MolDock: A New Technique for High-Accuracy Molecular Docking

René Thomsen* and Mikael H. Christensen

Molegro ApS, Hoegh-Guldbergs Gade 10, Bldg. 1090, DK-8000 Aarhus C, Denmark

Received November 29, 2005

In this article we introduce a molecular docking algorithm called MolDock. MolDock is based on a new heuristic search algorithm that combines differential evolution with a cavity prediction algorithm. The docking scoring function of MolDock is an extension of the piecewise linear potential (PLP) including new hydrogen bonding and electrostatic terms. To further improve docking accuracy, a re-ranking scoring function is introduced, which identifies the most promising docking solution from the solutions obtained by the docking algorithm. The docking accuracy of MolDock has been evaluated by docking flexible ligands to 77 protein targets. MolDock was able to identify the correct binding mode of 87% of the complexes. In comparison, the accuracy of Glide and Surflex is 82% and 75%, respectively. FlexX obtained 58% and GOLD 78% on subsets containing 76 and 55 cases, respectively.

Introduction

One application of molecular docking is to design pharmaceuticals in silico by optimizing lead candidates targeted against proteins. The lead candidates can be found using a docking algorithm that tries to identify the optimal binding mode of a small molecule (ligand) to the active site of a macromolecular target. Thus, the purpose of drug discovery is to derive drugs that more strongly bind to a given protein target than the natural substrate. By doing so, the biochemical reaction that the target molecule catalyzes can be altered or prevented.

Drugs are typically discovered by chance in a trial-and-error manner using high-throughput screening methods that use in vitro experiments to measure the activity of a large number of compounds against a given target. This process is very expensive and time consuming. If the 3D structure of the target is known, then simulated molecular docking can be a useful tool in the drug-discovery process. This in silico approach allows for a faster and cheaper identification of promising drug candidates by the virtual screening of compound databases. Afterward, lab experiments (synthesis), toxicological testing, clinical trials, and so forth can be conducted to further examine the drug candidates identified by the virtual screening process.

Docking methods typically use an energy-based scoring function to identify the energetically most favorable ligand conformation when bound to the target. The general hypothesis is that lower energy scores represent better protein−ligand bindings compared to higher energy values. Therefore, molecular docking can be formulated as an optimization problem, where the task is to find the ligand-binding mode with the lowest energy.

Unfortunately, the number of possible ligand bindings to be considered dramatically increases when ligand flexibility is taken into account. Despite advances in computing power, docking remains a very challenging problem because the high number of possible docking conformations prevents a systematic brute-force approach. To tackle docking problems and efficiently handle flexibility, search heuristics are required.

The most commonly used heuristic search algorithms that have been applied to molecular docking are simulated annealing,[1−2] tabu search,[3] and evolutionary algorithms.[4−5]

In this article, we introduce a docking algorithm called MolDock. MolDock is based on a new hybrid search algorithm, called guided differential evolution. The guided differential evolution algorithm combines the differential evolution optimization technique with a cavity prediction algorithm. Differential evolution (DE) was introduced by Storn and Price[6] in 1995 and has previously been successfully applied to molecular docking.[7] The use of predicted cavities during the search process, allows for a fast and accurate identification of potential binding modes (poses).

The docking scoring function of MolDock that we use is based on a piecewise linear potential (PLP) introduced by Gehlhaar et al.[8,9] and further extended in GEMDOCK by Yang et al.[10] In MolDock, the docking scoring function is extended with a new term, taking hydrogen bond directionality into account. Moreover, a re-ranking procedure (described below) is applied to the highest ranked poses to further increase docking accuracy.

To evaluate the docking accuracy of MolDock, we ran experiments with a selection of 77 protein−ligand complexes from the publicly available GOLD dataset.[11] Each of the energy-minimized ligands was docked to the corresponding protein target, and the root-mean-square deviation (RMSD) between the predicted and the cocrystallized structure was recorded. The experimental results were compared with published results from Glide,[12] GOLD,[4,11] FlexX,[13,14] and Surflex,[15] representing state-of-the-art docking programs.

The article is organized as follows. The next section describes the experimental setup, including the dataset that was used, how it was prepared, and how the docking results were evaluated. The third section presents the experimental results and provides a short analysis of the failed predictions. The fourth section discusses the experimental results and highlights future research topics. Finally, the Methods section provides a detailed description of the guided differential evolution algorithm and the scoring function used by MolDock.

Experimental Setup

Dataset. To benchmark the performance of MolDock, we used a set of 77 complexes. The complexes were taken from the GOLD benchmark set[11] and they have previously been used to evaluate Surflex[15] and Glide.[12] The original dataset used by Jain[15] contained 81 complexes but four of these (1ack, 1lpm, 3aah, 6rsa) were

* To whom correspondence should be addressed. Tel: +45 8942 3165. Fax: (+45) 8942 3077. E-mail: rt@molegro.com.

10.1021/jm051197e CCC: $33.50
Published on Web 04/29/2006

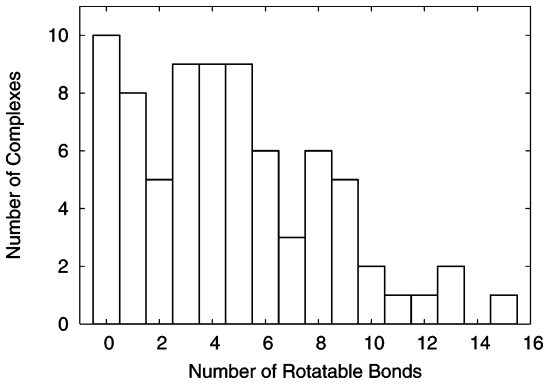

Figure 1. Distribution of rotatable bonds in dataset containing 77 complexes.

omitted because they were not included in the GOLD benchmark set. The GOLD dataset has been manually inspected to avoid redundancy, and the ligands were selected to represent drug-like compounds.[11]

The ligands in the dataset contain between 6 and 48 heavy atoms and between 0 and 15 rotatable bonds (not including bonds that only rotate hydrogen atoms). The distribution of rotatable bonds in the dataset is shown in Figure 1. The PDB codes of the complexes used are listed in the first column of Table 1.

Dataset Preparation. The GOLD dataset was chosen because of the previous significant effort put into the preparation and validation of these complexes. For example, hydrogens and bond orders were assigned to both ligand and protein molecules, and erroneous complexes were weeded out in the construction of the dataset.

For each complex, charges and protonation states were assigned using the templates described in the Methods section. All acyclic single bonds were set flexible except for bonds that only rotate hydrogens (e.g., bonds connected to hydroxyl and methyl groups). These bonds are kept rigid because the docking scoring function does not take explicit hydrogen positions into account while docking. Therefore, the number of rotatable bonds reported in this article does not include bonds capable of rotating hydrogens, unlike the values reported by Glide, GOLD, and FlexX. For ease of comparison between MolDock and the other programs, Table 1 includes the number of rotatable bonds that rotate heavy atoms (−H) and rotatable bonds that also include the rotation of hydrogens (+H).[16] During the docking experiments, structural water molecules were excluded. Cofactors and metal ions were retained. Furthermore, the ligands used in the docking experiments were energy-minimized using the MAXIMIN2 module, which is part of SYBYL.[17] These energy-minimized ligands were obtained from the GOLD dataset.

Automated Identification of Binding Site. MolDock automatically identifies potential binding sites (hereafter referred to as cavities) using the cavity detection algorithm described in the Methods section. To automate benchmarking, cavities within a $30 \times 30 \times 30$ Å^3 cube centered at the experimentally known ligand position were used. The cavities found by the cavity detection algorithm are actively used by the search algorithm (guided differential evolution) to focus the search during the docking simulation (see the Methods section for details regarding the search algorithm).

Evaluation Procedure. For each benchmark complex, we conducted 10 independent runs with the guided differential evolution algorithm, each of these runs returning one solution (pose). These 10 solutions were then re-ranked (see below), and the highest ranked solution was compared with the known experimental structure using the standard Cartesian root-mean-square deviation (RMSD) measure (between similar atoms in the pose and the experimental structure). A pose was considered successful if the RMSD between the pose and the experimentally known ligand was less than 2.0 Å. The same docking success criterion has been used in other docking studies to estimate the accuracy of Glide,[12] Surflex,[15] GOLD,[4,11] and FlexX.[13,14] Although other measures exist,[18] a RMSD with a 2.0 Å threshold is commonly used, and it was also used in this study because it is an objective measure that does not require manual inspection.

Re-Ranking Procedure. To further increase docking accuracy, the 10 solutions obtained from the 10 independent docking runs were re-ranked using a more complex scoring function. In addition to the docking scoring function terms, an sp^2-sp^2 torsion term and a Lennard-Jones 12−6 potential[5] were also used.

The coefficients for the terms were derived from the results obtained by docking each of the 77 complexes. Because this may introduce a bias in the re-ranking score, we cross-validated the results by splitting the set of 77 complexes into two subsets with 39 and 38 complexes (the split point was chosen by sorting the complexes alphabetically by PDB codes). By using coefficients obtained from one set to re-rank the other set (and vice versa), we obtained a combined docking accuracy on the 77 complexes of 83.1%. This slightly lower accuracy is caused either by a small bias in the coefficients or by an insufficient number of complexes in the two training sets. In either case, we concluded that the potential bias in the re-ranking process is not significant and because the re-ranking coefficients used in this study were based on the full dataset containing 77 complexes, we expect the re-ranking score to be almost nonbiased.

Because the re-ranking score is approximately 20 times slower than the docking scoring function it is not suited for docking. However, the execution time for re-ranking a few promising poses obtained from the docking simulation is negligible.

Results

Overall Accuracy. The docking accuracy of MolDock on the 77 complexes is 87.01% with an average RMSD of 1.38 Å. The detailed results are shown in Table 1, which also lists the RMSD results for Glide, Surflex, FlexX, and GOLD (adopted from previously published comparisons[12,15]). The comparison between MolDock and these programs is discussed below.

Comparison with Glide and Surflex. The results listed in Table 2 show the docking accuracy, average RMSDs for all highest ranked poses (all cases), and average RMSDs for successful docking runs (RMSD < 2.0 Å) using MolDock, Glide, and Surflex.

Overall, MolDock obtained a higher docking accuracy compared to that of Glide and Surflex on the dataset that we used in this experiment. The average RMSD over all tested complexes (all cases column) is included for comparison to Glide.[12] As pointed out by Cole et al.,[18] using this measure as an indication of overall success is problematic. For instance, a failed docking solution with an RMSD of 5 Å results in a lower average RMSD value than a docking solution with an RMSD of 8 Å. However, both solutions should be considered equally wrong. To complement the average RMSD measure, we also calculated the average RMSD of the successful docking solutions (RMSD < 2.0 Å column). Here, Glide obtains a lower RMSD to the cocrystallized native ligand than MolDock and Surflex. However, a direct comparison with Glide is complicated because Glide compares the RMSD of the predicted poses to energy-minimized ligands and not to the cocrystallized native ligands.[12] In addition, Glide includes structural waters in two cases (1lna and 1mdr).

Comparison with FlexX and GOLD. Table 3 and Table 4 show the docking accuracies and average RMSDs for all highest ranked poses (all cases) and successful docking runs (RMSD < 2.0 Å) using MolDock, FlexX, and GOLD. Because the published benchmark results[12] for FlexX and GOLD do not include all 77 complexes, the comparison is based on subsets

Table 1. Docking Results for MolDock, Glide, GOLD, FlexX, and Surflex[a]

pdb complex	# of bonds (−H)	# of bonds (+H)	RMSD (highest ranked pose) MolDock	Glide	GOLD	FlexX	Surflex	pdb complex	# of bonds (−H)	# of bonds (+H)	RMSD (highest ranked pose) MolDock	Glide	GOLD	FlexX	Surflex
1abe	0	4	0.26	0.17	0.86	1.16	0.27	1nco	8	15	0.39	6.99	n/a	5.85	8.26
1acj	0	1	0.78	0.28	4.00	0.49	3.89	1phg	3	5	1.38	4.32	1.35	4.74	4.44
1acm	6	6	0.56	0.29	0.81	1.39	1.43	1rds	8	13	4.34	3.75	4.78	4.89	9.83
1aco	4	4	0.42	1.02	0.86	0.96	3.39	1rob	4	7	1.13	1.85	3.75	7.70	0.82
1aha	0	1	0.32	0.11	0.51	0.56	0.37	1snc	6	7	1.69	1.91	n/a	7.48	4.92
1atl	9	13	1.59	0.94	n/a	2.06	7.01	1srj	3	4	0.44	0.58	0.42	2.36	0.39
1baf	7	12	1.60	0.76	6.12	8.27	6.52	1stp	5	5	0.76	0.59	0.69	0.65	0.51
1bbp	11	15	0.99	4.96	n/a	3.75	1.07	1tka	8	11	1.35	2.28	1.88	1.17	1.96
1bma	12	17	1.04	9.31	n/a	13.41	1.00	1tmn	13	15	5.58	2.80	1.68	0.86	1.30
1cbs	5	10	1.43	1.96	n/a	1.68	1.77	1tng	1	2	0.51	0.19	n/a	1.93	0.22
1cbx	5	5	1.06	0.36	0.54	1.35	0.70	1tni	4	5	1.28	2.18	n/a	2.71	2.97
1com	3	4	0.75	3.64	n/a	1.62	0.86	1tnl	1	2	0.46	0.23	n/a	0.71	2.26
1coy	0	3	0.66	0.28	0.86	1.06	0.54	1trk	8	11	0.73	1.64	n/a	1.57	1.22
1dbb	1	4	1.62	0.41	1.17	0.81	0.54	1ukz	4	7	0.36	0.37	n/a	0.94	0.77
1dbj	0	3	0.93	0.20	0.72	1.22	0.88	1ulb	0	1	0.68	0.28	0.32	3.37	0.77
1dr1	2	6	0.65	1.47	1.41	5.64	1.25	1wap	3	4	0.48	0.12	n/a	0.57	0.30
1dwd	9	10	1.07	1.32	1.71	1.66	1.68	2ada	2	6	0.55	0.53	0.40	0.67	0.32
1eap	10	11	2.52	2.32	3.00	3.72	4.89	2ak3	4	7	0.49	0.71	5.08	0.91	0.60
1epb	5	10	3.35	1.78	2.08	2.77	2.87	2cgr	7	7	0.92	0.38	0.99	3.53	1.63
1etr	9	11	1.96	1.48	4.23	7.24	4.05	2cht	2	3	0.43	0.42	0.59	4.58	0.42
1fen	4	10	0.89	0.66	n/a	1.39	1.18	2cmd	5	6	0.50	0.65	n/a	3.75	1.60
1fkg	10	13	1.89	1.25	1.81	7.59	1.81	2ctc	3	4	0.37	1.61	0.32	1.97	0.38
1fki	0	2	0.84	1.92	0.71	0.59	0.70	2dbl	6	9	1.55	0.69	1.31	1.49	0.81
1frp	6	8	0.92	0.27	n/a	1.89	0.75	2gbp	1	6	0.36	0.15	n/a	0.92	0.63
1glq	13	14	7.09	0.29	1.35	6.43	5.68	2lgs	4	5	0.57	7.55	n/a	4.63	1.22
1hdc	6	13	1.71	0.58	10.49	11.74	1.80	2phh	1	2	0.69	0.38	0.72	0.43	0.44
1hdy	0	0	1.73	1.74	0.94	n/a	0.66	2r07	8	10	1.81	0.48	8.23	11.63	1.35
1hri	9	10	6.33	1.59	14.01	10.23	1.98	2sim	5	10	1.29	0.92	0.92	1.99	1.10
1hsl	3	4	0.49	1.31	0.97	0.59	0.51	3cpa	5	7	1.63	2.40	1.58	2.53	1.90
1hyt	5	5	1.61	0.28	1.10	1.62	0.55	3hvt	1	2	0.35	0.77	1.12	10.26	1.64
1lah	4	6	0.32	0.13	n/a	0.28	0.30	3ptb	1	2	0.17	0.27	0.96	0.55	0.54
1lcp	3	6	1.59	1.98	n/a	1.65	2.01	3tpi	6	11	0.36	0.49	0.80	1.07	0.52
1ldm	1	1	0.73	0.30	1.00	0.74	0.44	4cts	3	3	0.60	0.19	1.57	1.53	2.20
1lic	15	16	2.44	4.87	10.78	5.07	3.46	4dfr	9	12	1.39	1.12	1.44	1.40	1.60
1lna	8	12	3.04	0.95	n/a	5.40	0.88	6abp	0	4	0.30	0.40	1.08	1.12	0.28
1lst	5	7	0.23	0.14	0.87	0.71	0.33	6rnt	4	8	7.48	2.22	1.20	4.79	7.03
1mdr	2	4	1.09	0.52	0.36	0.88	0.68	7tim	3	4	0.58	0.14	0.78	1.49	1.20
1mrg	0	1	0.45	0.30	n/a	0.81	0.70	8gch	7	9	4.07	0.30	0.86	8.91	4.51
1mrk	2	6	1.37	1.20	1.01	3.55	0.85								

[a] The RMSD values for the highest ranked pose is shown for each complex.

Table 2. Comparison of Docking Accuracy and Average RMSD Values of MolDock, Glide, and Surflex

method	docking accuracy	average RMSD (all cases) (Å)	average RMSD (Å) (RMSD < 2.0 Å)
MolDock	87.01%	1.38	0.90
Glide	81.82%	1.38	0.74
Surflex	75.32%	1.86	0.91

Table 3. Comparison of Docking Accuracy and Average RMSD Values of MolDock and FlexX

method	docking accuracy	average RMSD (all cases) (Å)	average RMSD (Å) (RMSD < 2.0 Å)
MolDock (76 cases)	86.84%	1.43	0.89
FlexX (76 cases)	57.89%	3.15	1.11

taking 76 complexes and 55 complexes into account for FlexX and GOLD, respectively.

On both subsets (76 and 55 cases), MolDock has a higher docking accuracy than FlexX and GOLD.

Search Robustness. The MolDock docking results presented in Table 1 are based on the highest ranked pose obtained in 10 independent docking runs for each benchmark complex. To test the robustness of MolDock in terms of the number of docking runs needed to identify a highest ranked pose below 2.0 Å (if it was possible), the docking accuracy was recorded while taking into account solutions obtained from 1−20 docking runs. Figure 2 shows the docking accuracy using the highest ranked solution (solid line) and the solution with lowest RMSD to the cocrystallized native ligand (dashed line) as a function of the number of runs conducted. On average, 10 docking runs are sufficient to obtain a high docking accuracy (87.01%).

Table 4. Comparison of Docking Accuracy and Average RMSD Values of MolDock and GOLD

method	docking accuracy	average RMSD (all cases) (Å)	average RMSD (Å) (RMSD < 2.0 Å)
MolDock (55 cases)	83.64%	1.63	0.93
GOLD (55 cases)	78.18%	2.17	0.99

Runtime Performance. On average, MolDock took 110 s for a single docking run on a Pentium IV 2.66 GHz PC running Windows XP (with 512 MB of memory). These run times are only based on successful docking runs. Taking unsuccessful docking runs (RMSD > 2.0 Å) into account, the average run-time was 158 s. In both cases, the run-time for re-ranking the poses was included. The current implementation has not been fully optimized for speed, and therefore further improvements are expected when optimizations such as energy approximations using precalculated grids are included. Moreover, the run-times

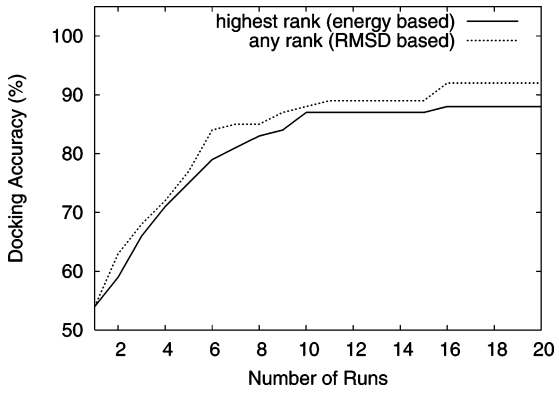

Figure 2. Docking accuracy (using MolDock on the 77 complexes) vs number of docking runs conducted.

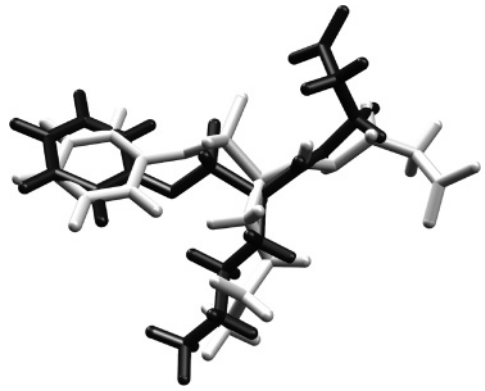

Figure 3. Example of a failed docking solution for Phenyl [1-(1-*N*-succinylamino)pentyl] phosphonate (PDB entry 1eap). The dark colored ligand is the highest ranked pose, and the other is the cocrystallized native ligand.

reported by other docking programs do not typically include the time used to perform the initialization of such energy grids.

Analysis of Failed Predictions. Out of the 77 complexes present in the dataset, all docking programs (MolDock, Glide, Surflex, FlexX, and GOLD) fail on the same set of 3 complexes (1eap, 1lic, 1rds). The 1eap and 1rds ligands are very exposed to the surface, which may explain why the docking programs fail to identify the correct binding modes. The 1lic ligand contains a long alkyl chain, which is very flexible (15 torsions), making it difficult for the docking programs to position it correctly. Regarding the remainder of the complexes in the dataset, MolDock succeeds on nine complexes where Glide fails. Likewise, Glide succeeds on five complexes where MolDock fails. Similarly, MolDock succeeds on 12 complexes where Surflex fails, and Surflex succeeds on three complexes where MolDock fails. These results indicate that no specific docking program outperforms the others on the entire dataset, and successful identification of binding modes might be further improved by combining the results from multiple programs. In fact, if the pose with lowest RMSD to the cocrystallized ligand among the highest ranked poses from MolDock or Glide is used, the docking accuracy increases to 93.5%.

Overall, MolDock failed to identify the binding mode for 10 out of the 77 complexes. Although two of the binding modes (1eap and 8gch) were classified as failures because of the RMSD threshold criterion, most of their functional groups were correctly placed in the binding site. However, the functional groups that were exposed to the surface were not correctly positioned. One example of a failed solution for 1eap is shown in Figure 3. Here, one of the terminal groups located on the molecule surface is not positioned correctly.

In general, it is difficult to predict the binding modes of ligands when there is no well-defined binding site present. For example, the binding mode of 6rnt was not correctly identified because the ligand was located on the protein surface (no cavities were found by the cavity prediction algorithm).

In one of the failed complexes (1epb), successful binding modes (RMSD < 2 Å) were observed among the 10 highest ranked poses. However, the highest ranked solution reported (after re-ranking the poses) was not the one with lowest RMSD to the cocrystallized native ligand.

Discussion and Conclusions

In this article, we introduce a new molecular docking method that we call MolDock, based on guided differential evolution and a force-field based docking scoring function. We evaluated its performance by a series of experiments using 77 complexes. The results of the experiments show that MolDock has a very high docking accuracy with respect to the identification of ligand-binding modes.

Interestingly, the docking experiments reported in this article also show that a simple docking scoring function followed by a re-ranking procedure is adequate for identifying high-quality binding modes in place of more advanced scoring schemes. This is consistent with previous studies confirming that simple scoring functions (e.g., PLP) are as good as or even better at estimating binding affinities than more advanced scoring functions.[19] Experiments with the inclusion of a sp^2-sp^2 torsion term, a solvation term,[20] and a Lennard-Jones 12−6 potential[5] did not improve the identification of correct binding modes compared with the simple scoring function presented in this article. However, the sp^2-sp^2 torsion term and a Lennard-Jones 12−6 potential were able to improve the docking accuracy when used in the re-ranking scoring function. The solvation term did not improve the docking accuracy and was omitted in both scoring schemes. However, the primary reason for the success of MolDock is its search algorithm and the re-ranking scoring function.

The re-ranking scoring function was calibrated from the same 77 complexes, but a cross-validation test showed that any potential bias is small. Although a small bias might be present, the docking accuracy is 88.31% when the docking solution with the lowest RMSD to the cocrystallized native ligand (in 10 runs) is used. If 20 runs are conducted, the docking accuracy (any rank) is 92.21% (see Figure 2). Combining the highest ranked solutions from different docking programs can also significantly increase the overall accuracy. For instance, taking the highest ranked pose from MolDock and Glide (and selecting the best one) results in a docking accuracy of 93.5%.

Another important issue is the preparation of the complexes. In this article, a very simple approach was taken in which the simplified charge and protonation schemes were used. Moreover, only the ligand was energy-minimized in contrast to the approach taken by Glide, where both the ligand and the protein are simultaneously energy minimized to resolve steric clashes. In the MolDock docking experiments, the protein and the ligand were automatically prepared (charges and protonation states were assigned). This automatic preparation combined with an automatic prediction of cavities makes it easy to fully automate the entire benchmarking process.

Future work will focus on further assessment of MolDock using other benchmark datasets, including docking to unbound protein structures, docking of covalently bound ligands, and cross docking between different targets. Currently, about 10 docking runs are needed to identify the binding mode as the

top-ranked solution. Because MolDock only returns the best scoring pose for each docking run, we expect that the robustness can be improved significantly if more (diverse) poses were reported for each docking run. Preliminary experiments taking structural waters into account significantly improves the overall docking accuracy, where 76 out of 77 binding modes are correctly identified (all binding modes were correctly identified except for 1tmn). The feasibility of including models for implicit or explicit water will be further investigated.

Finally, on the basis of the great performance of the re-ranking scoring function, we are currently working on a new re-ranking scheme combining binding-affinity estimation with a re-ranking measure. In addition, energy minimization of the found poses before re-ranking them is expected to further increase the docking accuracy. Work is currently in progress testing MolDock on each of these topics.

Methods

Evolutionary algorithms[21,22] (EAs) are iterative optimization techniques inspired by the Darwinian evolution theory. In EAs, the evolutionary process is simplified, and thus, it has very little in common with real world evolution. Nevertheless, during the last 50 years, EAs have proven their worth as powerful optimization techniques that can assist or replace traditional techniques when these fail or are inadequate for the task being solved.

Basically, an EA consists of a population of individuals (candidate solutions) exposed to random variation by means of variation operators, such as mutation and recombination. The individual being altered is often referred to as the parent, and the resulting solution after modification is called the offspring. Sometimes, more than one parent is used to make the offspring by recombination of solutions, which is referred to as crossover.

A. Guided Differential Evolution. The guided differential evolution algorithm used in MolDock is based on an EA variant called differential evolution (DE). The DE algorithm[6] was introduced by Storn and Price in 1995. Compared to more widely known EA-based techniques (e.g., genetic algorithms, evolutionary programming, and evolution strategies), DE uses a different approach to select and modify candidate solutions (individuals). The main innovative idea in DE is to create offspring from a weighted difference of parent solutions.

The DE works as follows. First, all individuals are initialized and evaluated according to the fitness function (see details below). Afterward, the following process will be executed as long as the termination condition is not fulfilled. For each individual in the population, an offspring is created by adding a weighted difference of the parent solutions, which are randomly selected from the population. Afterward, the offspring replaces the parent, if and only if it is fitter. Otherwise, the parent survives and is passed on to the next generation (iteration of the algorithm). The termination condition used stopped the search process when the current number of fitness (energy) evaluations performed exceeded the maximum number of evaluations allowed (max evaluations parameter setting). Moreover, early termination was allowed if the variance[23] of the population was below a certain threshold (0.01 here). A detailed outline of the generic DE algorithm and the offspring creation scheme is given in Figure S1 and Figure S2 (Supporting Information).

Additionally, guided differential evolution uses a cavity prediction algorithm (introduced below) to constrain predicted conformations (poses) during the search process. More specifically, if a candidate solution is positioned outside the cavity, it is translated so that a randomly chosen ligand atom will be located within the region spanned by the cavity. Naturally, this strategy is only applied if a cavity has been found. If no cavities are reported, the search procedure does not constrain the candidate solutions.

One of the reasons why DE works so well is that the variation operator exploits the population diversity in the following manner: initially, when the candidate solutions in the population are randomly generated, the diversity is large. Thus, when offspring are created, the differences between parental solutions are high, resulting in large step sizes being used. As the algorithm converges to better solutions, the population diversity is lowered, and the step sizes used to create offspring are correspondingly lowered. Therefore, by using the differences between other individuals in the population, DE automatically adapts the step sizes used to create offspring as the search process converges toward good solutions.

Despite the simplistic design of DE and the fact that it only utilizes one variation operator, it has shown great performance both on artificial benchmark problems[24] and real-world problems.[7,25−26]

A.1. Representation. Only the ligand properties were represented in the individuals because the protein remained rigid during the docking process. Thus, a candidate solution was encoded by an array of real-valued numbers representing ligand position, orientation, and conformation as Cartesian coordinates for the ligand translation, four variables specifying the ligand orientation (encoded as a rotation vector and a rotation angle), and one angle for each flexible torsion angle in the ligand (if any).

A.2. Initialization. For each individual in the initial population, each of the three translational parameters (encoded as a position relative to the cocrystallized native ligand) for x, y, and z was assigned a uniformly distributed random number between −15.0 and 15.0 Å, which was added to the center of the cocrystallized reference ligand. Initializing the orientation is more complicated. By just choosing uniform random numbers for the orientation axis (between −1.0 and 1.0 followed by normalization of the values to form a unit vector) and the angle of rotation (between −180° and +180°), the initial population would be biased toward the identity orientation (i.e., no rotation of the original coordinate system). To avoid this bias, we used the algorithm by Shoemake et al.[27] for generating uniform random quaternions and converted these quaternions to their rotation axis/rotation angle representation. The flexible torsion angles (if any) were given a random angle between −180° and +180°.

A.3. Fitness Evaluation. The fitness of a candidate solution is the sum of the intermolecular interaction energy between the ligand and the protein, and the intramolecular interaction energy of the ligand. In this study, the fitness was calculated using the docking scoring function described below.

A.4. Algorithmic Settings. The following parameters were used for the guided differential evolution algorithm: population size = 50, crossover rate = 0.9, scaling factor = 0.5, and max evaluations = 100 000. These settings were found by trial and error in a few preliminary runs and generally gave the best results across all the 77 complexes.

B. Cavity Prediction. To determine the potential binding sites, a simple grid-based cavity prediction algorithm was developed. The cavity prediction algorithm works as follows: first, a discrete grid with a resolution of 0.8 Å, covering the protein, is created. At every grid point, a sphere of radius 1.4 Å is placed. Whether this sphere will overlap with any of

Table 5. Charge Templates

charge	ligand atoms	protein atoms
0.5	N atoms in $-C(NH_2)_2$	His (ND1/NE2) Arg (NH1/NH2)
1.0	N atoms in $-N(CH_3)_2$, $-(NH_3)$	Lys (N)
−0.5	O atoms in $-COO$, $-SO_4$, $-PO_2$, $-PO_2-$	Asp (OD1/OD2) Glu (OE1/OE2)
−0.66	O atoms in $-PO_3$	
−0.33	O atoms in $-SO_3$	
−1.0	N atoms in $-SO_2NH$	

the spheres determined by the van der Waals radii of the protein atoms was checked. Grid points where the probe clashes with the protein atom spheres will be referred to as part of the inaccessible volume; all other points are referred to as accessible. Second, each accessible grid point is checked to see whether it is part of a cavity using the following procedure. From the current grid point, a random direction is chosen, and this direction (and the opposite direction) is followed until the grid boundaries are hit, checking if an inaccessible grid point is hit on the way. This is repeated a number of times, and if the percentage of lines hitting an inaccessible volume is larger than a given threshold, the point is marked as being part of a cavity. In this study, 16 different directions were tested, and a grid point was assumed part of a cavity if 12 or more of these lines hit an inaccessible volume. The threshold can be tuned according to how enclosed the found cavities should be. A value of 0% would only be possible far from the protein as opposed to a value of 100% corresponding to a binding site deeply buried in the protein. The final step is to determine the connected regions. Two grid points are connected if they are neighbors. Regions with a volume below 10.0 Å^3 are discarded as irrelevant (the volume of a connected set of grid points is estimated as the number of grid points times the volume of a unit grid cell). The cavities found are then ranked according to their volume.

The algorithm shares similarities with the LIGSITE algorithm written by Hendlich et al.[28] However, the introduced algorithm is more flexible because there is no dependence on the orientation of the target molecule and an arbitrary number of directions can be used. In contrast, LIGSITE uses flood-filling in seven fixed directions.

C. Scoring Function. The scoring function used by MolDock is derived from the PLP scoring functions originally proposed by Gehlhaar et al.[8,9] and later extended by Yang et al.[10] The scoring function used by MolDock further improves these scoring functions with a new hydrogen bonding term and new charge schemes. The docking scoring function, E_{score}, is defined by the following energy terms

$$E_{score} = E_{inter} + E_{intra}$$

where E_{inter} is the ligand−protein interaction energy:

$$E_{inter} = \sum_{i \in \text{ligand}} \sum_{j \in \text{protein}} \left[E_{PLP}(r_{ij}) + 332.0 \frac{q_i q_j}{4 r_{ij}^2} \right]$$

The summation runs over all heavy atoms in the ligand and all heavy atoms in the protein, including any cofactor atoms and water molecule atoms that might be present (in our benchmarks, all water molecules were removed before docking). The E_{PLP} term is the piecewise linear potential described below. The second term describes the electrostatic interactions between charged atoms. It is a Coulomb potential with a distance-dependent dielectric constant given by $D(r) = 4r$. The numerical value of 332.0 fixes the units of the electrostatic energy to kcal/mol. For distances less than 2.0 Å, the electrostatic energy is cut off at the level corresponding to a distance of 2.0 Å to ensure that no energy contribution can be higher than the clash penalty. Notice that although the electrostatic energy contribution has the theoretically predicted magnitude the other energy terms are empirically motivated, and the total energy does not necessarily correlate with the true binding affinity. The charges are set according to the scheme listed in Table 5. Metal ions are assigned a charge of +1 (e.g., Na) or +2 (e.g., Zn, Ca, Fe).

Table 6. PLP Parameters

	A_0	A_1	R_1	R_2	R_3	R_4
hydrogen bond	20.0	−2.5	2.3	2.6	3.1	3.6
steric	20.0	−0.4	3.3	3.6	4.5	6.0

Table 7. Hydrogen Bonding Atom Types

type	atoms
acceptor	N and O (with no H's attached)
donor	N and S (with one or more H's attached)
both	O (with one H attached) or O in structural water
nonpolar	all other atoms

E_{PLP} is a piecewise linear potential using two different sets of parameters: one set for approximating the steric (van der Waals) term between atoms and the other stronger potential for hydrogen bonds. The linear potential is defined by the following functional form: $E_{PLP}(0) = A_0$, $E_{PLP}(R_1) = 0$, $E_{PLP}(R_2) = E_{PLP}(R_3) = A_1$, $E_{PLP}(r) = 0$ for $r \geq R_4$ and is linearly interpolated between these values. The parameters used here (see Table 6) were adopted from GEMDOCK.[10]

A bond is considered a hydrogen bond if one of the atoms can donate a hydrogen atom and the other atom can accept it. The atom types are assigned according to the scheme shown in Table 7.

The PLP hydrogen bond term mentioned above only depends on the distance between atoms. To take into account the directionality of hydrogen bonding, the geometry of the hydrogen bond is examined, and the following factor, H_{factor}, is multiplied to the PLP hydrogen bond strength:

$$H_{factor} = \Phi(\angle_{D-H-A}; 90°; 100°) \cdot \Phi(\angle_{H-A-AA}; 90°; 100°) \cdot \Phi(\angle_{D-A-AA}; 90°; 150°)$$

Here, AA (Acceptor Antecedent) denotes a heavy atom connected to the acceptor (A), D denotes the donor, and H is the donated hydrogen atom. The ramp function Φ is defined as $\Phi(A; A_{min}; A_{max}) = 0$ for $A \leq A_{min}$ and $\Phi(A; A_{min}; A_{max}) = 1$ for $A \geq A_{max}$ and is linearly interpolated between these values for $A_{min} < A < A_{max}$. If it is not possible to calculate one of these factors, it is omitted. This is, for example, the case for hydroxyl groups where the exact location of the hydrogen is not investigated during docking, and the two first factors cannot be calculated. The angle checks above were motivated by the approach taken by McDonald and Thornton.[29]

Table 8. Torsional Parameters

	θ_0	m	A
sp^2-sp^3	0.0	6	1.5
sp^3-sp^3	π	3	3.0

E_{intra} is the internal energy of the ligand:

$$E_{\text{intra}} = \sum_{i \in \text{ligand}} \sum_{j \in \text{ligand}} E_{\text{PLP}}(r_{ij}) + \sum_{\text{flexible bonds}} A[1 - \cos(m \cdot \theta - \theta_0)] + E_{\text{clash}}$$

The double summation is between all atom pairs in the ligand, excluding atom pairs that are connected by two bonds or less. The second term is a torsional energy term, parametrized according to the hybridization types of the bonded atoms (see Table 8). θ is the torsional angle of the bond. Notice that this angle is not necessarily uniquely determined. In this study, the average of the torsional energy bond contribution was used if several torsions could be determined. The last term, E_{clash}, assigns a penalty of 1000 if the distance between two atoms (more than two bonds apart) is less than 2.0 Å. Thus, the E_{clash} term punishes infeasible ligand conformations.

Acknowledgment. We thank Nuevolution A/S and Birgit Schiøtt's Research Group, Department of Chemistry, University of Aarhus for valuable feedback and evaluation of MolDock.

Supporting Information Available: Pseudo-code for the differential evolution algorithm. This material is available free of charge via the Internet at http://pubs.acs.org. MolDock is part of a new molecular docking product, Molegro Virtual Docker (MVD), integrating the preparation of complexes, cavity prediction, and docking in a user-friendly GUI. A trial version of MolDock is available for Linux, Windows XP, and Mac OS X (see http://www.molegro.com for details on obtaining the software).

References

(1) Kirkpatrick, S.; Gelatt, C. D., Jr.; Vecchi, M. P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671−680.
(2) Goodsell, D. S.; Olson, A. J. Automated Docking of Substrates to Proteins by Simulated Annealing. *Proteins* **1990**, *8*, 195−202.
(3) Westhead, D. R.; Clark, D. E.; Murray, C. W. A Comparison of Heuristic Search Algorithms for Molecular Docking. *J. Comput.-Aided Mol. Des.* **1997**, *11*, 209−228.
(4) Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. Development and Validation of a Genetic Algorithm for Flexible Docking. *J. Mol. Biol.* **1997**, *267*, 727−748.
(5) 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. *J. Comput. Chem.* **1998**, *19*, 1639−1662.
(6) Storn, R.; Price, K. *Differential Evolution − A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*; Technical Report; International Computer Science Institute: Berkley, CA, 1995.
(7) Thomsen, R. Flexible Ligand Docking Using Differential Evolution, *Proceedings of the 2003 Congress on Evolutionary Computation* **2003**, *4*, 2354−2361.
(8) Gehlhaar, D. K.; Verkhivker, G.; Rejto, P. A.; Fogel, D. B.; Fogel, L. J.; Freer, S. T. Docking Conformationally Flexible Small Molecules into a Protein Binding Site through Evolutionary Programming. *Proceedings of the Fourth International Conference on Evolutionary Programming* **1995**, 615−627.
(9) Gehlhaar, D. K.; Bouzida, D.; Rejto, P. A. Fully Automated and Rapid Flexible Docking of Inhibitors Covalently Bound to Serine Proteases. *Proceedings of the Seventh International Conference on Evolutionary Programming* **1998**, 449−461.
(10) Yang, J.-M.; Chen, C.-C. GEMDOCK: A Generic Evolutionary Method for Molecular Docking. *Proteins* **2004**, *55*, 288−304.
(11) Nissink, J. W. M.; Murray, C.; Hartshorn, M.; Verdonk, M. L.; Cole, J. C.; Taylor, R. A New Test Set for Validating Predictions of Protein−Ligand Interaction. *Proteins* **2002**, *49*, 457−471.
(12) Friesner, R. A.; Banks, J. L.; Murphy, R. B.; Halgren, T. A. Glide: A New Approach for Rapid Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *J. Med. Chem.* **2004**, *47*, 1739−1749.
(13) Rarey, M.; Kramer, B.; Lengauer, T.; Klebe, G. A Fast Flexible Docking Method Using an Incremental Construction Algorithm. *J. Mol. Biol.* **1996**, *261*, 470−489.
(14) Kramer, B.; Rarey, M.; Lengauer, T. Evaluation of the FlexX Incremental Construction Algorithm for Protein−Ligand Docking. *Proteins* **1999**, *37*, 228−241.
(15) Jain, A. N. Surflex: Fully Automatic Molecular Docking Using a Molecular Similarity-Based Search Engine. *J. Med. Chem.* **2003**, *46*, 499−511.
(16) Glide treats bonds to methyl groups as rigid, resulting in a slightly lower number of rotatable bonds than the ones reported in Table 1.
(17) Clark, M.; Cramer, R. D.; Van Opdenbosch, N. Validation of the General-Purpose TRIPOS 5.2 Force Field. *J. Comput. Chem.* **1989**, *10*, 982−1012.
(18) Cole, C. J.; Murray, C. W.; Nissink, J. W. M.; Taylor, R. D.; Taylor, R. Comparing Protein−Ligand Docking Programs Is Difficult. *Proteins* **2005**, *60*, 325−332.
(19) Wang, R.; Lu, Y.; Wang, S. Comparative Evaluation of 11 Scoring Functions for Molecular Docking. *J. Med. Chem.* **2003**, *46*, 2287−2303.
(20) Stouten, P. F. W.; Frömmel, C.; Nakamura, H.; Sander, C. An Effective Solvation Term Based on Atomic Occupancies for Use in Protein Simulations. *Mol. Simul.* **1993**, *10*, 97−120.
(21) Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*; Springer-Verlag: Berlin, 1992.
(22) Michalewicz, Z.; Fogel, D. B. *How to Solve It: Modern Heuristics*; Springer-Verlag: Berlin, 2000.
(23) The variance is computed as the average of the distances between a mean individual and every individual in the population.
(24) Vesterstrøm, J.; Thomsen, R. A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems. *Proceedings of the 2004 Congress on Evolutionary Computation* **2004**, *2*, 1980−1987.
(25) Ursem, R. K.; Vadstrup, P. Parameter Identification of Induction Motors Using Differential Evolution. *Proceedings of the 2003 Congress on Evolutionary Computation* **2003**, *2*, 790−796.
(26) Paterlini, S.; Krink, T. Differential Evolution And Particle Swarm Optimization in Partitional Clustering. *Comput. Stat. Data. An.* **2005**, *50*, 1220−1247.
(27) Shoemake, K. Uniform Random Rotations. In *Graphics Gems III*, 1st ed.; Kirk, D., Ed.; AP Professional (Academic Press); Boston, MA, 1992; pp 124−132.
(28) Hendlich, M.; Rippmann, F.; Barnickel, G. LIGSITE: Automatic and Efficient Detection of Potential Small Molecule-Binding Sites in Proteins. *J. Mol. Graphics Modell.* **1998**, *15*, 359−363.
(29) McDonald, I. K.; Thornton, J. M. Satisfying Hydrogen Bonding Potential in Proteins. *J. Mol. Biol.* **1994**, *238*, 777−793.

JM051197E