

APPLICATION NOTE

Open Access



AMDock: a versatile graphical tool for assisting molecular docking with Autodock Vina and Autodock4

Mario S. Valdés-Tresanco^{1*}, Mario E. Valdés-Tresanco^{2,3}, Pedro A. Valiente^{2,4} and Ernesto Moreno^{1*} 

Abstract

AMDock (Assisted Molecular Docking) is a user-friendly graphical tool to assist in the docking of protein-ligand complexes using Autodock Vina and AutoDock4, including the option of using the AutoDock4Zn force field for metalloproteins. AMDock integrates several external programs (Open Babel, PDB2PQR, AutoLigand, ADT scripts) to accurately prepare the input structure files and to optimally define the search space, offering several alternatives and different degrees of user supervision. For visualization of molecular structures, AMDock uses PyMOL, starting it automatically with several predefined visualization schemes to aid in setting up the box defining the search space and to visualize and analyze the docking results. One particularly useful feature implemented in AMDock is the off-target docking procedure that allows to conduct ligand selectivity studies easily. In summary, AMDock's functional versatility makes it a very useful tool to conduct different docking studies, especially for beginners. The program is available, either for Windows or Linux, at <https://github.com/Valdes-Tresanco-MS>.

Reviewers: This article was reviewed by Alexander Krahl and Thomas Gaillard.

Keywords: AMDock, AutoDock4, AutoDock Vina, AutoDock4Zn, Docking, Graphical user interface

Background

Molecular docking has become a powerful tool for lead discovery and optimization. A large number of docking programs have been developed during the last three decades, based on different search algorithms and scoring functions. Aiming to make these docking programs more user-friendly, especially to beginners, different graphical user interfaces (GUIs) have been developed to assist in the preparation of molecular systems, the execution of the calculations and/or the analysis of the results. Examples of available GUIs (developed mostly for AutoDock [1] and/or Autodock Vina [2]) are AutoDock Tools (ADT), integrated into the PMV graphical package [1], BDT [3],

DOVIS [4, 5], VSDocker [6], AUDocker LE [7], WinDock [8], DockoMatic [9], PyMOL AutoDock plugin (PyMOL/AutoDock) [10], PyRx [11], MOLA [12], DockingApp [13] and JADOPPT [14].

We present here a new multi-platform tool, AMDock (Assisted Molecular Docking), whose main advantage over its predecessors is the integration of several valuable external tools within a simple and intuitive graphical interface that guides the users along well-established docking protocols - using either AutoDock4 or AutoDock Vina - from system preparation to analysis of results.

Functionalities and workflow

AMDock integrates functionalities from Autodock Vina and Autodock4, ADT scripts, AutoLigand [15], Open Babel [16], PDB2PQR [17] and PyMOL [18]. For proteins containing a zinc ion in the active site, AMDock

* Correspondence: mariosergiovaldes145@gmail.com; emoreno@udem.edu.co

¹Faculty of Basic Sciences, University of Medellín, Medellín, Colombia
Full list of author information is available at the end of the article



© The Author(s). 2020 **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

has the option of using the specially tailored AutoDock4Zn [19] parameters. AMDock is coded in Python 2.7 and is available for Windows and Linux. On Windows, it is packaged together with all the integrated tools, hence no additional software installation is required. On Linux, only Open Babel and PyMOL should be installed (both tools are included in most popular Linux repositories).

The AMDock main window has five tabs: 1) Home, 2) Docking Options, 3) Results Analysis, 4) Configuration, and 5) Info. A summary of AMDock's functionalities and workflow is presented below (Fig. 1) and discussed afterwards in more detail.

In the "Home" tab, the user can select the docking engine: Autodock Vina or Autodock4, with the additional option of using the Autodock4Zn parameters. Then the user is automatically directed to the "Docking Options" tab, which contains four panels that guide a sequential preparation of a docking simulation.

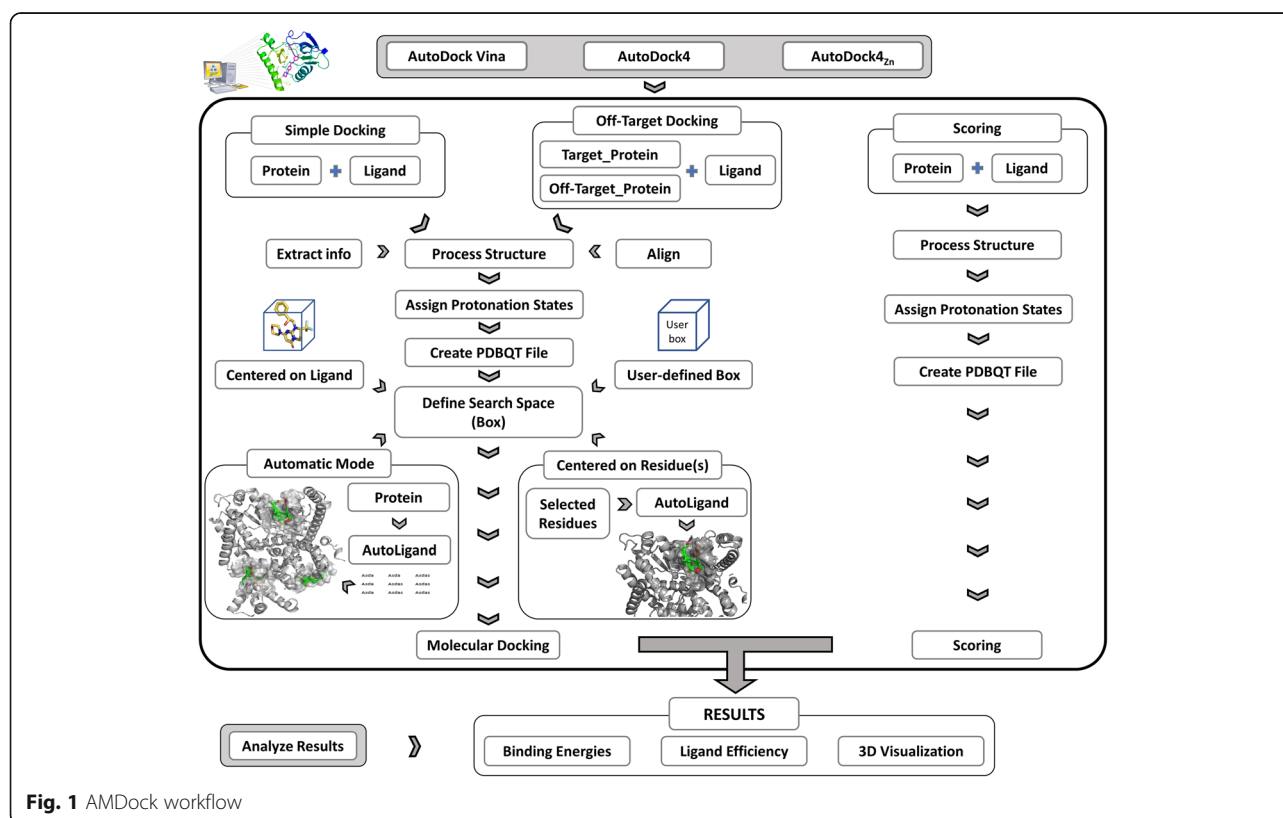
Input files for AMDock

Minimally, the Cartesian coordinates of the ligand and receptor molecules are needed, which can be provided in several common structure formats, e.g. PDB or PDBQT for the protein, and PDB, PDBQT or Mol2 for the ligand. If the protein coordinates come

together with a bound ligand, the coordinates of the later are stored and can be used afterward to define the search space.

The program works by following three main steps:

- 1- *Preparing the docking input files*: First, the user may set a pH value for the protonation of both the ligand (optional, default value 7.4), using Open Babel and the protein (default value: 7.4), using PDB2PQR. Two different docking options are available: a) "simple docking", for predicting the binding mode of a single protein-ligand complex, and b) "off-target docking", for predicting the binding poses of a ligand with two different receptors, i.e. the target and the off-target. Finally, the "Scoring" option included in this tab allows to score an already existing protein-ligand complex, using the Autodock Vina, Autodock4 or Autodock4Zn functions. Once the docking or scoring protocol has been selected, the input files are prepared using ADT scripts.
- 2- *Defining the search space*: Four different approaches can be used to define a box center and dimensions: a) "Automatic" - the program uses AutoLigand to predict possible binding sites and then a box with optimal dimensions is centered on each AutoLigand object,¹ at each predicted binding site. b) "Center



on Residue(s)” - AutoLigand is used to generate an object with a volume in correspondence with the ligand size, using as reference the geometric center of the selected residues. Then, a box with optimal dimensions is centered on the generated object. c) “Center on Hetero” - a box is placed on the geometric center of an existing ligand (if the receptor was given in complex with a ligand), and d) “Box” - the box center and dimensions are defined by the user. The box generated with any of these methods can be visualized in PyMOL and easily modified at the user’s convenience using the new AMDock plugin (adapted from [10]) embedded in the PyMOL menu window.

- 3- *Running the docking simulations and analyzing the results:* After running the molecular docking calculations (started by clicking the “Run” button), the user will be taken automatically to the “Results Analysis” tab, where the Affinity, Estimated Ki values and Ligand Efficiencies are listed for the different binding poses.

The estimated Ki is a very useful value as it is more related to usually measured experimental parameters, as compared to the affinity. Ligand efficiency (LE), on the other hand, is an important informative parameter when selecting a lead compound [20]. Here, LE is calculated by using the following equation:

$$LE = \frac{-\Delta G}{HA}, \quad (1)$$

where ΔG is the free energy of binding or the calculated score value and HA is the number of heavy (non-hydrogen) atoms of the ligand. Compounds with $LE > 0.3$ are highlighted as potential lead compounds [21].

The “Show in PyMOL” button starts PyMOL with a customized visualization of the complex between the receptor and the selected pose (the lowest-energy ligand pose is chosen by default). The resulting data throughout the process is stored in a file (*.amdock), which can be used to examine the results at any time later.

Different docking parameters can be set in the “Configuration” tab, while the “Info” tab, gives access to handy documentation, including a user manual and references.

Visualization

AMDock relies on PyMOL for visualization at two different stages: 1) setting up the grid box location and

dimensions (the search space), and 2) analysis of the docking results. PyMOL is a versatile and user-friendly molecular analysis program which, besides, allows to create high-quality images for publication. We have coded in AMDock several predetermined PyMOL representations for the two stages, selecting the visual design and information that we considered optimal in each case. These predefined representations can be modified by the user within PyMOL.

Search space

The predetermined representations (in descending order of complexity, according to the number of elements in the visualization contents) are the following: 1) *Box* - a simple representation where the protein under study appears as cartoon, together with the box with the specifications defined by the user (Fig. 2a); 2) *Centered on Hetero* - includes the receptor protein (cartoon) and the box with an optimal size centered on the selected previous ligand (sticks) (Fig. 2b); 3) *Centered on Residue(s)* - a representation that allows the user to identify the residues that were selected to define the search space. The protein is represented as cartoon, the selected residues as sticks and the AutoLigand object as points. The calculated box is also showed, so that the user can easily check and adjust (if necessary) its position and dimensions (Fig. 2c). 4) *Automatic* - Here we intended to create a simplified representation to show all the binding sites predicted by AutoLigand. The protein is in cartoon, each AutoLigand object is represented in sticks, surrounded by a surface constructed on its neighboring residues. Since docking simulations are to be performed for each site predicted by AutoLigand, a box is generated for each site, but showed only for a user-selected site (Fig. 2d). As mentioned above, in any of these variants the box center and size can be easily modified using the AMDock plugin implemented in PyMOL.

Results analysis

The protein is represented in cartoon. Each ligand pose is drawn in sticks and its polar contacts with the protein are shown as dashed lines. A similar visualization is also possible for both proteins if the “Off-Target Docking” procedure was chosen (Fig. 3c). This allows a simultaneous comparison of ligand poses for both the target and off-target proteins.

Case study: SAR405 binding selectivity - PI3K vs. Vps34

Phosphatidylinositol 3-kinase (PI3K) is an enzyme involved in growth, proliferation, motility, survival, and intracellular trafficking [22]. PI3K is also a promising cancer target, with several of its inhibitors being already in the clinical stage. A few of these inhibitors are currently in phase III clinical trials and one of

¹AutoLigand Object is equivalent to the term “envelope” used by the authors of AutoLigand and consists of a representation of a contiguous region in space, filled with points that represent potential atomic centers for ligand atoms [15]

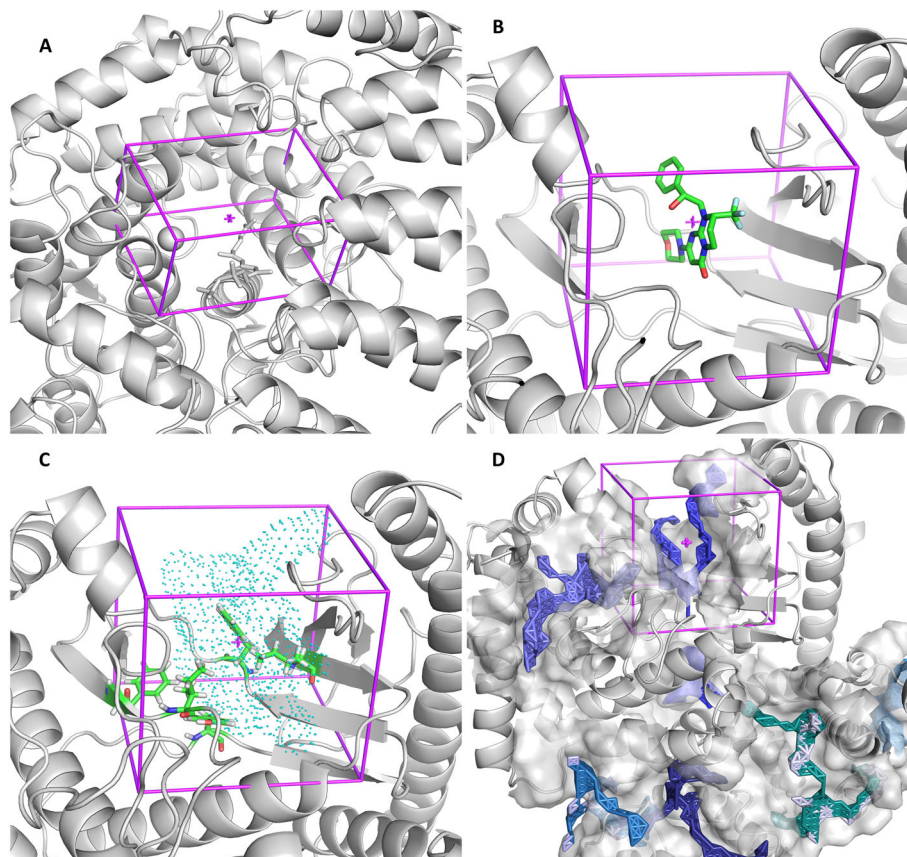


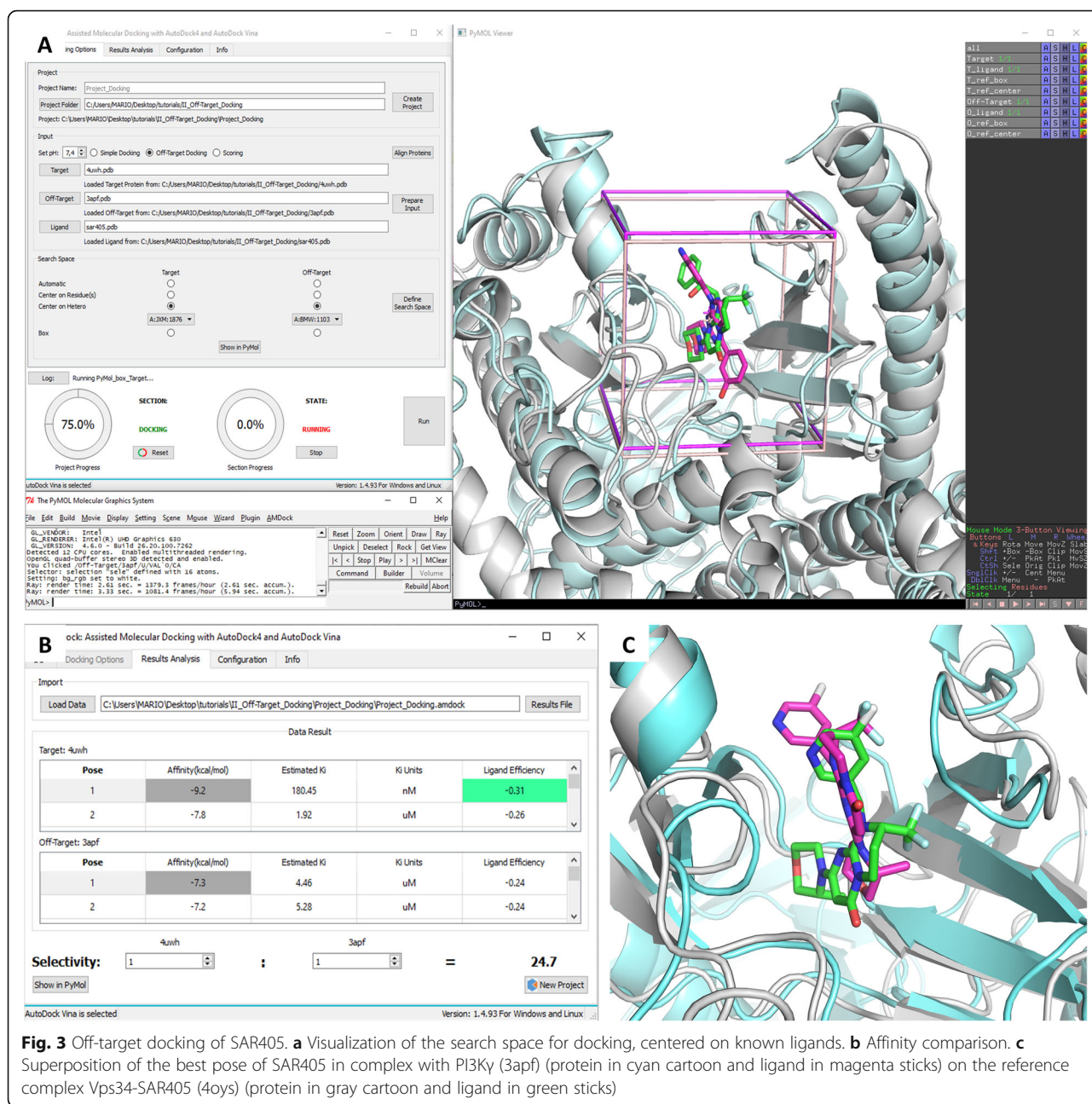
Fig. 2 Binding site visualization with PyMOL. **a** User-defined box. This is an example used in tutorials with AutoDock4Zn and farnesyltransferase (hFTase). **b** Centered on Hetero, **(c)** Centered on Residue(s) and **(d)** Automatic mode. Representations B, C and D correspond to Vps34 (PDB: 4uwh)

them, alpelisib, recently (May 2019) received the FDA approval for use in the treatment of metastatic breast cancer.

PI3K has several isoforms that are grouped in 3 different classes. Class I includes four different isoforms (α , β , γ and δ), while class III is composed of only one protein, called Vps34 [22]. Because of their sequence and structural similarities, some inhibitors may bind different isoforms, whereas several other inhibitors were designed to be isoform specific. Our research group is currently focused on the identification of PI3K orthologs found in different pathogenic microorganisms, which express only the ancestral Vps34 isoform. For this purpose, AMDock represents a valuable tool, particularly its “Off-Target Docking” option. Here we demonstrate its use with an exercise that resembles our own research work.

Sar405 is a highly specific inhibitor of Vps34 ($IC_{50} = 1.2$ nM), while its IC_{50} for other isoforms is $> 10^4$ nM [23]. A crystal structure of SAR405 in complex with human Vps34 is available in the Protein Data Bank [24]

(PDB code: 4oys). Here we use the human Vps34 as the “Target” receptor, while the PI3K gamma isoform (PDB: 3apf) is used as the “Off-Target” receptor. Both structures contain a bound ligand in the active site, which is convenient for generating the grid box. In the first step, we select the docking program (Autodock Vina) and thereafter a project folder is created in the computer hard disk. After loading both protein structures, we take advantage of their sequence similarity to use the available option of aligning and superimposing their structures using PyMOL, which makes possible defining a common search space and simplifying the subsequent analysis of the docking results. Next, the input files are prepared automatically, which includes protonation of titratable residues, merging of non-polar hydrogens and ion/water removal. The center of the box is defined based on the geometric center of the bound ligands (Fig. 3a), while the size of the box is defined based on the radius of gyration of the ligand to be docked [25], i.e. the SAR405 inhibitor in this case. The initial ligand conformation (its torsion angles) was randomized using ADT.



Once the process is completed, the results show that SAR405 is more selective for Vps34 (-9.2 kcal/mol) than for Pi3Kγ (-7.3 kcal/mol) as expected (Fig. 3b). The predicted binding pose for SAR405 in Vps34 is close to the crystal geometry (rmsd = 1.9 Å for all ligand atoms, rmsd = 0.5 Å for the ring core). Also, the predicted Ki value for this complex is in the nanomolar range, which agrees with the experimental value. On the other hand, a much higher Ki value is predicted for the Pi3Kγ-SAR405 complex, and the predicted binding pose differs significantly from the crystallographic structure (rmsd = 4.7), as shown in Fig. 3c, which may explain the poor affinity value predicted by

AutoDock Vina. This study case has been incorporated as a tutorial in the user manual, which is included in the AMDock installation folder, and the wiki on Github (<https://github.com/Valdes-Tresanco-MS/AMDock-win/wiki/4.3-Off-target-docking>).

Discussion

AMDock provides a novel, easy-to-use and versatile interface to work with two molecular docking engines, Autodock4 and Autodock Vina, having different functionalities and characteristics. AMDock should be very useful to researchers with little experience in working

with docking programs since no previous knowledge of the particular functioning of these programs is needed. Three different workflows (simple docking, off-target docking and scoring) are included in the AMDock environment. We find the off-target docking procedure particularly helpful for conducting ligand selectivity studies - a critical step in the drug design process.

Preparing the input files in a proper and consistent way, as well as correctly defining the search space, are critical issues when performing molecular docking studies. Several external programs/scripts are integrated into AMDock to allow preparing the input files with minimal effort while keeping control of the process. AMDock uses OpenBabel and PDB2PQR for ligand and receptor protonation, respectively, while the other GUIs mentioned in the Introduction use ADT for both receptor and ligand protonation (with the exception of DockingApp, which uses also OpenBabel for ligand protonation).

To define the search space, AMDock offers several options to set the position of the grid box in different scenarios, while the input ligand is used by default to determine the box optimal dimensions, which decreases the computational cost while optimizing the docking process [25]. In this regard, only ADT and the PyMOL/

AutoDock plugin offer some limited options other than a user-defined search space, but in any case the box size must be defined by the user. In some of these GUIs, as in DockingApp, the search space covers the entire receptor, which leads to additional computational costs and possibly compromises the accuracy of the simulations. With other GUIs, the user must use an external applications such as ADT to define the box parameters.

The “Centered on Residue(s)” option is preferable when the binding site residues are known. With this option, an object placed at the geometric center of the selected residues is generated with AutoLigand on the protein surface. This procedure optimizes both the location and size of the search space. If the box was centered instead on the geometric center of the selected residues, a significant part of it will likely be embedded in the protein, demanding a larger size to cover the needed sampling space (Fig. 4). The “Centered on Hetero” alternative is useful for redocking studies on complexes with crystallographic structures or when studying ligands with similar binding modes (Fig. 2b). The “Automatic” option, on the other hand, is desirable when no information regarding the binding site is available. In this case, an independent docking run is performed for every binding site predicted by AutoLigand (Fig. 2d).

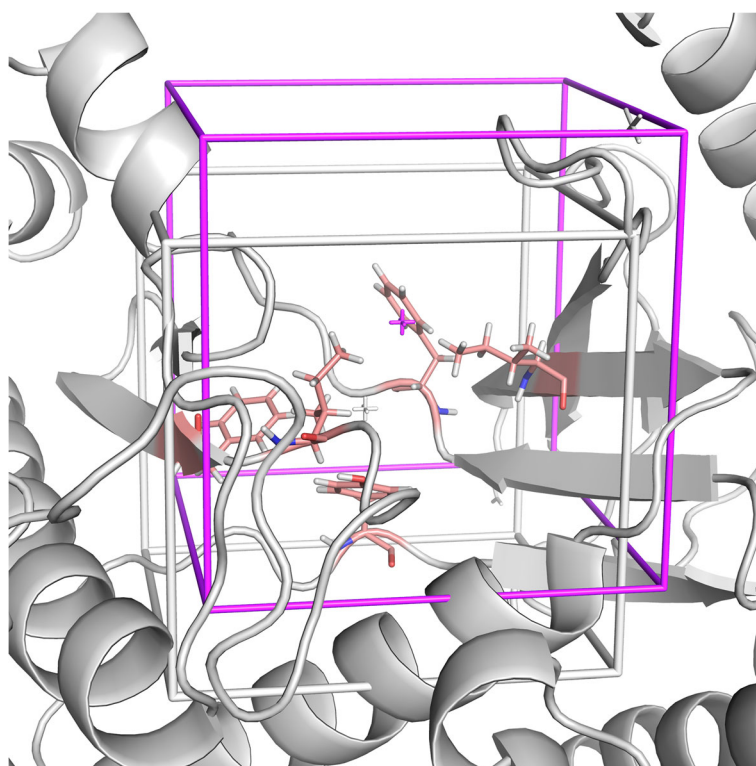


Fig. 4 Comparison between a box (white) located at the geometric center of the selected residues (A:ILE:634, A:TYR:670, A:PHE:684, A:PHE:758, A:ILE:760; in salmon) and an a box (magenta) centered on an object generated by AutoLigand from the geometric center of the selected residues. In the later case, the box defines a more optimal ligand sampling space

Table 1 Comparison of AMDock and AutoDock Tools features

Features		AMDock	AutoDock Tools
File formats	Receptor:	pdb, pdbqt	pdb, mol2, pdbq, pdbqs, pdbqt, pqr, cif
	Ligand:	pdb, pdbqt, mol2	pdb, pdbq, mol2
Protonation	Receptor:		
	PDB2PQR	Uses the last version ¹	Uses PDB2PQR v1.2.1
	pH value adjustment	Yes	No (default 7.0)
	Experimental protonation state	Only if the user enters a protonated structure ²	Only histidines or when the user enters a protonated structure
	Ligand:		
	Open Babel	Uses the last version ¹	Basic implementation of Open Babel v1.6
	pH value adjustment	Yes	No (default value: 7.0)
Structure manipulation	Flexible Side Chains	Not implemented ²	Yes
	Flexible Ligand	Active torsions not implemented ²	Yes
	Center Automatic ³	Possible binding sites are determined with AutoLigand. Docking is performed for each site.	The user must select a predicted site and prepare the search space. This should be repeated for each site to be tested.
	Center on Residues	Centers the box on an AutoLigand object, calculated for a group of selected residues	Only on a selected atom ⁴
	Center on Hetero	Centers the search space in the geometric center of a heteroatom set found in the defined receptor pdb.	On selected heteroatoms or on a ligand ⁵
	Custom Box	Box coordinates defined by the user.	Box coordinates defined by the user.
	Box Size	Determined from the radius of gyration of ligand, or set by the user ⁶	Defined by the user.
Docking programs	AutoDock4	Yes	Yes
	AutoDock Vina	Yes	Yes
Docking type	Simple	Yes	Yes
	Virtual Screening	No ²	No
	Off-target Docking	Yes	No
	Covalent Docking	No ²	Yes
	Using Autodock4ZN	Yes	Command line
	Hydrated docking	No ²	Command line
Analysis of Results	Simple docking	Yes	Yes
	Virtual Screening	No ²	Yes
	Off-target docking	Yes	No
	Covalent docking	No ²	Yes
	Autodock4 ZN docking	Yes	Yes
	Hydrated docking	No ²	Yes
Graphical Visualization	Engine	PyMOL	Python Molecular Viewer
	Capacity	All the options included in PyMOL	Protein-ligand interactions and cluster manager for AutoDock4 results
	Publication-quality images	Easy high-resolution and custom image generation	Easy low-resolution image generation. Difficult high-resolution image generation
	Maintenance	Active development	Inactive
Programming	Python base	Python 2.7.15 ⁷	Python 2.6 (Inactive development)

Table 1 Comparison of AMDock and AutoDock Tools features (Continued)

Features	AMDock	AutoDock Tools
Easy to use ⁸	Docking preparation	1
	Analysis of Results	2
	GUI simplicity	1
	Process Log	1
	Installation	2
	Platform	Linux and Windows
		Linux, Windows and Mac

¹ Last version with Python 2.x support

² Will be available in the next release

³ A common alternative is to do the so-called blind docking, in which the search space is defined to cover the entire receptor. This involves an increasing in the sampling number so as not to compromise the accuracy of the docking, which leads to an increase in computational cost. Additionally, it can introduce false positives by sampling sites with a different nature than the binding site. Results are usually questionable due to the non-convergence of the scoring functions

⁴ We describe the advantages of the method used in AMDock concerning this selection (Fig. 4)

⁵ It is possible to select an atom only if the heteroatoms of the receptor have not been removed. After that, these atoms must be removed and the receptor should be redefined. Another possibility is entering a set of heteroatoms and directly select the option "center on ligand". Both options have limitations and need a deeper understanding of the ADT program

⁶ We describe the advantages of the method used in AMDock

⁷ Next version in python 3.x under development (<https://github.com/Valdes-Tresanco-MS/AMDock-win-py3>)

⁸ Our own evaluation using a 1–5 scale, where 1 is very easy and 5 is very difficult

This way, the information from the AutoLigand ranking method is combined with that of the docking engine, without making an arbitrary selection of one of the predicted sites. This process is done automatically and the results for each of the predicted binding sites can be visualized in PyMOL. Overall, the definition and visualization of the box involves a minimal effort and can always be modified, thus representing an advantage not only for the novice user but also for experts.

It is worth noting that we standardized the box size to be in Angstroms to avoid commonly occurring errors, as reported in different forums and mailing-lists. These errors arise from the different ways in which the box dimensions are defined in AutoDock (number of points + grid spacing) and Autodock Vina (in angstroms), and may cause the search space to be very small or too large, leading ultimately to inconsistencies in the obtained docking results.

The integration of AMDock with PyMOL represents a significant advantage. Indeed, PyMOL is a widely used molecular viewer with great community support and active development. Within PyMOL, docking results can be analyzed with multiple tools, in particular with the powerful Protein-ligand Interaction Profiler [26]. Other applications such as ADT, PyRx or DockingApp have their own graphical viewers. PyRx and DockingApp offer simple solutions with limited analytical capabilities, while ADT allows only for simple analysis of protein-ligand interactions.

Furthermore, with AMDock it is possible to launch docking simulations for metalloproteins using the AutoDock's Zn force field, which is available in ADT only via command line. Its off-target docking option, very useful for drug repurposing studies, is available only in Dockomatic and PyRx (in the later, only in the payment version).

Most of the docking GUIs are focused on virtual screening. Currently, AMDock does not have support for virtual screening, however, we are currently working on its implementation, to make it available in the next program version.

Finally, and since ADT is probably the most widely used docking GUI, we provide a more detailed comparison between AMDock and ADT (Table 1).

Conclusions

AMDock is a user-friendly GUI that works in a highly intuitive and interactive manner, allowing to perform molecular docking studies with Autodock4 and AutoDock Vina with a minimal setup effort. These characteristics make AMDock an attractive tool also for teaching purposes. AMDock gathers features and procedures that are not present in other similar programs. It includes recent developments in AutoDock, such as the Autodock4Zn parameterization. For our group, AMDock has been very useful for estimating the selectivity profile of different PI3K inhibitors over orthologous proteins in several microorganisms. Further developments (hydrated ligand, covalent docking and virtual screening) will be included as docking options in future versions.

Reviewers' comments

Reviewer 1, Alexander Krahl

Summary: Valdés-Tresanco et al. describe in their manuscript entitled "AMDock: A versatile graphical tool for assisting molecular docking with Autodock Vina and Autodock4" the implementation of several molecular modelling tools in a graphical user interface, which allows to setup and perform docking simulations with Autodock4 or Autodock Vina. I think the tool is interesting and may allow individuals who just begin with

molecular docking to quickly go through the whole process. However, I would like to ask for clarifications.

Response: We thank Dr. Krah for his positive comment. Below we respond point-by-point his questions and critical comments.

Mayor recommendations

1) *Could the authors describe in more detail, how the pKa for titratable groups of the ligand is calculated (e.g. carboxylate or amine groups)?*

Response: Ligand hydrogen atoms are added either with Open Babel (default) or ADT. Open Babel uses a set of fragments with known pKa to estimate the protonation state of the ligand (Open Babel Documentation). ADT adds hydrogens depending on the atom type, its valence and minor chemical group considerations based on a reimplementation of the PyBabel v1.6 module of Open Babel (PyBabel documentation in ADT). The user can decide whether to use AMDock's internal tools or entering a protonated ligand. In the later case hydrogens will not be added.

This point was also a concern of Reviewer 2. We have added a statement on the optional protonation choice in the subsection "Functionalities and workflow". It reads: "... (optional, default value 7.4), using Open Babel ..."

2) *Can the user set experimentally know protonation states for protein residues? 3) Can protein residues be set flexible, as this possibility is incorporated in ADT/Autodock Vina.*

Response: At the moment these functionalities are not available. However, they will be incorporated in the next AMDock version (development version in <https://github.com/Valdes-Tresanco-MS/AMDOck-win-py3>)

4) *The authors report as an example of an off-target binding prediction, resulting in a higher score for the target than the off-target. Could the authors test two more examples, if targets with required structural and biophysical information can be found:*

a) *different inhibitors binding in the same range to the same protein bound to the same site?*

Response: Three docking exercises included either in the documentation or the manuscript involve three inhibitors having similar reported IC₅₀ values and the same protein crystallographic structure (PDB code 4UWH, a Vps34-inhibitor complex)

- The re-docking exercise (<https://github.com/Valdes-Tresanco-MS/AMDOck-win/wiki/4.2.2.1-re-Docking-experiment>) uses the crystallographic ligand from 4UWH: compound No. 4 (IC₅₀ = 3nM) described by Pasquier *et al.*, 2014.

- In the "similar docking ligand" exercise (<https://github.com/Valdes-Tresanco-MS/AMDOck-win/wiki/4.2.2.2-Docking-a-similar-ligand>), we use inhibitor No. 31 (IC₅₀ = 2 nM, <https://www.rcsb.org/ligand/7A5>) also described by Pasquier *et al.*, 2014.

- Finally, inhibitor SAR405 (IC₅₀ = 1.2nM, <https://www.rcsb.org/ligand/1TT>) is described as a case study in the manuscript (<https://github.com/Valdes-Tresanco-MS/AMDOck-win/wiki/4.3-Off-target-docking>).

All inhibitors have similar IC₅₀ values and bind to Vps34 in the same binding site. In all cases, AutoDock Vina was able to reproduce the crystallographic complex, with affinities of -9.2, -8.7 and -9.2 kcal/mol, respectively, and estimated Ki values in the nanomolar range.

It is important to mention that our case studies are only representative examples of the methodologies implemented in AMDock. As known from the literature, the accuracy of the prediction depends on several structural factors, *i.e.*, the protonation state, the quality of the receptor structure, the particular side chain orientations in the binding site (related to the induced fit effect), among others. AMDock provides a platform for preparing docking files and optimize the search space. However, it does not influence the predictability of the program used to perform the docking. As we remark in the documentation, the interpretation of the results must be based on empirical/experimental evidence, which must be carefully studied by the user.

b) *different inhibitors binding in the same range to the same target bound to a different (potentially allosteric) site?*

Response: A new tutorial has been included in the AMDock wiki (<https://github.com/Valdes-Tresanco-MS/AMDOck-win/wiki/4.5.2-Docking-to-allosteric-binding-sites>) for such a system.

5) *How does the program perform in comparison with Autodock Vina and other tools, which incorporate Autodock Vina?*

Response: To date, most tools that incorporate Autodock Vina are discontinued. We intend to provide a tool where routinary docking experiments (simple docking, off-target docking, redocking, etc.) can be performed with minimal effort. In terms of speed, AMDock does not provide a better performance since it uses the standard Autodock Vina engine. However, the integration of several external tools to prepare the docking files and define/optimize the search space saves time while avoiding commonly-made errors.

Reviewer 2, Thomas Gaillard

Summary: The manuscript of Valdés-Tresanco, Valdés-Tresanco, Valiente, and Moreno presents AMDock, a graphical tool aimed at facilitating molecular docking with Autodock Vina and Autodock4. AMDock integrates external programs (Open Babel, PDB2PQR, AutoLigand, ADT scripts). Molecular visualization is performed with PyMOL. The program is available for Windows and

Linux and is distributed on Github (<https://github.com/Valdes-Tresanco-MS>). Overall, the work presented is valid and well written. The significance and originality are however limited. There are indeed already many existing GUIs facilitating docking calculations. In particular, the AutoDock Tools (ADT) interface to AutoDock Vina and Autodock4 is well conceived and offers more functionalities than AMDock. The authors do not provide a detailed comparison of AMDock and other GUIs. To my opinion, the manuscript is not convincing at demonstrating how AMDock is advantageous, at least on some points, compared to existing tools. At last, some methodological points need to be clarified in the case study.

Response: We appreciate Dr. Gaillard's comments and agree with him in that other GUIs (ADT in particular) offer several functionalities that are not available in AMDock. Our goal, however, was not to outperform ADT, but to provide a simple, easy-to-use tool with a maximal optimization of the docking procedures. We are currently working on expanding AMDock's capabilities, though we do believe that in its current state the program can be useful in many aspects.

Mayor recommendations

1) *A list of other docking GUIs is provided by the authors in the Background section. A useful addition to the manuscript, maybe in the Discussion section, would be a comparative table of functionalities for these GUIs and AMDock.*

Response: In the Discussion section we now address in more details the comparison between AMDock and other tools mentioned in the Introduction section. Please see the response to next point (2).

2) *In particular, the authors should discuss what are the advantages of their tool with respect to AutoDock Tools (ADT), which is probably the most widely used GUI for Autodock and Vina. ADT also offers many options for ligand and receptor preparation, docking input file preparation, launching docking, results analysis, etc. In addition, ADT includes its own visualization interface, whereas AMDock depends on an external program (PyMOL). - The authors claim that the main advantage of their tool is "the integration of several valuable external tools within a simple and intuitive graphical interface that guides the users along well-established docking protocols - using either Autodock4 or AutoDock Vina - from system preparation to analysis of results" (p3). This is rather vague and it would be helpful to clarify which of these external tools offer a unique advantage over existing GUIs.*

Response: We have included in the Discussion section a more comprehensive comparison between AMDock and ADT – the new Table 1. We would like to emphasize that our aim in developing AMDock is not to

replace ADT, but to create a helpful complement. From our perspective, ADT can be a bit difficult to manipulate for non-expert users. On the other hand, we continue working to make AMDock a more versatile and robust tool. Finally, it is worth mentioning that, to our knowledge, ADT development is not currently active.

3) *In particular, a list of external tools integrated by AMDock is provided (p3). Most of these tools are described in the manuscript, at the exception of OpenBabel, whose role is not discussed.*

Response: Please see the response to point 1 by Reviewer 1.

4) *I may have missed the point but it is not clear to me what this procedure brings more than two standard dockings with the target and off-target receptors.*

Response: In principle, they are two separate standard dockings. However, we do believe that carrying out docking on both receptors at the same time is an advantage. As pointed out by the Reviewer, the comparison conditions should be similar in both cases. Here, we allow the user to protonate both receptors to the same pH, determine the optimal search space for both receptors, run the docking simulations with the same selected program and analyze the results in a simple comparative format. Furthermore, the user can superimpose the receptors and obtain a visual comparison of both the search space and the docking results. Carrying out all these steps separately requires additional effort and may lead to errors. This type of docking exercise is the basis for inverse virtual screening used primarily in drug repurposing. We do intend to implement this feature in future AMDock versions.

5) *In the case study, the authors are docking an inhibitor on target and off-target receptors and find that the inhibitor indeed prefers the target. In such tests, it is important to ascertain the fairness of the comparison. It is not clear how the initial conformation of the inhibitor is chosen. If the docking is in any way biased in favor of the known pose, the comparison with the off-target is not fair. The authors need to make sure that the initial conformation is randomized and that a sufficiently large search box is used.*

Response: We now describe in more details the starting conditions, stating that the initial ligand conformation was randomized. The search space has the same dimensions for both receptors, since the box size depends on the radius of gyration of the ligand (SAR405), which is a constant parameter in this exercise. The center of the box is determined by the geometric center of the co-crystallized ligands in both complexes. As can be observed in Figure 3A, the two boxes enclose their corresponding binding sites with sufficient margins, so no bias is introduced for any of the receptors.

Minor recommendations

1) *PyMOL is importantly used by AMDock. Could the authors give information on the compatibility of AMDock with the different versions of PyMOL?*

Response: Previously, AMDock worked with PyMOL 1.8.5. Recently, we verified that the AMDock plugin is compatible also with PyMOL v2.x. The current Windows distribution works with PyMOL version 2.1. In Linux, AMDock works with any installed PyMOL version.

PyMOL is a robust and widely used visualization program. Currently, its use as an external program in AMDock implies limitations on its full exploitation. We are currently working on its incorporation as a native viewer for AMDock. This will be a significant advantage since it will allow a direct exchange of information between AMDock and PyMOL (for example, to manipulate the receptor and the ligand, for active visualization of molecule preparation, creation, visualization and modification of the search space, flexible side chain selection, etc.).

2) *Why version 2.7 of Python is used and not 3.*? Note that 2.7 will not be maintained past 2020*

Response: AMDock is programmed in Python 2.7 to guarantee full compatibility with third-party programs (e.g.: AutoDockTools, PDB2PQR, AutoLigand, etc.). Some of these programs have been recently updated to Python 3. ADTs are critical to prepare the input files for docking with AutoDock or AutoDock Vina. We have recently created a version of ADT in Python3 available here (https://github.com/Valdes-Tresanco-MS/AutoDockTools_py3). We plan to migrate all the code to Python 3 in future versions (a developmental version in Python3 is available here (<https://github.com/Valdes-Tresanco-MS/AMDock-win-py3>)).

3) *In the case study, a step consists in "aligning and superimposing their structures". It is not clear to me if it is an AMDock or a PyMOL functionality*

Response: It is done with PyMOL. A remark was added in the Case study section.

4) *The concept of an "AutoLigand object" is used in the manuscript but not defined*

Response: It is now defined in a footnote.

5) *What "previous ligand" means on p7 l1?*

Response: We define "previous ligand" as any fragment of heteroatoms that appear co-crystallized with the protein in the PDB file. Ideally, the user should leave only the atom sets of interest, *i.e.* inhibitors, fragments or substrates found in the crystallographic structure. If a protein-ligand complex is introduced as the receptor, the coordinates of the ligand(s) are also stored. These coordinates can be used in the search space determination option called "Centered on Hetero", as explained in the manuscript.

Abbreviations

GUI: Graphical user interface; ADT: AutoDockTools; PMV: Python molecular viewer; AMDock: Assisted Molecular Docking; PI3K: Phosphatidylinositol 3-kinase; LE: Ligand efficiency

Acknowledgements

Special thanks to the JetBrains company (<https://www.jetbrains.com/>) for granting a free open source license to use their software.

Availability and requirements

Project name: AMDock: Assisted Molecular Docking with AutoDock4 and AutoDock Vina.

Project home page: <https://github.com/Valdes-Tresanco-MS>

Operating system(s): Windows and Linux.

Programming language: Python.

Other requirements: Python 2.7.

License: GPL version 3.

Any restrictions to use by non-academics: Restricted by the license and the other softwares.

Authors' contributions

MSVT designed and did most of the programming work, with important contributions from MEVT. They also produced a first draft of the manuscript. PAV and EM contributed to the project design, supervised the entire work, and undertook the final manuscript preparation. All authors read and approved the final manuscript.

Funding

This work was supported by the University of Medellin and Minciencias (grant 738-2016).

Availability of data and materials

The manual as well as tutorial files are included in the program installation folder. Additionally, information such as common errors, frequently asked questions, etc., can be found in the AMDock repository and the following mailing-list: (<https://groups.google.com/forum/#forum/amdock>).

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Faculty of Basic Sciences, University of Medellin, Medellin, Colombia. ²Center of Protein Studies, Faculty of Biology, University of Havana, 25 & J, 10400 La Habana, Cuba. ³Centre for Molecular Simulations and Department of Biological Sciences, University of Calgary, Calgary, Alberta T2N 1N4, Canada. ⁴Present address: Donnelly Centre for Cellular & Biomolecular Research, University of Toronto, 160 College St, Toronto, ON M5S 3E1, Canada.

Received: 13 May 2020 Accepted: 4 August 2020

Published online: 16 September 2020

References

- Morris GM, et al. AutoDock4 and AutoDockTools4: automated docking with selective receptor flexibility. *J Comput Chem.* 2009;30:174–82. <https://doi.org/10.1002/jcc>.
- Trott O, Olson AJ. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J Comput Chem.* 2012;32:174–82. <https://doi.org/10.1002/jcc>.
- Vaqué M, Arola A, Aliagas C, Pujadas G. BDT: an easy-to-use front-end application for automation of massive docking tasks and complex docking strategies with AutoDock. *Bioinformatics.* 2006;22(14):1803–4. <https://doi.org/10.1093/bioinformatics/btl197>.
- Zhang S, Kumar K, Jiang X, Wallqvist A, Reifman J. DOVIS: an implementation for high-throughput virtual screening using AutoDock. *BMC Bioinformatics.* 2008;9:1–4. <https://doi.org/10.1186/1471-2105-9-126>.

5. Jiang X, Kumar K, Hu X, Wallqvist A, Reifman J. DOVIS 2.0: an efficient and easy to use parallel virtual screening tool based on AutoDock 4.0. *Chem Cen J*. 2008;2(1):1–7. <https://doi.org/10.1186/1752-153X-2-18>.
6. Prakhov ND, Chernorudskiy AL, Gainullin MR. VSDocker: a tool for parallel high-throughput virtual screening using AutoDock on windows-based computer clusters. *Bioinformatics*. 2010;26(10):1374–5. <https://doi.org/10.1093/bioinformatics/btq149>.
7. Sandeep G, Nagasree KP, Hanisha M, Kumar MMK. AUDocker LE: a GUI for virtual screening with AUTODOCK Vina. *BMC Res Notes*. 2011;4(3):3–6. <https://doi.org/10.1186/1756-0500-4-445>.
8. Hu Z, Southerland W. WinDock: structure-based drug discovery on windows-based PCs. *J Comput Chem*. 2007;28:2347–51. <https://doi.org/10.1002/jcc>.
9. Bullock CW, Jacob RB, McDougal OM, Hampikian G, Andersen T. Dockomatic - Automated ligand creation and docking. *BMC Res Notes*. 2010;3(1):289. <https://doi.org/10.1186/1756-0500-3-289>.
10. Seeliger D, de Groot BL. Ligand docking and binding site analysis with PyMOL and Autodock/Vina. *J Comput Aided Mol Des*. 2010;24(5):417–22. <https://doi.org/10.1007/s10822-010-9352-6>.
11. Dallakyan S, Olson AJ. Small-Molecule Library Screening by Docking with PyRx. *Methods Mol Biol*. 2015;1263:243–50.
12. Abreu RMV, Froufe HJC, Queiroz MJRP, Ferreira ICFR. MOLA: a bootable, self-configuring system for virtual screening using AutoDock4/Vina on computer clusters. *J Cheminformatics*. 2010;2(1):2–7. <https://doi.org/10.1186/1758-2946-2-10>.
13. di Muzio E, Toti D, Polticelli F. DockingApp: a user friendly interface for facilitated docking simulations with AutoDock Vina. *J Comput Aided Mol Des*. 2017;31(2):213–8. <https://doi.org/10.1007/s10822-016-0006-1>.
14. García-Pérez C, Peláez R, Therón R, López-Pérez JL. JADOPPT: Java based AutoDock preparing and processing tool. *Bioinformatics*. 2017;33(4):583–5. <https://doi.org/10.1093/bioinformatics/btw677>.
15. Harris R, Olson AJ, Goodsell DS. Automated prediction of ligand-binding sites in proteins. *Proteins*. 2007;70:1506–17. <https://doi.org/10.1002/prot>.
16. Boyle NMO, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR. Open Babel: An open chemical toolbox. *J Cheminform*. 2011;3:33. <https://doi.org/10.1186/1758-2946-3-33>.
17. Dolinsky TJ, Nielsen JE, McCammon, Baker NA. PDB2PQR: An automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations. *Nucleic Acids Res*. 2004;32:665–7. <https://doi.org/10.1093/nar/gkh381>.
18. Schrödinger L. The PyMOL molecular graphics system. 2002.
19. Santos-Martins D, Forli S, Ramos MJ, Olson AJ. AutoDock4Zn: an improved AutoDock force field for small-molecule docking to zinc metalloproteins. *J Chem Inf Model*. 2014;54(8):2371–9. <https://doi.org/10.1021/ci500209e>.
20. Kenny PW. The nature of ligand efficiency. *J Cheminformatics*. 2019;11(1):1–18. <https://doi.org/10.1186/s13321-019-0330-2>.
21. Schultes S, de Graaf C, Haaksmä EEJ, de Esch IJP, Leurs R, Krämer O. Ligand efficiency as a guide in fragment hit selection and optimization. *Drug Discov Today Technol*. 2010;7(3):157–62. <https://doi.org/10.1016/j.ddtec.2010.11.003>.
22. Foster FM, Traer CJ, Abraham SM, Fry MJ. The phosphoinositide (PI) 3-kinase family. *J Cell Sci*. 2003;116(15):3037–40. <https://doi.org/10.1242/jcs.00609>.
23. Ronan B, et al. A highly potent and selective Vps34 inhibitor alters vesicle trafficking and autophagy. *Nat Chem Biol*. 2014;10(12):1013–9. <https://doi.org/10.1038/nchembio.1681>.
24. Burley SK, et al. RCSB Protein Data Bank: biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy. *Nucleic Acids Res*. 2019;47. <https://doi.org/10.1093/nar/gky1004>.
25. Feinstein WP, Brylinski M. Calculating an optimal box size for ligand docking and virtual screening against experimental and predicted binding pockets. *J Cheminform*. 2015;7(18). <https://doi.org/10.1186/s13321-015-0067-5>.
26. Salentin S, et al. PLIP: fully automated protein-ligand interaction profiler. *Nucl Acids Res*. 2015;43(W1):W443–7. <https://doi.org/10.1093/nar/gkv315>.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

