

# MAGI-Dock: a PyMOL companion to Autodock Vina

J. KAFTALLI<sup>1</sup>, A. BERNINI<sup>2</sup>, G. BONETTI<sup>3</sup>, S. CRISTONI<sup>4</sup>, G. MARCEDDU<sup>1</sup>, M. BERTELLI<sup>1,3,5</sup>

<sup>1</sup>MAGI EUREGIO, Bolzano, Italy

<sup>2</sup>Department of Biotechnology, Chemistry and Pharmacy, University of Siena, Siena, Italy

<sup>3</sup>MAGI'S LAB, Rovereto, Trento, Italy

<sup>4</sup>I.S.B – Ion Source & Biotechnologies SRL, Bresso, Milan, Italy

<sup>5</sup>MAGISNAT, Peachtree Corners, GA, USA

**Abstract.** – Molecular docking simulation of small molecule drugs to macromolecules is valuable in structural biology and medicinal chemistry research. Its spread is supported by freely available software and databases. Like many resources in the free domain, docking software is command-line based, which comes to a limitation when defining the volume encompassing an active site, the so-called docking box.

The box center and size, usually specified as cartesian coordinates, can be adjusted to correctly cover the active site only with a third-party molecular graphics program compatible with the docking input/output files, which reduces the choice to a few options. Moreover, the additional staff training may hamper the adoption of such software, e.g., in an enterprise environment.

We exposed the functionality of Autodock and Autodock Vina into a graphical user interface extending upon that of PyMOL. Both the functionality of PyMOL and Autodock are merged, synergizing the capabilities of each program.

To overcome such limitations, here we present MAGI-Dock. This graphical user interface combines the power of two of the most used free software for docking and graphics, Autodock Vina and PyMOL. MAGI-Dock is a free open-source software available under the GPL and can be downloaded from <https://github.com/gjonwick/MAGI-Dock>.

The coupling of Autodock Vina with PyMOL through a graphical interface removes the molecular modeling limitations that come with Autodock. Therefore, MAGI-Dock could be conducive to lowering the learning curve for molecular docking simulation, with benefits for trainees in both academia and enterprise environments.

*Key Words:*

MAGI-Dock, PyMOL, Autodock Vina, Molecular docking, Structural biology.

## Background

Molecular docking simulation of small molecules to bio-macromolecules is a powerful tool for structure-based drug discovery, which has reached widespread use also thanks to the many available resources in the free domain. AutoDock Vina<sup>1</sup> is one of the fastest and most widely used open-source docking engines in this field, providing flexible receptors, hydrated docking, support for macrocycle molecules, multiple scoring functions, etc. Moreover, simultaneous docking of multiple ligands and batch mode for virtual screening allows for exploiting large, customized virtual libraries of compounds and fragments<sup>2</sup> and orphan drugs<sup>3</sup>. Once the macro-molecular target and the ligands to be screened are defined, the docking process is carried out in four steps: (1) the preparation of the target and ligands (with setting up of rotatable bonds for the latter) by conversion into a suitable format (PDBQT); (2) the setting up of a search box defining the binding site on the target, including the set-up of flexible sidechains; (3) running the docking simulation; (4) analysis of the result. The command-line nature of Autodock allows for automating the whole process in a high throughput fashion<sup>4</sup> by, e.g., shell scripting, except for the set-up of the search box on the target. This step is usually carried out once and requires user-supervised molecular graphics interaction with the target to fine-tune the box boundaries and the flexibility of side chains forming the binding site. The support for such box drawing is limited to a few molecular graphics software<sup>5</sup> or discontinued<sup>6</sup>. Above all, a modern implementation is missing in one of the most widely used macro-molecular visualization tools, PyMOL<sup>7</sup>. Since PyMOL is written in Python, one of the most popular

programming languages<sup>8</sup>, it can easily be extended to Python plugins. Using this feature, we provide MAGI-Dock, a free PyMOL plugin for setting up ligands, search boxes, flexible sidechains, and molecular docking. The installation steps can be found in the section ‘Software requirements and installation’ in the [Supplementary File](#).

## MAGI-Dock Program

MAGI-Dock was developed in Python and PyQT<sup>9</sup>. It uses a graphical user interface program that runs on top of PyMOL and each functionality related to molecular docking is delegated to the scripts provided by MGLTools and Autodock Vina. The program uses the Python interpreter and libraries that come with PyMOL. PyMOL also provides its own interface to PyQT, allowing for easy integration of user scripts to its interface. The Auto-Dock Python scripts are run using the Python interpreter that comes with MGLTools, and its path can be specified when running MAGI-Dock.

## Interface and Process

MAGI-Dock is a PyMOL interface to Autodock Vina that fills the gap between the power of the popular docking software’s text-driven commands and the user interaction experience given by the most widespread molecular graphics software. The plugin provides a GUI for each preparation and simulation command. In particular, it provides an interactive drawing of the simulation box, a fundamental step needing user supervision. With the convenient interface that PyMOL provides, users can run docking effortlessly, without switching back and forth between PyMOL and other molecular modeling software. Moreover, such a feature allows users to use the molecular modeling power of PyMOL and run docking directly on customized structures. The plugin is conveniently organized into tabs, each managing a sequential step (Figure 1).

The first tab, *Config*, allows all configurations required by Autodock to be set. The Receptor tab deals with the generation of receptors ([Supplementary File](#)). Once the receptor is ready, the user may set selected amino acids as flexible. The receptor selected in the generated receptor list will also be automatically loaded into “memory”, ready to be used as a docking target. The *Ligands* tab deals with the preparation of ligands

([Supplementary File](#)): similar to the *Receptor* tab, users can choose the selections representing the ligands and click the *Add* button. This notifies the plugin about the ligands that are going to be prepared. Users may select the wanted ligands from the list and click the *Generate* button to prepare the ligands. The successfully prepared ligands will appear on the prepared ligands list. In the main form, there is a convenient interface to generate the docking box ([Supplementary File](#)). Users can make a selection in the PyMOL viewport, and by clicking the *Generate* box button, a box will automatically be created around the selection; alternatively, the object name can be typed in the *Selection* field, and a box will be generated around it with the same button. The box position and size can be finely tuned by modifying coordinates and edge size fields in the main form. Additional options are provided for the docking box: step size of the movement, useful for fine adjustment; hide/show the box boundaries; color the faces of the box (default transparent). In the *Docking* Tab ([Supplementary File](#)), the exhaustiveness, number of poses, minimum RMSD, and other options for the Vina run can be set by using the Parameters panel. Finally, the simulation can be started by clicking on the *Run Docking Job* button.

## Discussion

As computational biology is becoming a popular approach to tackling biological problems, there is a need for accessible software whose setup and usage will not bottleneck scientific research. Molecular Docking is one of the most popular approaches used in drug screening<sup>10-12</sup>, with Autodock<sup>5</sup> being a well known open-access software but with a rather limited 3D molecular modeling capability. On the other hand, PyMOL is one of the most widely used macromolecular visualization tools, with great molecular modeling facilities. Other works<sup>6</sup> have been important to merge these tools together but are outdated and discontinued. The coupling of Autodock Vina with PyMOL through a graphical interface is conducive to lowering the learning curve for molecular docking simulation, with benefits for trainees in both academia and enterprise environments. Therefore, in this work, we developed MAGI-Dock, a modern implementation of molecular docking into PyMOL, allowing users to dock molecules to their custom structures through the PyMOL interface.

## MAGI-Dock: a PyMOL companion to Autodock Vina

The screenshot displays the PyMOL interface with the MAGI-Dock plugin window. The PyMOL window shows a protein structure with a simulation box drawn in purple, red, and blue. The MAGI-Dock window has tabs for Config, Receptor, Ligands, Docking, and Results. The Config tab is active, showing a 'Refresh' button and a 'Generate Box from PyMOL Selection' section with a 'hPS5' selection and 'Generate Box' button. Below this are fields for Center X, Y, Z and Dim X, Y, Z. The bottom section contains fields for MGL python exe, AutoDock Tools Folder, Vina Executable, Autogrid Executable, Working Dir, and Box Path, each with a 'Browse' button. A status bar at the bottom of the PyMOL window shows 'PyMOL>show sticks, sele...'.

```

graph TD
    AutoDock[AutoDock Functionality] --- PrepareLigands[Prepare Ligands]
    AutoDock --- PrepareReceptors[Prepare Receptors]
    AutoDock --- RunDocking[Run Docking]
    RunDocking --- Batch[Batch]
    RunDocking --- Multiple[Multiple]
    RunDocking --- Single[Single]
    AutoDock --- Configuration[Configuration]
    Configuration --- WorkingDir[Working Directory]
    Configuration --- AutodockExecutables[Autodock Executables]
    Configuration --- BoxPath[Box Path]
    AutoDock --- DockingBox[Docking Box Interface]
    DockingBox --- ImportBox[Import Box]
    DockingBox --- EditCoordinates[Edit Coordinates]
    DockingBox --- ColorFaces[Color Faces]
    DockingBox --- SaveBox[Save Box]
    
```

**Figure 1.** MAGI-Dock interface organized in tabs, and the simulation box drawn in different colors to identify the XYZ faces. The diagram indicates the capabilities of the plugin.

## Conclusions

In conclusion, we believe that a tool such as MAGI-Dock will be very useful to researchers in the topics of Computational Biology, Drug Design, Bioengineering, etc., due to its ability to couple two of the most used software in these fields into a rational and easy to use interface. MAGI-Dock integrates all the steps needed to run molecular docking into the interface of PyMOL, from the preparation of receptors and ligands to running molecular docking. It remains an open-source software, and with the help of other developers, it can be further improved and become an important asset to the world of computational biology.

### Acknowledgments

The authors would like to acknowledge Daniel Seeliger and his work on merging Autodock and PyMOL as an inspiration for the work done in this article.

### Funding

This research was funded by the Provincia Autonoma di Bolzano in the framework of LP 15/2020.

### Informed Consent

Not applicable.

### Ethics Approval

Not applicable.

### Availability of Data and Materials

The data are within the test or in the supplementary materials document.

### Authors' Contributions

Conceptualization, J.K. and A.B.; Methodology, J.K.; Investigation, J.K. and A.B.; Writing- original draft preparation, J.K. and A.B.; Writing- review and editing, J.K., A.B., G.B., and S.C.; Project administration, J.K. and A.B.; Funding acquisition, G.M., and M.B. All authors have read and agreed to the published version of the manuscript.

### Conflicts of Interest

The authors of this manuscript J. Kaftalli, G. Marceddu, and M. Bertelli are affiliated with MAGI EUREGIO. MAGI EUREGIO released MAGI-Dock software on Git Hub in open source.

## References

- 1) 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* 2010; 31: 455-461.
- 2) Irwin JJ, Shoichet BK. ZINC--a free database of commercially available compounds for virtual screening. *J Chem Inf Model* 2005; 45: 177-182.
- 3) Wishart DS, Knox C, Guo AC, Cheng D, Shrivastava S, Tzur D, Gautam B, Hassanali M. DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res* 2008; 36: D901-906.
- 4) Eberhardt J, Santos-Martins D, Tillack AF, Forli S. AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings. *J Chem Inf Model* 2021 23; 61: 3891-3898.
- 5) Forli S, Huey R, Pique ME, Sanner MF, Goodsell DS, Olson AJ. Computational protein-ligand docking and virtual drug screening with the AutoDock suite. *Nat Protoc* 2016; 11: 905-919.
- 6) Seeliger D, de Groot BL. Ligand docking and binding site analysis with PyMOL and Autodock/Vina. *J Comput Aided Mol Des* 2010; 24: 417-422.
- 7) Delano WL. PyMOL: An Open-Source Molecular Graphics Tool. *CCP4 Newsletter Pro Crystallogr* 2002. Available at: [https://legacy.ccp4.ac.uk/newsletters/newsletter40/11\\_pymol.pdf](https://legacy.ccp4.ac.uk/newsletters/newsletter40/11_pymol.pdf).
- 8) Srinath KR. Python—The Fastest Growing Programming Language. *International Research Journal of Engineering and Technology*. IRJET 2017; 4: 354-357. Available at: <https://www.irjet.net/archives/V4/i12/IRJET-V4I1266.pdf>.
- 9) Summerfield M. *Rapid GUI programming with Python and Qt: the definitive guide to PyQt programming*. Addison-Wesley Pro 2007.
- 10) Paolacci S, Kiani AK, Shree P, Tripathi D, Tripathi YB, Tripathi P, Tartaglia GM, Farronato M, Farronato G, Connelly ST, Ceccarini MR, Coatto M, Ergoren MC, Sanlidag T, Dautaj A, Bertelli M. Scoping review on the role and interactions of hydroxytyrosol and alpha-cyclodextrin in lipid-raft-mediated endocytosis of SARS-CoV-2 and bioinformatic molecular docking studies. *Eur Rev Med Pharmacol Sci* 2021; 25: 90-100.
- 11) Joshi T, Joshi T, Sharma P, Mathpal S, Pundir H, Bhatt V, Chandra S. In silico screening of natural compounds against COVID-19 by targeting Mpro and ACE2 using molecular docking. *Eur Rev Med Pharmacol Sci* 2020; 24: 4529-4536.
- 12) Yuan S, Chan HCS, Hu Z. Using PyMOL as a platform for computational drug design. *WIREs Comput Mol Sci* 2017; 7: e1298. Available at: <https://wires.onlinelibrary.wiley.com/doi/10.1002/wcms.1298>.