J Comput Aided Mol Des (2006) 20:601–619
DOI 10.1007/s10822-006-9060-4

ORIGINAL PAPER

Development and validation of a modular, extensible docking program: DOCK 5

Demetri T. Moustakas · P. Therese Lang · Scott Pegg · Eric Pettersen · Irwin D. Kuntz · Natasja Brooijmans · Robert C. Rizzo

Received: 12 April 2006 / Accepted: 22 July 2006 / Published online: 6 December 2006
© Springer Science+Business Media B.V. 2006

Abstract We report on the development and validation of a new version of DOCK. The algorithm has been rewritten in a modular format, which allows for easy implementation of new scoring functions, sampling methods and analysis tools. We validated the sampling algorithm with a test set of 114 protein–ligand complexes. Using an optimized parameter set, we are able to reproduce the crystal ligand pose to within 2 Å of the crystal structure for 79% of the test cases using our rigid ligand docking algorithm with an average run time of 1 min per complex and for 72% of the test cases using our flexible ligand docking algorithm with an average run time of 5 min per complex. Finally, we perform an analysis of the docking failures in the test set and determine that the sampling algorithm is generally sufficient for the binding pose prediction problem for up to 7 rotatable bonds; i.e. 99% of the rigid ligand docking cases and 95% of the flexible ligand docking cases are sampled successfully. We point out that success rates could be improved through more advanced modeling of the receptor prior to docking and through improvement of the force field parameters, particularly for structures containing metal-based cofactors.

Keywords Automated docking · Scoring functions · Structure-based drug design · Flexible docking · Binding mode prediction · Incremental construction · Validation

Introduction

Transient non-covalent interactions are critical for biological processes. The sequencing of a variety of genomes and the development of proteomics techniques have enabled scientists to study these interactions on the widest scales [1]. Advances in X-ray crystallography, nuclear magnetic resonance spectroscopy, and other experimental structure techniques provide the ability to study these interactions at an atomic level of detail [2]. One important application of these advances is the design of small molecules that interact with cellular processes to modify biological activity and treat disease.

D. T. Moustakas and P. T. Lang are joint first authors

Electronic Supplementary Material The structure files for the test set and the optimized input files used to generate this data can be found at the DOCK web site (http://dock.compbio.ucsf.edu).

D. T. Moustakas
Joint Graduate Program in Bioengineering, University of California, San Francisco, 600 16th Street, Genentech Hall, Box 2240, San Francisco, CA 94143, USA

D. T. Moustakas
Joint Graduate Program in Bioengineering, University of California, Berkeley, Berkeley, CA, USA

P. T. Lang · N. Brooijmans
Graduate Program in Chemistry and Chemical Biology, University of California, San Francisco, 600 16th Street, Genentech Hall, Box 2240, San Francisco, CA 94143, USA

S. Pegg · E. Pettersen · I. D. Kuntz (✉)
Department of Pharmaceutical Chemistry, University of California, San Francisco, 600 16th Street, Genentech Hall, Box 2240, San Francisco, CA 94143, USA
e-mail: kuntz@cgl.ucsf.edu

R. C. Rizzo
Department of Applied Mathematics and Statistics, Stony Brook University, Room 1-101, Stony Brook, NY 11794-3600, USA

The drug discovery process typically requires between 10 years and 15 years from early discovery until FDA approval [3]. Computational tools—such as virtual screening, homology modeling and cheminformatics—are applied both to facilitate various stages of research and to create models that explain experimental data [4–6]. Molecular docking, which can broadly be defined as the prediction of the orientation of two molecules with respect to one another, is a computational technique that has been successfully used in both of these capacities [7]. In drug design applications, one molecule is typically a protein or nucleic acid drug target—the receptor—and the other is a potential ligand. In these applications, docking is used to identify novel ligands that interact with a biomolecular target and to predict the geometric position (binding mode) of ligands with respect to the target of interest.

DOCK background

DOCK is one example of a family of molecular docking packages available, which includes Glide, FlexX, and GOLD (Table 1) [8–11]. Each of these programs consists of two key parts: a search algorithm and a scoring function. The search algorithm samples both the relative orientations of the two objects as well as their conformations. It must be thorough enough to ensure adequate coverage of the binding free energy landscape in order to find the global minimum of the scoring function. The scoring function ranks the various geometries generated by the search algorithm, proposing the top-scoring pose as the global minimum. It must rapidly evaluate receptor–ligand complex stability with sufficient accuracy such that the global minimum of the scoring function agrees with experimental data.

The number of degrees of freedom in receptor–ligand interactions is very large, and several approximations must be made to ensure that the docking problem is tractable. Many different approaches, ranging from freezing non-essential motions to the use of preferred conformations, have been developed to reduce the number of degrees of freedom sampled [12]. In the DOCK algorithm, for example, the receptor is considered to be conformationally rigid, requiring only the ligand conformational, translational and rotational degrees of freedom to be sampled during complex formation. This assumption is reasonable in docking applications in which either the receptor conformation does not change dramatically upon ligand binding or in which the aim is to stabilize a particular receptor conformation.

In order to guide the search for ligand orientations with respect to the receptor, a negative image of the active site volume is created by placing spheres on the solvent accessible surface area of the receptor, thus restricting the ligand orientational sampling to the most relevant region on the surface of the receptor [13]. To sample the internal degrees of freedom of the ligand, DOCK uses the incremental construction algorithm, anchor-and-grow, which separates the ligand flexibility into two steps [14, 15], (Fig. 1). First, the largest rigid substructure of the ligand (anchor) is identified and rigidly oriented in the active site by matching its heavy atoms centers to the receptor sphere centers (orientation). The anchor orientations are evaluated and optimized using the scoring function and the energy minimizer. The orientations are then ranked according to their score, spatially clustered by heavy atom root mean squared deviation (RMSD), and prioritized (pruning). Next, the remaining flexible portion of the ligand is built onto the best anchor orientations within the context of the receptor (grow). It is assumed that the shape of the binding site will help restrict the sampling of ligand conformations to those that are most relevant for the receptor geometry.

Table 1 Summary of scoring functions and sampling algorithms for commonly used docking programs

Method	Ligand sampling method[a]	Receptor sampling method[a]	Scoring function[b]	Solvation scoring[c,d]
DOCK 4/5	IC	SE	MM	DDD, GB, PB
FlexX/FlexE	IC	SE	ED	NA
Glide	CE + MC	TS	MM + ED	DS
GOLD	GA	GA	MM + ED	NA

[a] Sampling methods are defined as Genetic Algorithm (GA), Conformational Expansion (CE), Monte Carlo (MC), incremental construction (IC), merged target structure ensemble (SE), torsional search (TS)

[b] Scoring functions are defined as either empirically derived (ED) or based on molecule mechanics (MM)

[c] If the package does not accommodate this option, the symbol NA (Not Available) is used

[d] Additional accuracy can be added to the scoring function using implicit solvent models. The most commonly used options are distance dependent dielectric (DDD), a parameterized desolvation term (DS), generalized Born (GB) and linearized Poisson Boltzmann (PB)

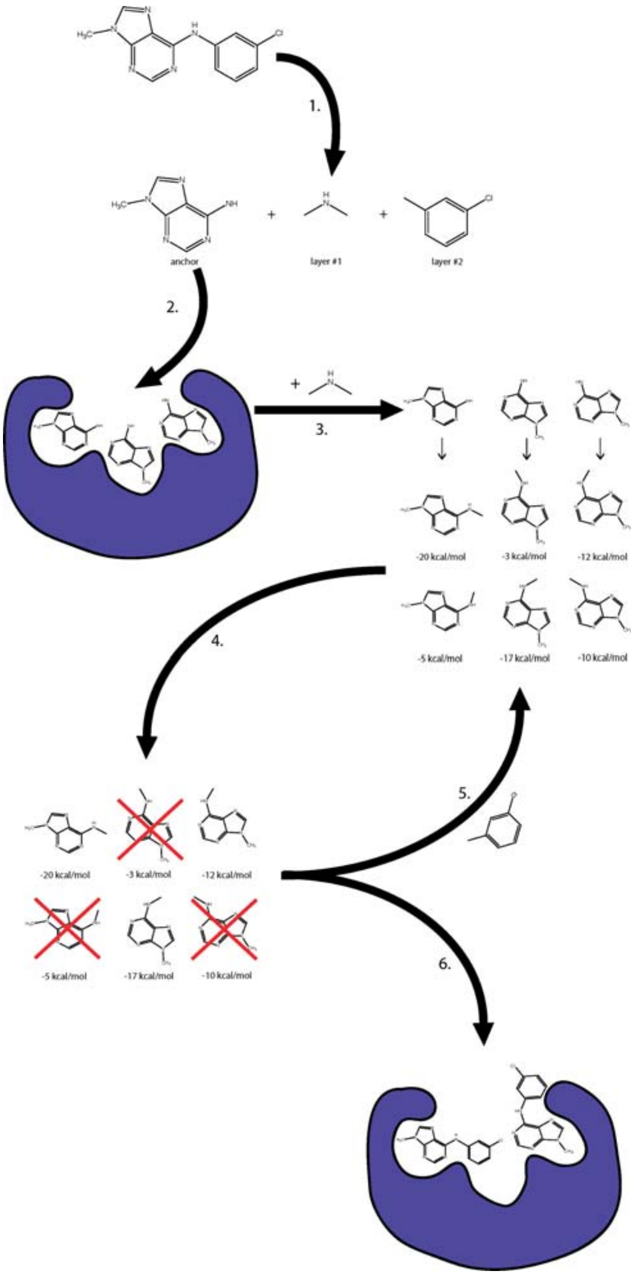

Fig. 1 The "anchor-and-grow" conformational search algorithm. The algorithm performs the following steps: **(1)** DOCK perceives the molecule's rotatable bonds, which it uses to identify an anchor segment and overlapping rigid layer segments. **(2)** Rigid docking is used to generate multiple poses of the anchor within the receptor. **(3)** The first layer atoms are added to each anchor pose, and multiple conformations of the layer 1 atoms are generated. An energy score within the context of the receptor is computed for each conformation. **(4)** The partially grown conformations are ranked by their score and are spatially clustered. The least energetically favorable and spatially diverse conformations are discarded. **(5)** The next rigid layer is added to each remaining conformation, generating a new set of conformations. **(6)** Once all layers have added, the set of completely grown conformations and orientations is returned

In order to evaluate a large number of ligand poses in a reasonable amount of time, approximate scoring functions must be used. Once again, numerous solutions to this problem have been proposed, including a variety of empirical and physics-based terms [12]. DOCK uses an energy scoring function based on the AMBER molecular mechanics force field [14, 16]. Only the interactions between the ligand and protein are considered, leaving only intermolecular van der Waals (VDW) and electrostatic components in the function. Since the receptor is considered to be rigid, the receptor contribution to the potential energy can be pre-calculated and stored on a grid [16]. These approximations enable the program to evaluate large libraries of small molecules against a receptor in a reasonable period of time.

This paper describes a new version of the DOCK program and explores the critical variables that control its ability to find correct binding modes in a suite of test problems. Our motivation is to provide a modular docking package that permits the easy development of new scoring functions, search algorithms, and analysis tools. Thus, each functional unit of the DOCK algorithm was implemented as a self-contained and portable module that interacts with the user through a well-defined interface (Fig. 2). The object-oriented language C++ was chosen to allow each component of the DOCK algorithm to be implemented as a class, which encapsulates both the data structures and functions [17]. DOCK 5 incorporates several new routines, including parallelization of the algorithm through an external library, modification of the ligand structural class to enable greater user control over sampling, and clustering of the final results by root mean square deviation. The implications of these additions will be discussed in this paper. Additional scoring functions and alternate sampling techniques have been implemented as well and will be discussed in future papers (http://dock.compbio.ucsf.edu).

Previous studies have examined the scoring function and the matching algorithm of DOCK in detail ([14] and equations 1–6 in [16]). In this paper, we pay particular attention to the robustness of the anchor-and-grow portion of the DOCK algorithm. We seek to maximize the success of complex structure prediction by independently optimizing the various steps in the anchor-and-grow algorithm. In the process, we also quantify and bound the errors for cases in which flexible docking fails and provide direction for potential areas of improvement.

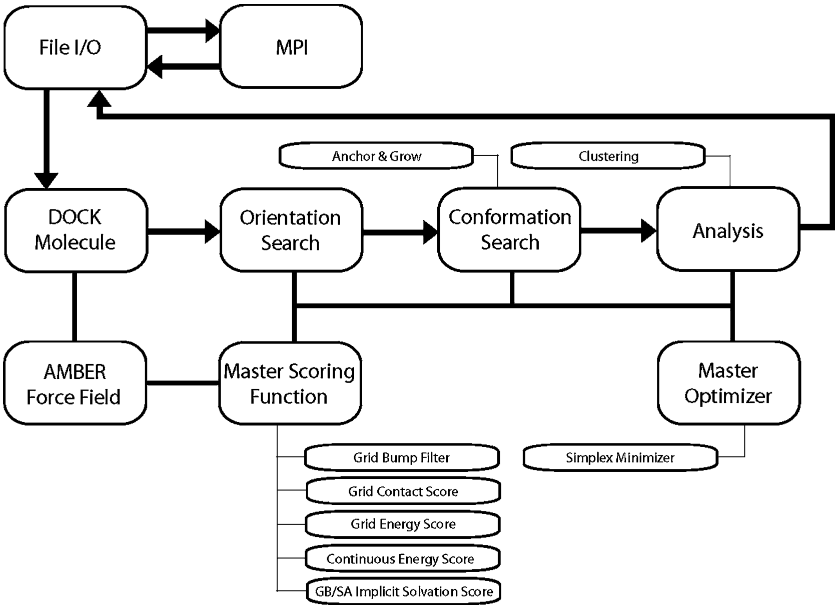

Fig. 2 The major DOCK 5 classes and their interconnections. The bold arrows denote the connections between the classes that implement the DOCK sampling algorithm. The path traced by the arrows illustrates the sequence of operations performed upon a ligand molecule during docking. The bold lines (without arrowheads) denote functional connections between classes. These connections allow one class to call functions implemented in another. This diagram demonstrates that the classes implementing the DOCK sampling methods are heavily connected to a layer of classes that implement the physics engine: the force field, the scoring functions, and the energy minimizers. The thin lines denote hierarchical relationships between a master class and modular subclasses. These hierarchical arrangements allow new functional classes (scoring functions, energy minimizers, etc.) to be plugged into the existing DOCK algorithm in a modular fashion

Overview of test set

The validation of any software program requires careful testing of all aspects of the algorithm and assessment of its utility in all anticipated applications of the software. Molecular docking is commonly used in several modes, namely ligand binding mode prediction, virtual screening, and prioritization of a set of related compounds based on their affinity. However, predicting the correct binding mode of a ligand–receptor complex is a requisite step for the successful comparison of different ligands and therefore will be the focus of this paper. It is important to note, however, that predicting binding orientations is not the only metric for the accuracy and utility of docking algorithms. Optimizing DOCK for applications, including ranking libraries of small molecules and calculating absolute free energies of binding, will be addressed in other papers (http://dock.compbio.ucsf.edu).

Large-scale validation of docking algorithms was long hampered by the lack of a large number of high quality protein–ligand complex crystal structures. Thanks to advances in automation in molecular biology and crystallography, the number of structures in the Protein Data Bank (PDB) continues to grow at a rapid pace [18]. The developers of GOLD were first to test their program on a large number of available structures [19]. Their test set was compiled using a number of criteria to select candidate protein–ligand complex structures. The protein must be of pharmacological interest and the ligands must be drug-like. In addition, complexes were chosen that exhibited interesting and unusual interactions between the ligand and the protein. The final set of 100 (more recently expanded to 134) protein–ligand complexes has served as the basis for other, larger test sets [11, 20–22].

More recently, the CCDC/Astex set compiled 305 protein–ligand complex structures by expanding the original GOLD test set [22]. However, the authors note that many of the new entries contain larger ligands that have more rotatable bonds, making this set less drug-like. The crystal structures in the CCDC/Astex set were evaluated for crystallographic errors and inconsistencies, yielding a “clean” set of 224 protein–ligand complexes. To create the test set for the DOCK validation studies, we filtered out 84 complexes with eight or more rotatable ligand bonds. In addition, several of the complexes had properties that we felt made them inappropriate for a validation set. These issues included ligands that were covalently bound to the receptor (PDB code 1ASE), ligands with missing electron density (PDB code 1EED), and known sequence misregistry in the receptor (PDB code 3HVT). Ligands with vanadium that required VDW types in which we were not completely confident were also removed. The final test set contained 114 drug-like complexes (see Methods, Table 2).

Methods

DOCK 4 to DOCK 5 conversion

The new DOCK rigid body orienting code was written as a direct implementation of the isomorphous subgraph matching method of Kuhl et al. [23]. All receptor sphere pairs and atom center pairs are considered for inclusion in a matching clique. This is more computationally demanding than the clique matching algorithm implemented in previous versions of DOCK that used a distance binning algorithm to restrict the clique search, in which pairs of spheres and atom centers were binned by distance. Only sphere pairs and center pairs that were within the same distance bin were considered as potential matches [14]. The new DOCK clique matching implementation avoids bin boundaries that prevent some receptor sphere and ligand atom pairs from matching, and, as a result, it can find good matches missed by previous versions of DOCK. The rigid body rotation code was also corrected to avoid a singularity that occurred if the spheres in the match lay within the same plane. Both of these changes improved orientational sampling.

Table 2 Complexes used in the test set (total of 114 complexes)

Protein data bank identifier					
1A28	1COM	1FLR	1OKL	1TYL	2MCP
1A6W	1COY	1HAK	1PBD	1UKZ	2PCP
1A9U	1CPS	1HDC	1PDZ	1ULB	2PHH
1ABE	1D3H	1HSL	1PHD	1WAP	2PK4
1ABF	1D4P	1HYT	1PHG	1XID	2TMN
1ACJ	1DBB	1IMB	1PTV	1XIE	2YPI
1ACM	1DBJ	1IVB	1QCF	1YDR	3CPA
1ACO	1DG5	1LAH	1QPE	2AAD	3ERD
1AI5	1DID	1LCP	1QPQ	2ACK	3GPB
1AOE	1DOG	1LDM	1RNT	2ADA	3HVT
1AQW	1DR1	1LST	1ROB	2AK3	4AAH
1AZM	1DWB	1LYL	1RT2	2CHT	4COX
1BYG	1EBG	1MDR	1SNC	2CMD	4CTS
1C5C	1ETT	1MLD	1SRJ	2CPP	4FBP
1C5X	1F0R	1MRG	1TDB	2CTC	4LBD
1C83	1F0S	1MRK	1TNG	2DBL	5ABP
1CBX	1F3D	1MUP	1TNH	2GBP	5CPP
1CIL	1FGI	1NGP	1TNI	2H4N	6RNT
1CKP	1FKI	1NIS	1TNL	2LGS	7TIM

The anchor-and-grow algorithm in the new version of DOCK was also modified to prevent premature pruning of the growth tree. The DOCK 5 anchor-and-grow code was completely rewritten with several differences in the implementations between DOCK 4 and 5. The anchor-and-grow implementation in DOCK 5 fixed a series of bugs that caused some branches of the search to be pruned when they should have been preserved for the next round of growth. The mechanism of minimization of partially grown conformers was also changed to allow the entire partial conformer to move, instead of just the latest layer, enabling more accurate ranking and pruning of the partially grown conformers.

In addition, the simplex minimizer was re-coded based on the original Nelder and Mead algorithm [24]. The new minimizer implementation consistently found lower energy minima when using the same set of 1,000 ligand orientations in a receptor, indicating that it was performing better than the previous version (data not shown). In addition, we changed the mechanism of minimization of partially grown ligand conformers to allow all atoms in the partial conformer to be minimized, rather than only the outermost layer of atoms. These changes may explain why DOCK 4 performs more poorly when run with the DOCK 5 optimized parameters (see below).

The final version of the new DOCK code, including all functions described below and all bug fixes, was posted to the DOCK web site as version 5.4.0 (http://dock.compbio.ucsf.edu). All experiments performed with the new implementation of DOCK used this version and will be referred to as DOCK 5 for convenience. All experiments performed with the previous version of DOCK used version 4.0.1 and will be referred to as DOCK 4.

Conversion of the DOCK codebase from C to C++

The design of the new DOCK 5 architecture balances the speed of the code, or computational performance, against its modularity and extensibility. The code was developed using ANSI C++ to ensure portability across multiple platforms [17]. The only external library used by DOCK 5 is MPICH for parallel processing [25]. To enable easy modification or replacement of DOCK 5 algorithm components, the DOCK 5 class structure was designed so that there are classes for each major DOCK algorithm function, and these classes interface with each other by passing instances of the DOCK 5 molecule class. Within the major functions, there are two layers of classes: those that implement the ligand sampling functions—rigid orienting, conformational searching, and minimizing—and those that implement the underlying physics engine—the force field definitions and the scoring functions. The sampling classes are applied sequentially to the ligand molecule; the physics engine classes are utilized by the sampling classes to score the ligand–receptor interaction after each step.

As a specific example of modularity, the DOCK 5 scoring functions are implemented as a master score class with five scoring function subclasses. The master score class acts as an interface to the scoring subclasses, enabling the user to designate primary and secondary scoring functions at runtime. This design was chosen because the individual scoring functions were best implemented as individual classes; they each require different input and use different internal data structures. While they could have been implemented into one large scoring class, the result would have been quite large and disjoint. This solution was also applied to the ligand conformational search, energy minimization and post-docking analysis classes.

The DOCK 5 molecule class was designed to contain the minimum information required to specify a three-dimensional ligand conformation (atom coordinates, bond connectivity, atom partial charges, atom types and bond types) to minimize the memory required to store a molecule, allowing large arrays of molecules to be stored in RAM. Standard C-style arrays were used to store the molecular data to maximize the speed of accessing this information.

Test set preparation

The proteins and ligands were extracted from the PDB files, which were downloaded from the PDB website (www.rcsb.org, Table 2). The ligands were assigned atom types and bond types manually, and hydrogens were added using Sybyl [26]. Subsequently, AM1-BCC partial electrostatic charges were calculated using the Antechamber package distributed with Amber 8 [27, 28]. The number of rotatable bonds of each of the ligands was measured using DOCK, and ligands with > 7 rotatable bonds were eliminated from the test set. We choose seven or fewer bonds to give a reasonable representation of DOCK's performance using compounds similar to those of most interest in drug discovery [29–31]. The final test set that was used consisted of 114 non-covalent protein–ligand complexes [32] (Table 2).

For the proteins, we removed all waters, covalently linked sugars, sulfates, and halogens that were not part of the ligand. Co-factors, such as heme, ATP, and NADPH, were kept, atom and bond types were assigned manually, and Gasteiger–Hückel partial electrostatic charges were calculated using the "Compute" module in Sybyl [26, 33, 34]. Ions, such as calcium and zinc, were considered to be part of the protein and the correct charge was assigned manually. Different VDW parameters for zinc were used depending on the coordination state of the zinc atom in the protein–ligand complex (Table 3). Hydrogens were added to the protein residues using the "Biopolymer" module in Sybyl, as were AMBER partial charges and VDW parameters [26, 37]. No additional optimization of the protein structure was carried out at this point.

The GRID accessory program of DOCK was used to pre-calculate scoring function potential grids [16]. All parameters were set to default parameters, except for the "energy_cutoff_distance," which was set to 9,999, resulting in the inclusion of all protein atoms in the energy calculation. For matching, the dms program was used to generate a molecular surface for each receptor [38]. The SPHGEN accessory program of DOCK was used to create a negative image of the surface using spheres [39, 40]. For the purpose of this validation study, a general procedure was established to generate a sphere cluster for every protein in the test set. In this procedure, we select all the spheres found within 10 Å of any ligand atom. The receptor box delimiting the active side was calculated with the accessory program SHOWBOX using the sphere set with an additional 5 Å boundary. We have explored additional box sizes ranging from 1 Å to 9 Å padding and found that there is little sensitivity to the exact padding amount (i.e. success rate for rigid ligand docking of 80 ± 1%, time increase 10% with padding size increase, and an average test set energy of -50 ± 0.1 DOCK units). The final procedure creates sphere sets with an average of 101 docking spheres and boxes of ~20 $Å^3$. These receptor sphere sets are larger than what one would typically use in most docking applications. This adds stringency to our testing of DOCK 5 by increasing the orientational and translational space that it must search.

Table 3 Zinc VDW parameters used to generate grids

Tetra-coordinated Zinc[a]	
Radius	1.700 Å
Well depth	0.067 kcal/mol
Penta-coordinated Zinc[b]	
Radius	1.100 Å
Well depth	0.0125 kcal/mol

[a] Parameters used for receptors with tetra coordinated zinc ions [35]

[b] Parameters used for receptors with penta coordinated zinc ions [36]

Optimized hydrogen locations for test set receptors

To assess the effect of hydrogen placements on docking outcomes, we also optimized the hydrogen atom placement and hydrogen-bonding network for the receptor using the ‘‘Dock Prep’’ module in Chimera [41]. In this module, the hybridization states of the non-hydrogen atoms of a PDB structure are determined by an enhanced version of the IDATM atom-typing algorithm [42]. Then, all hydrogens that can be unambiguously positioned are added to the file. To assist in positioning ambiguous hydrogens, hydrogen-bonding interactions are examined. The definitions of hydrogen-bonding donors and acceptors as well as hydrogen-bonding angle and distance criteria are based on the values found in Mills and Dean [43]. Relevant hydrogen bonds (H-bonds) are examined from shortest to longest, with satisfaction of shorter bonds having priority. For H-bonds where it is unclear which end is acting as the donor (e.g. water–water), use of that bond is postponed until either end is resolved further, though any lower-priority bonds that conflict geometrically with the postponed bond are eliminated from consideration at that time. If neither end is resolved by other interactions, the ambiguity is decided arbitrarily. Should examination of H-bond interactions not completely determine the positions of all of the hydrogens bound to a heavy atom, they are positioned to first satisfy potential H-bond interactions, then any remaining hydrogens are positioned to avoid steric clashes with other atoms. For histidine residues, normally one nitrogen will be protonated (chosen based on H-bond/steric considerations); however if both ring nitrogens are H-bond donors, they will both be protonated.

Selection of active site waters

All waters within 3 Å RMSD of any ligand heavy atom were selected. These waters were included as part of the receptor. The new receptor–water complexes were then subjected to the same hydrogen bonding optimization as above.

DOCK parameter optimization

To characterize the performance of DOCK 5 in regenerating known complex structures, we explored the optimum parameters for use with rigid and flexible ligand docking strategies (see Appendix 1). Unless otherwise stated, all docking experiments were carried out on 2.2 GHz dual processor Opteron 828s running Linux Fedora Core 3. The code was compiled using open-source GNU compilers (http://www.gnu.org). The optimized parameters have been implemented as the defaults. We note that our primary criterion for optimization was success in finding the proper ligand geometry and not the CPU time required per compound. Unless otherwise stated, these parameters were used for all experiments in this paper.

Greedy clustering of conformational ensemble

The greedy clustering algorithm is designed to eliminate redundant ligand orientations from consideration. DOCK generates a set of ligand orientations that are ranked by the scoring function. The RMSD between each ligand orientation in the list is calculated. If the RMSD between two ligand orientations falls within the clustering threshold, the second orientation is assigned to a cluster with the first. The first ligand orientation is selected and compared to all subsequent unclustered orientations in the list; this process is repeated until the last unclustered orientation has been selected. Once the entire list has been processed, only the best scoring ligand pose in each cluster, designated as the cluster head, is retained.

Evaluation of MPI functionality

Parallel processing is fully integrated into the DOCK calculation. The DOCK program starts a single master node and a set of processing nodes. The master node performs file processing and molecule input/output, whereas the processing nodes perform the actual docking calculations. If the number of processors is set to 1, the code defaults to non-MPI behavior. As a result of this configuration, there will be minimal difference in performance between 1 and 2 processors. Improved performance will only become evident with more than two nodes. It should be emphasized that the primary benefit in using DOCK 5 in parallel mode is to reduce bookkeeping tasks associated with manually splitting up a database into multiple chunks, which then must be submitted to different processors individually. DOCK 5 automatically partitions out subsets of a database to various nodes, collates and ranks the final results, and takes care of all intermediate bookkeeping.

To gauge the performance of parallelization of the DOCK 5 algorithm, two small subsets of the NCI database from the ZINC database were constructed [25, 44]. The two subsets, one containing 500 and the other 1,000 small molecules, were filtered to have ≤5 and ≤14 rotatable bonds, respectively. The receptor used as a target for this study was HIV-1 reverse transcriptase in complex with nevirapine (PDB code 1VRT). Because the receptor was not part of the test set, nevirapine was flexibly redocked using the optimized parameters, which yielded a ligand orientation 0.28 Å RMSD from the crystal structure orientation. In addition, a library consisting of 1,000 copies of neviripine was generated to remove dependence on the order and size of the compound library. All parallelization study calculations were executed at the Computational Science Center at Brookhaven National Laboratory (http://www.bnl.gov/csc) on a cluster consisting of 34 nodes with dual 3.2 GHz Xeon processors running Linux. Tests were performed using between 2 and 68 nodes. The code was compiled using open-source GNU compilers and MPI software mpich version 1.2.7 from Argonne National Laboratory (http://wwwunix.mcs.anl.gov/mpi/mpich).

Results

We first consider the results of rigidly docking ligands, which used a conformation taken directly from the complex crystal structure, to the complex crystal structure conformation of the receptor. We then present the results of flexible ligand docking tests. In each case, we consider (a) the overall performance of each sampling algorithm, (b) the ability of each algorithm to reproduce the crystal ligand orientation as the top-scoring pose, (c) the effect of the initial ligand conformation on the performance of the algorithm, (d) any additional information contained in the set of all sampled ligand orientations, and (e) the ability to extract additional information by clustering docking results. We also compare the performance of DOCK 5 to equivalent DOCK 4 experiments. Finally, we analyze the cases in which DOCK 5 fails to reproduce the crystal structure and propose some directions for improvement of both the DOCK algorithm and our test set preparation method.

Rigid ligand docking

Overall performance

Unless otherwise noted, all experiments described in this section involved rigid docking of the complex crystal structure ligand conformation to the receptor complex crystal structure. For each case in the test set, the heavy atom RMSD between the top-scoring docked ligand pose and the complex crystal structure ligand pose was evaluated. A DOCK 5 run was considered to be successful for cases in which the RMSD between for the top-scoring ligand orientation and the crystal ligand orientation was less than 2.0 Å. DOCK 5 selects the correct pose as the lowest energy structure for 79% (90/114) of the test cases using the rigid docking protocol with an average time of 55 s per complex.

Dependence on ligand conformation

An ensemble of ligand conformations was generated using the anchor-and-grow algorithm to apply changes of each of the ligand's rotatable bonds. This expansion generated a conformation ensemble for each ligand that covered all torsional parameters that DOCK samples. Each generated conformation was rigidly docked to the receptor, and the results from all the dockings were binned according to the magnitude of the ligand's conformational perturbation (Fig. 3a). The curve shows dramatic and continual decrease in the success rate as the perturbation magnitude increases with little success for any ligand conformations greater than 0.5 Å heavy atom RMSD away from the crystal conformation. Therefore, any conformation generation method must generate ligand conformations within 0.5 Å heavy atom RMSD of the crystal conformation for rigid docking to have a reasonable chance to succeed.

Analysis of total orientational ensemble

To this point, we have disregarded "near misses," which we define as any generated orientations within 2 Å RMSD from the crystal structure that are close to the top of the ranked conformation list, but are not the best scoring poses. We can examine the remaining poses either by including all poses that differ by a fixed energy unit from the most favorable geometry or by including those that differ by a fixed number of ranked poses from the most favorable energy. In order to quantify the extent of these partial successes, all generated ligand poses for each test case were preserved and sorted by their energy scores.

An energy gap is defined as the difference between the DOCK score of the top scoring ligand orientation and the score of a ligand ranked further down the list. Considering all docked ligand orientations with an energy gap of 2.5 DOCK units—an average of five ligand orientations—increases the rigid ligand docking success rate to 90% for the entire test set, while an average of 50 orientations increase the rigid docking success rate to 99% (Fig. 4a, b). These results indicate that the orienting method samples near-crystal ligand orientations well, but the current energy scoring func-

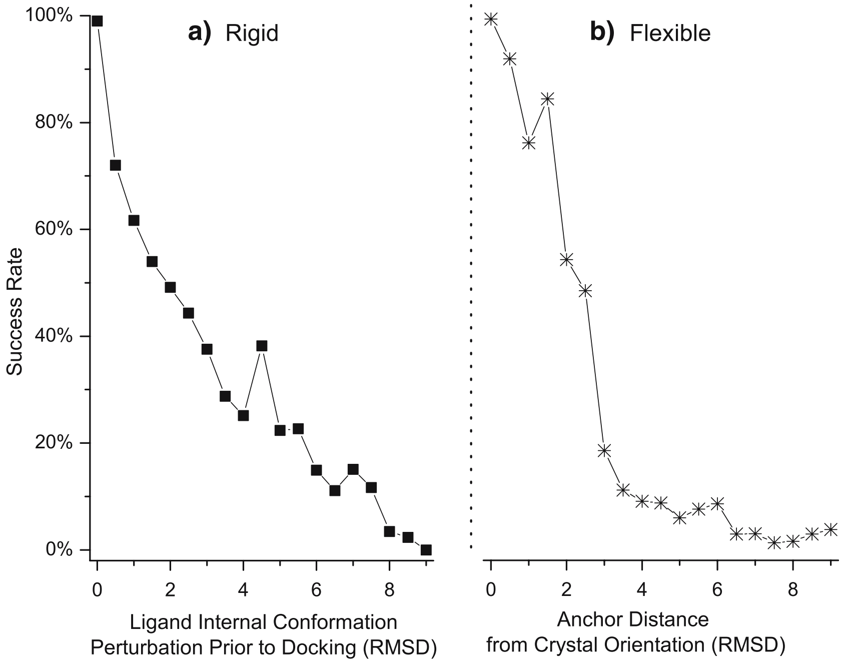

Fig. 3 (**a**) Rigid docking success rates (■)—as calculated by any conformation being within 2 Å heavy atom RMSD of the complex crystal orientation—shown as a function of the ligand internal conformation perturbation magnitude (RMSD). (**b**) Flexible growth success rates (✳)—as calculated by any conformation being within 2 Å heavy atom RMSD of the complex crystal orientation—shown as a function of the magnitude of the anchor perturbation (RMSD)

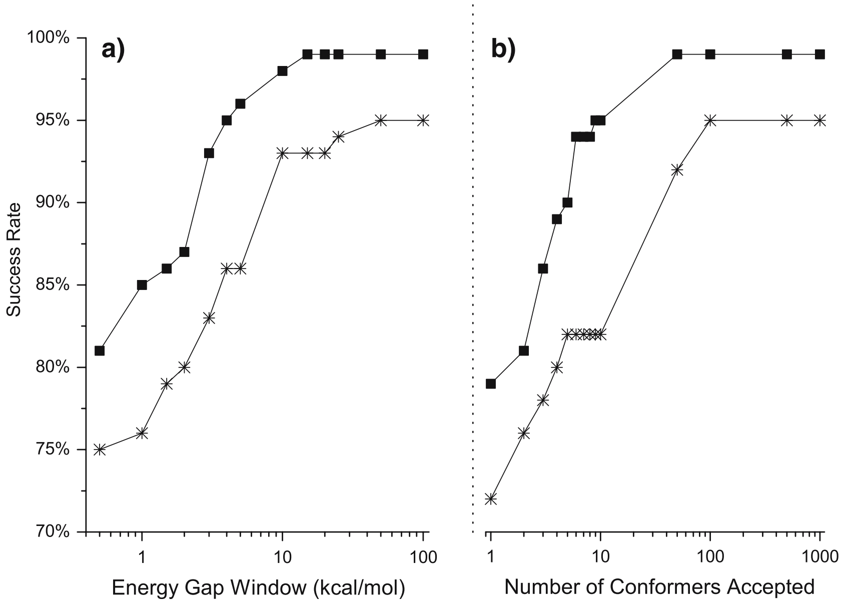

Fig. 4 (**a**) The rigid (■) and flexible (✳) docking success rate as a function of the DOCK score energy gap (kcal/mol) for all conformers generated. (**b**) The rigid and flexible docking success rate as a function of the number of ranked conformers examined

tion cannot discriminate well between the top-ranked orientations.

Geometric clustering of poses

Each ligand conformational ensemble was spatially clustered according to inter-pose RMSD values (see Methods section for algorithm details). After examining a range of potential cut-offs, an optimal value of 1.0 Å was chosen (Fig. 5). Using this clustering threshold, only 15 clusterheads are required to achieve a success rate of 99%, compared with the top 50 ranked unclustered orientations. This result is encouraging, suggesting that the clustering helps sort through the conformers efficiently.

Flexible ligand docking

Overall performance

Unless otherwise noted, all experiments described in this section involved flexible docking of the ligand to the receptor complex crystal structure. As with the rigid docking tests, the heavy atom RMSD between the

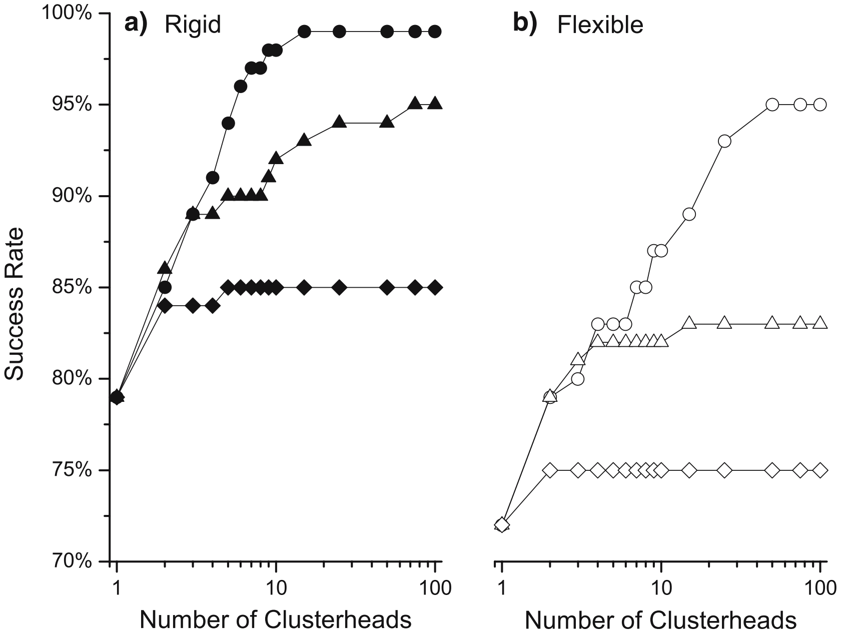

Fig. 5 The rigid (filled) and flexible (open) docking success rate as a function of the number of cluster heads examined. Clusters with heavy atom RMSD cutoffs of 1.0 Å (●), 3.0 Å (▲), and 5.0 Å (◆) were compared

top-scoring docked ligand pose and the complex crystal structure ligand pose was evaluated for each complex in the test set. The success rate over the entire test set using the optimized flexible ligand anchor-and-grow protocol was 72% (82/114) with an average time of 314 s per complex.

Dependence on anchor position

The anchor-and-grow algorithm belongs to the set of incremental construction algorithms for searching ligand conformational space [14, 15]. It uses a rigid docking step for the ‘‘anchors’’ to identify likely anchor positions (anchor orienting), and a torsion angle search step to generate ligand conformations rooted at the previously identified anchor positions (flexible growth). In order for flexible docking to succeed, both of these individual steps must be successful.

To measure the dependence of success rate on the precision of the anchor location, the crystal position of the anchor for each complex in the test set was perturbed randomly from 0 Å to more than 10 Å. Each perturbed anchor position was then considered as the starting point for flexible growth (Fig. 3b). With the anchor starting less than 0.5 Å heavy atom RMSD from the crystal orientation, the growth algorithm can find the experimental orientation 99% of the time. However, the results demonstrate a rapid decrease in success rate as the anchor is moved further away from its crystal structure position, decreasing to 76% at 1.0 Å perturbation down to 54% at 2.0 Å. These data imply that if the flexible ligand docking algorithm can place the anchor within 0.5 Å heavy atom RMSD of the crystal anchor position, DOCK 5 has a very high probability of successfully predicting the full binding mode correctly.

Analysis of total conformational ensemble

We examined the entire ensemble of conformers generated by flexible docking, as we described previously in the rigid ligand docking analysis. Considering all docked ligand conformations with a 2.5 DOCK unit energy gap—an average of five ligand orientations—increases the success rate to 82%, while an average of 100 orientations increasing the success rate to 95% (Fig. 4a, b). Again, these results indicate that the sampling density produced by the optimized parameters is quite high, but there is little discrimination between very similar poses by the current scoring function.

Geometric clustering of poses

As with the rigid ligand docking tests, each conformational ensemble was spatially clustered according to interpose RMSD (see Methods section for algorithm details). A clustering threshold of 1.0 Å, as determined in the rigid docking section, was used (Fig. 5). Using this clustering threshold, only 50 clusterheads must be examined to reach a success rate of 95% as compared to 100 purely ranked orientations. Once again, this result is encouraging, as it requires a small number of ligand poses to be retained for rescoring with more advanced scoring functions that are better at discriminating between very similar ligand poses.

Comparison to DOCK 4

Using the optimized DOCK 5 parameters, we performed the same rigid and flexible ligand docking experiments on the entire test set using the last available version of DOCK 4. The performance of the current implementation of DOCK 5 compared favorably with the DOCK 4 performance (Table 4). We attribute the improved accuracy in performance to improvements outlined in the Methods Section. However, when comparing the speed of docking experiments between DOCK 4 and DOCK 5, DOCK 4 is fivefold faster for rigid docking and 30-fold faster for flexible ligand docking than DOCK 5 (Table 5). We attribute this increased calculation time to extra stages of minimization and sampling in DOCK 5, as well as additional overhead necessary to preserve the modularity of the code (see Methods).

Comparison to other docking methods

Developers of Glide, GOLD and FlexX have also evaluated their methods using similar test sets and made some of their analyses available [9, 45, 46]. Based on this data, we note that DOCK's flexible docking success rate of 70% is comparable to Glide's and FlexX's success rates of 82% and 61%, respectively (Table 6). Unfortunately, GOLD has not posted the results for the entire CCDC/Astex test set, so a complete comparison could not be made. However, for the subset of the test set they did report, DOCK's success rate of 67% is once again reasonable as compared to the success rate of 77% for GOLD, considering that the DOCK scoring function does not use either empirically weighted parameters or adjustable parameters.

Table 4 Success based on DOCK version (see Methods)

DOCK version	Rigid ligand	Flexible ligand
4.0.1	71.9%	42.1%
5.4.0	79.0%	71.9%

Table 5 Average length of time in seconds for docking calculation using the optimized parameter set (see Appendix 1)

	Average	Minimum	Maximum
DOCK 4 rigid lig	10.9 ± 12.1	0.99	66.8
DOCK 4 flexible lig	7.1 ± 6.04	0.44	33.5
DOCK 5 rigid lig	55.4 ± 37.5	6.0	198.0
DOCK 5 flexible lig	314.7 ± 449.8	2.0	2638.0

Table 6 Comparison of DOCK success rates to other docking programs for flexible ligand docking

Program	No. of complexes	Success	DOCK success
GOLD	43	77%	67%
Glide	71	82%	70%
FlexX	71	61%	70%

Analysis of successes and failures of docking protocols

Docking failures can be categorized into two categories: sampling (soft) and scoring (hard) failures [47]. For scoring failures, an orientation near the crystal structure was sampled in the course of the DOCK run, but the scoring function failed to rank it at the top of the list. A sampling failure indicates that the DOCK run failed to sample any orientations within 2 Å RMSD of the crystal structure. The major caveat of this classification scheme is the assumption that the model of both the receptor and the ligand, including the VDW parameters, electrostatics, and hydrogen orientations and protonation states, reflect those that occur in the experimental structure [48]. Here, we analyze the flexible docking ligand failures within the sampling-scoring classification scheme.

Failures resulting from receptor modeling/structural problems

The original CCDC/Astex test set was filtered for experimental errors using a variety of metrics [22]. We plotted the flexible ligand success rate as a function of various metrics of the quality of the X-ray structures to determine if the selection criteria were appropriate for testing the DOCK algorithm (Fig. 6). There appears to be at best a weak correlation between the RMSD of the best scoring DOCK pose and either crystal resolution or b-factor of active site or backbone atoms, indicating that the cut-offs chosen for the original set were reasonable for docking purposes.

We next explored whether specific atom types caused problems with the DOCK force field terms by correlating the test set success rate with the presence and type of active site cofactor (Table 7). The only clear problem involved metal ions in the receptor. These structures showed a much lower success rate, accounting for nearly half of both the rigid and flexible ligand docking failures. However, there still are a number of failures in the portion of the test set without cofactors in the active site that require further characterization. Unless otherwise mentioned, all studies below were performed on this subset, referred to as the Cofactor Free (CF) subset.

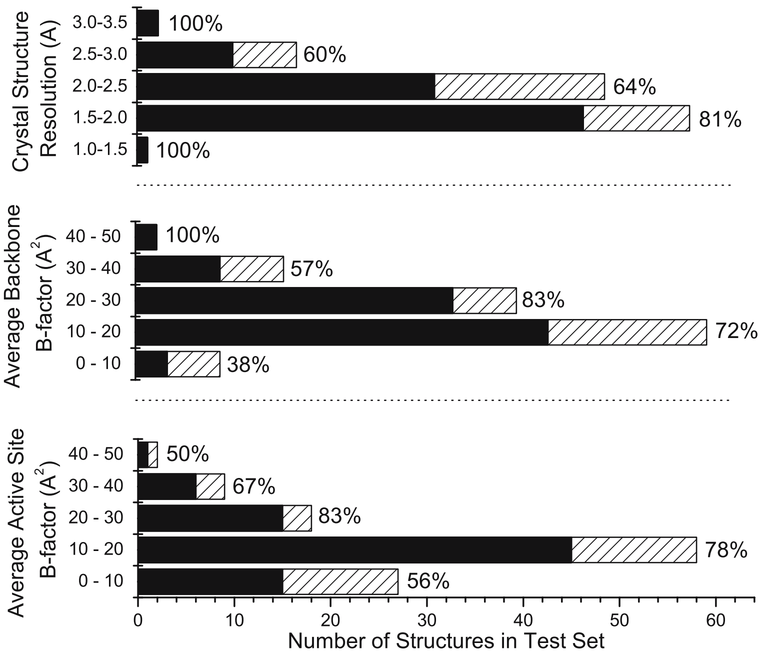

Fig. 6 Correlation of flexible ligand success (filled) and failure (striped) rates with crystallographic resolution (Å) and experimental B-factor (Å^2). For active site B-factors, the active site was defined as any atom within 9 Å of the experimental ligand orientation

Table 7 Success as function of active site cofactor

	Total count	Rigid success	Flexible success
Entire test set	114	79.0%	71.9%
CF subset	76	81.6%	76.3%
Active site cofactor	38	73.7%	63.2%
Active site metal cofactor	28	64.3%	50.0%

For all members of the test set, the experimental resolution of the crystal structures was too poor to identify hydrogen atom locations. We originally modeled the hydrogen atom positions using a rule-based method. To test this scheme, we applied a more advanced hydrogen addition procedure that accounted for steric clashes and hydrogen-bonding networks to the CF subset (see Methods). As a follow-up, we assumed all crystallographically bound waters found within 3 Å of any ligand heavy atom were critical for binding and included them in the receptor model as well. We found that both of these procedures improved the flexible ligand docking success rate (Table 8).

Table 8 Flexible ligand success as function of CF test set preparation (total of 76 complexes)

Test set preparation technique	Success
Standard	76.3%
Hydrogen optimization	78.9%
Active site waters + hydrogen optimization	80.3%

Failures resulting from ligand flexibility

In addition to the selection criteria imposed on the original test set, we also filtered out complexes in which the ligand had greater than seven rotatable bonds (see Methods). We reexamined this choice on the CF subset by plotting the rigid and flexible ligand docking success rate as a function of the number of flexible bonds (Fig. 7). As expected, the results show a decrease in the success rate with increasing ligand size, but with no dramatic drop-off.

Sampling versus scoring failures

We now return to classification of DOCK failures based on scoring and sampling classifications [47]. First, we examined the test set failure cases with active site cofactors (Table 9). Within this set, nine examples were scoring failures for both rigid and flexible ligand docking, indicating that new VDW and electrostatic parameters need to be developed for magnesium, heme groups, and some coordination states of zinc. In addition, there were three flexible ligand scoring failures that were rigid successes, thus suggesting that the flexible algorithm was able to identify additional orientations with better scores than the experimental ligand orientation. Only two flexible ligand docking cases were sampling failures. We expected flexible ligand docking sampling failures due to the increased ligand degrees of freedom compared with rigid ligand docking, but it does not appear to be a severe problem in this test set containing ligands with less than eight

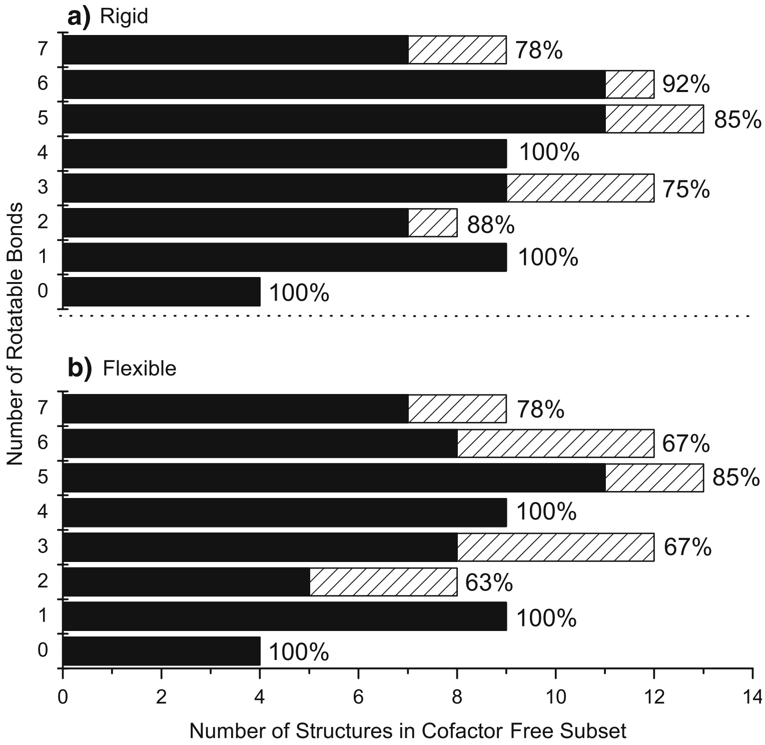

Fig. 7 Rigid and flexible docking success (filled) and failure (striped) rates as a function of the number of rotatable bonds in each ligand in CF test set

Table 9 Comparison of success and failure cases of both rigid and flexible docking for complexes in test set with cofactors in active site (total of 36 complexes)

	Rigid sampling failure	Rigid scoring failure	Rigid success
Flexible sampling failure	0	0	2
Flexible scoring failure	0	9	3
Flexible success	0	1	23

Table 10 Comparison of success and failure cases of both rigid and flexible docking for complexes in CF subset (total of 76 complexes)

	Rigid sampling failure	Rigid scoring failure	Rigid success
Flexible sampling failure	1	1	2
Flexible scoring failure	0	7	7
Flexible success	0	5	53

rotatable bonds. Finally, one of the rigid ligand docking scoring failures was a flexible ligand success. In this case, there was a large VDW clash between one of the ligand atoms and the receptor. The anchor-and-grow algorithm was able to build the ligand in the active site to avoid this clash, which the rigid ligand docking algorithm could not accommodate.

We repeated this analysis with the CF subset (Table 10). Here, there was one rigid ligand docking sampling failure, which also failed for flexible ligand docking. Upon closer examination of the receptor site, a residue making critical interactions with the ligand was not resolved in the experimental complex structure (PDB code 1A6W). We anticipate that there may not be enough contacts to correctly place the molecule. Seven examples were scoring failures for both rigid and flexible ligand docking. In this subset, though, we cannot attribute the failure to unusual atom types, indicating that the scoring function is incorrectly modeling some portion of the energy landscape. There were also seven scoring failures for flexible ligand docking that were successes for rigid ligand docking, once again suggesting that the flexible docking algorithm identified additional orientations that scored better than the experimental orientation.

As in the cofactor set above, there were only three additional flexible ligand docking sampling failures. One of these was also a scoring failure in rigid ligand docking, implying that this failure case may actually be due to a combination of both sampling and scoring factors. The remaining two flexible ligand docking sampling failures once again indicate that the flexible algorithm was able to identify alternative orientations that scored better than the crystal complex orientation. Finally, five rigid ligand docking scoring failures were flexible ligand dockings successes, signifying that the flexible ligand docking algorithm is able to compensate for intermolecular clashes in the active site of the experimental structure that the rigid ligand algorithm simple cannot accommodate (data not shown).

Analysis of DOCK score for docking protocols

To analyze the ability of DOCK to reproduce the ligand–receptor interaction energy as measured by the DOCK scoring function, we plotted the score from the top-ranking pose for both rigid and flexible ligand docking that were successful against the DOCK score of the complex crystal structure (Fig. 8a, b). Each crystal structure ligand was minimized with 1,000 steps of the DOCK simplex minimizer. The significant feature of both plots is that the docked pose generally scores more favorably than the minimized crystal structure. When rigid ligand docking is compared with flexible ligand docking, the flexibly docked ligand conformations almost always have a lower score (Fig. 8c). These results indicate that increasing the amount of ligand orientational and conformational sampling increasingly identifies deeper wells in the binding energy landscape. When we plotted the flexible ligand success rate against the minimized crystal score, there was little correlation, though DOCK was observed to perform better using crystal structures with scores more negative than –20 DOCK units (Fig. 8d). This lack of correlation indicates that, while having a negative interaction energy for the crystal structure will increase the probability of DOCK finding the correct binding orientation, this metric is not a good predictive indicator of DOCKing success.

Database docking using MPI

Substantial speedup is observed for up to about 14 processors for the 500 compound library and 18 processors for the 1,000 compound library (Fig. 9). Interestingly, the library with 1,000 copies of neviripine shows almost perfectly parallel behavior up to 68 processors. We hypothesize that the speedup for the heterogeneous libraries will continue to approach ideal as larger libraries with increased numbers of rotatable bonds are used, but will never be completely linear due to overhead from input and output and lag resulting from communication between the nodes.

Discussion

In this paper we have described a new version of the DOCK program. Our main purpose was to develop modular code that was straightforward to modify and

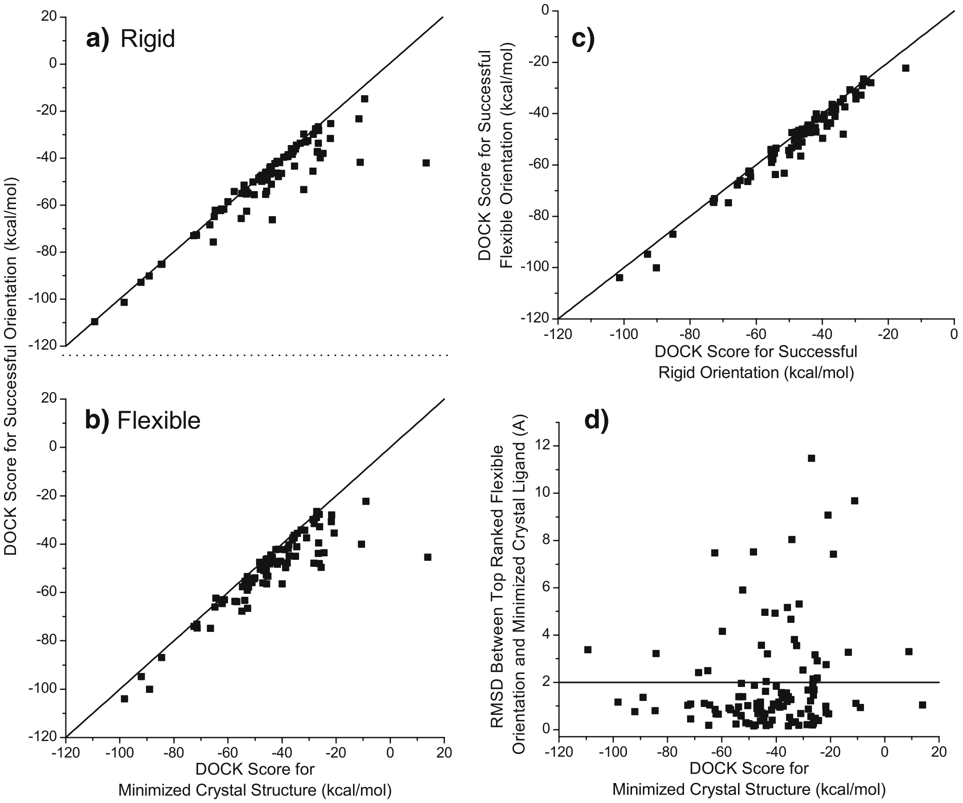

Fig. 8 (**a**) Successful rigid ligand docking scores (kcal/mol) as a function of minimized crystal structure ligand scores (kcal/mol), (**b**) Successful flexible ligand docking scores (kcal/mol) as a function of minimized crystal structure ligand scores (kcal/mol), (**c**) Successful flexible ligand docking energy scores (kcal/mol) as a function of successful rigid ligand docking energy scores (kcal/mol), (**d**) Comparison of the RMSD between all top ranked flexible ligand orientations and the minimized crystal ligand orientations to the minimized crystal interaction energy as measured by the DOCK score (kcal/mol)

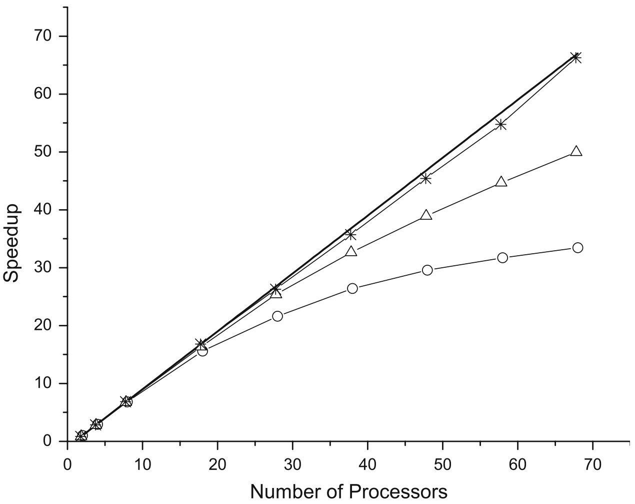

Fig. 9 Speedup (calculated as length of time for calculation on a single processor/length of time for calculation on n processors) for docking a library of 500 different small molecules (○), 1,000 different small molecules (△), and 1,000 copies of nevaripine (✳) using flexible ligand docking as a function of the number of processors in MPI mode. A perfectly parallel calculation (–) is plotted for comparison

which showed improved performance over the old version. By using an object-oriented language for DOCK 5, we were able to accomplish this goal, and we demonstrate, here, how routines such as the simplex minimizer and the clustering algorithm can be added or replaced without changes in other parts of the program. The successful parallelization of the calculation and the addition of post-processing clustering were simple but useful modifications to the algorithm, which encourages further investigations and algorithm experimentation.

The performance of DOCK 5 on a curated test set of 114 protein–ligand complexes proved to be superior to DOCK 4, with an over-all success rate of 79% for rigid ligand docking and 72% for flexible ligand docking, compared with 72 and 42%, respectively for DOCK 4. We ascribe the improvements to significant changes in the flexible search sampling and pruning procedures and to code corrections. The difference in performance of DOCK 5 for rigid and flexible docking is relatively modest (79% vs. 72%) even though the search for flexible ligands includes both configurational and conformational spaces. Using the receptor structure to prune the conformational search tree is clearly a reasonably efficient procedure. Although, the DOCK 5 code takes longer on average to run a calculation than DOCK 4, we feel this drawback is balanced by the improved results and the modularity of DOCK 5. Efforts to increase throughput are underway.

We also wish to stress the importance of having a high quality test set for evaluation of docking programs. X-ray crystallography typically provides essential but incomplete data for the calculations we wish to carry out. For example, in the majority of cases, hydrogen positions must be determined. In other cases, critical water molecules must be placed and some residues need to be modeled where experimental data is lacking. The ligand conformations may also contain significant uncertainties. Finally, we must be aware of the inherent assumptions underlying the force field parameters used in the molecular modeling steps. All of these considerations speak to the need for careful inspection of test set complexes. Our results demonstrate this issue: the success rate for reconstitution of the complex geometries was shown to depend on the nature of the cofactors, the optimization of hydrogen placements, and the inclusion of critical waters.

The primary result that emerges from the analysis of the docking failures is that the current force field requires improvement, particularly in the treatment of metal-containing cofactors. We also note that binding conformations and configurations are determined by the free energy of the system while we are only, at best, estimating the enthalpy. Finally, we do identify a few situations in flexible ligand docking where the conformational sampling is insufficient. A test set with ligands containing more than seven rotatable bonds would, presumably, show an increase in these sampling failures. We hypothesize that the key weakness is the pruning algorithm, which we will explore in future studies.

What are the routes to improvement? An obvious starting point is the use of more accurate methods for preparing experimental structures, including tools for accurate pK_a prediction and *de novo* identification of critical waters. For the docking calculation itself, it would be helpful to improve VDW and electrostatic parameters for all atoms heavier than oxygen, particularly for metal atoms. Ideally, one would directly include charge polarization and ligation geometry in the force field. In addition, modifications to the force field to better approximate the free energy—e.g. generalized Born or Poisson Boltzmann implicit solvation electrostatics with surface area corrections to account for the hydrophobic effect—would also improve modeling accuracy. The DOCK 5 platform is positioned to enable future developments and work is underway to incorporate them into future releases.

Conclusions

In this study, we have evaluated a new version of DOCK. We have found that it predicts binding geometries of a structurally diverse test set comparably to similar algorithms and better than the previous version of DOCK. Simultaneously, we have thoroughly explored the sampling portions of the algorithm and found that the majority of binding pose prediction failures is a result of scoring function deficiencies. In further exploration of these failures, we have determined that the docking success seems to be a function of whether there are alternative orientations that score well—as defined by the scoring function—rather than the interaction energy of the experimental structure itself. Finally, we have implemented new functionalities and shown that they improve the success rates of both rigid and flexible ligand docking. In general, we have a new tool that not only performs well on a typical test set but is an ideal tool to explore any number of new algorithms in the context of the molecular docking problem.

Acknowledgements Gratitude is expressed to Dr. Bentley Strockbine and Sudipto Mukherjee for computational assistance with MPI calculations. Demetri Moustakas, Natasja Brooijmans, P. Therese Lang and Irwin D. Kuntz would like to thank the NIH grant GM 56531 (Paul Ortiz de Montellano, PI) for support. P. Therese Lang would also like to thank the Burroughs Welcome Foundation and the American Foundation for Pharmaceutical Education for additional support. The authors would like to thank Scott Brozell, Mathew Jacobson, and Brian Shoichet and members of his group for helpful conversations.

Appendix 1

Rigid docking parameter optimization

The parameters listed in Appendix 1 control the sampling of ligand poses within the receptor active site during rigid ligand docking. The parameters that control the step sizes for the simplex minimizer (simplex_trans_step, simplex_rot_step, and simplex_tors_step) were optimized in a previous study and were held at those values [14, 49]. For the remaining parameters—the number of orientations (max_orientations) and the number of minimization steps (simplex_final_max_iterations)—a series of rigid ligand docking experiments were performed to optimize the DOCK score for the top ranking pose averaged over the entire test set and the success rate, defined as the orientation of the top ranking pose being within 2 Å heavy atom RMSD from the crystal ligand. The success rate and DOCK scores initially improved as the number of orientations and the amount of minimization increased and then converged (Fig. 10). We selected the lowest converged values—1,000 orientations and 1,000 minimization steps—as optimal.

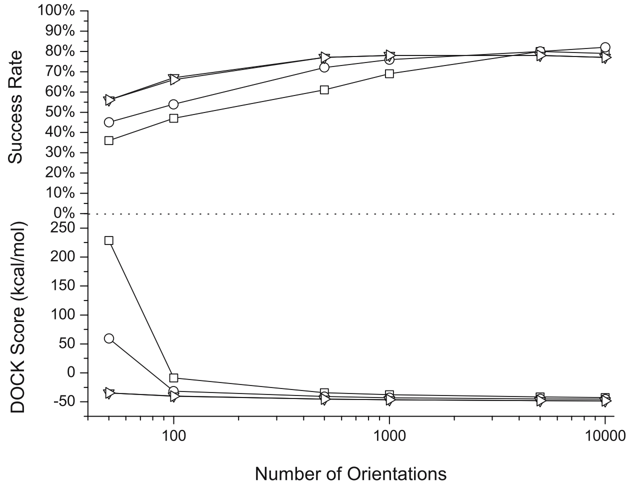

Fig. 10 Optimization of parameters for rigid ligand docking. Parameters of 50 (□), 100 (○), 1,000 (▽), and 10,000 (▷) minimization steps (simplex_final_max_iterations) are examined as a function of the number of orientations (max_orientations)

Flexible docking parameter optimization

For the more complex flexible ligand algorithm, the parameter optimization was performed first on the anchor docking, and the best parameters were then used for optimizing the growth. The parameters that control the sampling in both these steps are listed in Appendix 2. As for rigid ligand docking, the parameters that control step sizes for the simplex minimizer were set to the previously defined optimal values.

Appendix 1 Description of and optimized default values for parameters that affect rigid ligand docking

Parameter name	Parameter description	Value
max_orientations	The number of ligand poses sampled by the rigid orienting algorithm	1,000
simplex_score_converge	The score threshold used to determine simplex convergence	0.1
simplex_trans_step	The maximum initial translation step size for the simplex minimizer	1.0 Å
simplex_rot_step	The maximum initial rotational euler angle step size for the simplex minimizer	0.1 radian
simplex_tors_step	The maximum initial dihedral angle step size	10°
simplex_final_max_iterations	The maximum number of simplex iterations	1,000

Appendix 2 Description of and optimized default values for parameters that affect flexible ligand docking

Parameter name	Parameter description	Value
max_orientations	The number of anchor poses sampled by the rigid orienting algorithm	500
num_anchor_orients_for_growth	The maximum number of anchor orientations promoted to the conformational search	100
num_confs_for_next_growth	The number of partially grown ligand conformers stored at each stage of the flexible growth procedure	100
simplex_anchor_max_iterations	The maximum number of simplex iterations applied to the ligand anchor during anchor docking	500
simplex_grow_max_iterations	The maximum number of simplex iterations applied to the ligand during the flexible growth procedure	500

The first step in the anchor-and-grow algorithm is ring identification or anchor segmentation. All bonds within molecular rings are treated as rigid. This classification scheme is a first-order approximation of molecular flexibility, since some amount of flexibility can exist in non-aromatic rings. To treat such phenomena as sugar puckering and chair-boat hexane conformations, the user needs to supply each ring conformation as a separate input molecule. If the molecule does not have a ring, the largest rigid segment is specified as the anchor. Additional bonds may be specified as rigid by the user. For simplicity, all runs in this study used the default of largest anchor only. If the molecule had multiple anchors of the same size, the first anchor on the anchor list was used. Once the anchor had been identified, the parameters that control the number of anchor orientations (max_orientations), the number of anchor minimization steps (simplex_anchor_max_iterations), and the cutoff for the anchor pruning (num_confs_for_next_growth) were explored. Because the anchors are substructures of the ligand, the parameter convergence was monitored as a function of the RMSD between the anchor orientation and the corresponding substructure of the crystal ligand averaged over all generated orientations before the pruning function. When the number of anchor orientations and minimization steps were varied systematically, the number of minimization steps converged at 500 (Fig. 11a). We expected this optimized value to be lower than rigid docking because anchors are typically smaller than the final ligand.

Because the anchor orientations are pruned before the growth step, we used the optimized number of minimization steps while exploring the number of anchor orientations and the pruning cutoff. The optimal anchor pruning cutoff of 100 was chosen as a balance between convergence and the length of the calculation, which remained fixed for the final exploration of the number of orientations. The optimal number of orientations was selected to be 500 because the combination of these three variables generated the highest number of anchors near the crystal structure (Fig. 11a). Note that if the number of orientations was increased beyond the selected value, the number of anchors near the crystal structure dropped dramatically. We hypothesized that this resulted from a combination of increased sampling and pruning. The pruning function was designed to identify a representative orientation from each energy well that the matching algorithm finds (see Introduction: DOCK background). As sampling increased, the ranked orientations began to converge toward the bottom of the deepest energy wells, sampling less of the alternative high energy wells. Because the pruning function is designed to supply the most diverse ligands, fewer orientations made it through the pruning step as the sampling is increased. We felt that this effect was reducing the potential sampling for the algorithm and plan to explore alternatives in future studies.

The next step in the anchor-and-grow algorithm is flexible bond identification. Each flexible bond is associated with a label defined in an editable file. The parameter file is identified with the flex_definition_file parameter. Each label in the file contains a definition based on the atom types and chemical environment of the bonded atoms. Typically, bonds with some degree of double bond character are excluded from minimization so that planarity is preserved. Each label is also associated with a set of preferred torsion positions. The location of each flexible bond is used to partition the molecule into rigid segments. A segment is the largest local set of atoms that contains only non-flexible bonds.

Using the optimal anchor parameters, we varied number of minimization steps for each layer of growth (simplex_grow_max_iterations) and the cutoff of number of conformers for the growth pruning function (num_confs_for_next_growth). Because the dock run now creates a complete pose, we return to using a combination of the score for the top ranking pose averaged over the entire test set and the success rate to

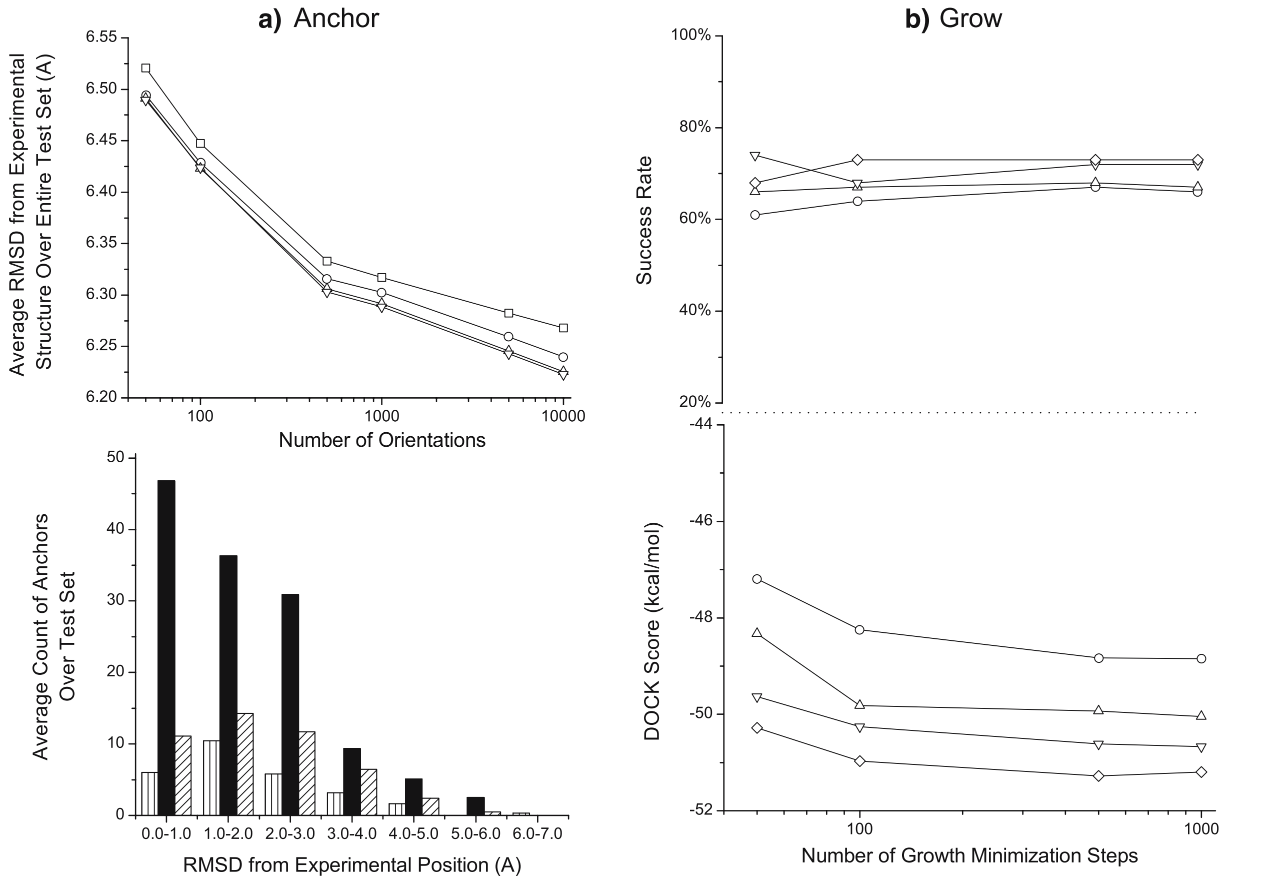

Fig. 11 Optimization of parameters for flexible ligand docking. (**a**) Parameter optimization for anchor sampling portion of flexible ligand docking. TOP: Parameters of 0 (□), 50 (○), 100 (△), and 500 (▽) anchor minimization steps (simplex_anchor_max_iterations) are plotted as a function of the number of orientations (max_orientations). BOTTOM: Parameters of 50 (vertical stripes), 500 (filled), and 5,000 (diagonal stripes) anchor orientations (max_orientations) are compared using an anchor pruning cutoff (num_confs_for_next_growth) of 100. (**b**) Parameter optimization for growth sampling portion of flexible ligand docking. Growth pruning cutoffs (num_confs_for_next_growth) of 25 (○), 50 (△), 100 (▽), and 200 (◇) are plotted as a function of the number of growth minimization steps (simplex_grow_max_iterations)

monitor convergence. As with rigid ligand docking, the success rate improves modestly with improved sampling and eventually converges (Fig. 11). However, although DOCK scores improved as the number of orientations and the amount of minimization increased, the values do not converge. We once again attribute this phenomenon to the pruning function. Therefore, we used the success rate to select the lowest converged values—500 minimization steps and the cutoff for the number of conformers for the growth section as 100—as optimal.

References

1. Kopec KK, Bozyczko-Coyne D, Williams M (2005) Biochem Pharmacol 69:1133
2. Congreve M, Murray CW, Blundell TL (2005) Drug Discovery Today 10:895
3. Kraljevic S, Stambrook PJ, Pavelic K (2004) EMBO Rep 5:837
4. Schnecke V, Bostrom J (2006) Drug Discovery Today 11:43
5. Hillisch A, Pineda LF, Hilgenfeld R (2004) Drug Discovery Today 9:659
6. Posner BA (2005) Curr Opin Drug Discovery Dev 8:487
7. Alvarez JC (2004) Curr Opin Chem Biol 8:365
8. Verdonk ML, Cole JC, Hartshorn MJ, Murray CW, Taylor RD (2003) Proteins 52:609
9. Friesner RA, Banks JL, Murphy RB, Halgren TA, Klicic JJ, Mainz DT, Repasky MP, Knoll EH, Shelley M, Perry JK, Shaw DE, Francis P, Shenkin PS (2004) J Med Chem 47:1739
10. Halgren TA, Murphy RB, Friesner RA, Beard HS, Frye LL, Pollard WT, Banks JL (2004) J Med Chem 47:1750
11. Kramer B, Rarey M, Lengauer T (1999) Proteins 37:228
12. Kitchen DB, Decornez H, Furr JR, Bajorath J (2004) Nat Rev Drug Discovery 3:935
13. Shoichet BK, Bodian DL, Kuntz ID (1992) J Comput Chem 13:380
14. Ewing TJA, Kuntz ID (1997) J Comput Chem 18:1175
15. Leach AR, Kuntz ID (1992) J Comput Chem 13:730
16. Meng EC, Shoichet BK, Kuntz ID (1992) J Comput Chem 13:505
17. Lischner R (2003) C++ in a nutshell. 1st edn. O'Reilly Media, Inc, Sebastopol, CA

18. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) Nucleic Acids Res 28:235
19. Jones G, Willett P, Glen RC, Leach AR, Taylor R (1997) J Mol Biol 267:727
20. Pang YP, Perola E, Xu K, Prendergast FG (2001) J Comput Chem 22:1750
21. Perola E, Walters WP, Charifson PS (2004) Proteins 56:235
22. Nissink JW, Murray C, Hartshorn M, Verdonk ML, Cole JC, Taylor R (2002) Proteins 49:457
23. Kuhl FS, Crippen GM, Friesen DK (1984) J Comput Chem 5:24
24. Nelder JA, Mead R (1965) Comput J 7:308
25. Gropp W, Lusk E, Doss N, Skjellum A (1996) Parallel Computing 22:789
26. SYBYL, Tripos, Inc., St. Louis, Missouri, 63144
27. Case DA, Darden TA, Cheatham III, TE, Simmerling CL, Wang J, Duke RE, Luo R, Merz KM, Wang B, Pearlman DA, Crowley M, Brozell S, Tsui V, Gohlke H, Mongan J, Hornak V, Cui G, Beroza P, Schafmeister C, Caldwell JW, Ross WS, Kollman PA (2004) AMBER 8, University of California, San Francisco
28. Jakalian A, Bush BL, Jack DB, Bayly CI (2000) J Comput Chem 21:132
29. Hann MM, Oprea TI (2004) Curr Opin Chem Biol 8:255
30. Oprea TI (2002) J Comput-Aided Mol Des 16:325
31. Oprea TI, Davis AM, Teague SJ, Leeson PD (2001) J Chem Inf Model 41:1308
32. Brooijmans N (2003) Theoretical studies of molecular recognition, Graduate Department of Chemistry and Chemical Biology, University of California, San Francisco, San Francisco, CA
33. Purcell WP, Singer JA (1967) J Chem Eng Data 12:235
34. Gasteiger J, Marsili M (1980) Tetrahedron 36:3219
35. Aqvist J, Warshel A (1990) J Am Chem Soc 112:2860
36. Merz KM, Murcko MA, Kollman PA (1991) J Am Chem Soc 113:4484
37. Cornell WD, Cieplak P, Bayly CI, Gould IR, Merz KM, Ferguson DM, Spellmeyer DC, Fox T, Caldwell JW, Kollman PA (1995) J Am Chem Soc 117:5179
38. Richards FM (1977) Ann Rev Biophys Bioeng 6:151
39. DesJarlais RL, Sheridan RP, Seibel GL, Dixon JS, Kuntz ID, Venkataraghavan R (1988) J Med Chem 31:722
40. Kuntz ID, Blaney JM, Oatley SJ, Langridge R, Ferrin TE (1982) J Mol Biol 161:269
41. Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, Ferrin TE (2004) J Comput Chem 25:1605
42. Meng EC, Lewis RA (1991) J Comput Chem 12:891
43. Mills JEJ, Dean PM (1996) J Comput-Aided Mol Des 10:607
44. Irwin JJ, Shoichet BK (2005) J Chem Inf Model 45:177
45. The results for the FlexX test set are available at http://www.biosolveit.de/FlexX/
46. The results for the GOLD test set are available at http://www.ccdc.cam.ac.uk/products/life_sciences/validate/gold_validation/value.html
47. Verkhivker GM, Bouzida D, Gehlhaar DK, Rejto PA, Arthurs S, Colson AB, Freer ST, Larson V, Luty BA, Marrone T, Rose PW (2000) J Comput-Aided Mol Des 14:731
48. Kuntz ID, Agard DA (2003) Adv Protein Chem 66:1
49. Gschwend DA, Kuntz ID (1996) J Comput-Aided Mol Des 10:123