JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

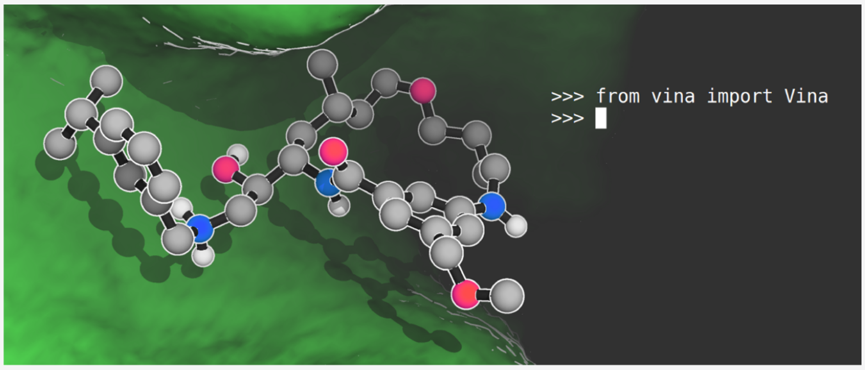
AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings

Jerome Eberhardt,[*,†] Diogo Santos-Martins,[†] Andreas F. Tillack, and Stefano Forli[*]

Cite This: https://doi.org/10.1021/acs.jcim.1c00203

Read Online

ACCESS | Metrics & More | Article Recommendations | SI Supporting Information



```
>>> from vina import Vina
>>> █
```

ABSTRACT: AutoDock Vina is arguably one of the fastest and most widely used open-source programs for molecular docking. However, compared to other programs in the AutoDock Suite, it lacks support for modeling specific features such as macrocycles or explicit water molecules. Here, we describe the implementation of this functionality in AutoDock Vina 1.2.0. Additionally, AutoDock Vina 1.2.0 supports the AutoDock4.2 scoring function, simultaneous docking of multiple ligands, and a batch mode for docking a large number of ligands. Furthermore, we implemented Python bindings to facilitate scripting and the development of docking workflows. This work is an effort toward the unification of the features of the AutoDock4 and AutoDock Vina programs. The source code is available at https://github.com/ccsb-scripps/AutoDock-Vina.

INTRODUCTION

AutoDock Vina (Vina)[1] is one of the docking programs in the AutoDock Suite,[2] together with AutoDock4 (AD4),[3] AutoDock-GPU,[4] AutoDockFR,[5] and AutoDock-CrankPep.[6] Vina is arguably among the most widely used programs, probably because of its ease of use and speed, when compared to the other docking programs in the suite and elsewhere, as well as being open source.

Research groups around the world have modified and built upon the Vina source code, improving the search algorithm (QuickVina2[7]), made the interface more user friendly allowing modification of scoring terms through the user interface (Smina[8]), and improved the scoring function for carbohydrate docking (Vina-Carb[9]) and halogen bonds (VinaXB[10]), as well as ranking and scoring (Vinardo[11]).

Besides these valuable developments, there are still several methods within the AutoDock Suite that are not available in the Vina program because they have been implemented specifically for either the AD4 scoring function or the AD4 program. Examples of such methods include docking with macrocyclic flexibility,[12] specialized metal coordination models,[13] modeling of explicit water molecules,[14] coarse-grained ligand models,[15] and ligand-irreversible binding.[16] Despite being a less-efficient program, AD4 allows the user to modify a large number of docking parameters, providing direct access to some of the engine internals, making it well suited for the development of new docking methods. Conversely, the Vina interface is highly specialized and optimized, and one of its hallmarks is the very limited amount of user input necessary to perform a docking. In turn, this makes it impossible to implement additional functionality without significant changes in the source code.

The usefulness of such specialized methods is hindered by the poor search efficiency of the AD4 program. In fact, AD4 can be up to 100× slower than Vina,[1] depending on the search complexity. The large performance difference is due to the better search algorithm used in Vina, a Monte-Carlo (MC) iterated search combined with the BFGS[17] gradient-based optimizer. In comparison with the Lamarckian genetic algorithm and Solis-Wets local search of AD4,[3] the search

Received: February 22, 2021

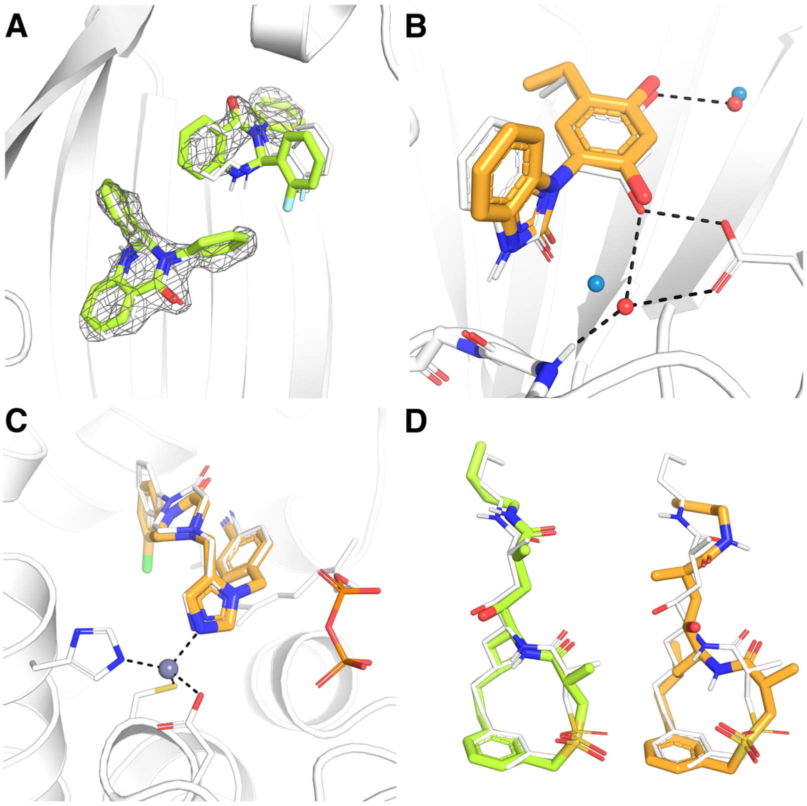


Figure 1. Example applications of AutoDockVina 1.2.0 for docking (A) multiple ligands (PDB 5x72), (B) with water molecules using the hydrated docking protocol from AutoDock4 (PDB 4ykq), (C) in the presence of zinc using the AutoDock4$_{Zn}$ forcefield (PDB 1s63), or (D) flexible macrocycles (compound 19 from the BACE dataset of the D3R Grand Challenge 4). Proteins are represented in white cartoon and crystal poses and protein residues in white thin sticks. The 2Fo−Fc electron-density map, contoured at 2.0σ, is colored gray. The docking poses are represented in sticks, and colored in green and orange when docked using the Vina or AutoDock4 scoring function, respectively. Docking with zinc was done in the presence of the farnesyl disphosphate molecule, represented in sticks and colored in white.

efficiency of Vina leads to better docking results with fewer scoring function evaluations.

Here, we implemented the AD4 scoring function in the Vina program. Furthermore, some of the specialized features available in AD4 were also ported to the Vina source code, enabling their use with the Vina's powerful MC/BFGS search algorithm. Then, we further extended the Vina program enabling simultaneous docking of multiple ligands, and added Python bindings to facilitate programmatic access to the docking engine functionalities.

■ SCORING FUNCTION EXTENSIONS AND IMPROVEMENTS

AutoDock4.2 Scoring Function. One major improvement is the availability of the AD4 scoring function in Vina. This allows users to access it using the same Vina MC-based search algorithm and explore with equal efficiency its energy landscape. This will likely facilitate large-scale consensus docking virtual screening campaigns.[18,19]

The AD4 and Vina scoring functions are quite different. AD4 uses a physics-based[3] model with van der Waals, electrostatic, and directional hydrogen-bond potentials derived from an early version of the AMBER force field,[3,20] a pairwise-additive desolvation term based on partial charges, and a simple conformational entropy penalty. On the other hand, Vina lacks electrostatics and solvation,[1] and consists of a van der Waals-like potential (defined by a combination of a repulsion term and two attractive Gaussians), a nondirectional hydrogen-bond term, a hydrophobic term, and a conformational entropy penalty.

Performance wise, in Vina 1.2.0, the average time required to perform energy evaluations with the AD4 scoring function is nearly 3× larger than with the Vina scoring function. This is due to the presence of additional electrostatic and desolvation maps that need to be interpolated for each movable atom.

Grid Map File Support. Both the AD4 and Vina programs calculate intermolecular interactions by performing trilinear interpolations of grid maps precalculated on the target structure. Vina also uses the target structure to perform a postprocessing minimization of the docked poses. In AD4, maps are precalculated using a separate program (AutoGrid[2]) prior to docking and loaded at runtime, while Vina calculates them on-the-fly prior to running the MC search. The availability to accessible grid map files generated by AutoGrid provided the foundations for a number of specialized methods, such as the zinc-coordination potentials in the AutoDock4$_{Zn}$ force field,[13] biasing docking using information from molecular dynamics (MD) simulations in AutoDock-Bias,[21] and the integration of Grid Inhomogenous Solvation Theory (GIST)[22−24] in AutoDock-GIST.[25] GIST is a method to analyze MD simulations and characterize thermodynamic properties of water molecules. It provides a more accurate representation of water molecule interactions with the receptor within ligand-binding sites, but at the expense of a much higher computational cost compared to implicit solvent models.

https://doi.org/10.1021/acs.jcim.1c00203

In AutoDock Vina 1.2.0, we added the support to optionally load external grid map files, enabling all these methods in both the AD4 and Vina scoring functions. These methods can be applied by following the existing protocols to prepare target structures and the corresponding grid maps, then replace the AutoDock4 binary with the new version of Vina. The availability of reading and writing maps facilitates the development of similar methods for the Vina scoring function.

New Atom Types. We extended both the Vina and AD4 scoring functions to support new atom types for atoms and pseudoatoms as required by the hydrated docking method (atom type W) and the macrocycle sampling methods (G*x* and CG*x* atom types, with *x* ranging from 0 to 3 for handling ligands containing multiple macrocycles). These atom types are implemented in the source code. Additionally, we also added parameters for silicon (atom type Si) to address user requests for better supporting the chemical space covered in public repositories such as the ZINC database.[26]

NEW DOCKING METHODS

We increased the number of docking methods available in Vina leveraging the availability of new atom types, the possibility of specifying grid map files to be used during docking, and by extending the existing code.

Simultaneous Multiple Ligand Docking. Vina is now able to dock simultaneously multiple ligands. This functionality may find application in fragment-based drug design, where small molecules that bind the same target can be grown or combined into larger compounds with potentially better affinity.

The protein PDEδ in complex with two inhibitors (PDB 5x72)[27] was used as a proof of concept to test the ability of Vina to successfully dock multiple ligands simultaneously. The two inhibitors in this structure are stereoisomers, and only the *R*-isomer is able to bind in a specific region of the pocket, while both the *R*- and *S*-isomers can bind to the second location. Using the Vina scoring function, the best set of poses (top 1) shows an excellent overlap with the crystallographic coordinates for one of the isomers, and reasonable overlap with the electron density for the other isomer, which shows some degree of ambiguity (Figure 1A). Using the AutoDock4 scoring function, a similar performance in overlapping the crystallographic poses is found, but only when considering the first two sets of poses (top 2).

Hydrated Docking. The hydrated docking protocol was developed to model water molecules directly involved in the ligand−receptor interaction.[14] The method is based on docking ligands explicitly hydrated with spherical water molecules, and can be used to predict the position and the role (i.e., bridging or displaced) of individual water molecules and generally improve ligand pose predictions. Water molecules are represented by a single atom of type W, and are added to the ligand molecule at the end of each hydrogen bond vector. During docking, W atoms move along with the ligand, do not contribute to intramolecular interactions, and are allowed to overlap with the protein. In fact, when that happens, a water molecule is considered displaced (i.e., removed from the system), and a reward is added to the score to reflect the entropy gain resulting from releasing the water molecule to the bulk solvent. This entropy reward does not take the receptor into account. It is a constant value that was calibrated to 0.2 kcal/mol in the original study.[14] Following the standard hydrated docking protocol,[14] the W map, which represents water−receptor interactions, is obtained by combining the oxygen-acceptor (OA) and hydrogen-donor (HD) maps of the AD4 force field.

To validate the implementation of this docking protocol in Vina 1.2.0, we used six HSP90 protein−ligand complexes from the D3R Grand Challenge 2015.[28] This is an interesting system for the hydrated docking because different ligands bind with a different number of water molecules bridging hydrogen bonds with the protein. Details about the screening library and receptor preparation and analysis method are discussed in Supporting Information. The docking success rate is reported in Figure 2, and a hand-picked system (PDB 4ykq) is depicted in Figure 1B. The results show that the docking success rate is higher using the hydrated docking protocol (Figure 2). Considering only the top pose, the success rate is increased by 17 percentage points, from 50 to 67%, and goes up to 83% with the top three poses. For two of the six ligands (4yku and 4ykx), a net improvement is observed with RMSD below 1 Å compared to average RMSD values above 2 Å 6.1 and 2.3 Å for 4yku and 4ykx, respectively (see Table S1). However, a decrease in accuracy in observed for 4kyz, with an average RMSD of 5.2 Å for the top 1 pose using the hydrated docking, compared to an average RMSD of 1.9 Å without water molecules. The correct pose is found in only one of the 10 docking replicates as the top 2 poses, and in only two replicates as the top 3 poses.

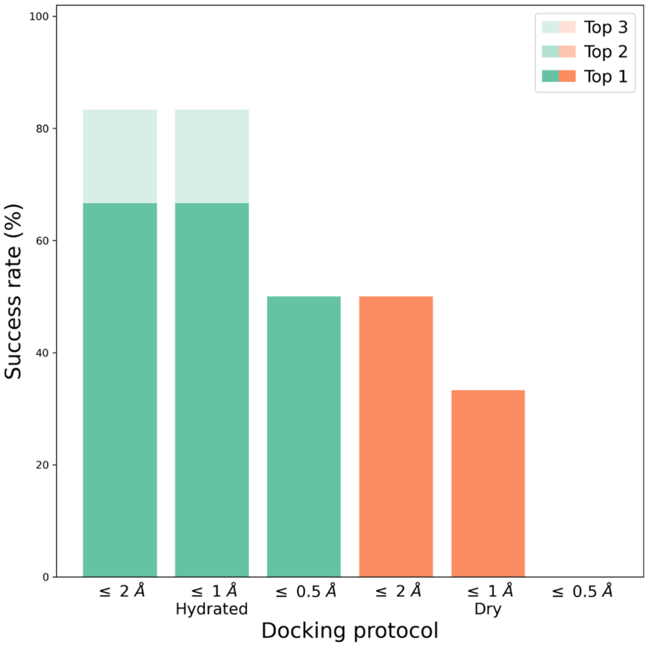


Figure 2. Docking success rate of six ligands redocked against HSP90, using the AutoDock4.2 scoring function, with and without the hydrated docking protocol considering the top 1, top 2, and top 3 poses. The pose prediction was considered as successful if the RMSD was inferior than 2, 1, or 0.5 Å from the crystal pose.

AutoDock4$_{Zn}$. One of the most used methods developed for AD4 is the AutoDock4$_{Zn}$, a specialized force field to model zinc-coordinating ligands.[13] It is based on the use of pseudoatoms to describe the optimal tetrahedral coordination geometry of the zinc ion complexed in proteins, and the definition of improved potentials to describe its interaction with coordinating elements in the ligand (i.e., nitrogen, oxygen, and sulfur). The coordination geometry is encoded in the grid maps for the standard AD4 atom types. The results of the implementation of this method in Vina are shown in Figure 1C. The method is capable of reproducing the improved docking performance reported for the original work with AD4, showing an excellent overlap with the crystallographic pose of the ligand and optimal zinc coordination geometry.

Macrocycle Conformational Sampling. Docking of macrocycles is a challenging task because of the difficulty of sampling the ring flexibility by modeling the correlated torsional changes resulting in different conformations. AD4 has a specialized protocol to dock macrocycles while modeling their flexibility on-the-fly.[12] One of the bonds in the ring structure is broken, resulting in an open form of the macrocycle that removes the need for correlated torsional variations, enabling torsional degrees of freedom to be explored independently. During the docking, a linear potential is applied to restore the bond resulting in the closed ring form. Thus, macrocycle conformations are sampled while adapting to the binding pocket, at the cost of increased search complexity with the added extra rotatable bonds. This method was successfully applied in the D3R Grand Challenge 4,[29] both by us[30,31] and others.[32]

The current implementation of macrocycle sampling in AutoDock Vina 1.2.0 is the same as in AutoDock-GPU,[4] which differs from the original approach[12] by the use of dummy atoms. The dummy atom implementation was previously described,[30] and is summarized herein. To each of the atoms previously connected by the broken bond, a dummy atom is added. The distance between each dummy atom and its parent atom corresponds to the length of the broken bond, and the 1−3 angle matches the original bond geometry. During docking, a linear potential attracts each dummy atom to overlap with the opposite parent atom, restoring the broken bond with the proper distance and 1−3 angles.

To validate our implementation in Vina, we used 19 macrocycles from the BACE-1 set of the D3R Grand Challenge 4 (Figure 1D). We tested both the AD4 and Vina scoring functions, a linear potential of 5 or 50 kcal/mol/Å,[30] and search exhaustiveness of 8 or 64 (Table 1). The lowest RMSD with respect to the experimental coordinates were achieved with the Vina scoring function, using exhaustiveness of 64 and a linear potential of 50 kcal/mol/Å. The AD4 scoring function seemed to perform better at lower exhaustiveness, while the Vina scoring function required higher exhaustiveness values to achieve good performance. Overall, the magnitude of the linear potential is the least important parameter for reproducing crystallographic poses, and the exhaustiveness is the most important one. The relatively large number of rotatable bonds (about 20) may explain why the search exhaustiveness is the most important factor for predicting the crystallographic pose of these molecules.

Table 1. Redocking of 19 Macrocycles of the BACE-1 Set from the D3R Grand Challenge 4

scoring function	exhaust	linear pot (kcal/mol/Å)	RMSD average	RMSD median
AD4	8	5	2.33	1.52
AD4	8	50	3.03	1.74
AD4	64	5	2.11	1.54
AD4	64	50	2.04	1.50
Vina	8	5	5.93	7.71
Vina	8	50	5.10	5.73
Vina	64	5	1.82	1.02
Vina	64	50	1.22	0.77

PYTHON BINDINGS

Leveraging the popularity and utility of the Python language,[33] we added bindings for the language in the version 1.2.0. In order to generate a Python interface as compliant (i.e., *pythonic*) as possible with the language guidelines, the Vina code was refactored as a library. A Python extension module was created automatically from the C++ code using Simplified Wrapper and Interface Generator (SWIG).[34] Most of the features are provided either by binding directly to the existing the C++ code, or via additional convenience functions to simplify the access from the Python environment.

The availability of Python bindings facilitates the use and integration of the Vina docking engine in complex and articulated pipelines, reducing the code burden necessary to integrate the docking process with the numerous Python packages and other software suites that support the language. Through these bindings, users can embed the docking engine directly in any Python pipeline by importing directly the Vina package instead of spawning and managing external processes. We anticipate that this will allow users from the community to more rapidly design, implement, and distribute multistep docking protocols, as well as facilitate their integration in web services.

The Python interface provides the following features:

- create an instance of the AutoDock Vina engine (scoring function choice, CPU cores, and random seed)
- read/write one or more PDBQT files
- compute Vina affinity maps
- read/write Vina affinity maps and read AutoDock affinity maps
- randomize orientation and position of the input ligand(s) (randomize_only)
- evaluate the energy of the current pose or poses (score_only)
- perform local optimization (local_only)
- set Monte-Carlo global search parameters (exhaustiveness, number of output poses, maximum evaluations, etc...)

Thus, a basic Vina calculation can be configured and performed as follows:

The code is documented using Python docstrings, and the documentation is automatically generated using Sphinx.[35]

MISCELLANEOUS IMPROVEMENTS

Batch Ligand Docking. AutoDock Vina 1.2.0 can dock an arbitrary number of ligands with a single launch of the program. Multiple ligand file names can be specified with the new option --batch and each ligand is docked without recalculating or loading the maps every time for each ligand. This improves computing efficiency when running very large virtual screenings.

Setting the Number of Evaluations. Vina performs eight independent MC runs by default. For more complex searches (i.e., more flexible ligands and larger binding sites), this number can be modified with the exhaustiveness parameter. Conversely, the number of energy evaluations performed in each run is determined using heuristics that take into account the number of atoms and rotatable bonds. In this new version,

https://doi.org/10.1021/acs.jcim.1c00203

```python
#!/usr/bin/env python
# Simple example with Vina Python bindings
#

from vina import Vina


v = Vina()

v.set_receptor("protein.pdbqt")
v.set_ligand_from_file("ligand.pdbqt")

v.compute_vina_maps([0., 0., 0.], [30, 30, 30])
v.dock(exhaustiveness=32)

v.write_poses("docking_results.pdbqt")
```

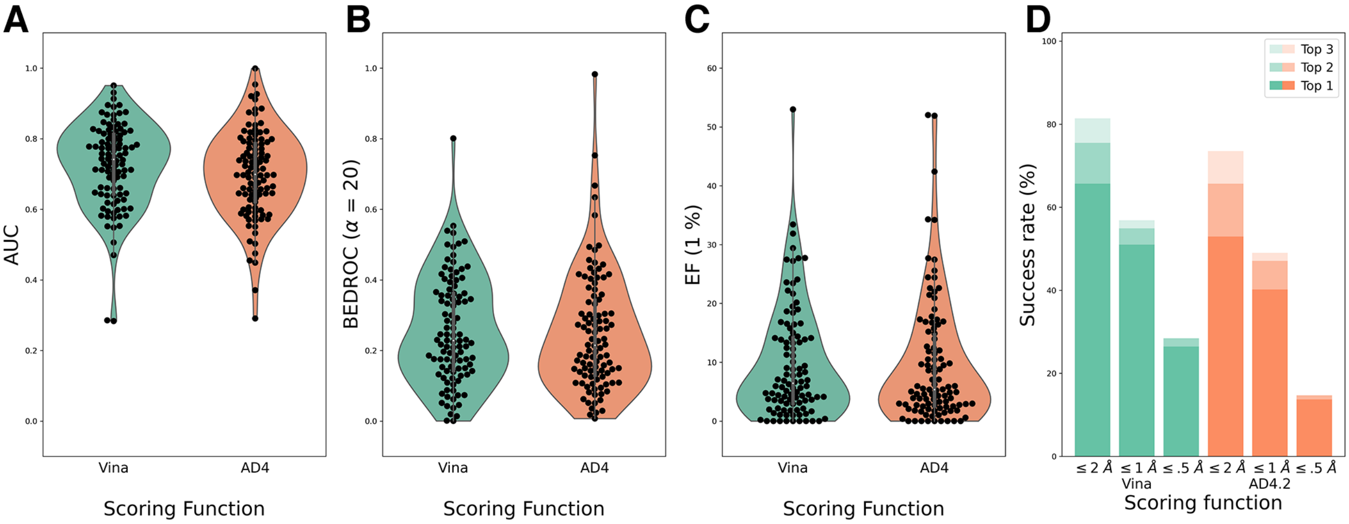


Figure 3. Early recognition of active compounds from the DUD-E dataset and crystal pose prediction. All 102 targets from the DUD-E dataset were selected and used to compare Vina and AutoDock4.2 scoring functions in AutoDock Vina. Violin plots of (A) AUC, (B) BEDROC using an α of 160.9 and (C) EF at 1%. (D) Docking success rate for Vina and AutoDock4.2 scoring functions using crystal poses considering the top 1, top 2, and top 3 poses. The pose prediction was considered as successful if the RMSD was inferior than 2, 1, or 0.5 Å from the crystal pose.

we added an option --max_evals that allows users to specify the number of evaluations to be performed (analogous to the ga_num_evals option in AD4), providing more control over the search algorithm.

Optionally Disable Pose Refinement. By default, Vina uses the receptor structure prior to docking to precalculate grid maps, and after dockings are completed to minimize poses using direct pairwise interactions with the receptor (instead of using the precalculated grid maps as during docking). However, when map files are loaded instead of calculated internally, the refinement with receptor atoms is disabled because there is no way to guarantee consistency between the internal energy potentials used for docking and those used for calculating the grid maps. In fact, one of the purposes of loading maps from external files is to explicitly allow the user to modify them. To avoid any ambiguity, the rigid receptor file and maps are not allowed to be specified at the same time. When docking with the AD4 scoring function, the post-processing minimization is never available, and grid maps must be provided. Postdocking refinement for the Vina scoring function can now be disabled with the --no_refine option.

■ VIRTUAL SCREENING PERFORMANCE COMPARISON

With the possibility of using the same search method for both AD4 and Vina scoring functions, it is now possible to homogeneously assess their screening performance (i.e., without the uncertainty of the different search methods). Therefore, we performed virtual screenings using all 102 targets from the DUD-E dataset,[36] for a total of 22,840 actives and 1,388,885 decoys compounds. For each target, active and decoy sets, and cocrystallized ligands, were docked. Details about the screening library and receptor preparation and analysis are discussed in Supporting Information.

The results (Figure 3) show that overall both Vina and AD4 scoring function perform similarly in early recognition, but the Vina scoring function reproduces crystal poses with a higher accuracy. For Vina and AD4, respectively, the average AUC values were 0.72 ± 0.12 and 0.70 ± 0.12, BEDROC 0.26 ± 0.15 and 0.25 ± 0.17, and for EF (1%) 9.70 ± 9.43 and 9.70 ± 10.60. However, the two scoring functions show different performance depending on the targets. Based on the BEDROC metric, the AD4 scoring function outperforms the Vina scoring function for the following targets: *pur2*, *fpps*, *tryb1*, *xiap*, and *nram* but underperforms for *thb*, *fak1*, *ada17*, *hdac8*, and *pgh2*. For 25 out of 112 targets, both scoring functions perform poorly, with BEDROC metrics lower than 0.1 (see Tables S2 and S3). In terms of the success rate in reproducing experimental coordinates within 2 Å RMSD, 68 and 54% of them are correctly predicted when considering only the top pose (top 1) for the Vina and AD4 scoring functions, respectively. When considering the first two poses (top 2), the success rates increase to 78 and 68%, and using the first three poses (top 3) to 84 and 76% for Vina and AD4, respectively. When using a more stringent cutoff of 0.5 Å RMSD, only 27 and 14% of the top poses (top 1) are correctly predicted for the Vina and AD4 scoring functions, respectively. Those results are aligned with recent studies showing that on average the Vina scoring function outperforms the AD4 scoring function for pose prediction.[37,38] However, a more accurate ranking using the AD4 scoring function was not observed as results from a previous study shown.[38] These results show that the scoring functions' performance is target-dependent, and the availability of the two scoring functions in the same docking engine simplifies the process of testing and selecting the most effective one for a given target.

AutoDock-Vina has been used in many studies as a reference for benchmarking new or already existing docking programs and scoring functions. Here, we compare our results to published benchmarks done on the full DUD-E dataset using commercial docking programs (GOLD,[39] GLIDE,[40] Surflex,[41] and FlexX[42]) and academic programs (EDOCK,[43] DOCK6,[44] AutoDock4.2,[3] ΔvinaRF20,[45] ID-Score,[46] X-Score,[47] DLIGAND,[48] and DLIGAND2[49]). Vieira and Sousa compared AutoDock Vina 1.1.2 and AutoDock4.2.[50] They obtained AUC values of 68 ± 4.7 and 66.4 ± 10.2, and EF (1%) values of 7.6 ± 4.7 and 8.9 ± 5.6 for Vina 1.1.2 and AutoDock4.2, respectively, in perfect agreement with our results. Because the search algorithm implemented in the AutoDock4.2 program is inferior to that of Vina, we had hypothesized that running the AD4 scoring function in Vina 1.2.0, which uses the superior MC/BFGS search algorithm, would increase the performance of the AD4 scoring function. However, as our results are identical to the study from Vieira and Sousa, which used the AutoDock4.2 program, that does not seem to be the case. Chaput et al. compared multiple commercial docking programs.[51] Their results showed that GOLD, GLIDE, FlexX, and Surflex obtained BEDROC ($\alpha = 20$) values above 0.5 for 37, 24, 15, and 13 target systems, respectively. Our results show that the Vina and AD4 scoring functions achieve a BEDROC ($\alpha = 20$) above 0.5 for only seven and six target systems, respectively. The superior performance observed in the Chaput et al. benchmark may by explained, in part, by the inclusion of crystallographic water molecules that were considered to be important by the original authors of the DUD-E set. Chen et al.[49] compared the Vina scoring function to ΔvinaRF20, ID-Score, X-score, DLIGAND, and DLIGAND2 scoring functions. The logAUC values calculated on the DUD-E dataset show that the Vina scoring function performs similarly to other scoring functions. They reported the following results: 10.14 for DLIGAND2, 9.96 for Vina, 9.00 for ΔvinaRF20, 7.61 for DLIGAND, 7.25 for X-score, and 2.47 for ID-score. Finally, Vina was compared to EDock and DOCK6 in terms of pose prediction accuracy by Zhang et al.[43] All programs perform similarly when the ligand-binding site is known, with an average RMSD of 1.28, 1.38, and 1.41 Å for EDock, Vina, and DOCK6, respectively. In comparison, the average RMSD was about 5.12 and 4.56 Å for the Vina and AD4 scoring functions in this present study, well above the reported values by Zhang et al.

DISCUSSION AND CONCLUSIONS

This work is an effort toward the unification of the different functionalities developed within the AutoDock Suite. AutoDock Vina 1.2.0 allows users to access the powerful iterated local search of Vina with many of the features implemented in the AutoDock4 program, among which the AutoDock4.2 scoring function itself, and the capability of reading and writing grid maps with precalculated target interactions. This latter feature unlocked the possibility of porting a number of existing methods and specialized scoring functions to Vina, such as hydrated docking,[14] the AutoDock4$_{\mathrm{Zn}}$[13] force field, and the AutoDock-Bias docking.[21]

For other methods, such as sampling of macrocycle conformations during docking, that require the definition of ad hoc intramolecular terms, the modifications have been implemented in the source code. This was necessary because AutoDock Vina 1.2.0 does not allow the user to create new atom types or modify pairwise interactions without changes to the source code.

AutoDock Vina 1.2.0 facilitates the design and execution of simple and complex docking simulations. The new version provides Python bindings, enabling easier scripting for virtual screening and other advanced applications. We also implemented batch processing to streamline high-throughput virtual screenings, as well as simultaneous multiple-ligand docking against a single target structure. All new features can be accessed both from the command line interface when using a compiled Vina binary, or from Python.

Having both Vina and AD4 scoring functions available with a common search algorithm allowed a direct comparison of their screening power in a number of targets from the DUD-E set. The scoring functions performed similarly overall across all targets that we considered. However, when considering individual targets, either scoring function can outperform the other, highlighting the need for a better scoring function that performs consistently well for every target.

Due to the added functionality and the array of scoring methods that are now available, we believe that AutoDock Vina 1.2.0 is a useful tool for molecular docking for both novice and expert users.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.1c00203.

Impact of attractive potential on docking scores; description of the preparation and analysis methods for the hydrated docking test and the DUD-E benchmark runs; per target results for the hydrated docking test and the DUD-E benchmark runs (PDF)

Docked poses of the macrocycles from the D3R Grand Challenge 4 BACE-1 dataset (ZIP)

AUTHOR INFORMATION

Corresponding Authors

Jerome Eberhardt − *Department of Integrative Structural and Computational Biology, Scripps Research, La Jolla 92037 California, United States*; Email: jerome@scripps.edu

Stefano Forli − *Department of Integrative Structural and Computational Biology, Scripps Research, La Jolla 92037 California, United States;* orcid.org/0000-0002-5964-7111; Email: forli@scripps.edu

Authors

Diogo Santos-Martins − *Department of Integrative Structural and Computational Biology, Scripps Research, La Jolla 92037 California, United States;* orcid.org/0000-0003-4622-3747

Andreas F. Tillack − *Department of Integrative Structural and Computational Biology, Scripps Research, La Jolla 92037 California, United States;* orcid.org/0000-0002-1832-3030

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.1c00203

Author Contributions

[†]J.E. and D.S.-M. contributed equally to this work.

Notes

The authors declare no competing financial interest.

AutoDock-Vina is released as open source under an Apache license. The source code, the documentation, and updates are available on GitHub: https://github.com/ccsb-scripps/AutoDock-Vina.

■ ACKNOWLEDGMENTS

We thank David Goodsell for the insightful discussions, and Paolo Governa for the help with the beta testing. This work was supported by the NIH grant GM069832. This is manuscript #30066 from Scripps Research. We acknowledge the use of NumPy,[52] Matplotlib,[53] Seaborn,[54] Pandas,[55] and Jupyter Notebook.[56] This manuscript is dedicated to the memory of Prof. Maurizio Botta, whose encouragement to "*always do more*" was instrumental to the development of some of the methods described here.

■ REFERENCES

(1) Trott, O.; Olson, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* **2010**, *31*, 455.

(2) Forli, S.; Huey, R.; Pique, M. E.; Sanner, M. F.; Goodsell, D. S.; Olson, A. J. Computational protein-ligand docking and virtual drug screening with the AutoDock suite. *Nat. Protoc.* **2016**, *11*, 905−919.

(3) Huey, R.; Morris, G. M.; Olson, A. J.; Goodsell, D. S. A semiempirical free energy force field with charge-based desolvation. *J. Comput. Chem.* **2007**, *28*, 1145−1152.

(4) Santos-Martins, D.; Solis-Vasquez, L.; Tillack, A. F.; Sanner, M. F.; Koch, A.; Forli, S. Accelerating autodock4 with gpus and gradient-based local search. *J. Chem. Theory Comput.* **2021**, *17*, 1060−1073.

(5) Ravindranath, P. A.; Forli, S.; Goodsell, D. S.; Olson, A. J.; Sanner, M. F. AutoDockFR: Advances in Protein-Ligand Docking with Explicitly Specified Binding Site Flexibility. *PLoS Comput. Biol.* **2015**, *11*, No. e1004586.

(6) Zhang, Y.; Sanner, M. F. Docking flexible cyclic peptides with AutoDock CrankPep. *J. Chem. Theory Comput.* **2019**, *15*, 5161−5168.

(7) Alhossary, A.; Handoko, S. D.; Mu, Y.; Kwoh, C.-K. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics* **2015**, *31*, 2214−2216.

(8) Koes, D. R.; Baumgartner, M. P.; Camacho, C. J. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. *J. Chem. Inf. Model.* **2013**, *53*, 1893−1904.

(9) Nivedha, A. K.; Thieker, D. F.; Makeneni, S.; Hu, H.; Woods, R. J. Vina-Carb: Improving Glycosidic Angles during Carbohydrate Docking. *J. Chem. Theory Comput.* **2016**, *12*, 892−901.

(10) Koebel, M. R.; Schmadeke, G.; Posner, R. G.; Sirimulla, S. AutoDock VinaXB: implementation of XBSF, new empirical halogen bond scoring function, into AutoDock Vina. *J. Cheminf.* **2016**, *8*, 27.

(11) Quiroga, R.; Villarreal, M. A. Vinardo: A Scoring Function Based on Autodock Vina Improves Scoring, Docking, and Virtual Screening. *PLoS One* **2016**, *11*, No. e0155183.

(12) Forli, S.; Botta, M. Lennard-Jones potential and dummy atom settings to overcome the AUTODOCK limitation in treating flexible ring systems. *J. Chem. Inf. Model.* **2007**, *47*, 1481−1492.

(13) Santos-Martins, D.; Forli, S.; Ramos, M. J.; Olson, A. J. AutoDock4Zn: an improved AutoDock force field for small-molecule docking to zinc metalloproteins. *J. Chem. Inf. Model.* **2014**, *54*, 2371−2379.

(14) Forli, S.; Olson, A. J. A force field with discrete displaceable waters and desolvation entropy for hydrated ligand docking. *J. Med. Chem.* **2012**, *55*, 623−638.

(15) Serrano, P.; Aubol, B. E.; Keshwani, M. M.; Forli, S.; Ma, C.-T.; Dutta, S. K.; Geralt, M.; Wüthrich, K.; Adams, J. A. Directional phosphorylation and nuclear transport of the splicing factor SRSF1 is regulated by an RNA recognition motif. *J. Mol. Biol.* **2016**, *428*, 2430−2445.

(16) Bianco, G.; Forli, S.; Goodsell, D. S.; Olson, A. J. Covalent docking using autodock: Two-point attractor and flexible side chain methods. *Protein Sci.* **2016**, *25*, 295−301.

(17) Nocedal, J.; Wright, S. *Numerical Optimization*; Springer Science & Business Media, 2006.

(18) Houston, D. R.; Walkinshaw, M. D. Consensus docking: improving the reliability of docking in a virtual screening context. *J. Chem. Inf. Model.* **2013**, *53*, 384−390.

(19) Cuzzolin, A.; Sturlese, M.; Malvacio, I.; Ciancetta, A.; Moro, S. DockBench: an integrated informatic platform bridging the gap between the robust validation of docking protocols and virtual screening simulations. *Molecules* **2015**, *20*, 9977−9993.

(20) Weiner, S. J.; Kollman, P. A.; Case, D. A.; Singh, U. C.; Ghio, C.; Alagona, G.; Profeta, S.; Weiner, P. A new force field for molecular mechanical simulation of nucleic acids and proteins. *J. Am. Chem. Soc.* **1984**, *106*, 765−784.

(21) Arcon, J. P.; Modenutti, C. P.; Avendaño, D.; Lopez, E. D.; Defelipe, L. A.; Ambrosio, F. A.; Turjanski, A. G.; Forli, S.; Marti, M. A. AutoDock Bias: improving binding mode prediction and virtual screening using known protein-ligand interactions. *Bioinformatics* **2019**, *35*, 3836−3838.

(22) Lazaridis, T. Inhomogeneous Fluid Approach to Solvation Thermodynamics. 1. Theory. *J. Phys. Chem. B* **1998**, *102*, 3531−3541.

(23) Lazaridis, T. Inhomogeneous fluid approach to solvation thermodynamics. 2. Applications to simple fluids. *J. Phys. Chem. B* **1998**, *102*, 3542−3550.

(24) Nguyen, C. N.; Young, T. K.; Gilson, M. K. Grid inhomogeneous solvation theory: Hydration structure and thermodynamics of the miniature receptor cucurbit[7]uril. *J. Chem. Phys.* **2012**, *137*, 044101.

(25) Uehara, S.; Tanaka, S. AutoDock-GIST: Incorporating Thermodynamics of Active-Site Water into Scoring Function for Accurate Protein-Ligand Docking. *Molecules* **2016**, *21*, 1604.

(26) Irwin, J. J.; Tang, K. G.; Young, J.; Dandarchuluun, C.; Wong, B. R.; Khurelbaatar, M.; Moroz, Y. S.; Mayfield, J.; Sayle, R. A. ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery. *J. Chem. Inf. Model.* **2020**, *60*, 6065.

(27) Jiang, Y.; Zhuang, C.; Chen, L.; Lu, J.; Dong, G.; Miao, Z.; Zhang, W.; Li, J.; Sheng, C. Structural biology-inspired discovery of novel KRAS−PDEδ inhibitors. *J. Med. Chem.* **2017**, *60*, 9400−9406.

(28) Gathiaka, S.; Liu, S.; Chiu, M.; Yang, H.; Stuckey, J. A.; Kang, Y. N.; Delproposto, J.; Kubish, G.; Dunbar, J. B.; Carlson, H. A.; Burley, S. K.; Walters, W. P.; Amaro, R. E.; Feher, V. A.; Gilson, M. K. D3R grand challenge 2015: evaluation of protein−ligand pose and affinity predictions. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 651−668.

(29) Parks, C. D.; Gaieb, Z.; Chiu, M.; Yang, H.; Shao, C.; Walters, W. P.; Jansen, J. M.; McGaughey, G.; Lewis, R. A.; Bembenek, S. D.; Ameriks, M. K.; Mirzadegan, T.; Burley, S. K.; Amaro, R. E.; Gilson,

M. K. D3R grand challenge 4: blind prediction of protein−ligand poses, affinity rankings, and relative binding free energies. *J. Comput.-Aided Mol. Des.* **2020**, *34*, 99−119.

(30) Santos-Martins, D.; Eberhardt, J.; Bianco, G.; Solis-Vasquez, L.; Ambrosio, F. A.; Koch, A.; Forli, S. D3R Grand Challenge 4: prospective pose prediction of BACE1 ligands with AutoDock-GPU. *J. Comput.-Aided Mol. Des.* **2019**, *33*, 1071−1081.

(31) El Khoury, L.; Santos-Martins, D.; Sasmal, S.; Eberhardt, J.; Bianco, G.; Ambrosio, F. A.; Solis-Vasquez, L.; Koch, A.; Forli, S.; Mobley, D. L. Comparison of affinity ranking using AutoDock-GPU and MM-GBSA scores for BACE-1 inhibitors in the D3R Grand Challenge 4. *J. Comput.-Aided Mol. Des.* **2019**, *33*, 1011−1020.

(32) Lam, P. C.-H.; Abagyan, R.; Totrov, M. Macrocycle modeling in ICM: benchmarking and evaluation in D3R Grand Challenge 4. *J. Comput.-Aided Mol. Des.* **2019**, *33*, 1057−1069.

(33) van Rossum, G. Python Programming Language. *USENIX Annual Technical Conference*, 2007; p 36.

(34) Beazley, D. M. *SWIG: An Easy to Use Tool for Integrating Scripting Languages with C and C++*; Tcl/Tk Workshop, 1996; p 74.

(35) Brandl, G. Sphinx: Python documentation generator. https://www.sphinx-doc.org/ (accessed Feb 19, 2021).

(36) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. *J. Med. Chem.* **2012**, *55*, 6582−6594.

(37) Gaillard, T. Evaluation of AutoDock and AutoDock Vina on the CASF-2013 benchmark. *J. Chem. Inf. Model.* **2018**, *58*, 1697−1706.

(38) Nguyen, N. T.; Nguyen, T. H.; Pham, T. N. H.; Huy, N. T.; Bay, M. V.; Pham, M. Q.; Nam, P. C.; Vu, V. V.; Ngo, S. T. Autodock vina adopts more accurate binding poses but autodock4 forms better binding affinity. *J. Chem. Inf. Model.* **2019**, *60*, 204−211.

(39) 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.

(40) 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.; et al. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J. Med. Chem.* **2004**, *47*, 1739−1749.

(41) Jain, A. N. Surflex: fully automatic flexible molecular docking using a molecular similarity-based search engine. *J. Med. Chem.* **2003**, *46*, 499−511.

(42) 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.

(43) Zhang, W.; Bell, E. W.; Yin, M.; Zhang, Y. EDock: blind protein−ligand docking by replica-exchange monte carlo simulation. *J. Cheminf.* **2020**, *12*, 37.

(44) Allen, W. J.; Balius, T. E.; Mukherjee, S.; Brozell, S. R.; Moustakas, D. T.; Lang, P. T.; Case, D. A.; Kuntz, I. D.; Rizzo, R. C. DOCK 6: Impact of new features and current docking performance. *J. Comput. Chem.* **2015**, *36*, 1132−1156.

(45) Wang, C.; Zhang, Y. Improving scoring-docking-screening powers of protein−ligand scoring functions using random forest. *J. Comput. Chem.* **2017**, *38*, 169−177.

(46) Li, G.-B.; Yang, L.-L.; Wang, W.-J.; Li, L.-L.; Yang, S.-Y. ID-Score: a new empirical scoring function based on a comprehensive set of descriptors related to protein−ligand interactions. *J. Chem. Inf. Model.* **2013**, *53*, 592−600.

(47) Wang, R.; Lai, L.; Wang, S. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *J. Comput.-Aided Mol. Des.* **2002**, *16*, 11−26.

(48) Zhang, C.; Liu, S.; Zhu, Q.; Zhou, Y. A knowledge-based energy function for protein- ligand, protein- protein, and protein-DNA complexes. *J. Med. Chem.* **2005**, *48*, 2325−2335.

(49) Chen, P.; Ke, Y.; Lu, Y.; Du, Y.; Li, J.; Yan, H.; Zhao, H.; Zhou, Y.; Yang, Y. DLIGAND2: an improved knowledge-based energy function for protein−ligand interactions using the distance-scaled, finite, ideal-gas reference state. *J. Cheminf.* **2019**, *11*, 52.

(50) Vieira, T. F.; Sousa, S. F. Comparing AutoDock and Vina in ligand/decoy discrimination for virtual screening. *Appl. Sci.* **2019**, *9*, 4538.

(51) Chaput, L.; Martinez-Sanz, J.; Quiniou, E.; Rigolet, P.; Saettel, N.; Mouawad, L. Benchmark of four popular virtual screening programs: construction of the active/decoy dataset remains a major determinant of measured performance. *J. Cheminf.* **2016**, *8*, 56.

(52) Harris, C. R.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357−362.

(53) Hunter, J. D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90−95.

(54) Waskom, M. et al. *mwaskom/seaborn*: v0.8.1 (September 2017), 2017; https://doi.org/10.5281/zenodo.883859.

(55) McKinney, W. Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 2010; pp 56−61.

(56) Kluyver, T.; Ragan-Kelley, B.; Pérez, F.; Granger, B.; Bussonnier, M.; Frederic, J.; Kelley, K.; Hamrick, J.; Grout, J.; Corlay, S.; Ivanov, P.; Avila, D.; Abdalla, S.; Willing, C. Jupyter Notebooks—A Publishing Format for Reproducible Computational Workflows. *Positioning and Power*; Academic Publishing: Players, Agents and Agendas, 2016; pp 87−90.