter (CL) and proximity clustering (PC) are activated after shape complementarity (SC), **Middle:** electrostatics & charge complementarity (EL) after SC+LJ+CL+PC, **Bottom:** interface propensity (IP) after SC+LJ+CL+PC+EL.
Figure S8: Changes in the number of hits in the top 1000 docking poses generated by F^2Dock 2.0 as various options in F^2Dock 2.0 are activated one after another (on the rigid-body test cases from Zlab benchmark 2.0 [2]). Continued from S7. Top: interface propensity filter (PF) after SC+LJ+CL+PC+EL+IP, Middle: residue-residue contact filter (RC) after SC+LJ+CL+PC+EL+IP+PF, and Bottom: antibody contact filter (AF) or glycine filter (GF) after SC+LJ+CL+PC+EL+IP+PF+RC.
Figure S9: Changes in the minimum RMSD in the top 1000 docking poses generated by F²Dock 2.0 as various options in F²Dock 2.0 are activated one after another (on the rigid-body test cases from Zlab benchmark 2.0 [2]). **Top:** Lennard-Jones Filter (LJ), clash filter (CL) and proximity clustering (PC) are activated after shape complementarity (SC), **Middle:** electrostatics & charge complementarity (EL) after SC+LJ+CL+PC, **Bottom:** interface propensity (IP) after SC+LJ+CL+PC+EL.
Figure S10: Changes in the minimum RMSD in the top 1000 docking poses generated by F^2Dock 2.0 as various options in F^2Dock 2.0 are activated one after another (on the rigid-body test cases from Zlab benchmark 2.0 [2]). Continued from S9. **Top:** interface propensity filter (PF) after SC+LJ+CL+PC+EL+IP, **Middle:** residue-residue contact filter (RC) after SC+LJ+CL+PC+EL+IP+PF, and **Bottom:** antibody contact filter (AF) or glycine filter (GF) after SC+LJ+CL+PC+EL+IP+PF+RC.
5 Additional Results Comparing ZDock 3.0.2 and F²Dock 2.0

<table>
<thead>
<tr>
<th>Antibody</th>
<th>Easy</th>
<th>Medium</th>
<th></th>
<th></th>
<th>Top ranked near-native solution</th>
<th>Best ranked near-native solution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ZDock</td>
<td>F²Dock</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rank</td>
<td>RMSD</td>
</tr>
<tr>
<td>1AHW</td>
<td>45</td>
<td>0.30</td>
<td>Acc</td>
<td>8</td>
<td>4.4</td>
<td>0.47</td>
</tr>
<tr>
<td>1B1J</td>
<td>1.9</td>
<td>0.69</td>
<td>Med</td>
<td>63</td>
<td>2.3</td>
<td>0.53</td>
</tr>
<tr>
<td>1BK</td>
<td>1.9</td>
<td>0.69</td>
<td>Med</td>
<td>63</td>
<td>2.3</td>
<td>0.53</td>
</tr>
<tr>
<td>1DQ5</td>
<td>4.0</td>
<td>0.56</td>
<td>Acc</td>
<td>74</td>
<td>4.9</td>
<td>0.21</td>
</tr>
<tr>
<td>1E6J</td>
<td>4.1</td>
<td>0.20</td>
<td>Acc</td>
<td>126</td>
<td>5</td>
<td>0.19</td>
</tr>
<tr>
<td>1FSK</td>
<td>2.9</td>
<td>0.52</td>
<td>Med</td>
<td>1</td>
<td>3.2</td>
<td>0.48</td>
</tr>
<tr>
<td>1HR</td>
<td>3.9</td>
<td>0.42</td>
<td>Acc</td>
<td>9</td>
<td>3.9</td>
<td>0.42</td>
</tr>
<tr>
<td>1IQL</td>
<td>4.3</td>
<td>0.48</td>
<td>Acc</td>
<td>4</td>
<td>3</td>
<td>0.38</td>
</tr>
<tr>
<td>1JPS</td>
<td>2.1</td>
<td>0.65</td>
<td>Med</td>
<td>186</td>
<td>2.7</td>
<td>0.61</td>
</tr>
<tr>
<td>1JKC</td>
<td>2.9</td>
<td>0.56</td>
<td>Med</td>
<td>105</td>
<td>4.4</td>
<td>0.47</td>
</tr>
<tr>
<td>1KXQ</td>
<td>2.7</td>
<td>0.78</td>
<td>Med</td>
<td>1</td>
<td>1.6</td>
<td>0.75</td>
</tr>
<tr>
<td>1MLC</td>
<td>2.0</td>
<td>0.41</td>
<td>Acc</td>
<td>11</td>
<td>3.8</td>
<td>0.32</td>
</tr>
<tr>
<td>1NCA</td>
<td>1.7</td>
<td>0.56</td>
<td>Med</td>
<td>168</td>
<td>3.7</td>
<td>0.55</td>
</tr>
<tr>
<td>1SN</td>
<td>1.6</td>
<td>0.67</td>
<td>Med</td>
<td>1267</td>
<td>1.6</td>
<td>0.67</td>
</tr>
<tr>
<td>1QFW</td>
<td>3.1</td>
<td>0.43</td>
<td>Acc</td>
<td>191</td>
<td>4.8</td>
<td>0.34</td>
</tr>
<tr>
<td>1VFB</td>
<td>1.5</td>
<td>0.75</td>
<td>Med</td>
<td>5</td>
<td>3.2</td>
<td>0.65</td>
</tr>
<tr>
<td>2FD6</td>
<td>5.0</td>
<td>0.18</td>
<td>Inc</td>
<td>62</td>
<td>4.4</td>
<td>0.26</td>
</tr>
<tr>
<td>2G5</td>
<td>3.0</td>
<td>0.55</td>
<td>Med</td>
<td>122</td>
<td>3.9</td>
<td>0.61</td>
</tr>
<tr>
<td>2JEL</td>
<td>3.5</td>
<td>0.50</td>
<td>Med</td>
<td>1</td>
<td>3.3</td>
<td>0.45</td>
</tr>
<tr>
<td>2VPS</td>
<td>4.0</td>
<td>0.67</td>
<td>Inc</td>
<td>1</td>
<td>3.9</td>
<td>0.31</td>
</tr>
<tr>
<td>9QFW</td>
<td>2</td>
<td>0.40</td>
<td>Inc</td>
<td>1</td>
<td>3.9</td>
<td>0.31</td>
</tr>
</tbody>
</table>

Figure S11: Comparison of the performance of F²Dock 2.0 and ZDock 3.0.2 for each of the 25 antibody-antigen and antigen-bound antibody complexes from ZLab’s benchmark 4.0 in terms of the rank, RMSD, Fnat and CAPRI classification of the top hit and the best hit.
Figure S12: Comparison of the performance of F²Dock 2.0 and ZDock 3.0.2 for each of the 52 enzyme-inhibitor and enzyme-substrate complexes from ZLab’s benchmark 4.0 in terms of the rank, RMSD, $F_{nat}$ and CAPRI classification of the top hit and the best hit.

### References


Figure S13: Comparison of the performance of F2Dock 2.0 and ZDock 3.0.2 for each of the 99 other type of complexes from ZLab’s benchmark 4.0 in terms of the rank, RMSD, $F_{nat}$ and CAPRI classification of the top hit and the best hit. Continued as Figure S14.


Figure S14: Comparison of the performance of $F^2$Dock 2.0 and ZDock 3.0.2 for each of the 99 other type of complexes complexes from ZLab’s benchmark 4.0 in terms of the rank, RMSD, $F_{nat}$ and CAPRI classification of the top hit and the best hit.


