AutoDock-based incremental docking protocol improves docking of large ligands

Ankur Dhanik,[†] John S McMurray,[‡] and Lydia E Kavraki[*,†]

Department of Computer Science, Rice University, Houston, Texas, USA, and Department of Experimental Therapeutics, University of Texas MD Anderson Cancer Center, Houston, Texas, USA

E-mail: kavraki@rice.edu

Phone: +1 713 3485737. Fax: +1 713 3485930

Abstract

It is well known that computer-aided docking of large ligands, with many rotatable bonds, is extremely difficult. *AutoDock* is a widely used docking program that can dock small ligands, with upto 5 or 6 rotatable bonds, accurately and quickly. Docking of larger ligands, however, is not very accurate and is computationally expensive. In this paper we present an *AutoDock*-based incremental docking protocol which docks a large ligand to its target protein in increments. A fragment of the large ligand is first chosen and then docked. Best docked conformations are incrementally grown and docked again, and this process is repeated until all the atoms of the ligand are docked. Each docking operation is performed using *AutoDock*. However, in each docking operation only a small number of rotatable bonds are allowed to rotate. We did a systematic docking study on a dataset of 73 protein-ligand complexes derived from the core set of PDBbind database. The number of rotatable bonds in the ligands vary

[*] To whom correspondence should be addressed

[†] Rice University

[‡] University of Texas MD Anderson Cancer Center

from 7 to 30. Multiple docking experiments were done to evaluate the docking performance of the incremental protocol in comparison to *AutoDock*'s standard protocol. Results from the study show that, on average over the dataset, docking of large ligands using our incremental protocol is upto 23-fold computationally faster than docking using *AutoDock*'s standard protocol and also has better or comparable accuracy. We propose that, for docking large ligands, our incremental protocol can be used as an alternative to *AutoDock*'s standard protocol.

Introduction

Computer-aided docking is an important tool for gaining understanding of the binding interactions between a ligand and its target receptor (mainly, a protein).[1–3] Over the years, several docking programs have been developed (see representative examples[4–12]) and many docking studies have been performed (e.g.,[13–21]). Typically, docking programs explore the conformation space of the ligand to generate a conformation bound to the target protein. A conformation is usually defined by the torsional degrees of freedom around the rotatable bonds of the ligand and the translation and orientation degrees of freedom. In general, the docking programs are fairly accurate and computationally fast when docking small ligands with 5 or 6 rotatable bonds.[21,22] However, accurate and fast docking of large ligands with many rotatable bonds is still very challenging mainly because of the high-dimensionality of the conformation spaces which need to be explored to dock such large ligands.

Our strategy for docking large ligands derives inspiration from fragment-based docking methods. Some of the fragment-based methods place several fragments of a ligand in the binding cavity and then construct the full ligand from the best placed fragments, while other methods anchor a fragment of the ligand in the binding site and then build the full ligand incrementally. Some common docking programs that utilize such fragment-based approaches include LUDI,[23] FlexX,[7] GROWMOL,[24] HOOK,[25] Q-fit,[26] and SURFLEX.[9] Multistep docking strategies have also been employed for flexible ligand docking. For example, in docking approach by Wang and coworkers,[27] the conformation space corresponding to the ligand's rotatable bonds and that corresponding to the rigid body (translation,orientation) degrees of freedom are explored separately. The exploration of the conformation space corresponding to the unbound ligand's rotatable bonds is done using a multistep approach where the ligand is explored a few rotatable bonds at a time. Docking programs such as DOCK[28] and MS-DOCK[29] use a similar multistep approach for exploring the ligand conformations.

In this paper, we present an incremental protocol for docking large ligands where at each increment the docking operation is done using *AutoDock*.[8,30] *AutoDock* is an excellent non-commercial docking program that is widely used. It employs a stochastic Lamarckian genetic algorithm for computing ligand conformations and simultaneously minimizing its scoring function which approximates the thermodynamic stability of the ligand bound to the target protein. *AutoDock* performs well when docking small ligands;[21] it is accurate as well as computationally fast. For docking large ligands, *AutoDock* recommends increasing the limit on the maximum number of energy evaluations (parameter called *ga_num_evals*). Increasing the limit essentially amounts to a more exhaustive search for the docked conformation in the conformation space. For the large ligands considered in this paper, docking with the recommended parameter settings results in modest gains in the accuracy while significantly increasing computational time (see Results section). In our docking protocol, instead of exploring a very high-dimensional conformation space of the large ligand, we harness the strengths of *AutoDock* in a way such that it always explores low-dimensional subspaces of the conformation space.

Starting from an initial fragment of the ligand, our protocol repeatedly docks and grows fragments of the ligand. Each docking operation is done using *AutoDock*. First an initial fragment of the ligand, composed of a small number of rotatable bonds and atoms that are directly rotated by them, is chosen and all the rotatable bonds are set active. Only the active bonds are allowed to rotate in each docking operation which ensures that only the low-dimensional subspaces of the full conformation space of the ligand are explored. The initial fragment is docked and the lowest-scoring docked conformations are selected. The selected docked conformations are then grown by adding a few more rotatable bonds and corresponding atoms. The grown fragments are then docked but only a few of the rotatable bonds are set active. The best docked conformations are grown and docked again until all the atoms of the ligand are docked. We present the details of our incremental docking protocol in the Methods section.

Using our protocol, we performed a systematic docking study on a dataset of 73 protein-ligand complexes (7 to 30 rotatable bonds in the ligands) from the core set of the PDBbind database.[31] To compare the docking performance of our protocol, we did four different docking experiments as part of our study. One experiment was done using our protocol and three were done using *AutoDock*'s standard protocol in different parameter settings. The docking performance was mainly evaluated on the basis of docking accuracy and computational cost. Through the study we show that our protocol results in a significantly faster docking performance as compared to *AutoDock*'s standard protocol and the docking accuracy is either better or comparable.

Materials and Methods

Incremental docking protocol

We present an incremental protocol for docking large ligands using *AutoDock*.[8,30] In a typical docking operation done using *AutoDock*'s standard protocol, the conformation space of the ligand is explored to generate a docked conformation that is bound to the target protein and has minimum binding energy (computed using *AutoDock*'s scoring function). The conformation space is defined by the set of torsion angles corresponding to all rotatable bonds in the ligand and the translation and orientation degrees of freedom of the ligand. Therefore, in the case of a large ligand, with many rotatable bonds, the exploration of the conformation space for the minimum-energy docked conformation becomes very challenging. *AutoDock* recommends docking the large ligand by using the standard protocol with parameter setting that results in a more exhaustive exploration of the conformational space. Our protocol, on the other hand, docks the large ligand incrementally such that, at each increment, only a subspace of the conformation space is explored. We present a brief overview of our incremental docking protocol followed by a detailed description.

Overview Given a large ligand and a target protein, first an initial fragment of the ligand, composed of a small number of rotatable bonds and atoms directly rotated by the bonds, is chosen. All rotatable bonds in the initial fragment are set active. Only the active bonds are allowed to rotate and the fragment is docked to the target protein using *AutoDock*. Some of the docked conformations are then selected and grown by adding a few more bonds as well as atoms that are directly rotated by the new rotatable bonds. A small number of rotatable bonds in each grown conformation are set active and the conformations are docked. The docked conformations are grown and docked again, until all the atoms in the ligand are docked. Note that, in each docking operation, *AutoDock* explores a small number of rotatable bonds as well as the translation and orientation degrees of freedom, i.e., a subspace of the conformation space.

Description Our incremental protocol consists of several steps as shown in Figure 1. The protocol accepts as input:

(a) a ligand structure with bonds labeled as rotatable and non-rotatable,

(b) a target protein structure to which the ligand has to be docked,

(c) the center and dimensions of the *AutoDock* grid that encompasses the binding pocket in the protein,

(d) *ga_run*, number of conformations that are output after a docking operation done using *AutoDock*,

(e) *ga_num_evals*, maximum number of energy evaluations that are permitted by *AutoDock* when searching for the minimum of the binding energy,

(f) n_s (maxrot size), maximum number of rotatable bonds that are allowed to rotate and are, therefore, explored in each docking operation of our protocol, and

(g) k, number of docked conformations that are selected after a docking operation.

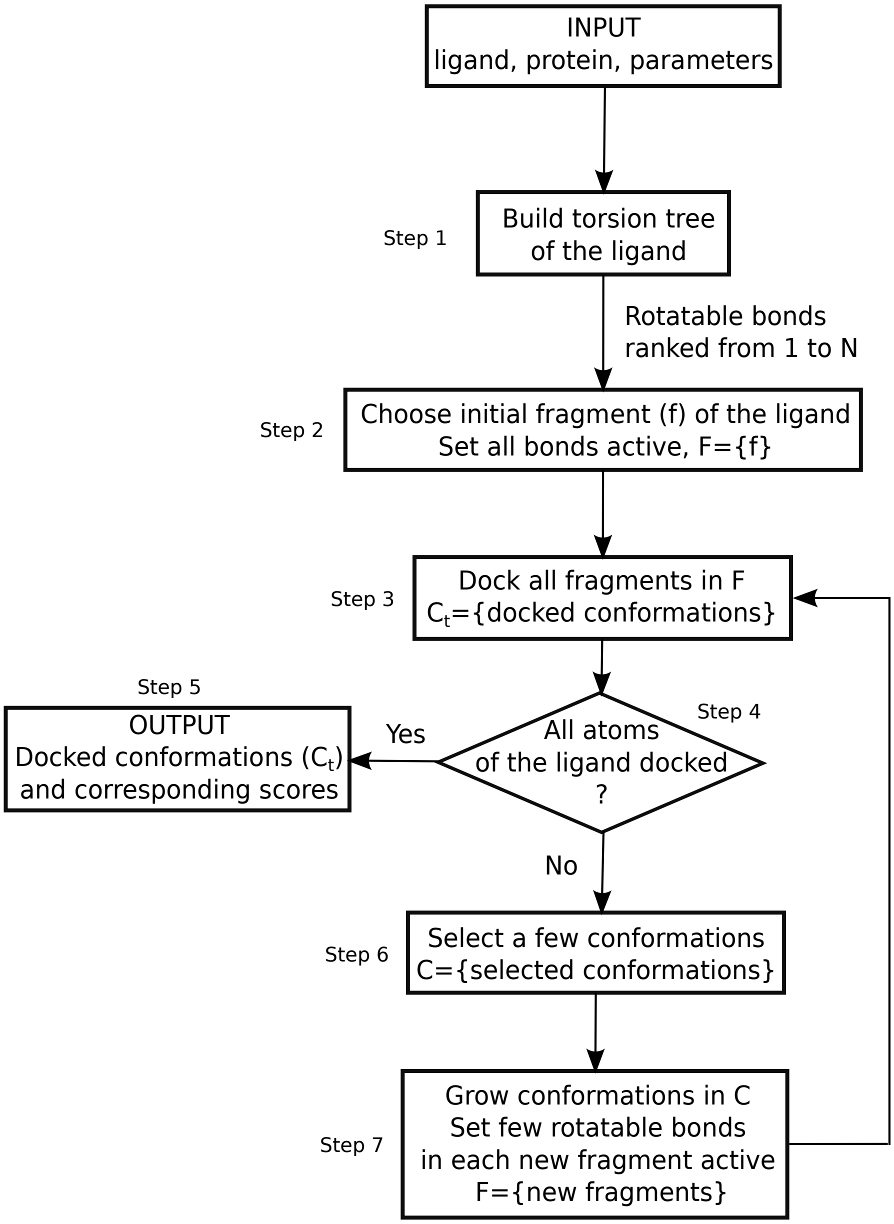

Figure 1: **Incremental docking protocol.** The flowchart shows the steps involved in our docking protocol. Given input ligand, a torsion tree is built and an initial fragment is chosen. The initial fragment is docked to the target protein and the best conformations are selected from the docked conformations. These conformations are grown into new fragments, and are docked again. Selecting best docked conformations, growing the selected conformations and docking is repeated until all the atoms of the ligand are docked. The fully docked conformations of the ligands and corresponding scores are then output.

Note that the parameters (c)-(e) are required by *AutoDock* for a standard docking operation and the parameter names are thus borrowed from *AutoDock* user guide.[32] The details of the steps in the incremental protocol that we propose are as follows.

Step 1 A torsion tree is constructed where a node represents a list of atoms and an edge represents a rotatable bond of the ligand. The tree is used for ranking the rotatable bonds, and the ranked bonds and the tree nodes are utilized in the next steps of the protocol. To construct the tree, first a root atom is selected. The root atom is selected from the set of all heavy atoms in the ligand such that the initial fragment derived from the torsion tree has the largest number of hydrogen bond donors and acceptors. The idea is to identify an initial fragment of the ligand that is likely to be involved in many binding interactions, thereby increasing its docking accuracy. The root atom can also be selected such that the initial fragment contains atoms that are involved in already known binding interactions, or it can be selected randomly. In the docking study described later, we show the impact of the choice of the root atom on the docking performance of our protocol. Let S_1 be a set that contains the root atom and atoms in the ligand that are connected to the root atom through a sequence of non-rotatable bonds. Let S_2 be a set that contains any atom that is not contained in S_1 and is bonded to at least one atom in S_1. All atoms in S_1 and S_2 are inserted into the root node. For each rotatable bond connected to the root atom, an edge and a node is added to the tree. The edge corresponds to the rotatable bond and the new node corresponds to the list of atoms that are directly rotated by the bond.

Suppose that an edge between nodes a and b represents a rotatable bond from atom A to atom B. Let S_1 be a set that contains atoms connected to B through a sequence of non-rotatable bonds. Let S_2 be a set that contains any atom that is not contained in S_1 and is bonded to either B or at least one atom in S_1. Then b contains atoms in S_1 and S_2. The atoms contained in b, thus, form the set of atoms that are directly rotated by the bond from A to B. The leaf nodes of the tree are recursively expanded, until all atoms and rotatable bonds in the ligand have been appended to the tree. Figure 2 shows the torsion tree for the ligand from a protein-ligand complex deposited in the PDB (1NDZ). Starting from the root node, the torsion tree is traversed in a breadth-first fashion and the edges are ranked in the order that they were visited. Suppose there are N edges in the torsion tree. The breadth-first traversal, thus, returns N edges or rotatable bonds ranked from 1 to N.

Step 2 The initial fragment (say, f), first to be docked with our protocol, is composed of the atoms in the torsion tree nodes that are connected by the edges ranked from 1 to r_u, where $r_u = min(n_s, N)$, and n_s is the input maxrot size. All the rotatable bonds in the initial fragment are set active and a set F is initialized such that $F = \{f\}$.

Step 3-5 AutoDock energy grid maps are computed as is typical in docking using *AutoDock*. A map contains, for each grid point location, values of interaction energies between the protein and the various atom types in the fragment. The atom types are assigned by *AutoDock*. Since each fragment in F contains atoms of the same atom types, the maps are computed only once in this step.

Each fragment in F is then docked to the target protein. Note that, by default, at the start of a docking operation *AutoDock* assigns random values to all the active rotatable bonds and the translation and orientation degrees of freedom. A specified number (*ga_run*) of docked conformations and their *AutoDock* scores are obtained in each docking operation. If all the atoms in the ligand are docked, then the set of all docked conformations (say, C_t) and corresponding scores are returned as the output of our docking protocol. While all the atoms in the ligand are not docked, then *Steps 6-7* are performed.

Step 6-7 The docked conformations in C_t are ranked by their *AutoDock* scores such that a conformation with lower score is ranked higher. The k top-ranked docked conformations are selected and each docked conformation is grown by adding rotatable bonds ranked from $r_u + 1$ to $min(r_u + \lfloor n_s/2 \rfloor, N)$ and all the atoms (derived from the torsion tree nodes) that are directly rotated by the added bonds. In each new fragment that is created by growing the docked conformations, the newly added rotatable bonds are set active. Few ($\lceil n_s/2 \rceil$) of the already explored bonds are also set active because re-exploration of the bonds improves the docking accuracy. Thus, the rotatable bonds ranked from $r_u - \lceil n_s/2 \rceil + 1$ to $min(r_u + \lfloor n_s/2 \rfloor, N)$ are set active, and all the other rotatable

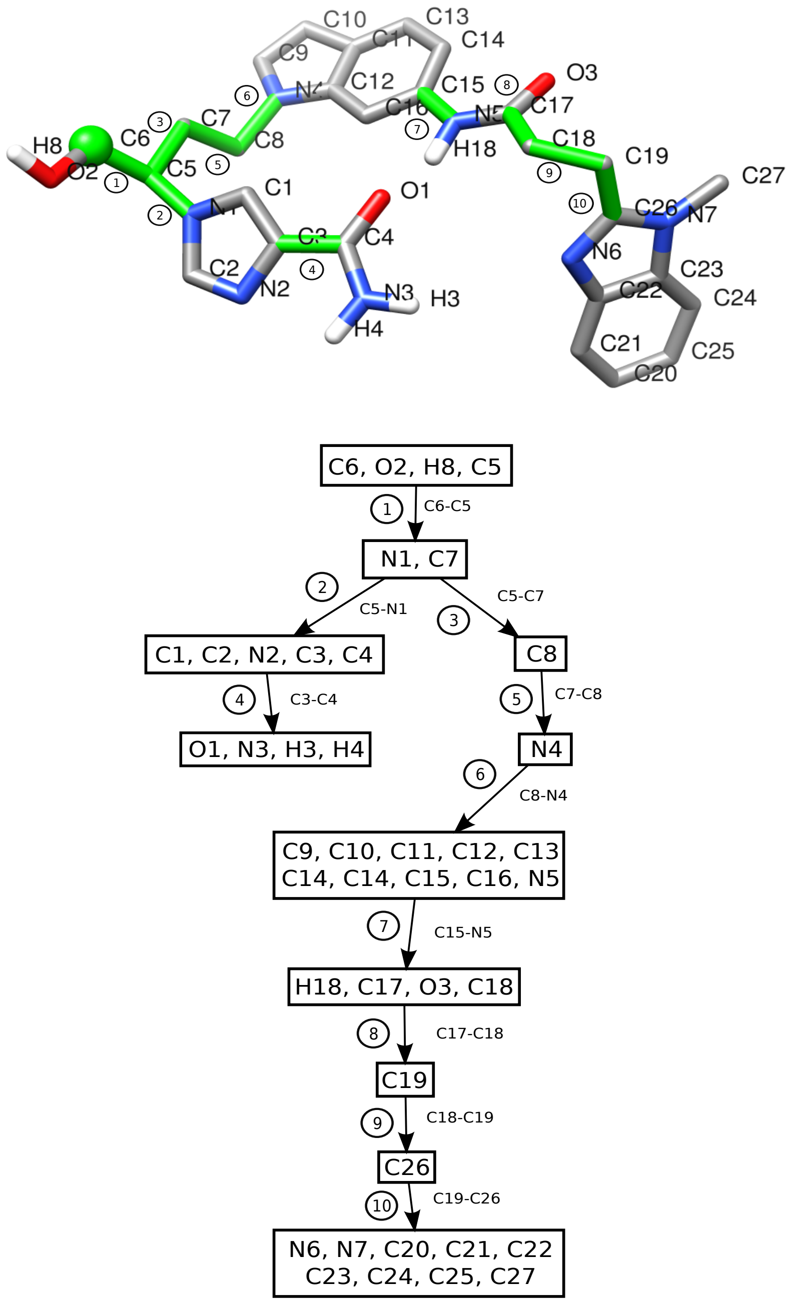

Figure 2: **Torsion tree.** The top figure shows the ligand from a protein-ligand complex deposited in the PDB (1NDZ). The ligand has 10 rotatable bonds shown as green sticks. The torsion tree shown in the bottom figure is rooted at the carbon atom C6 (green sphere). An edge of the tree represents a rotatable bond and a node represents a list of the atoms that are directly rotated by the bond. The tree is traversed in a breadth-first fashion and the edges are ranked (circled numbers) by the order in which they were visited. Note that a bond, if it is an amide, or is in a cycle, or rotates only the hydrogen atoms, then it is considered non-rotatable.

bonds are set non-active. The set F and parameter r_u are updated such that $F = \{\text{new fragments}\}$, and $r_u = min(r_u + \lfloor n_s/2 \rfloor, N)$. *Steps 3-7* are repeated.

The incremental growth of the ligand from a protein-ligand complex deposited in the PDB (1NDZ) is shown in Figure 3. Since our protocol involves selecting k docked conformations at each increment, and then growing and docking them again, it lends itself very easily to a parallel implementation. Each of the k conformations in *Step 6-7* are, therefore, docked in parallel.

Docking study

The purpose of our docking study is twofold: (a) to determine how the selection of the root atom and the maxrot size (n_s) affects docking performance of our protocol, and (b) compare the docking performance of our incremental protocol with the docking performance of *AutoDock*'s[8,30] standard protocol. Here we establish a dataset of protein-ligand complexes for our docking study and describe various docking experiments. In each docking experiment, all of the ligands in the dataset were computationally docked to their target proteins.

Dataset The dataset is derived from the PDBbind (v2007) database.[31] The PDBbind database is constructed from protein-ligand complexes available in the PDB. It contains a refined set of 1300 complexes that includes the structures of the protein and the ligand in each complex, as well as the experimentally measured binding affinity of each complex. The refined set is further grouped in clusters based on protein sequence similarity. Representatives of these clusters form the core set which contains 210 complexes in total. In each of the 210 complexes, the bonds in the ligand are labeled as rotatable or non-rotatable. Any bond that is an amide bond, or is in a cycle, or rotates only hydrogen atoms is labeled non-rotatable, and the rest of the bonds are labeled rotatable. Out of the 210 complexes, 73 complexes have greater than 6 rotatable bonds in the ligand. We consider ligands in the 73 complexes as large ligands for the purpose of our study and these 73 complexes form our dataset. The distribution of the number of ligands in the dataset with respect to the number of rotatable bonds is shown in Figure 4.

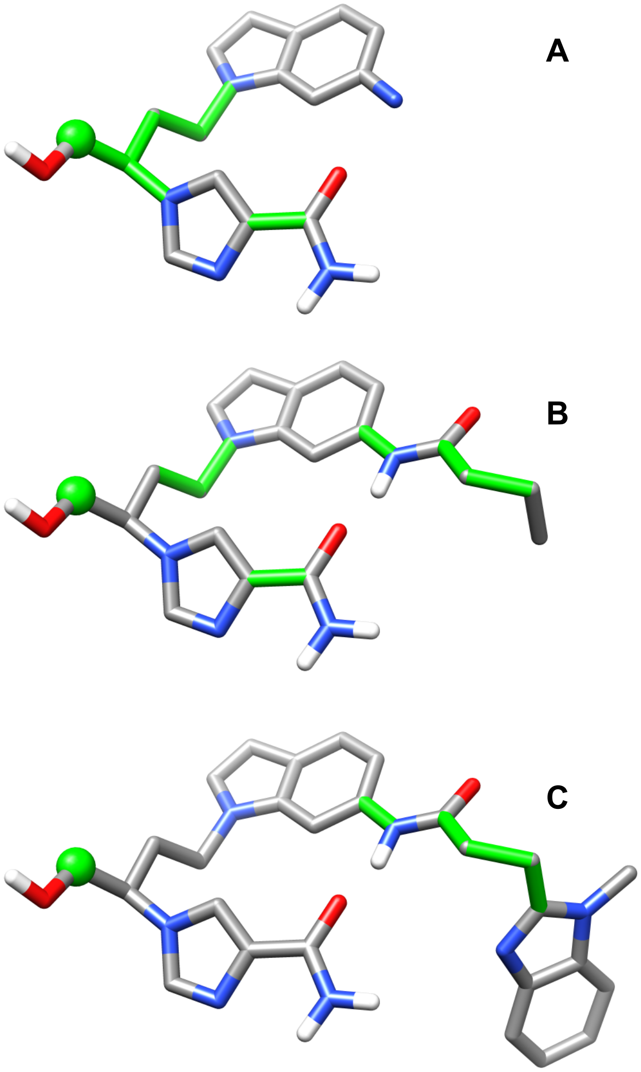

Figure 3: **Incremental growth of the ligand in our incremental docking protocol.** After the construction of the torsion tree (Figure 2), the ligand is incrementally grown and docked. For this illustration, the maxrot size (n_s) is set to 6. (A) The first fragment has atoms that are contained in the nodes connected by the edges ranked from 1 to 6. Bonds corresponding to these edges are set active and the fragment is docked. Active bonds are shown as green sticks. (B) The docked conformations of the fragment are grown by adding atoms that are contained in the nodes connected by the edges ranked from 7 to 9. The 3 newly added bonds and the 3 bonds corresponding to the edges ranked from 4 to 6 are set active. The new fragments are docked again. (C) The fragment is grown again by adding atoms that are contained in the node connected by the edge ranked 10. The newly added bond and the bonds corresponding to the edges ranked from 7 to 9 are set active. The fragment is docked and we obtain docked conformations of the full ligand.

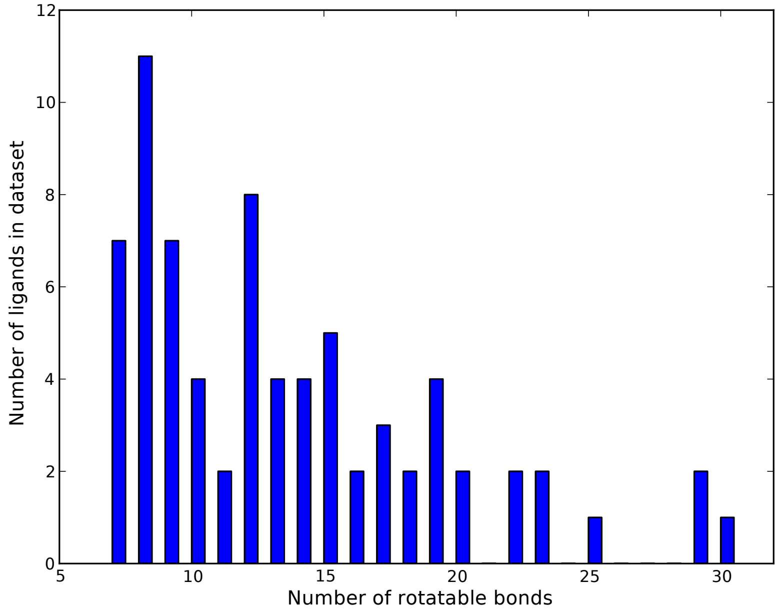

Figure 4: **Distribution of ligands in the dataset.** The dataset that was used for docking experiments contains 73 protein-ligand complexes derived from the PDBbind database. Each bar in the plot represents the total number of ligands that have a specified number of rotatable bonds.

Each protein and ligand in the dataset is processed in the following manner. The ligand atoms are assigned atom types and Gasteiger charges.[33] All non-polar hydrogen atoms and lone-pair charges are removed. The charge of each removed non-polar hydrogen is added to the carbon atom to which it is bonded and the charge of each lone pair is added to the atom it is associated with. The protein atoms are also assigned atom types and Gasteiger charges. The non-polar hydrogen atoms and lone-pairs of the protein are processed in the same way as those of the ligand. Waters and non-standard residues are removed as well. Similar to some other docking studies[19,21] that use the PDBbind database, structural optimization of the ligand and the protein was not performed.

Root atom selection mode and maxrot size For examining the robustness of our proposed protocol, two different modes for the selection of the root atom, required for the construction of the torsion tree in our incremental protocol, were tested: (a) a heavy atom was picked such that the tree rooted at that atom results in the initial fragment with the largest number of hydrogen bond donors and acceptors (*std* mode), and (b) a heavy atom was picked randomly (*rand* mode). We also tested various maxrot sizes, $n_s = 4, 6$, and 8. Using all combinations of selection modes and maxrot sizes, we performed six docking experiments using our protocol. The input parameter *ga_num_evals* was set to 250000, and *ga_num* was set to 50 for the first docking operation and to 20 for subsequent docking operations. The input parameter k was set to 5.

Docking performance comparison For comparing the docking performance of our protocol and *AutoDock*'s standard protocol, we did multiple docking experiments. AD^{inc} refers to the docking experiment done using our incremental protocol with $n_s = 6$ and *std* mode for the selection of the root atom. Other input parameters to AD^{inc} were same as in the six docking experiments discussed above. AD^1 refers to the docking experiment done using *AutoDock*'s standard protocol with the maximum number of energy evaluations (*ga_num_evals*) set to 2.5 million and the number of output docked conformations (*ga_run*) set to 50. AD^2 refers to the docking experiment done using *AutoDock*'s standard protocol with *ga_num_evals* set to 5 million and *ga_run* set to 50, and AD^3 refers to the experiment with *ga_num_evals* set to 25 million and *ga_run* set to 50. The settings in AD^3 are the recommended settings for *AutoDock*'s standard protocol when docking large ligands and have been used previously for docking performance comparison.[12]

In dockings done with *AutoDock*, the *AutoDock* grid was centered on the average coordinates of the atoms of the input ligand structure (obtained from the dataset) and the dimensions of the grid were chosen such that the grid box fully encompasses the input ligand structure. The size of each dimension of the grid box was extended by 15Å. The size of any dimension, if it was less than 60Å, was set to 60Å.[12] The rest of the parameters of *AutoDock*, that are not explicitly mentioned, were set to their default values. Since *AutoDock* employs a stochastic algorithm, all experiments were repeated 5 times. Version 4.2 of *AutoDock* was used in all of the docking experiments. The docking study was done on a cluster at Rice University that has 192 computing nodes and 2304 (in total) processor cores, with each core running at 2.83 GHz.

Results

Here we present the results from the various docking experiments described in the Materials and Methods section. Each docking of a ligand to its target protein produced multiple docked conformations of the ligand and corresponding docking scores. We computed root mean squared distance (RMSD) between the docked conformations and the input conformation of the ligand that is obtained from the structure of the protein-ligand complex in the dataset. The RMSD values were computed using only the heavy atoms of the ligand. We identified two docked conformations of each ligand: (a) a *Top-scoring* conformation (*CS*), and (b) a *Top-RMSD* conformation (*CR*). *CS* is the docked conformation that has the minimum score and the *CR* is the one with the lowest RMSD value. It is to be noted that since the dataset contains structures of the ligands bound to their respective proteins, it is possible to compute the RMSD values. In the general case of ligands for which the structures of the bound conformations are not known, only the *Top-scoring* conformation of each ligand can be identified. The docking performance is, therefore, evaluated on the basis of the score (S_{CS}) and RMSD value (R_{CS}) associated with *CS* of each ligand, and the computational expense of docking (DT). Better docking performance means lower values of R_{CS} and DT. We also analyze the score (S_{CR}) and RMSD value (R_{CR}) associated with *CR* of each ligand.

Root atom selection mode and maxrot size

Six docking experiments were performed to determine the effect of root atom selection mode (*rand*, *std*) and maxrot size on the docking performance ($n_s = 4, 6, 8$) of our incremental protocol. The results from the six experiments are presented in Table 1. The values of S_{CS}, S_{CR}, R_{CS}, and R_{CR} were obtained for the 73 (number of protein-ligand complexes in the dataset) dockings done in each experiment. Average of the values over the 73 dockings were computed for each experiment. Each experiment was repeated 5 times and Table 1 lists the average (over the 5 experiments) of the average values, denoted by S_{CS}^{avg}, S_{CR}^{avg}, R_{CS}^{avg}, and R_{CR}^{avg}. DT^{avg} represents the total computational expense of the 73 dockings averaged over the 5 repeated experiments.

Table 1: Root atom selection mode and maxrot size (n_s).

Mode	n_s	S_{CS}^{avg} (kcal/mol)	R_{CS}^{avg} (Å)	S_{CR}^{avg} (kcal/mol)	R_{CR}^{avg} (Å)	$DT^{avg}(h)$
std	4	-8.93	5.50	-6.84	3.49	41.57
rand	4	-8.99	5.73	-6.79	3.39	42.02
std	6	-8.83	5.24	-6.55	3.03	31.70
rand	6	-8.84	5.44	-6.56	2.99	31.58
std	8	-8.34	5.73	-6.17	3.15	25.26
rand	8	-8.27	5.76	-6.21	3.01	24.88

Six docking experiments were done using our incremental protocol to evaluate the effect of root atom selection mode and maxrot size on the docking performance. In each experiment different maxrot sizes (f_s) and selection modes were used. This table lists, for each experiment, average *AutoDock* scores (S_{CS}^{avg}, S_{CR}^{avg}) average RMSD values (R_{CS}^{avg}, R_{CR}^{avg}), and average total docking times (DT^{avg}). S_{CS}^{avg} and R_{CS}^{avg} correspond to the *Top-scoring* docked conformations, and S_{CS}^{avg} and R_{CS}^{avg} correspond the *Top-RMSD* docked conformations. RMSD is computed from the ligand conformation in the dataset and includes only the heavy atoms. In selection mode *std*, the root atom is selected using a heuristic and in *random* mode it is selected randomly from the set of all heavy atoms in the ligand.

Most accurate docking performance, i.e., lowest R_{CS}^{avg} was obtained in the experiment done with *std* selection mode and maxrot size $n_s = 6$. Experiments with $n_s = 8$ were done in the lowest docking time (DT^{avg}). The maxrot size determines number of docking operations that are performed in each docking using our protocol. A bigger maxrot size thus results in fewer docking operations per docking which means experiments with bigger maxrot size require lower docking times. Similarly, experiments with smaller maxrot size require higher docking times as in the case of experiments with $n_s = 4$.

The *Top-scoring* conformations were, in general, less accurate ($R_{CS}^{avg} > R_{CR}^{avg}$ in all experiments) than *Top-RMSD* conformations. The docked conformation with the lowest score is not necessarily the one with the lowest RMSD value because the use of approximate scoring functions in docking programs results in inaccurate estimation of the thermodynamic stabilities of the complexes. Comparison of scores ($S_{CR}^{avg} > S_{CS}^{avg}$) re-emphasizes that the conformations with the lowest RMSD values do not necessarily have the lowest scores.

Table 2: Docking performance comparison.

Experiment	S_{CS}^{avg} (kcal/mol)	R_{CS}^{avg} (Å)	S_{CR}^{avg} (kcal/mol)	R_{CR}^{avg} (Å)	DT^{avg} (h)
AD^{inc}	-8.83	5.24	-6.55	3.03	31.70
AD^1	-8.65	5.69	-7.20	3.16	72.30
AD^2	-9.10	5.56	-7.75	3.12	143.67
AD^3	-9.66	5.17	-8.44	2.92	725.85

Four experiments were done to compare the docking performance of our incremental protocol with *AutoDock*'s standard docking protocol. This table lists, for each experiment, average *AutoDock* scores (S_{CS}^{avg}, S_{CR}^{avg}) average RMSD values (R_{CS}^{avg}, R_{CR}^{avg}), and average total docking times (DT^{avg}). S_{CS}^{avg} and R_{CS}^{avg} correspond to the *Top-scoring* docked conformations, and S_{CS}^{avg} and R_{CS}^{avg} correspond the *Top-RMSD* docked conformations. AD^{inc} refers to the experiment done using our protocol ($n_s = 6$, *std* selection mode). AD^1 ($ga\_num\_evals = 2.5$ *million*, $ga\_run = 50$), AD^2 ($ga\_num\_evals = 5$ *million*, $ga\_run = 50$), and AD^3 ($ga\_num\_evals = 25$ *million*, $ga\_run = 50$) refer to the experiments done using *AutoDock*'s standard protocol with different parameter settings. RMSD is computed from the ligand conformation in the dataset and includes only the heavy atoms.

Docking performance comparison

To compare the performance of our incremental protocol, we did docking experiments using *AutoDock*'s standard protocol as described in the Methods section. For each experiment, values of S_{CS}^{avg}, S_{CR}^{avg}, R_{CS}^{avg}, R_{CR}^{avg}, and DT^{avg} were computed as above. The results from the experiments are listed in Table 2. The experiment done using our protocol took, on average, 31.70 hours to finish. It is to be noted that this is the total CPU time spent. Due to the parallel implementation of our protocol (see Methods section), the real time spent was much lower. Comparison of docking times DT^{avg} in Table 2 reveals that the docking time spent in the experiment done using our protocol (AD^{inc}) is much lower than experiments AD^1, AD^2, and AD^3. The *Top-scoring* conformations computed in AD^{inc} were overall more accurate (lower or comparable R_{CS}^{avg} values) than those computed in all other experiments. As expected, in all experiments, R_{CR}^{avg} values were lower and S_{CR}^{avg} values were higher than corresponding R_{CS}^{avg} and S_{CS}^{avg} values, thus re-emphasizing the limitations of the scoring functions. For experiments AD^{inc}, AD^1, AD^2, and AD^3, the median of the 73 R_{CS} values, corresponding to the 73 protein-ligand complexes in our dataset, was 4.56Å, 5.48Å, 5.42Å, and 4.88Å respectively. The median values also reflected the better docking performance of our protocol.

Experiment AD^3 was done using *AutoDock*'s standard docking protocol, and the parameter settings recommended by *AutoDock*, for docking large ligands, were used. Compared to the docking time spent in AD^3, the experiment done using our protocol was completed in approximately 23-fold lower total docking time. Comparison of R_{CS}^{avg}, and DT^{avg} in experiments AD^{inc} and AD^3 makes it clear that our protocol results in significantly lower total docking time as well as comparable RMSD values. Tweaking of parameter settings in experiments with *AutoDock*'s standard protocol could be done in various ways and could possibly reduce docking time. However, in our docking experiments, the incremental protocol performed approximately 2-fold to 23-fold faster, did not sacrifice accuracy, and improved the accuracy in most of the cases. Thus, for docking large ligands, our protocol was clearly superior to *AutoDock*'s standard protocol.

Figure 5 plots the RMSD values (R_{CS}^{avg}), and docking times (DT^{avg}) respectively against varying cut-off values of the number of rotatable bonds. Each plotted value of R_{CS}^{avg}, and DT^{avg} represents the average value computed from the R_{CS} and DT values for ligands that have the number of rotatable bonds less than or equal to the cut-off. Different trend lines on the plots correspond to different docking experiments (AD^{inc}, AD^1, AD^2, or AD^3). Figure 5A shows that, on average, AD's standard protocol produced slightly more accurate (lower R_{CS}^{avg} values) conformations for ligands with smaller number of rotatable bonds, but as the number of rotatable bonds increased, our protocol produced more accurate conformations than the standard protocol used in AD^1 and AD^2. Comparison of RMSD values obtained in AD^{inc} and AD^3 reveals comparable docking accuracy variation with the cut-off values of the number of rotatable bonds. Figure 5B shows that the experiments done using our protocol required lower docking time as compared to the other experiments and this was consistent for all cut-off values.

Figure 6 plots the scores (S_{CS}^{avg}), RMSD values (R_{CS}^{avg}), and docking times (DT^{avg}), as obtained in experiments AD^{inc} and AD^3, for each ligand in the dataset. Ligands are numbered from 1 to 73 in the increasing order of the number of rotatable bonds as shown in Figure 7. Figure 6A reveals that AD^{inc} and AD^3 produced comparable scores. This is an important result because it shows

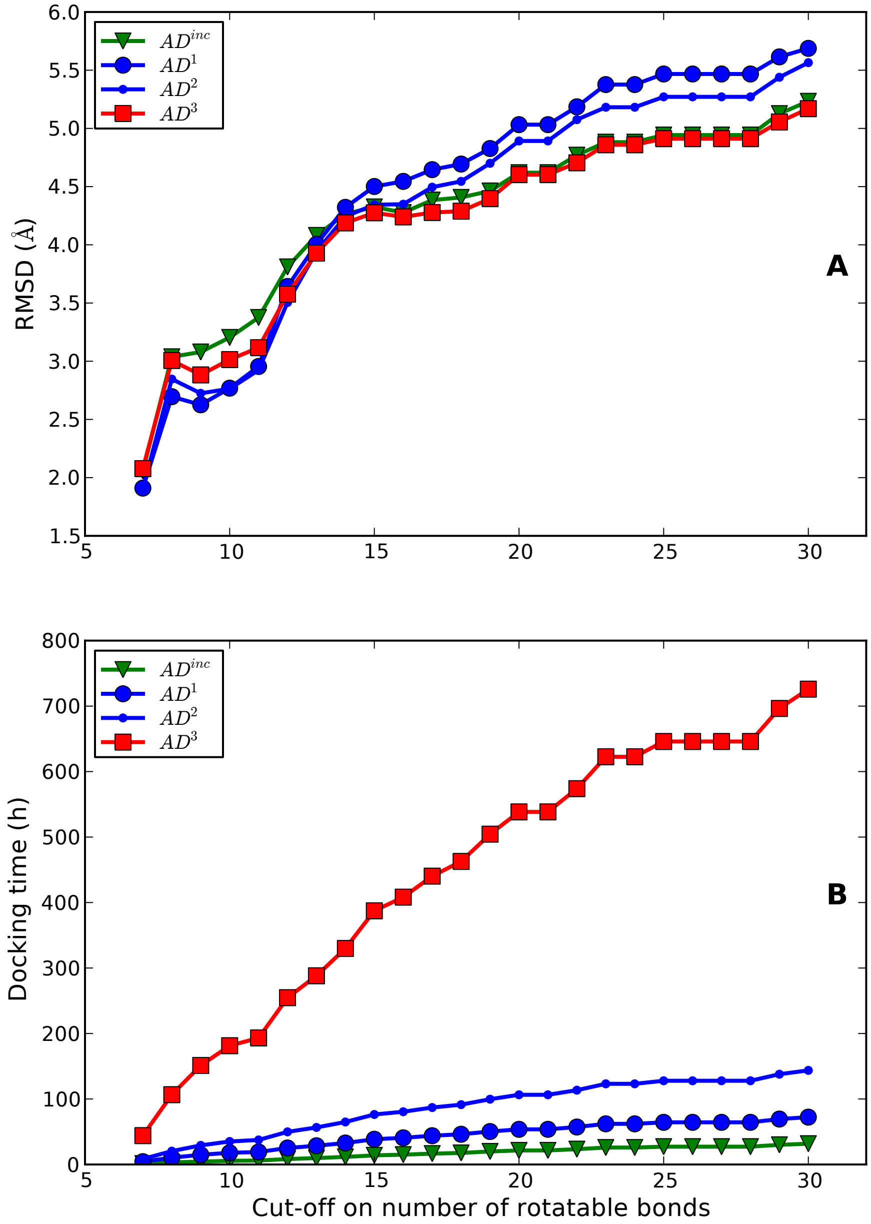

Figure 5: **Docking performance variation with the number of rotatable bonds.** This plot shows: (A) average RMSD values, and (B) average docking times, versus cut-off values of the number of rotatable bonds for the following four experiments. All plotted values correspond to *Top-scoring* docked conformations and are average over the 73 ligands in the dataset. AD^{inc} refers to the experiment done using our incremental docking protocol ($n_s = 6$, *std* selection mode). AD^1 ($ga\_num\_evals = 2.5$ *million*, $ga\_run = 50$), AD^2 ($ga\_num\_evals = 5$ *million*, $ga\_run = 50$), and AD^3 ($ga\_num\_evals = 25$ *million*, $ga\_run = 50$) refers to the experiments done using *AutoDock*'s standard docking protocol with different parameter settings. RMSD is computed from the ligand conformation in the dataset and includes only the heavy atoms.

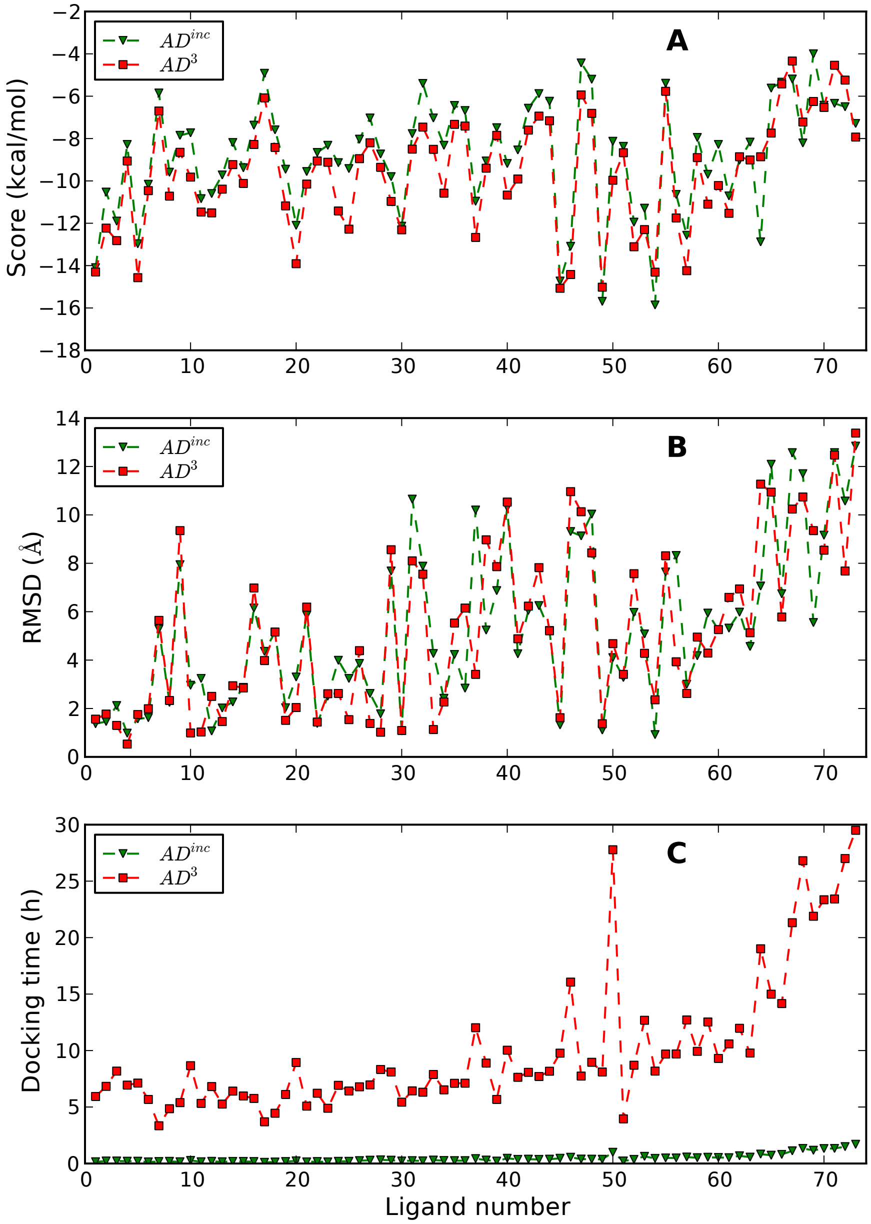

Figure 6: **$\mathbf{AD^{inc}}$ versus $\mathbf{AD^3}$.** This plot shows: (A) average *AutoDock* scores, (B) average RMSD values, and (C) average docking times for the 73 ligands in the dataset. The ligands are numbered from 1 to 73 in the increasing order of the number of rotatable bonds. All plotted values correspond to *Top-scoring* docked conformations. AD^{inc} refers to the experiment done using our incremental protocol ($n_s = 6$, *std* selection mode). AD^3 ($ga\_num\_evals = 2.5$ *million*, $ga\_run = 50$) refers to the experiment done using *AutoDock* with recommended parameter settings for large ligands. RMSD is computed from the ligand conformation in the dataset and includes only the heavy atoms.

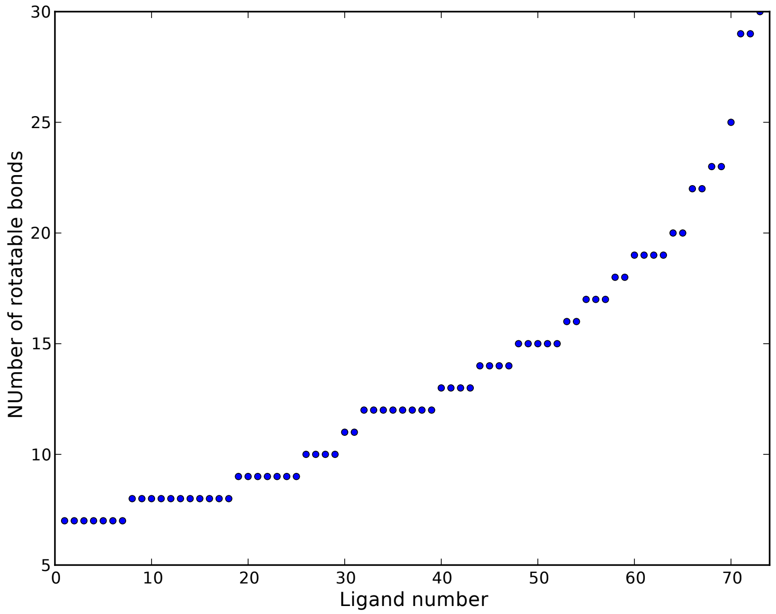

Figure 7: **Ligand numbers.** Docking experiments were done on a dataset of 73 protein-ligand complexes derived from the core set of PDBbind database. The ligands in the dataset are numbered from 1 to 73 in the increasing order of the number of rotatable bonds in the ligands.

that, for large ligands, our protocol was able to compute similar minima of the *AutoDock*'s scoring function in approximately 23-fold lower computational time. A comparison of RMSD values in Figure 6B shows that, for some ligands, our protocol resulted in slightly better docking accuracy and, for other ligands, the standard protocol resulted in slightly better docking accuracy. Figure 6C shows that, for each ligand in our dataset, docking using our protocol was significantly faster than docking using *AD*'s standard protocol and this result is a big strength of our protocol.

Discussion

AutoDock is a popular non-commercial docking program that docks a ligand to its target protein and performs well (accurate and computationally fast) when the number of rotatable bonds in the ligand is small. Increase in the number of rotatable bonds in the ligand, however, severely affects the docking performance of *AutoDock*. A ligand with a large number of rotatable bonds has a high-dimensional conformation space which makes exploring for the docked conformation, that corresponds to the global minimum of the scoring function, extremely difficult. *AutoDock*, therefore, recommends parameter settings that result in a more exhaustive search of the global minimum. In our experiments however, the more exhaustive search, results in modest improvement in docking accuracy and substantial increase in the computational docking time.

In this paper we propose an incremental docking protocol that utilizes *AutoDock* for docking operations. Instead of searching for the docked conformation in the high-dimensional conformation space, the search is done in the subspaces of the conformation space. Starting from a carefully chosen initial fragment of the ligand, the ligand is incrementally docked and grown until all the atoms of the ligand are docked. At each increment a few bonds are set active and are allowed to rotate. By limiting the number of active bonds (termed as maxrot size) to a small value, we ensure that in all the docking operations using *AutoDock*, the search for the docked conformation is done in a subspace of the conformation space.

We performed a systematic docking study with docking experiments on a dataset of 73 protein-ligand complexes (7 to 30 rotatable bonds in the ligands) from the core set of the PDBbind database. Six different experiments were done using our protocol to determine the effect of the choice of the root atom and maxrot size on the docking performance. The choice of the root atom leading to an initial fragment of the ligand with highest number of hydrogen bond donors and acceptors, and maxrot size of 6 resulted in the best docking performance. To compare our protocol with *AutoDock*'s standard protocol, we did experiments using our protocol and *AutoDock*'s standard protocol with three different parameter settings. One of the experiments done using *AutoDock* had parameters set to the values that are recommended when docking large ligands.

The docking experiments demonstrate that our protocol is computationally fast, it is upto 23-fold faster than docking using *AutoDock*'s standard protocol in the recommended parameter settings. Each docking operation in our protocol explores a subspace of the conformation space and is, therefore, fast. Even though our protocol consists of multiple docking operations, less computational time is spent in each individual operation and this results in lower docking time overall.

For a given ligand and target protein, the main goal of a docking program is to search for a docked conformation of the ligand that corresponds to the global minimum of the program's scoring functions. From the results of our docking study, it is clear that the accuracy of docking using our protocol is comparable to the accuracy of docking using *AutoDock*'s standard protocol in the recommended parameter settings. The accuracy is measured by the average RMSD value computed from the RMSD values between *Top-scoring* (lowest score) docked conformation of each ligand and the ligand conformation in the dataset. In experiments, our protocol also results in average docking scores that are comparable to the scores obtained using *AutoDock*'s standard protocol.

The average (over the dataset) RMSD value of the *Top-scoring* conformations computed in the experiment done using our incremental protocol is 5.24Å which is either lower than or comparable to the average RMSD values computed in all of the experiments done using *AutoDock*'s standard protocol. The docked conformations can be further refined by using, for example, molecular dynamics. The strength of our protocol is that docked conformations required for further refinement can be quickly computed. Consensus docking programs[11,34] that utilize multiple docking programs, such as *AutoDock* and others, to obtain docked conformations can also benefit from the quick computation of docked conformations using our protocol. Such consensus docking programs can probably further improve the accuracy of docking large ligands. Thus, for docking large ligands, our incremental protocol can provide an excellent alternative to *AutoDock*'s standard protocol.

Acknowledgement

This work has been supported in part by a training fellowship from the Keck Center Computational Cancer Biology Training Program of the Gulf Coast Consortia (CPRIT RP101489), The Texas Higher Education Coordinating Board (NHARP 01907), The National Cancer Institute (RO1 CA096652), The John and Ann Doerr Fund for Computational Biomedicine at Rice University, and the Data Analysis and Visualization Cyberinfrastructure funded by NSF under grant OCI-0959097.

References

(1) Anderson, A. C. The process of structure-based drug design. *Chemistry & Biology* **2003**, *10*, 787–797.

(2) Sousa, F.; Fernandes, P. A.; Ramos, M. J. Protein-ligand docking: current status and future challenges. *Proteins: Structure, Function, and Bioinformatics* **2006**, *65*, 15–26.

(3) Schneider, G. Virtual screening: an endless staircase? *Nature Reviews Drug Discovery* **2010**, *9*, 273–276.

(4) Kuntz, I. A geometric approach to macromolecule-ligand interactions. *Journal of Molecular Biology* **1982**, *161*, 269–288.

(5) Abagyan, R.; Totrov, M.; Kuznetsov, D. ICM–a new method for protein modeling and design: applications to docking and structure prediction from the distorted native conformation. *Journal of Computational Chemistry* **1994**, *15*, 488–506.

(6) Jones, G.; Willett, P.; Glen, R. C. Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation. *Journal of Molecular Biology* **1995**, *245*, 43–53.

(7) Rarey, M.; Kramer, B.; Lengauer, T.; Klebe, G. A fast flexible docking method using an incremental construction algorithm. *Journal of Molecular Biology* **1996**, *261*, 470–489.

(8) Morris, G. M.; Goodsell, D. S.; Halliday, R. S.; Huey, R.; Hart, W. E.; Belew, R. K.; Olson, A. J. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *Journal of Computational Chemistry* **1998**, *19*, 1639–1662.

(9) Jain, A. N. Morphological similarity: a 3D molecular similarity method correlated with protein-ligand recognition. *Journal of Computer-aided Molecular Design* **2000**, *14*, 199–213.

(10) Friesner, R. A.; Banks, J. L.; Murphy, R. B.; Halgren, T. A.; Klicic, J. J.; Mainz, D. T.; Repasky, M. P.; Knoll, E. H.; Shelley, M.; Perry, J. K.; Shaw, D. E.; Francis, P.; Shenkin, P. S.

Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *Journal of Medicinal Chemistry* **2004**, *47*, 1739–1749.

(11) Plewczynski, D. VoteDock: consensus docking method for prediction of protein-ligand interactions. *Journal of Computational Chemistry* **2010**, *32*, 1–14.

(12) Trott, O.; Olson, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry* **2010**, *31*, 455–461.

(13) Kellenberger, E.; Rodrigo, J.; Muller, P.; Rognan, D. Comparative evaluation of eight docking tools for docking and virtual screening accuracy. *Proteins: Structure, Function, and Bioinformatics* **2004**, *57*, 225–242.

(14) Chang, M. W.; Lindstrom, W.; Olson, A. J.; Belew, R. K. Analysis of HIV wild-type and mutant structures via in silico docking against diverse ligand libraries. *Journal of Chemical Information and Modeling* **2007**, *47*, 1258–1262.

(15) Onodera, K.; Satou, K.; Hirota, H. Evaluations of molecular docking programs for virtual screening. *Journal of Chemical Information and Modeling* **2007**, *47*, 1609–1618.

(16) Rogers, J. P.; Beuscher,; Flajolet, M.; McAvoy, T.; Nairn, A. C.; Olson, A. J.; Greengard, P. Discovery of protein phosphatase 2C inhibitors by virtual screening. *Journal of Medicinal Chemistry* **2006**, *49*, 1658–1667.

(17) Cross, J. B.; Thompson, D. C.; Rai, B. K.; Baber, J. C.; Fan, K. Y.; Hu, Y.; Humblet, C. Comparison of several molecular docking programs: pose prediction and virtual screening accuracy. *Journal of Chemical Information and Modeling* **2009**, *49*, 1455–1474.

(18) Chang, M. W.; Ayeni, C.; Breuer, S.; Torbett, B. E. Virtual screening for HIV protease inhibitors: a comparison of AutoDock 4 and Vina. *PloS ONE* **2010**, *5*, e10926.

(19) Li, X.; Li, Y.; Cheng, T.; Liu, Z.; Wang, R. Evaluation of the performance of four molecular docking programs on a diverse set of protein-ligand complexes. *Journal of Computational Chemistry* **2010**, *31*, 2109–2125.

(20) Li, N.; Thompson, S.; Schultz, D. C.; Zhu, W.; Jiang, H.; Luo, C.; Lieberman, P. M. Discovery of selective inhibitors against EBNA1 via high throughput in silico virtual screening. *PloS ONE* **2010**, *5*, e10126.

(21) Plewczynski, D.; Lziniewski, M.; Augustyniak, R.; Ginalski, K. Can we trust docking results? Evaluation of seven commonly used programs on PDBbind database. *Journal of Computational Chemistry* **2011**, *32*, 742–755.

(22) Hetényi, C.; van der Spoel, D. Efficient docking of peptides to proteins without prior knowledge of the binding site. *Protein Science* **2002**, *11*, 1729–1737.

(23) Böhm, H.-J. The computer program LUDI: a new method for the de novo design of enzyme inhibitors. *Journal of Computer-Aided Molecular Design* **1992**, *6*, 61–78, 10.1007/BF00124387.

(24) Bohacek, R. S.; McMartin, C. Multiple highly diverse structures complementary to enzyme binding sites: results of extensive application of a de novo design method incorporating combinatorial growth. *Journal of the American Chemical Society* **1994**, *116*, 5560–5571.

(25) Eisen, M. B.; Wiley, D. C.; Karplus, M.; Hubbard, R. E. HOOK: a program for finding novel molecular architectures that satisfy the chemical and steric requirements of a macromolecule binding site. *Proteins: Structure, Function, and Bioinformatics* **1994**, *19*, 199–221.

(26) Jackson, R. M. Q-fit: a probabilistic method for docking molecular fragments by sampling low energy conformational space. *Journal of Computer-aided Molecular Design* **2002**, *16*, 43–57.

(27) Wang, J.; Kollman, P. a.; Kuntz, I. D. Flexible ligand docking: a multistep strategy approach. *Proteins: Strcuture, Function, and Bioinformatics* **1999**, *36*, 1–19.

(28) Ewing, T. J.; Makino, S.; Skillman, A. G.; Kuntz, I. D. DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases. *Journal of Computer-Aided Molecular Design* **2001**, *15*, 411–428, 10.1023/A:1011115820450.

(29) Sauton, N.; Lagorce, D.; Villoutreix, B. O.; Miteva, M. A. MS-DOCK: accurate multiple conformation generator and rigid docking protocol for multi-step virtual ligand screening. *BMC Bioinformatics* **2008**, *9*, 184.

(30) Morris, G. M.; Huey, R.; Lindstrom, W.; Sanner, M. F.; Belew, R. K.; Goodsell, D. S.; Olson, A. J. AutoDock4 and AutoDockTools4: automated docking with selective receptor flexibility. *Journal of Computational Chemistry* **2009**, *30*, 2785–2791.

(31) Wang, R.; Fang, X.; Lu, Y.; Yang, C.-Y.; Wang, S. The PDBbind database: methodologies and updates. *Journal of Medicinal Chemistry* **2005**, *48*, 4111–4119, PMID: 15943484.

(32) *AutoDock 4.2 User Guide*, `http://autodock.scripps.edu/downloads/autodock-registration/autodock-4-2-download-page/`.

(33) Gasteiger, J.; Marsili, M. Iterative partial equalization of orbital electronegativity–a rapid access to atomic charges. *Tetrahedron* **1980**, *36*, 3219–3228.

(34) Teramoto, R.; Fukunishi, H. Supervised Consensus Scoring for Docking and Virtual Screening. *Journal of Chemical Information and Modeling* **2007**, *47*, 526–534.