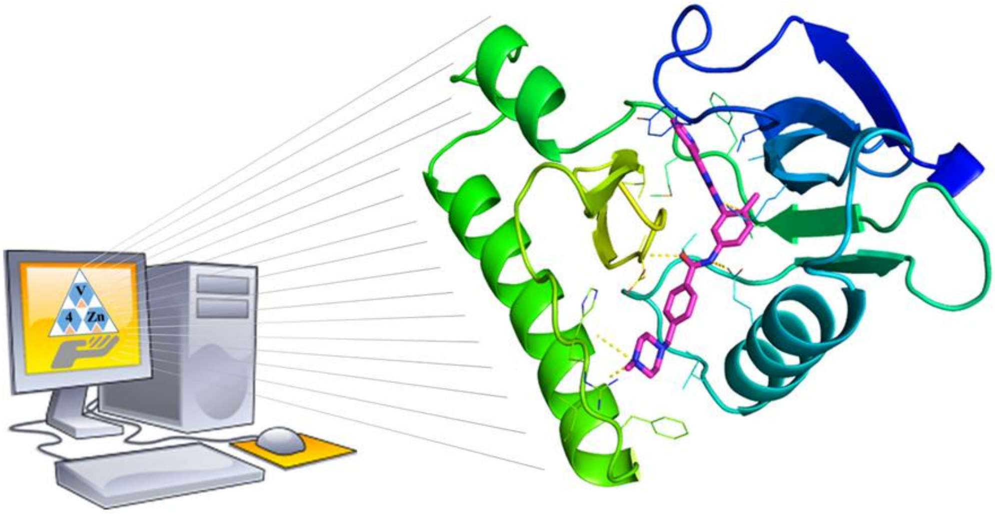
AMDock



Assisted Molecular Docking with AutoDock Vina and AutoDock4

USER MANUAL

Version 1.4.x

Last revision on 25/04

2020

This page intentionally blank

AMDock

USER MANUAL

Contributions from

Mario S. Valdés-Tresanco, BSc

Mario E. Valdés-Tresanco, MSc

Pedro A. Valiente, PhD

Ernesto Moreno, PhD

2020

TABLE OF CONTENTS

1. INTRODUCTION 2

1.1 How to cite AMDock? 4

2. DOWNLOAD AND INSTALLATION 5

3. AMDock WORKFLOW and GUI 12

3.1 Main tab 13

3.1.1 Defining a working directory 13

3.1.2 Preparing input files 14

3.1.3 Defining a search space 15

3.1.4 Run docking 16

3.1.5 Analysis of results 17

3.2 Results Analysis tab 17

3.3 Configuration and Info tab 18

4. TUTORIALS 13

4.1 Before beginning… 13

4.2 Simple docking 13

4.2.1 “Custom box” with Autodock4Zn 13

4.2.2 “Box centered on Hetero” using Autodock Vina 17

4.2.2.1 re-Docking experiment 17

4.2.2.2 Docking a similar ligand 20

4.2.3 “Centered on residue(s)” using AutoDock Vina 22

4.2.4 “Automatic” mode using AutoDock Vina 25

4.3 Off-target docking 29

4.4 Scoring 33

5. APPENDIX 38

5.1 Change Box specifications on AMDock PyMol plugin 38

5.2 Making a high-quality image for publications 41

AMDock

Getting Started

1. INTRODUCTION

Here we introduce a manual showing all the functionalities of AMDock program. AMDock is a GUI program that assists docking runs with AutoDock Vina and AutoDock4 (including AutoDock4Zn force field for zinc-containing metalloproteins). The program includes: preparation of protein and ligand input files, the definition of a proper search space and docking run itself. There are multiple options to perform these tasks. That way, the user will be free to choose how automatic the docking run will be. Several programs are used and their functionalities on the AMDock context are explained across the manual. However, other applications of these programs are omitted here. All information about these programs can be reviewed in:

AutoDock Vina 1.2.1:

Manual: http://vina.scripps.edu/manual.html

Original Publication: Trott, O., Olson, A. J. (2010) AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J Comput Chem*, **31**: 455-461.

AutoDock 4.2.6:

Manual: http://autodock.scripps.edu/faqs-help/manual/autodock-4-2-user-guide

Original Publication: Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S., Olson, A. J. (2009) Autodock4 and AutoDockTools4: automated docking with selective receptor flexibility. *J Comput Chem*, **30**: 2785-2791.

AutoDock4$_{Zn}$ Force Field:

Manual: http://mgldev.scripps.edu/AutoDockZN/AutoDockZN_userguide.pdf

Original Publication: Santos-Martins, D., Forli, S., João Ramos, M., Olson, A. J. (2014) AutoDock4$_{Zn}$: an improved AutoDock force field for small-molecule docking to zinc metalloproteins. *J Chem Info Model*, **54**: 2371-2379.

Open Babel 2.4.1:

Manual: http://openbabel.org/docs/dev/OpenBabel.pdf

Original Publication: O'Boyle, N. M., Banck, M., James, C. A., Morley, C., Vandermeersch, T., Hutchison, G. R. (2011) Open Babel: An open chemical toolbox. *J Cheminf*, **3**: 33.

PDB2PQR 2.1:

Manual: http://www.ics.uci.edu/~dock/pdb2pqr/userguide.html

Original Publication: Dolinsky, T. J., Nielsen, J. E., McCammon, J. A., Baker, N. A. (2004) PDB2PQR: an automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations. *Nucleic Acids Res*, **32**: W665-7.

AutoDockTools Module 1.5.7:

*Manual:*http://autodock.scripps.edu/faqs-help/tutorial/using-autodock-4-with-autodocktools

Original Publication: Sanner, M. F. (1999) Python: A Programming Language for Software Integration and Development. *J Mol Graphics Mod*, **17**: 57-61

AutoLigand:

Manual: http://autodock.scripps.edu/faqs-help/tutorial/using-autoligand-with-autodocktools

Original Publication: Harris, R., Olson, A., Goodsell, D. (2008) Automated prediction of ligand-binding sites in proteins. *Proteins*, **70**: 1505-1517

Optimal Box Size 1.1:

Manual: http://www.brylinski.org/eboxsize

Original Publication: Feinstein, W. P., Brylinski, M. (2015) Calculating an optimal box size for ligand docking and virtual screening against experimental and predicted binding pockets. *J Cheminf*, **7**: 18

PyMOL 1.8.5:

Manual: http://pymol.sourceforge.net/newman/userman.pdf

Original Publication: DeLano, W. L. (2002) The PyMOL Molecular Graphics System.

***All the programs included in AMDock are open source*.**

In any case, we have tried hard to be as clear and concise as possible. Comments and questions are welcome, please send them to the AMDock mailing list.

1.1 How to cite AMDock?

When using AMDock program, please reference:

Coming soon!...

and the original publication of the corresponding module.

Thanks!

2. DOWNLOAD AND INSTALLATION

AMDock is distributed freely to the public. The current versions for Linux and Windows are available at https://github.com/Valdes-Tresanco-MS

Usually, the installation of a program becomes a “headache” to naive users. So, we have tried hard to make it as easy as possible. In this sense, the program has been written in Python and has been tested on Windows 10, Ubuntu 18.04 and Debian 8.0. It has not tested on MAC platform yet.

The Windows version of AMDock is distributed as a compressed file with all dependencies and functionalities. That means you can execute it quickly and without installing nothing more.

On the other hand, the UNIX version of AMDock is distributed as a compressed file. So, you must decompress this file, open the “README” file and follow the instructions.

AMDock

Basis

3. AMDock WORKFLOW and GUI

To understand how AMDock works and what kind of information is needed, it is necessary to know which programs are included in the AMDock environment and which procedures are performed (Figure 1).

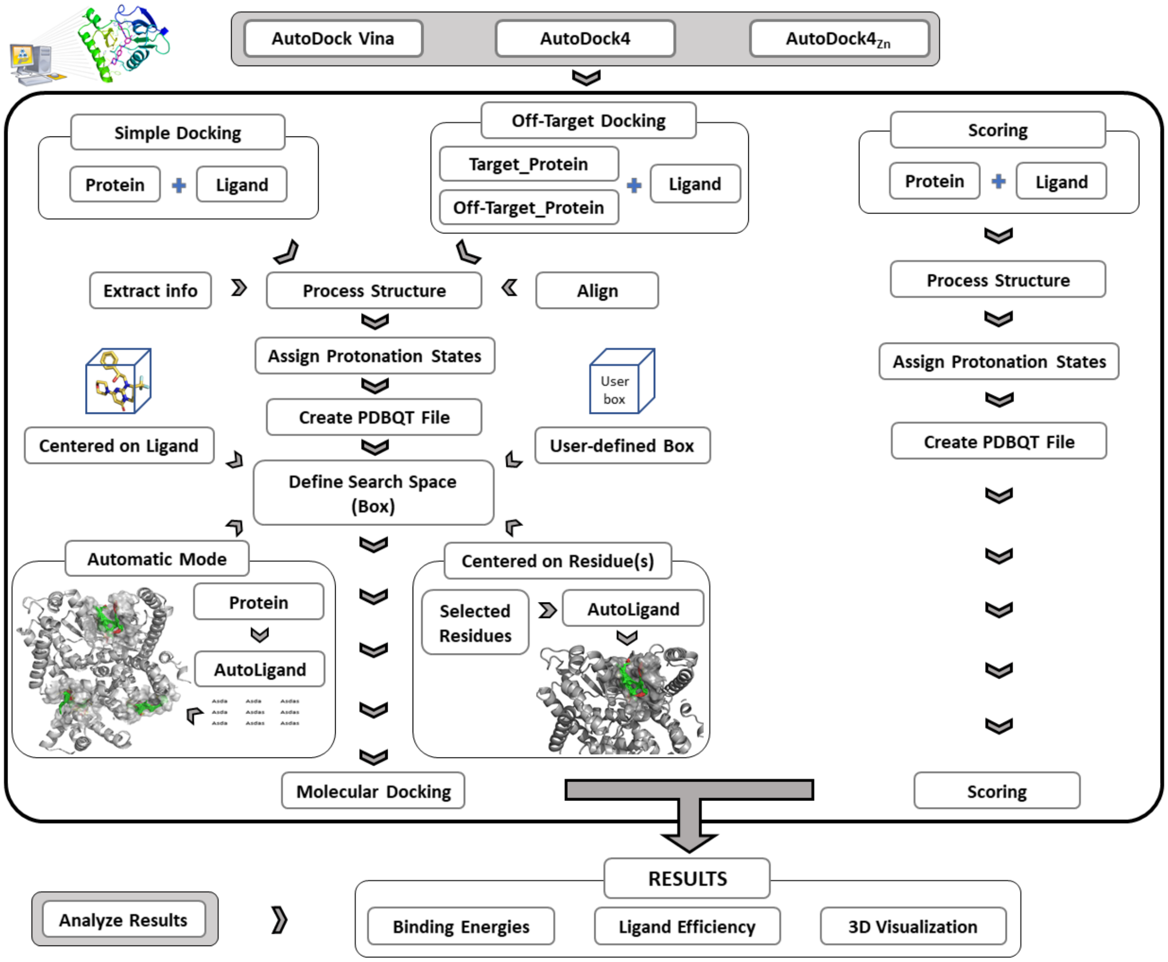
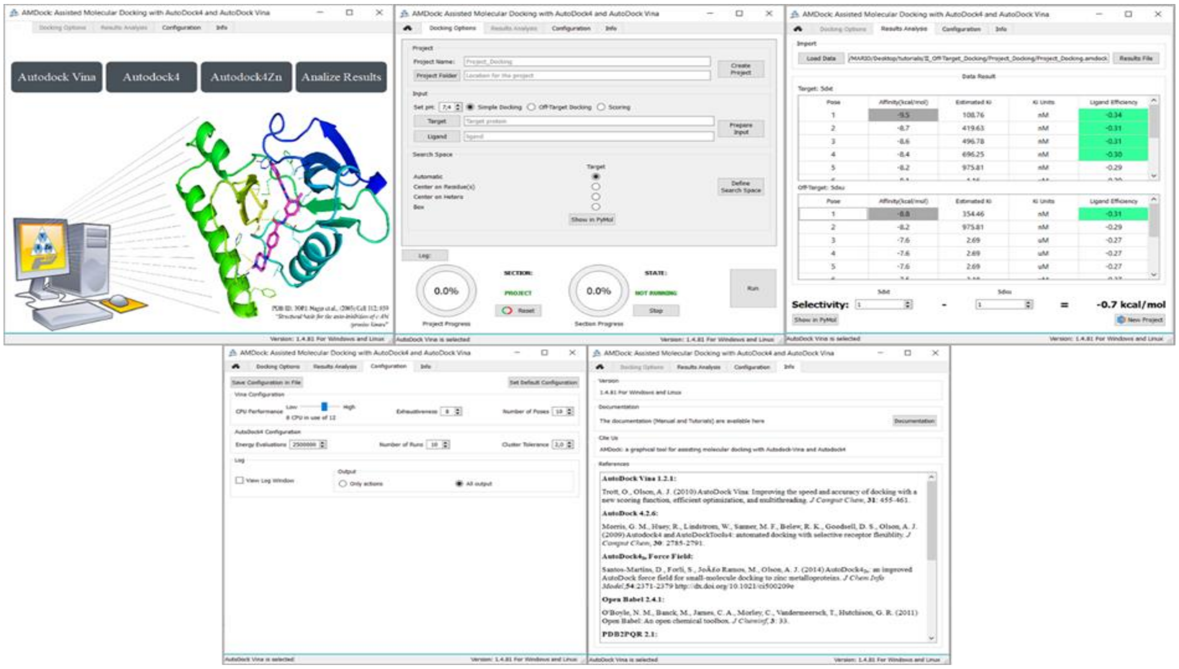


Figure 1. Workflow in AMDock

The AMDock main window has five tabs: 1) Home, 2) Docking Options, 3) Results Analysis, 4) Configuration and 5) Info. (Figure 2)



Project Name: Project_Docking

Project Folder: Location for the project

Set pH: 7,4 ● Simple Docking ○ Off-Target Docking ○ Scoring

Target: Target protein

Ligand: ligand

Target: ● Automatic ○ Center on Residue(s) ○ Center on Hetero ○ Box

/MARIO/Desktop/tutorials/II_Off-Target_Docking/Project_Docking/Project_Docking.amdock

Target: 5dxt

Pose	Affinity(kcal/mol)	Estimated Ki	Ki Units	Ligand Efficiency
1	-9.5	108.76	nM	-0.34
2	-8.7	419.63	nM	-0.31
3	-8.6	496.78	nM	-0.31
4	-8.4	696.25	nM	-0.30
5	-8.2	975.81	nM	-0.29

Off-Target: 5dxu

Pose	Affinity(kcal/mol)	Estimated Ki	Ki Units	Ligand Efficiency
1	-8.8	354.46	nM	-0.31
2	-8.2	975.81	nM	-0.29
3	-7.6	2.69	uM	-0.27
4	-7.6	2.69	uM	-0.27
5	-7.6	2.69	uM	-0.27

Selectivity: 5dxt 1 - 5dxu 1 = -0.7 kcal/mol

CPU Performance: 8 CPU in use of 12; Exhaustiveness: 8; Number of Poses: 10

Energy Evaluations: 2500000; Number of Runs: 10; Cluster Tolerance: 2,0

□ View Log Window; Output: ○ Only actions ● All output

Version: 1.4.81 For Windows and Linux

Documentation: The documentation (Manual and Tutorials) are available here

Cite Us: AMDock: a graphical tool for assisting molecular docking with Autodock-Vina and Autodock4

References:

AutoDock Vina 1.2.1:

Trott, O., Olson, A. J. (2010) AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J Comput Chem*, **31**: 455-461.

AutoDock 4.2.6:

Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S., Olson, A. J. (2009) Autodock4 and AutoDockTools4: automated docking with selective receptor flexiblity. *J Comput Chem*, **30**: 2785-2791.

AutoDock4$_{Zn}$ Force Field:

Santos-Martins, D., Forli, S., JoÃ£o Ramos, M., Olson, A. J. (2014) AutoDock4$_{Zn}$: an improved AutoDock force field for small-molecule docking to zinc metalloproteins. *J Chem Info Model*, **54**:2371-2379 http://dx.doi.org/10.1021/ci500209e

Open Babel 2.4.1:

O'Boyle, N. M., Banck, M., James, C. A., Morley, C., Vandermeersch, T., Hutchison, G. R. (2011) Open Babel: An open chemical toolbox. *J Cheminf*, **3**: 33.

PDB2PQR 2.1:

Figure 2. AMDock components. From left to right: Home, Docking Options, Results Analysis, Configuration and Info.

3.1 Main tab

AMDock assists docking runs with AutoDock Vina and AutoDock4 (including AutoDock4Zn force field for zinc-containing metalloproteins). Once the docking engine has been selected, you can perform a Simple Docking, Off-Target Docking or Scoring procedure.

AMDock proceeds by following 5 steps:

3.1.1 Defining a working directory

Firstly, a new directory is created (default name: Docking_Project) in a location specified by the user. This directory contains 2 folders: *i*) "input", where the protein (*.pdb, *.ent, *.pdbqt) and ligand (*.pdb, *.mol2, *.pdbqt) files are stored and *ii*) "results", where docking results are stored. Docking results will depend on the selected procedure (Simple Docking, Off-Target Docking or Scoring). Please, see the TUTORIAL section for more information.

A *.amdock file is generated and placed in the working directory. This file contains the summarized data of the docking procedure. A log file containing all the output from all the programs and algorithms used can be also generated.

3.1.2 Preparing input files

For Protein: The cartesian coordinates of the protein are needed. These can be taken from the protein file or a protein-ligand complex. Coordinates usually come from X-ray crystallography, NMR spectroscopy cryo-EM or model-building. AMDock deals with one of the three formats listed below:

*.pdbqt - This means you have prepared the protein input file by yourself and AMDock will not perform any modification. This is recommended only if you know the minimum procedure to generate a proper *.pdbqt file.

*.pdb; *.ent - Files in these formats are processed by AMDock in order to generate a proper .pdbqt file (prepare_receptor4.py). AMDock will perform several tasks for you:

- ✓ remove heteroatoms. That includes water molecules, ions, crystallization reagents, ligands. *Important! Even when ligands atoms are removed from the protein-ligand file, ligand coordinates are stored and can be used to define a search space (see TUTORIAL section) If some cofactor or ligand should be conserved at the active site because of its importance for docking process, it is recommended to prepare a .pdbqt file in an external program since AMDock will remove all ligand heteroatoms present in the receptor (see TUTORIAL section).*
- ✓ delete alternates (prepare_receptor4.py)
- ✓ determine protonation states (PDB2PQR v. 2.01)
- ✓ complete missing side chains (PDB2PQR v. 2.01) (*Important! This software is not able to model missing residues. These residues can be modeled with external software, i.e. MODELLER*)
- ✓ merge charges and remove non-polar hydrogens (prepare_receptor4.py)
- ✓ align proteins (if “off-target” docking is selected)

For Ligand: The cartesian coordinates of the ligand are needed. These usually come from X-ray crystallography, NMR spectroscopy or model-building. AMDock deals with one of the three formats listed below:

*.pdbqt - This means that you have prepared a protein input file by yourself and AMDock will not perform any modification. This is recommended only if you know the minimum procedure to generate a proper *.pdbqt file.

*.pdb; *.mol2 - Files in these formats are processed by AMDock in order to generate a proper .pdbqt file (prepare_ligand4.py). AMDock will perform several tasks for you:

- ✓ determines protonation states (OpenBabel v. 2.4.1 or ADT)
- ✓ merge charges and remove non-polar hydrogens (prepare_ligand4.py)

pH: A pH value can be set for determining the protonation states of both the receptor and the ligand.

As can be seen, AMDock should be able to handle successfully most protein and ligand files. Even so, it is always recommendable checking the input files and remove unnecessary ions, solvent, cofactors; checking for “connects” on ligand files.

3.1.3 Defining a search space

A search space is defined when the coordinates of the geometrical center and the dimensions of the box have been defined. There are multiple options in AMDock for defining the search space (box’s center and dimensions). These options are listed below and comprise from *automatic definition* to *user guide definition* of the search space.

- ✓ Automatic (known as “*blind docking*”): potential ligand-binding sites are identified and characterized by using the AutoLigand tool. The AutoLigand code generates objects (“FILL”) with the ligand’s dimensions. Since the binding site is unknown, boxes with optimal dimensions (see **Optimal Box Size 1.1**) are placed on the geometric center of each of the generated objects and an independent docking run is performed for each of the predicted binding sites. Binding sites are sorted at the end according to the results from the docking run. *This option is recommendable only when you have no idea where the binding site is*.

- ✓ Center on Residue(s): Centering on residues usually leads to small/big search space which compromises the reproducibility of the results. Therefore, instead of centering directly over the selected residues, we employed Autoligand for generating an object located at the geometric center of the selected residues. Then, a box with optimal dimensions (see **Optimal Box Size 1.1**) is placed at the geometric center of the generated object. This way, the optimal size of the box is guaranteed without increasing the size of the box. *This option is recommendable when you know which residue(s) is/are located at the binding site. This information usually comes from mutagenesis experiments, comparing proteins belonging to the same family, in silico prediction of binding sites, etc.*
- ✓ Center on Ligand: A box with optimal dimensions (see **Optimal Box Size 1.1**) is placed on the geometric center of the ligand. This option will be available only if a protein-ligand complex was used as the protein input file. *This option is recommendable when you are interested in doing redocking experiments, docking ligands belonging to the same family or ligands with the same binding mode.*
- ✓ Box: Box's center and dimensions are defined by the user. *This option is usually preferred by expert users. Remember, the box's center and dimensions should be as proper as possible to ensure a suitable search space.*

The search space can be visualized and modified at the user convenience using PyMOL. A convenient representation is showed in PyMOL, highlighting the most important elements for each method. This constitutes an advantage as compared to traditional programs with fewer options.

3.1.4 Run docking

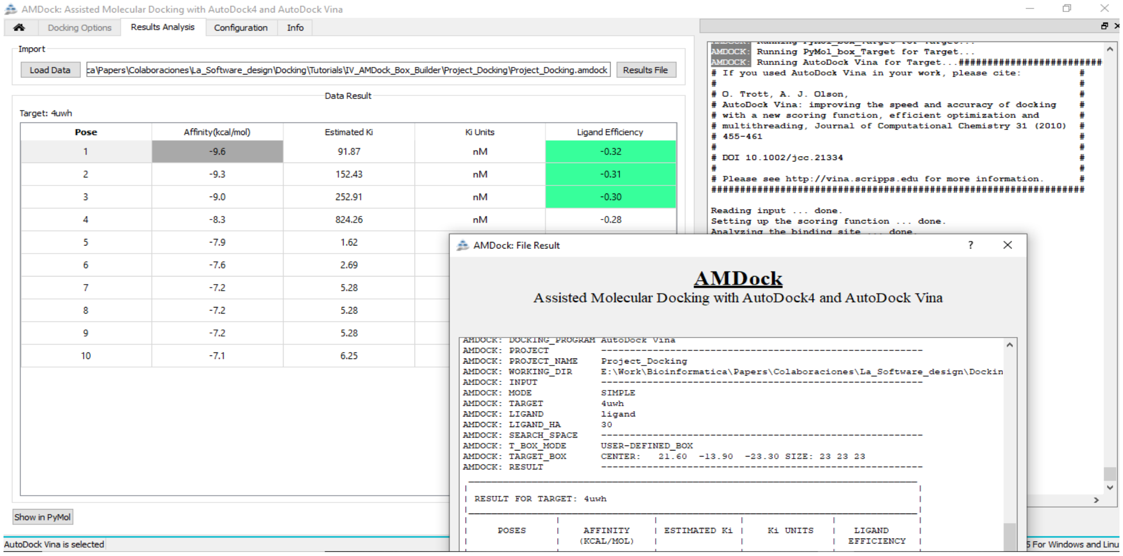
Running time will depend on several factors such as: the number of poses, number of rotatable bonds of ligand, etc., although AutoDock Vina tends to be faster than AutoDock 4 by orders of magnitude. The whole process can be followed on the right side of the screen.

3.1.5 Analysis of results

After docking run ends, you will get automatically into the “Results Analysis” tab. There, you will observe a summarizing table with Binding Energies, Estimated K*i* values, and Ligand Efficiencies.

3.2 Results Analysis tab

The result tab contains the final results of the docking run. This includes: scores values, predicted Ki values and Ligand Efficiencies. Pressing the “Show in PyMOL” button in the left bottom corner you will visualize in PyMOL the complex between the receptor and the ligand pose with the lowest energy. When performing “off-target” docking, both receptors can be visualized at the same time, which allows for a better comparison between the two predicted complexes. Polar contacts are shown as dashed yellow lines and an image with publication standards could be generated after few clicks (See Appendix). Keep in mind that the view is created automatically by a PyMOL script. Since it is automatic, the view cannot be in the most informative angle.



ca\Papers\Colaboraciones\La_Software_design\Docking\Tutorials\IV_AMDock_Box_Builder\Project_Docking\Project_Docking.amdock

Target: 4uwh

Pose	Affinity(kcal/mol)	Estimated Ki	Ki Units	Ligand Efficiency
1	-9.6	91.87	nM	-0.32
2	-9.3	152.43	nM	-0.31
3	-9.0	252.91	nM	-0.30
4	-8.3	824.26	nM	-0.28
5	-7.9	1.62		
6	-7.6	2.69		
7	-7.2	5.28		
8	-7.2	5.28		
9	-7.2	5.28		
10	-7.1	6.25		

```
AMDOCK: Running PyMol_box_Target for Target...
AMDOCK: Running AutoDock Vina for Target...##########################
# If you used AutoDock Vina in your work, please cite:               #
#                                                                    #
# O. Trott, A. J. Olson,                                             #
# AutoDock Vina: improving the speed and accuracy of docking         #
# with a new scoring function, efficient optimization and            #
# multithreading, Journal of Computational Chemistry 31 (2010)       #
# 455-461                                                            #
#                                                                    #
# DOI 10.1002/jcc.21334                                              #
#                                                                    #
# Please see http://vina.scripps.edu for more information.           #
######################################################################

Reading input ... done.
Setting up the scoring function ... done.
Analyzing the binding site ... done.
```

AMDock

Assisted Molecular Docking with AutoDock4 and AutoDock Vina

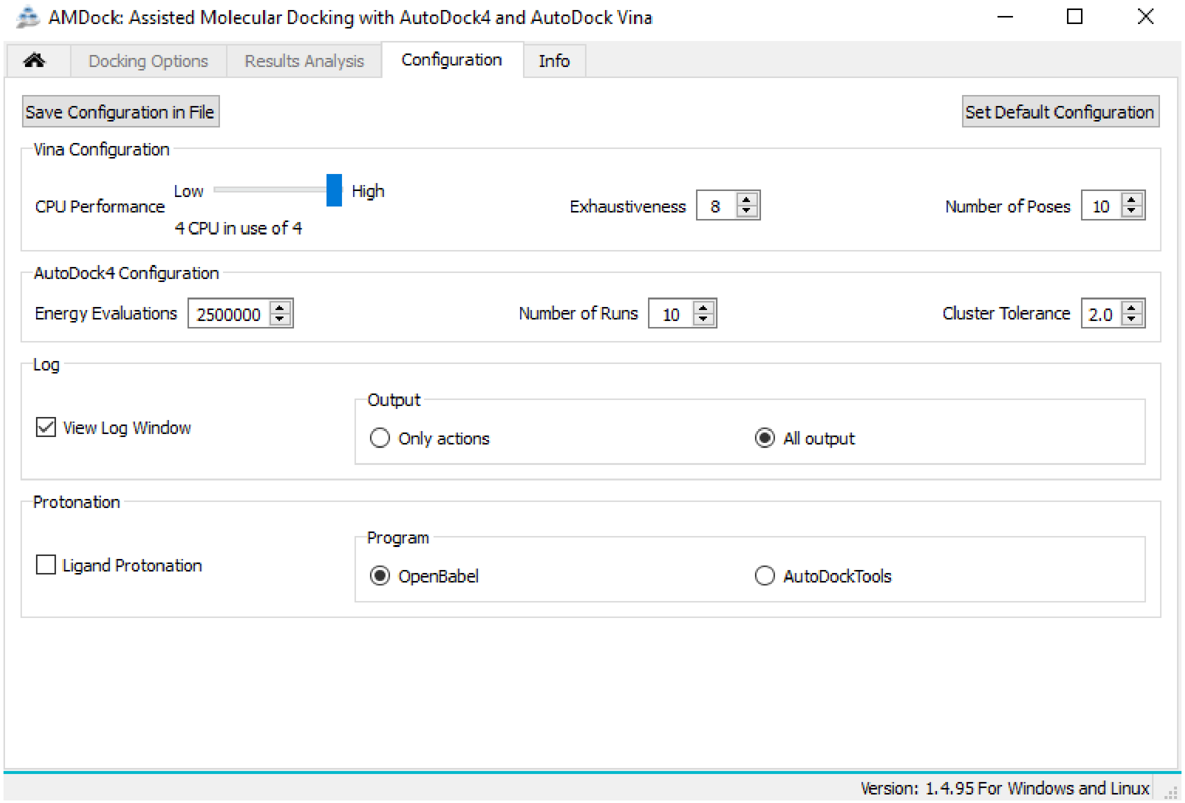
```
AMDOCK: DOCKING_PROGRAM AutoDock Vina
AMDOCK: PROJECT         -----------------------------------------------------
AMDOCK: PROJECT_NAME    Project_Docking
AMDOCK: WORKING_DIR     E:\Work\Bioinformatica\Papers\Colaboraciones\La_Software_design\Dockin
AMDOCK: INPUT           -----------------------------------------------------
AMDOCK: MODE            SIMPLE
AMDOCK: TARGET          4uwh
AMDOCK: LIGAND          ligand
AMDOCK: LIGAND_HA       30
AMDOCK: SEARCH_SPACE    -----------------------------------------------------
AMDOCK: T_BOX_MODE      USER-DEFINED_BOX
AMDOCK: TARGET_BOX      CENTER:   21.60  -13.90  -23.30 SIZE: 23 23 23
AMDOCK: RESULT          -----------------------------------------------------

 ____________________________________________________________________
|                                                                    |
| RESULT FOR TARGET: 4uwh                                            |
|____________________________________________________________________|
|               |                |              |           |              |
|     POSES     |    AFFINITY    | ESTIMATED Ki |  Ki UNITS |    LIGAND    |
|               |   (KCAL/MOL)   |              |           |  EFFICIENCY  |
```

Figure 3. AMDock Result Analysis Tab. Behind left to right: Result table and Log record. The AMDock output table is shown in the front.

3.3 Configuration and Info tab

In the configuration tab, it is possible to change several parameters related to the programs and algorithms in AMDockuch: the number of processors for running the docking calculation, the number of poses, etc.



AMDock: Assisted Molecular Docking with AutoDock4 and AutoDock Vina
Docking Options
Results Analysis
Configuration
Info
Save Configuration in File
Set Default Configuration

Vina Configuration

CPU Performance Low — High
4 CPU in use of 4
Exhaustiveness 8
Number of Poses 10

AutoDock4 Configuration

Energy Evaluations 2500000
Number of Runs 10
Cluster Tolerance 2.0

Log

☑ View Log Window
Output: ○ Only actions ● All output

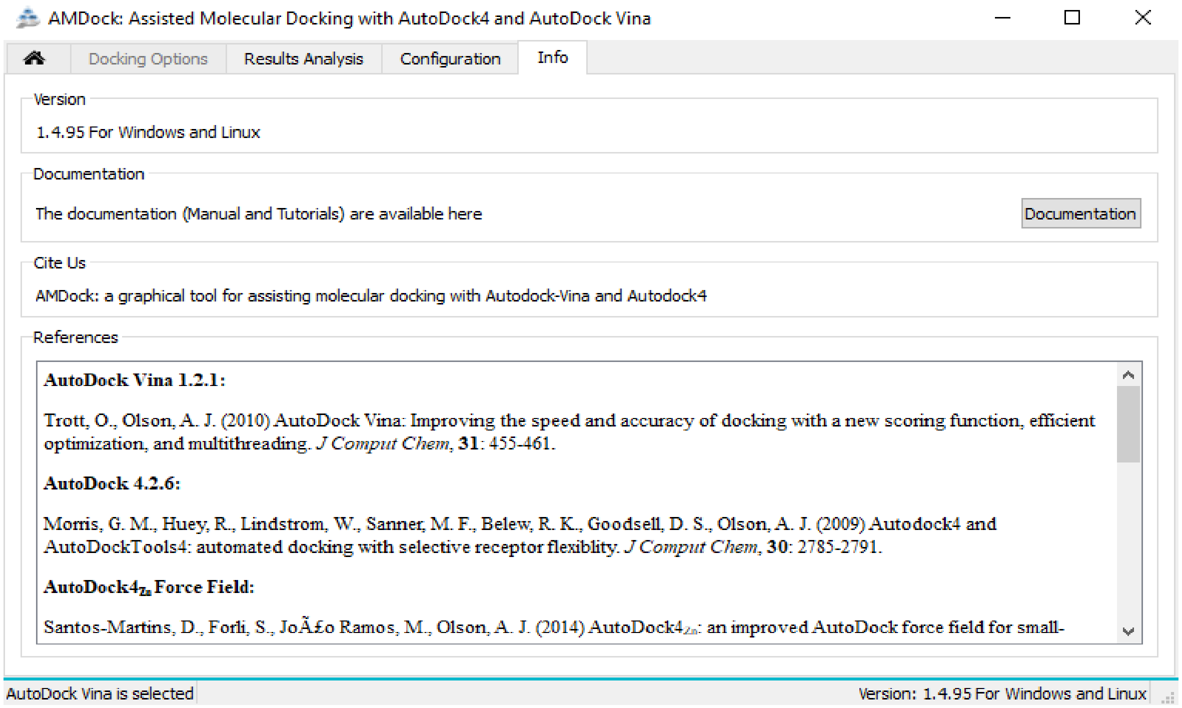
Protonation

□ Ligand Protonation
Program: ● OpenBabel ○ AutoDockTools

Version: 1.4.95 For Windows and Linux

Figure 4. AMDock Configuration Tab.

The info tab contains all the references to the documentation and original publications of the programs and algorithms included in AMDock.



AMDock: Assisted Molecular Docking with AutoDock4 and AutoDock Vina
Docking Options
Results Analysis
Configuration
Info
Documentation

Version

1.4.95 For Windows and Linux

Documentation

The documentation (Manual and Tutorials) are available here

Cite Us

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

References

AutoDock Vina 1.2.1:

Trott, O., Olson, A. J. (2010) AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J Comput Chem*, **31**: 455-461.

AutoDock 4.2.6:

Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S., Olson, A. J. (2009) Autodock4 and AutoDockTools4: automated docking with selective receptor flexiblity. *J Comput Chem*, **30**: 2785-2791.

AutoDock4$_{Zn}$ Force Field:

Santos-Martins, D., Forli, S., JoÃ£o Ramos, M., Olson, A. J. (2014) AutoDock4$_{Zn}$: an improved AutoDock force field for small-

AutoDock Vina is selected

Version: 1.4.95 For Windows and Linux

Figure 5. AMDock Info Tab.

AMDock

Tutorials

4. TUTORIALS

4.1 Before beginning…

Here we describe how AMDock can be used to predict the protein-ligand complex structure from the coordinates of receptor and ligand separately. We are interested in showing only the AMDock procedure. Thus, we avoid discussions about the advantages and disadvantages of docking programs. We used one or another program for docking purposes at our convenience as well as structure files used in previous tutorials. All information about these programs can be reviewed in their respective manuals and original publications. Reading "AMDock WORKFLOW and GUI" section is encouraged before starting the tutorials.

4.2 Simple docking

This procedure is focused on predicting the protein-ligand complex structure using different methods to define the search space. We chose 4 study cases and each of the methods aims to define the search space in AMDock (*i.e.* "Custom Box", "Box centered on Hetero", "Box centered on Residue(s)", and "Automatic Mode"**.**) will be used accordingly to predict the complex structure.

4.2.1 "Custom box" with Autodock4Zn

Some applications and plugins have been designed for helping researchers in performing molecular docking run under a uniform and user-friendly graphical interface. These applications have been mainly focused on Autodock Vina, Autodock or both. To our knowledge, the Autodock4Zn force field has not been incorporated in any of those applications. Therefore, we present a tutorial for Autodock4Zn in AMDock, although this procedure is valid for the rest of the docking engines included in AMDock.

"Custom Box" is the simplest option since the box specifications (*i.e.* location and dimensions) are defined by the user. This tutorial is based on a previous one available here. In the Autodock4Zn tutorial the authors selected as a study case the human farnesyltransferase (hFTase) complexed with the inhibitor L-778,123 (PDB ID: 1S63).

Receptor (hFTase.pdbqt) and ligand (L-778_123.pdb) files are available in **Installation_path/Doc/Tutorials/I_Simple_Docking/Custom_Box** folder. The crystal structure of the complex (complex.pdb) is available as well for comparison purposes. Let's begin!

1. Open AMDock program

2. Select **Autodock4Zn**

**hFTase is a Zn-metalloenzyme. Autodock4Zn includes a specialized potential describing the interactions for zinc-coordinating ligands.*

3. Set the Project Name (default **Docking_Project**)

4. Set the Location for the Project and push the button "**Create Project**"

5. Check that "**Simple Docking**" checkbox is selected (It is checked by default)

6. Choose the receptor file (**Installation_path/Doc/Tutorials/I_Simple_Docking/ Custom_Box/hFTase.pdbqt**)

**In this case, we are using a .pdbqt file for the receptor. That means AMDock will no perform any change in your file if you use a .pdbqt file. The receptor file we are using contains a "ligand" which is important for the receptor function and the ligand binding. You can check later what happens if this cofactor is removed from the structure.*

7. Choose the ligand file (**Installation_path/Doc/Tutorials/ I_Simple_Docking/ Custom_Box/L-778_123.pdb**)

***** *The ligand structure already contains the hydrogen atoms. That way, AMDock will do only the merge of the non-polar hydrogens. If the ligand structure doesn't contain any hydrogen, you can use OpenBabel or AutodockTools for adding the hydrogens. To do so, go to the Configuration tab, check the box "Ligand Protonation" and choose the method for ligand protonation*

8. Press "**Prepare Input**" button

9. Then, for defining a search space, press "**Box**" and set the box center (x = 18; y = 134; z =-1) and size (15; 11; 19)

Autodock4Zn is based on Autodock4. In AutoDock4, the search space size is specified by "grid points" (0.375 Angstroms). In the Autodock4Zn tutorial, the box sizes are 40, 30, and 50 npts. The AutoDock Vina box sizes are given in Angstroms instead. Because both programs are included in the AMDock environment, we decided to standardize the space size definition.* ***AMDock defines the search space size in Angstroms.* *Thus, a box with npts = 40, 30, 50 in Autodock4Zn tutorial, corresponds to a box with the following dimensions in Å units: 15Å x 11Å x 19Å. These dimensions have been calculated by multiplying npts x 0.375Å.*

<u>Note:</u> *A warning message will pop up since AMDock estimates automatically an "ideal" size for the box based on ligand's radius of gyration (see* ***Optimal Box Size 1.1****). We can dismiss the message in this case since we know in advance the box dimensions.*

10. Press "**Define Search Space**" button

** Once the search space is defined, you can see the box in the receptor by pressing the button "Show in PyMOL"*

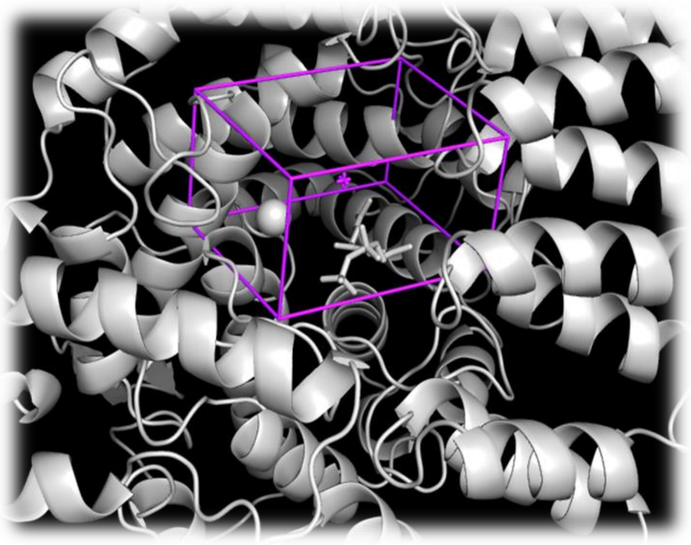


Figure 6. Custom Box. The protein (farnesyltransferase) is represented in cartoon and gray color. Note that the Zn atom and the cofactor are also represented, in sphere and sticks respectively. The box is shown in cylinders and magenta color.

Note: If you want to modify the box, you can use the AMDdock plugin in PyMOL. More details about this procedure here.

11. Press the "**Run**" button

12. When docking run ends, the "**Results Analysis**" tab will appear automatically. There, you will observe a summarizing table with Binding Energies, Estimated Ki values, and Ligand Efficiencies.

* *The number of poses and energies can vary since this is docking engine dependent.*

13. In the results table, just the first pose is selected. Press the button "**Show in PyMOL**" in the bottom left corner to visualize the complex.

* *If you want to see all the predicted poses, just select all the poses in the result table and press "Show in PyMOL"*

14. The crystal structure can be superposed on the predicted structure. Press "Open" in the upper left corner in PyMOL and find the X-ray structure (**Installation_path/Doc/Tutorials/I_Simple_Docking/Custom_Box/complex.pdb**)

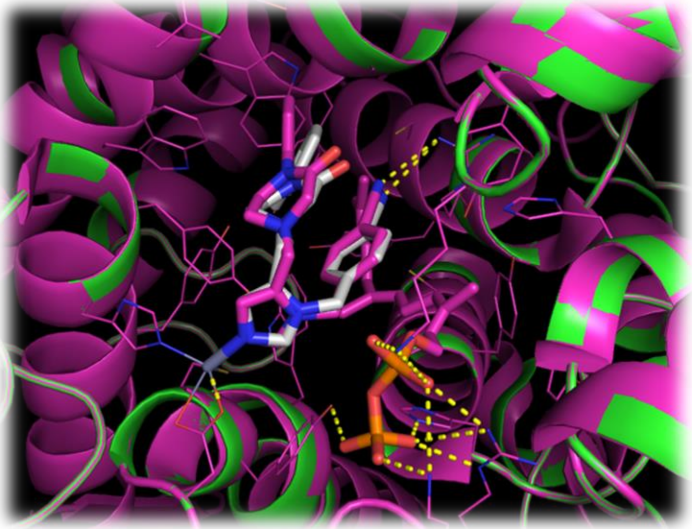


Figure 7. Superposition of the predicted complex from custom box using AutoDock4Zn with a reference complex (PDB ID: 1S63). Receptor is represented as cartoon while X-ray and predicted ligand binding pose are repreented as white and magenta sticks respectively.

15. A detailed report of the entire process can be seen on the right side of the screen, which allows for tracking all the results from different programs and algorithms. This log file can be saved in case an error occurred and sent to the developers.

4.2.2 “Box centered on Hetero” using Autodock Vina

When using this option, it is assumed the receptor structure contains a ligand that is going to be used for determining the box position. This is a useful way to define the box position in redocking experiments or when aiming at the prediction of the complex structure of a similar ligand, presumably with the same binding mode.

4.2.2.1 re-Docking experiment

This procedure is widely used for testing how accurately certain docking engine can predict the correct structure of a given complex of interest. In this case, we chose as a study case the lipid kinase Vps34 involved in vesicle trafficking and autophagy (PDB ID: 4UWH). In a standard re-docking experiment, the ligand coordinates should be extracted from the complex and randomized.

Receptor (4uwh.pdb) and ligand (ligand.pdb) files are available at **Installation_path/Doc/Tutorials/I_Simple_Docking/Box_Hetero/2.1.re-Docking** folder. The crystal structure of the complex (complex.pdb) is available as well for comparison purposes.

1. Open AMDock program or just create a new project if you are at “**Results Analysis**” tab

2. Select Autodock Vina

3. Set the Project Name (default **Docking_Project**)

4. Set the Location for Project and push the button “**Create Project**”

5. Check that “**Simple Docking**” checkbox is selected (It is checked by default)

6. Choose the receptor file(**Installation_path/Doc/Tutorials/I_Simple_Docking/ Box_Hetero/2.1.re-Docking/4uwh.pdb**)

7. Choose the ligand file (**Installation_path/Doc/Tutorials/I_Simple_Docking/Box_Hetero/2.1.re-Docking/ligand.pdb**)

** Before going to the next step, go to the configuration tab and check "Ligand Protonation" with OpenBabel*

8. Press the "**Prepare Input**" button

9. Then, for defining a search space, pick "**Center on Hetero**" and select the ligand "**A:JXM:1876**"

10. Press "**Define Search Space**" button

** Once the search space is defined, you can see the box in the receptor by pressing the button "Show in PyMOL"*

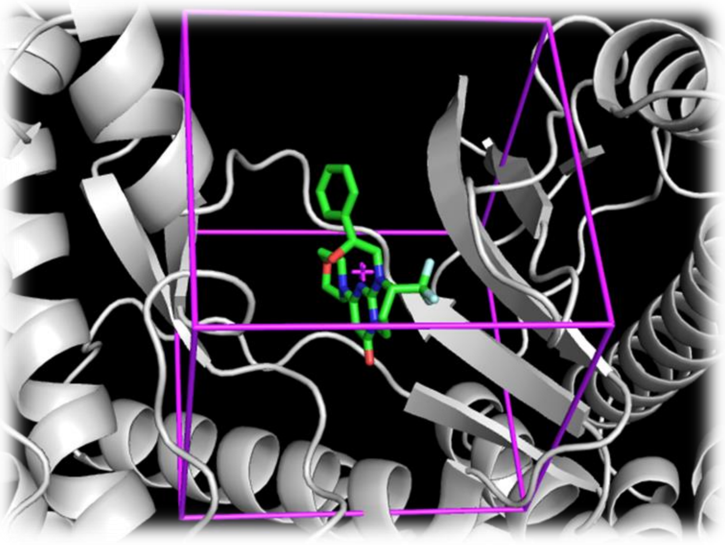


Figure 8. Box Centered on Hetero. The protein (Vps34) is represented in cartoon and gray color and previous ligand in green sticks. The box is shown in cylinders and magenta color.

<u>Note</u>: If you want to modify the box, you can use the AMDdock plugin in PyMOL. More details about this procedure here.

11. Press the "**Run**" button

12. When docking run ends, the “**Results Analysis**” tab will appear automatically. There, you will observe a summarizing table with Binding Energies, Estimated Ki values, and Ligand Efficiencies.

** The number of poses and energies can vary since this is docking engine dependent*

13. In the results table, just the first pose is selected. Press the button “**Show in PyMOL**” in the bottom left corner to visualize the complex.

** If you want to see all the predicted poses, just select all the poses in the result table and press “Show in PyMOL”*

14. The crystal structure can be superposed on the predicted structure. Press “Open” in the upper left corner in PyMOL and find the X-ray structure (**Installation_path/Doc/Tutorials/I_Simple_Docking/Box_Hetero/2.1.re-Docking/complex.pdb**)

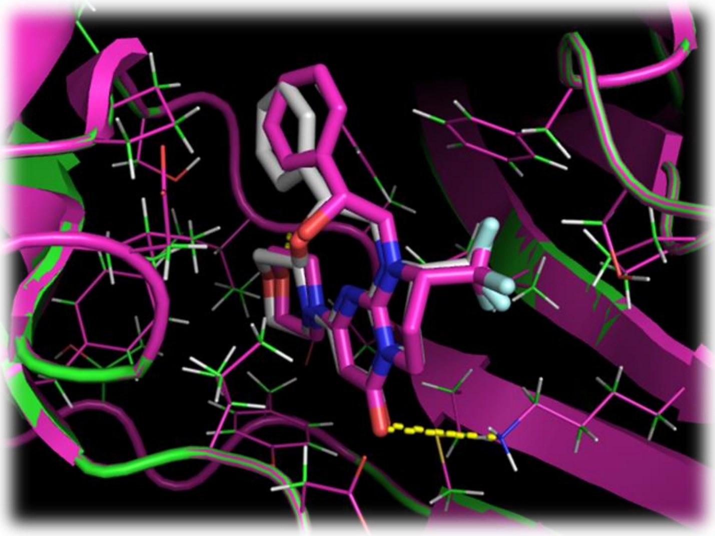


Figure 9. Superposition of predicted complex from box Centered on Hetero with a reference complex (PDB ID: 4UWH). Receptor is represented as cartoon while X-ray and predicted ligand binding pose are repreented as white and magenta sticks respectively.

15. A detailed report of the entire process can be seen on the right side of the screen, which allows for tracking all the results from different programs and algorithms. This log file can be saved in case an error occurred and sent to the developers.

4.2.2.2 Docking a similar ligand

This procedure is used when a structure of a complex with a ligand similar to that of your interest is available. In this case, we chose as a study case the lipid kinase Vps34 (PDB ID: 4UWL). This ligand is similar to that of 4UWH, so we can use it as receptor.

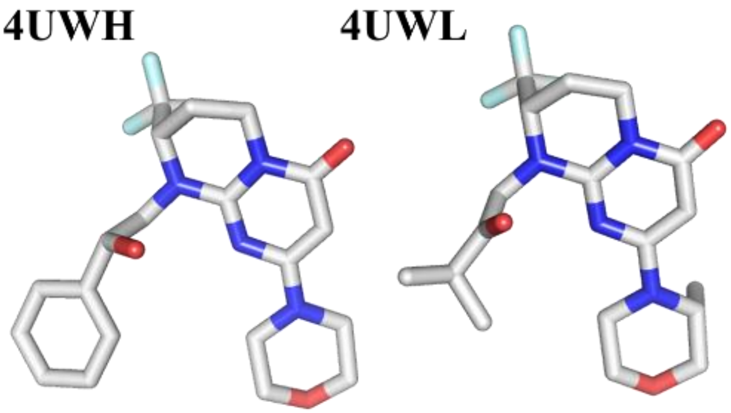


Figure 10. Vps34 inhibitors with similar structure.

Receptor (4uwh.pdb) and ligand (ligand.pdb) files are available at **Installation_path/Doc/Tutorials/I_Simple_Docking/Box_Hetero/2.2.Similar_Ligand** folder. The crystal structure of the complex (complex.pdb) is available as well for comparison purposes.

1. Open AMDock program or just create a new project if you are at "**Results Analysis**" tab
2. Select Autodock Vina
3. Set the Project Name (default **Docking_Project**)
4. Set the Location for Project and push the button "**Create Project**"
5. Check that "**Simple Docking**" checkbox is selected (It is checked by default)

6. Choose the receptor file (**Installation_path/Doc/Tutorials/I_Simple_Docking/ Box_Hetero/2.2.Similar_Ligand/4uwh.pdb**)

7. Choose the ligand file (**Installation_path/Doc/Tutorials/I_Simple_Docking/ Box_Hetero/2.2.Similar_Ligand /ligand.pdb**)

** Before going to the next step, go to the configuration tab and check "Ligand Protonation" with OpenBabel*

8. Press the "**Prepare Input**" button

9. Then, for defining a search space, pick "**Center on Hetero**" and select the ligand "**A:JXM:1876**"

10. Press "**Define Search Space**" button

** Once the search space is defined, you can see the box in the receptor by pressing the button "Show in PyMOL"(Showed in **Figure 8)***

Note: If you want to modify the box, you can use the AMDdock plugin in PyMOL. More details about this procedure here.

11. Press the "**Run**" button

12. When docking run ends, the "**Results Analysis**" tab will appear automatically. There, you will observe a summarizing table with Binding Energies, Estimated Ki values, and Ligand Efficiencies.

** The number of poses and energies can vary since this is docking engine dependent*

13. In the results table, just the first pose is selected. Press the button "**Show in PyMOL**" in the bottom left corner to visualize the complex.

** If you want to see all the predicted poses, just select all the poses in the result table and press "Show in PyMOL"*

14. The crystal structure can be superposed on the predicted structure. Press "Open" in the upper left corner in PyMOL and find the X-ray structure

(**Installation_path/Doc/Tutorials/I_Simple_Docking/Box_Hetero/2.2.Similar_Ligand/complex.pdb**)

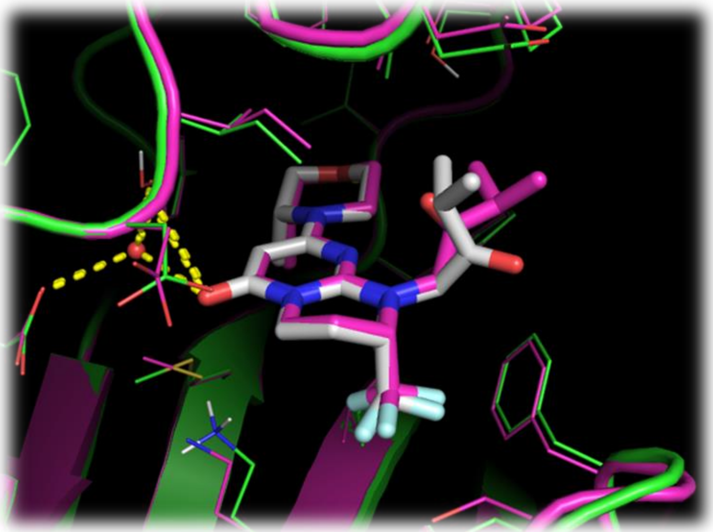


Figure 11. Superposition of predicted complex from box Centered on Hetero with a reference complex (PDB ID: 4UWL). Receptor is represented as cartoon while X-ray and predicted ligand binding pose are repreented as white and magenta sticks respectively.

15. A detailed report of the entire process can be seen on the right side of the screen, which allows for tracking all the results from different programs and algorithms. This log file can be saved in case an error occurred and sent to the developers.

4.2.3 "Centered on residue(s)" using AutoDock Vina

This option is recommended when residues belonging to the binding site are known. This information usually comes from mutagenesis experiments or comparing protein belonging to the same family.

Receptor (4uwh.pdb) and ligand (ligand.pdb) files are available in **Installation_path/Doc/Tutorials/I_Simple_Docking/3.Box_Residues** folder. The crystal structure of the complex (complex.pdb) is available as well for comparison purposes.

1. Open AMDock program or Create a new project in "**Results Analysis**"

2. Select Autodock Vina

3. Set the Project Name (default **Docking_Project**)

4. Set the Location for Project and push the button "**Create Project**"

5. Check that "**Simple Docking**" checkbox is selected (It is checked by default)

6. Choose the receptor file

(**Installation_path/Doc/Tutorials/I_Simple_Docking/3.Box_Residues /4uwh.pdb**)

7. Choose the ligand file

(**Installation_path/Doc/Tutorials/I_Simple_Docking/3.Box_Residues/ligand.pdb**)

** Before going to the next step, go to the configuration tab and check "Ligand Protonation" with OpenBabel*

8. Press the "**Prepare Input**" button

9. Then, for defining a search space, pick "**Center on Residue(s)**". A text box will appear where residues should be specified by following this notation:

CHAIN:RESIDUE1:NUMBER; CHAIN:RESIDUE2:NUMBER…

AMDock checks automatically if those residues exist in the receptor.

In this tutorial, the following residues were chosen:

A:ILE:634; A:TYR:670; A:PHE:684; A:PHE:758; A:ILE:760

10. Press "**Define Search Space**" button

** Once the search space is defined, you can see the box by pressing the button "Show in PyMOL". In this case, the receptor is represented as cartoon, binding site residues in sticks and the object generated in AutoLigand as mesh representation.*

<u>Note</u>*: If you want to modify the box, you can use the AMDdock plugin in PyMOL. More details about this procedure here.*

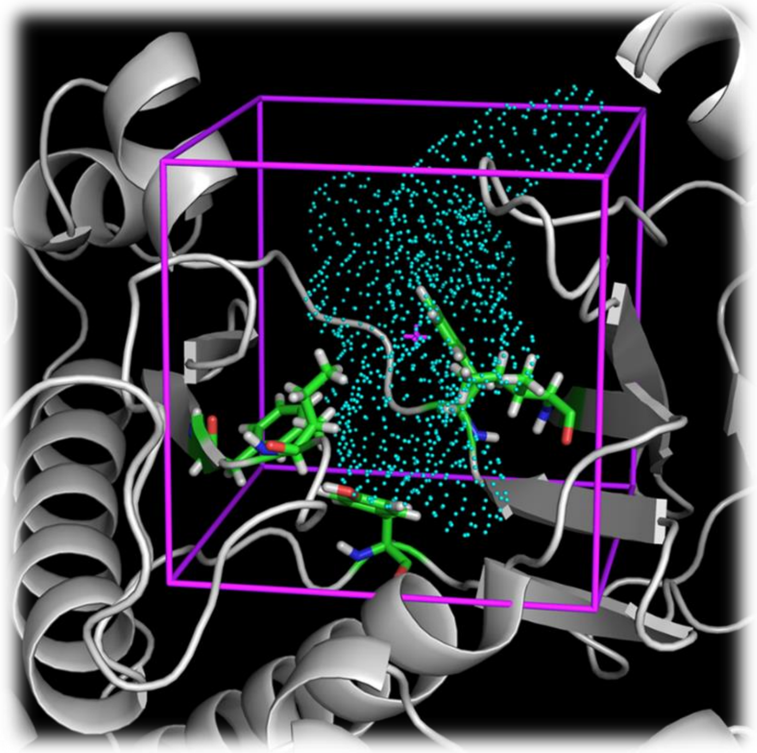


Figure 12. Box Centered on Residues. The protein (Vps34) is represented in gray cartoon, the selected residues in green sticks and the object generated by AutoLigand in cyan dots. The box is shown in magenta cylinders

11. Press the "**Run**" button

12. When docking run ends, the "**Results Analysis**" tab will appear automatically. There, you will observe a summarizing table with Binding Energies, Estimated Ki values, and Ligand Efficiencies.

* *The number of poses and energies can vary since this is docking engine dependent*

13. In the results table, just the first pose is selected. Press the button "**Show in PyMOL**" in the bottom left corner to visualize the complex.

* *If you want to see all the predicted poses, just select all the poses in the result table and press "Show in PyMOL"*

14. The crystal structure can be superposed on the predicted structure. Press "Open" in the upper left corner in PyMOL and find the X-ray structure (**Installation_path/Doc/Tutorials/I_Simple_Docking/3.Box_Residues/complex.pdb**)

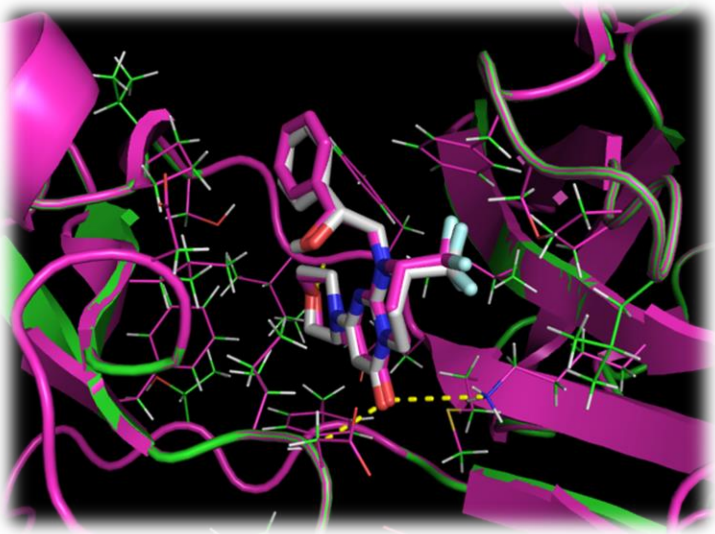


Figure 13. Superpsition of predicted complex from box Centered on Residue(s) with a reference complex (PDB ID: 4UWH). Receptor is represented as cartoon while X-ray and predicted ligand binding pose are repreented as white and magenta sticks respectively.

15. A detailed report of the entire process can be seen on the right side of the screen, which allows for tracking all the results from different programs and algorithms. This log file can be saved in case an error occurred and sent to the developers.

4.2.4 “Automatic” mode using AutoDock Vina

This option (also known as “blind docking”) is recommendable only when you have no idea where the binding site is.

Receptor (4uwh.pdb) and ligand (ligand.pdb) files are available in **Installation_path/Doc/Tutorials/I_Simple_Docking/4.Box_Auto** folder. The crystal structure of the complex (complex.pdb) is available as well for comparison purposes.

1. Open AMDock program or Create a new project in “**Results Analysis**”
2. Select Autodock Vina
3. Set the Project Name (default **Docking_Project**)
4. Set the Location for Project and push the button "**Create Project**"

5. Check that “**Simple Docking**” checkbox is selected (It is checked by default)

6. Choose the receptor file

(**Installation_path/Doc/Tutorials/ I_Simple_Docking/4.Box_Auto/4uwh.pdb**)

7. Choose the ligand file

(**Installation_path/Doc/Tutorials/ I_Simple_Docking/4.Box_Auto/ligand.pdb**)

** Before going to the next step, go to the configuration tab and check “Ligand Protonation” with OpenBabel*

8. Press “**Prepare Input**” button

9. Then, for defining a search space, choose “**Automatic**”.

10. Press “**Define Search Space**” button (and go for a coffee)

11. Once AutoLigand is done, a table showing both the volume and the EPV (Energy per Volume) for every predicted binding site will appear.

** You can see the box by pressing the button “Show in PyMOL”. In this case, the receptor is represented as cartoon, binding site in transparent surface and the objects generated in AutoLigand as sticks. Only one box is showed for better clarity*

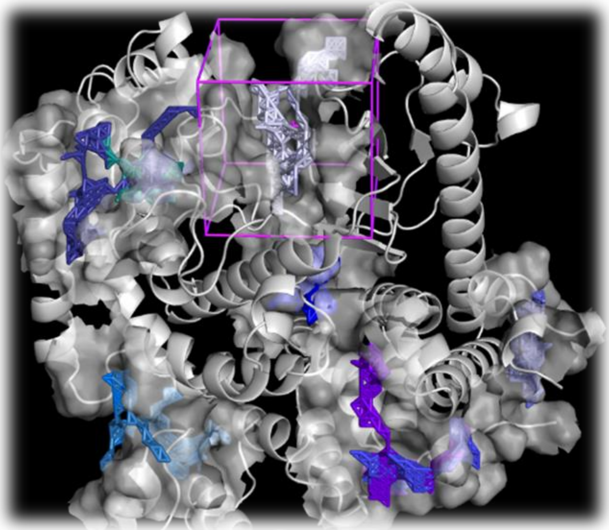


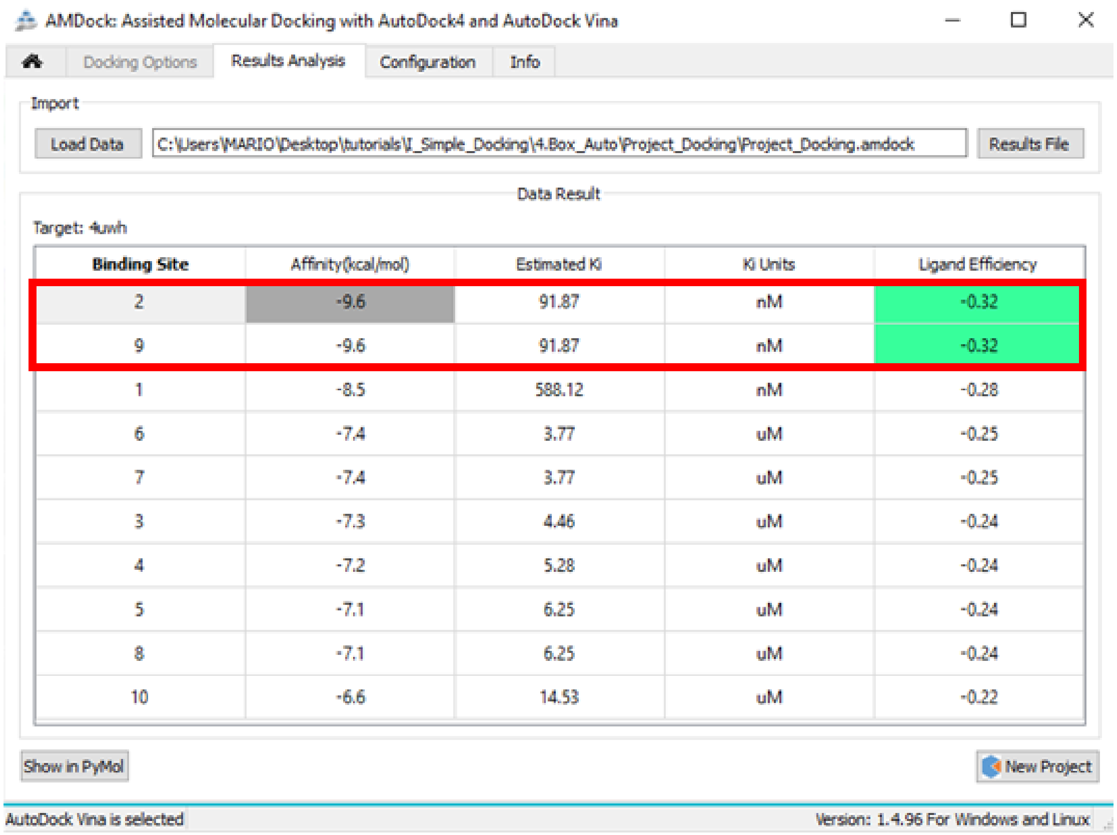
Figure 14. Automatic mode Search Space determination. The protein (Vps34) is represented gray cartoon. Each binding site predicted by AutoLigand is repsented as transparent surface and also contains an object represented as sticks. The box is shown in cylinders and magenta color. Only one box is shown for better clarity.

12. Press "**Run**" button (and go for a coffee again)

** An independent docking run will be performed in every site predicted by AutoLigand*

13. When docking run ends, "**Results Analysis**" tab will appear automatically. There, you will observe a summarizing table with Binding Energies, Estimated Ki values, and Ligand Efficiencies.

** Each of the rows in the final table corresponds to the better pose predicted for that binding site.*



C:\Users\MARIO\Desktop\tutorials\I_Simple_Docking\4.Box_Auto\Project_Docking\Project_Docking.amdock

Target: 4uwh

Binding Site	Affinity(kcal/mol)	Estimated Ki	Ki Units	Ligand Efficiency
2	-9.6	91.87	nM	-0.32
9	-9.6	91.87	nM	-0.32
1	-8.5	588.12	nM	-0.28
6	-7.4	3.77	uM	-0.25
7	-7.4	3.77	uM	-0.25
3	-7.3	4.46	uM	-0.24
4	-7.2	5.28	uM	-0.24
5	-7.1	6.25	uM	-0.24
8	-7.1	6.25	uM	-0.24
10	-6.6	14.53	uM	-0.22

AutoDock Vina is selected

Version: 1.4.96 For Windows and Linux

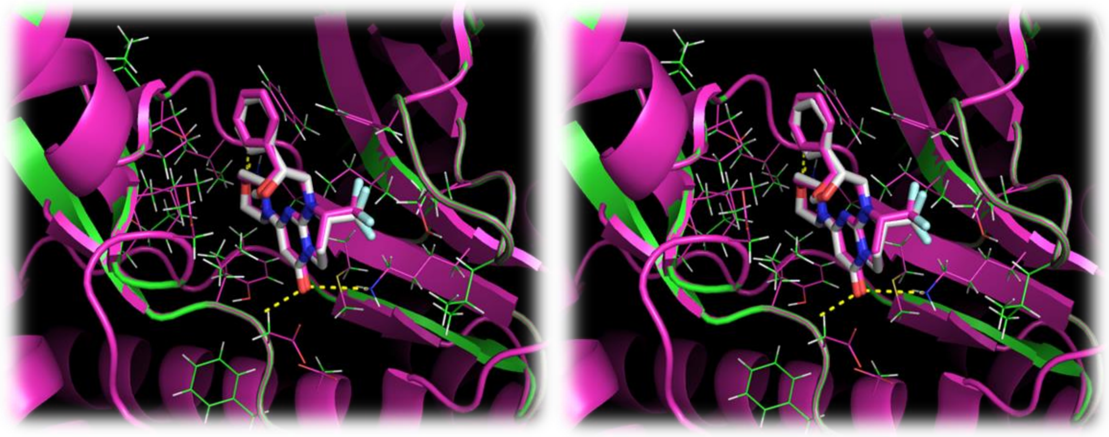


Figure 15. Results from Automatic mode selection. Up: The table show the sorted list of affinity for each predicted binding site. Down: Superpsition of predicted complex from predicted site 2 (left) and 9 (rigth) with a reference complex (PDB ID: 4UWH). Receptor is represented as cartoon (green) while X-ray and predicted ligand binding pose are represented as white and magenta sticks respectively.

13. In the results table, just the first pose is selected. Press the button "**Show in PyMOL**" in the bottom left corner to visualize the complex.

** If you want to see all the predicted poses, just select all the poses in the result table and press "Show in PyMOL"*

14. The crystal structure can be superposed on the predicted structure. Press "Open" in the upper left corner in PyMOL and find the X-ray structure (**Installation_path/Doc/Tutorials/I_Simple_Docking/4.Box_Auto/complex.pdb**)

15. A detailed report of the entire process can be seen on the right side of the screen, which allows for tracking all the results from different programs and algorithms. This log file can be saved in case an error occurred and sent to the developers.

4.3 Off-target docking

For this tutorial, we selected as a study case the PI3K system, as in the previous cases. We intend to demonstrate how AMDock allows for a comparison between the two binding modes and predicted affinities for both complexes, *i.e.* target and off-target. The binding mode of the ligand SAR405 will be predicted for both the target (PDB ID:4UWH (Vps34)) and the off-target (PDB ID: 3APF (PI3Kγ)). It is known this inhibitor binds Vps34 preferentially with IC50=1.2nM, while IC50 for the other isoforms is > 10 000 nM.

The receptors (4uwh.pdb and 3apf.pdb) and ligand (sar405.pdb) files are available in (**Installation_path/Doc/Tutorials/II_Off-Target_Docking**) folder. The structure of Vps34 complexed with SAR405 (complex.pdb, PDB ID:4OYS) is available as well for comparison purposes.

1. Open AMDock program

2. Select Autodock Vina

3. Set the Project Name (default **Docking_Project**)

4. Set the Location for Project and push the button "**Create Project**"

5. Check that "**Off-Target Docking**" checkbox is selected

** Automatically, a new text box for introducing off-target receptor will be available and as well as a new button "**Align Proteins**" which allows for aligning both receptors*

6. Choose the target receptor file

(**Installation_path/Doc/Tutorials/II_Off-Target_Docking/4uwh.pdb**)

7. Choose the off-target receptor file

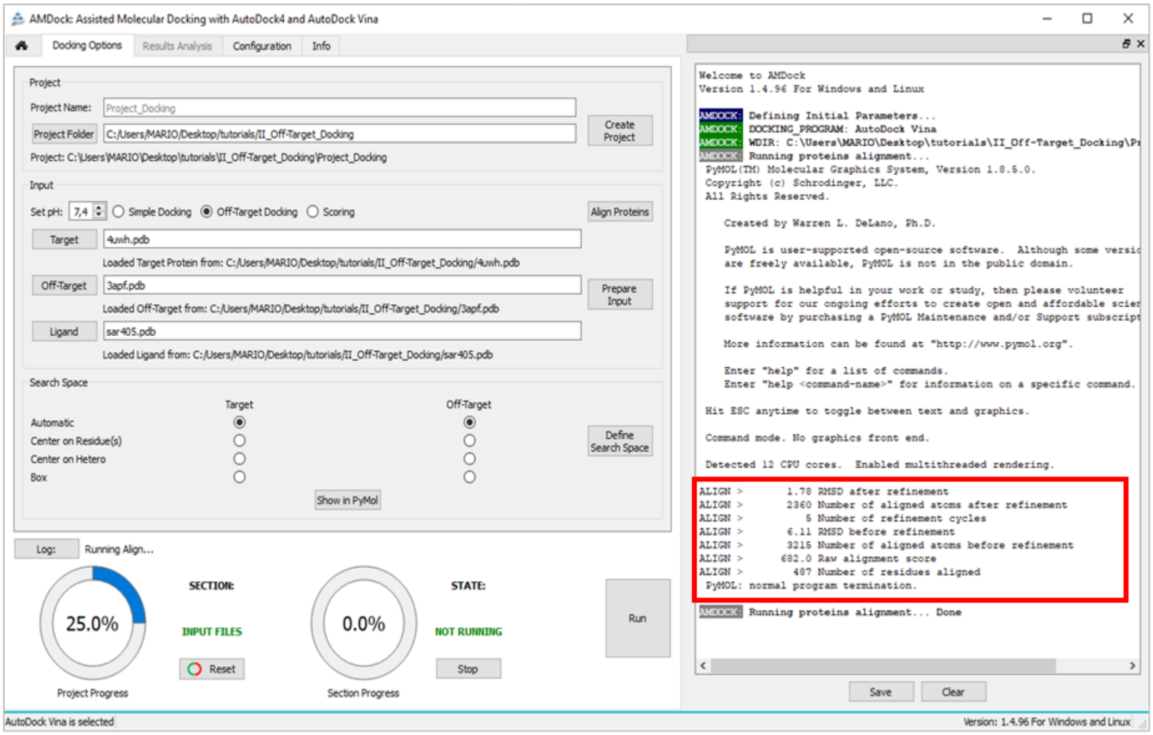
(**Installation_path/Doc/Tutorials/II_Off-Target_Docking/5ncy.pdb**)

8. Choose the ligand file

(**Installation_path/Doc/Tutorials/II_Off-Target_Docking/sar405.pdb**)

9. Press "Align Proteins"

** This will be critical to define a similar search space in both proteins. The method used to align the proteins is quite simple and is performed by PyMOL. That way, it is mandatory to check the quality of the alignment. In general, both proteins should be similar, hence we should expect a low value of RMSD (you can check the log in the left side of the screen nin the RMSD value or the alignment).*



Project Name: Project_Docking

Project Folder: C:/Users/MARIO/Desktop/tutorials/II_Off-Target_Docking

Project: C:\Users\MARIO\Desktop\tutorials\II_Off-Target_Docking\Project_Docking

Set pH: 7,4 ○ Simple Docking ● Off-Target Docking ○ Scoring

Target: 4uwh.pdb

Loaded Target Protein from: C:/Users/MARIO/Desktop/tutorials/II_Off-Target_Docking/4uwh.pdb

Off-Target: 3apf.pdb

Loaded Off-Target from: C:/Users/MARIO/Desktop/tutorials/II_Off-Target_Docking/3apf.pdb

Ligand: sar405.pdb

Loaded Ligand from: C:/Users/MARIO/Desktop/tutorials/II_Off-Target_Docking/sar405.pdb

	Target	Off-Target
Automatic	●	●
Center on Residue(s)	○	○
Center on Hetero	○	○
Box	○	○

Log: Running Align...

25.0% INPUT FILES

0.0% NOT RUNNING

```
Welcome to AMDock
Version 1.4.96 For Windows and Linux

AMDOCK: Defining Initial Parameters...
AMDOCK: DOCKING_PROGRAM: AutoDock Vina
AMDOCK: WDIR: C:\Users\MARIO\Desktop\tutorials\II_Off-Target_Docking\Pr
AMDOCK: Running proteins alignment...
 PyMOL(TM) Molecular Graphics System, Version 1.8.5.0.
 Copyright (c) Schrodinger, LLC.
 All Rights Reserved.

    Created by Warren L. DeLano, Ph.D.

    PyMOL is user-supported open-source software.  Although some versic
    are freely available, PyMOL is not in the public domain.

    If PyMOL is helpful in your work or study, then please volunteer
    support for our ongoing efforts to create open and affordable scier
    software by purchasing a PyMOL Maintenance and/or Support subscript

    More information can be found at "http://www.pymol.org".

    Enter "help" for a list of commands.
    Enter "help <command-name>" for information on a specific command.

 Hit ESC anytime to toggle between text and graphics.

 Command mode. No graphics front end.

 Detected 12 CPU cores.  Enabled multithreaded rendering.
ALIGN >       1.78 RMSD after refinement
ALIGN >       2360 Number of aligned atoms after refinement
ALIGN >          5 Number of refinement cycles
ALIGN >       6.11 RMSD before refinement
ALIGN >       3215 Number of aligned atoms before refinement
ALIGN >      682.0 Raw alignment score
ALIGN >        487 Number of residues aligned
 PyMOL: normal program termination.
AMDOCK: Running proteins alignment... Done
```

AutoDock Vina is selected

Version: 1.4.96 For Windows and Linux

Figure 16. Proteins alignment from "Off-Target Docking". The red box highlights the alignment results from PyMOL.

10. Press the “Prepare Input” button

11. Then, for defining a search space, pick “**Center on Hetero**” and select the ligand “**A:JXM:1876**” for target receptor and “**A:BMW:1103**” for the off-target.

12. Press “**Define Search Space**” button

* *Once the search space is defined, you can see the box by pressing the button “Show in PyMOL”. In this case, both receptors are aligned and represented as cartoon with ligands in sticks.*

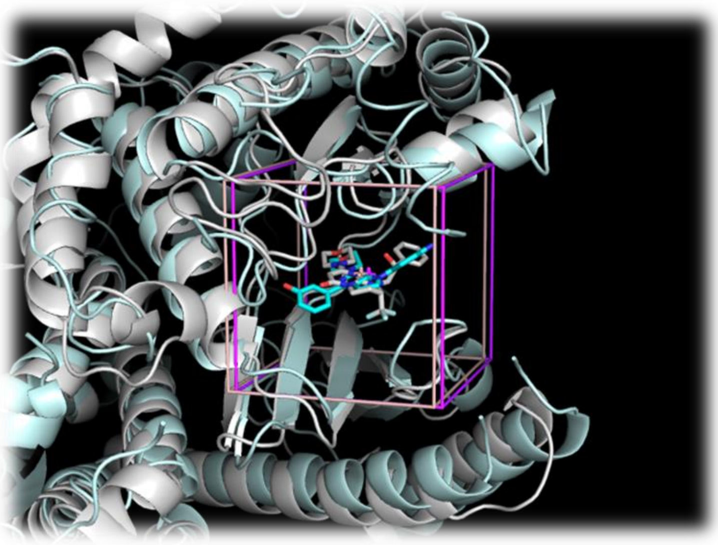
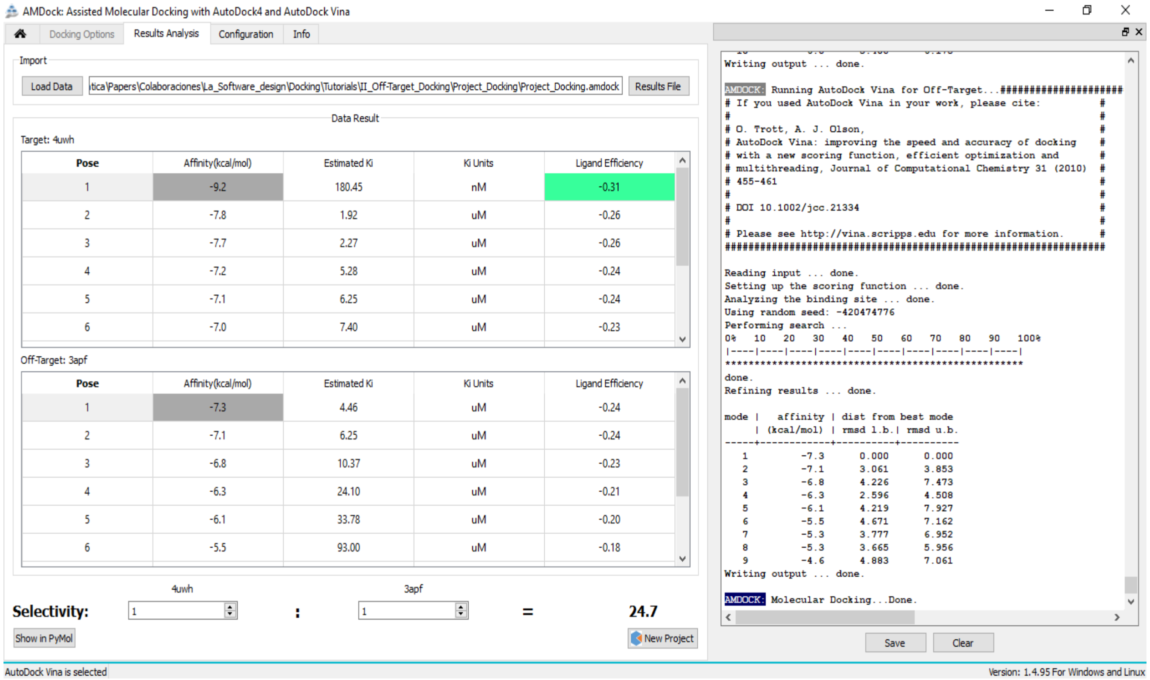


Figure 17. Search space definition in Target (Vps34) and Off-Target (PI3Kγ). Both boxes are Centered on Hetero.

14. Press “Run” button

15. When docking run ends, the “Results Analysis” tab will appear automatically. There, you will observe a summarizing table with Binding Energies, Estimated Ki values, and Ligand Efficiencies. Also, the selectivity between the target and off-target receptors is shown.


AMDock: Assisted Molecular Docking with AutoDock4 and AutoDock Vina
Docking Options
Results Analysis
Configuration
Info
Import
Load Data
Results File
Data Result
Show in PyMol
New Project
Save
Clear
AutoDock Vina is selected
Version: 1.4.95 For Windows and Linux

tica\Papers\Colaboraciones\La_Software_design\Docking\Tutorials\II_Off-Target_Docking\Project_Docking\Project_Docking.amdock

Target: 4uwh

Pose	Affinity(kcal/mol)	Estimated Ki	Ki Units	Ligand Efficiency
1	-9.2	180.45	nM	-0.31
2	-7.8	1.92	uM	-0.26
3	-7.7	2.27	uM	-0.26
4	-7.2	5.28	uM	-0.24
5	-7.1	6.25	uM	-0.24
6	-7.0	7.40	uM	-0.23

Off-Target: 3apf

Pose	Affinity(kcal/mol)	Estimated Ki	Ki Units	Ligand Efficiency
1	-7.3	4.46	uM	-0.24
2	-7.1	6.25	uM	-0.24
3	-6.8	10.37	uM	-0.23
4	-6.3	24.10	uM	-0.21
5	-6.1	33.78	uM	-0.20
6	-5.5	93.00	uM	-0.18

Selectivity: 4uwh 1 : 3apf 1 = 24.7

```
Writing output ... done.

AMDOCK: Running AutoDock Vina for Off-Target...#####################
# If you used AutoDock Vina in your work, please cite:               #
#                                                                    #
# O. Trott, A. J. Olson,                                             #
# AutoDock Vina: improving the speed and accuracy of docking         #
# with a new scoring function, efficient optimization and            #
# multithreading, Journal of Computational Chemistry 31 (2010)       #
# 455-461                                                            #
#                                                                    #
# DOI 10.1002/jcc.21334                                              #
#                                                                    #
# Please see http://vina.scripps.edu for more information.           #
######################################################################

Reading input ... done.
Setting up the scoring function ... done.
Analyzing the binding site ... done.
Using random seed: -420474776
Performing search ...
0%   10   20   30   40   50   60   70   80   90   100%
|----|----|----|----|----|----|----|----|----|----|
***************************************************
done.
Refining results ... done.

mode |   affinity | dist from best mode
     | (kcal/mol) | rmsd l.b.| rmsd u.b.
-----+------------+----------+----------
   1         -7.3      0.000      0.000
   2         -7.1      3.061      3.853
   3         -6.8      4.226      7.473
   4         -6.3      2.596      4.508
   5         -6.1      4.219      7.927
   6         -5.5      4.671      7.162
   7         -5.3      3.777      6.952
   8         -5.3      3.665      5.956
   9         -4.6      4.883      7.061
Writing output ... done.

AMDOCK: Molecular Docking...Done.
```

Figure 19. Off-Target Docking results format of SAR405 againts Vps34 and PI3Kγ. The best pose of SAR405 in complex with Vps34 has an affinity of -9.2 kcal / mol, while with gamma it is -7.3 kcal / mol. According to the experimental data, SAR405 has a higher affinity (~ 25 times) for Vps34 than for PI3Kγ.

16. In each one of the tables, only the first pose is selected. Press the button "**Show in PyMOL**" in the bottom left corner to visualize the complex.

** If you want to see all the predicted poses, just select all the poses in the result table and press "Show in PyMOL". On the other hand, if you want only to visualize one of the complexes, use Ctrl+click to deselect the poses you are not interested in.*17. The crystal structure can be superposed on the predicted structure. Press "Open" in the upper left corner in PyMOL and find the X-ray structure (**Installation_path/Doc/Tutorials/II_Off-Target_Docking /complex.pdb**)

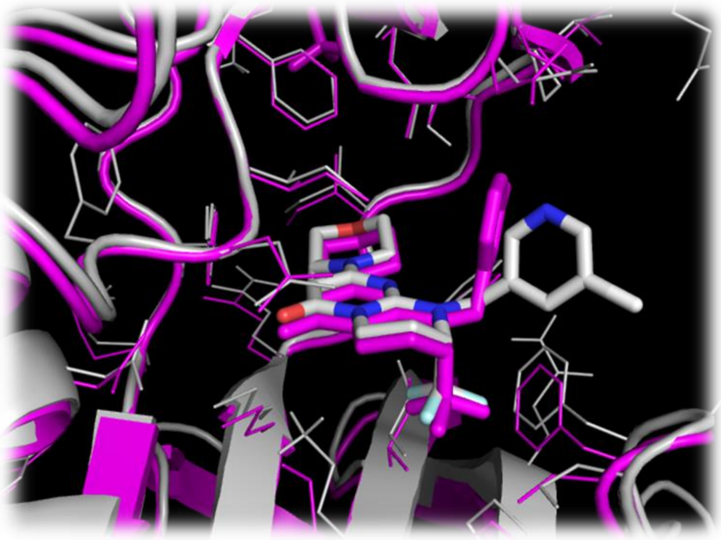


Figure 20. Superposition of predicted complex Vps34-SAR405 and a reference structure (PDB ID: 4UWL) (magenta).

18. A detailed report of the entire process can be seen on the right side of the screen, which allows for tracking all the results from different programs and algorithms. This log file can be saved in case an error occurred and sent to the developers.

4.4 Scoring

In this tutorial, we will study the effect of point mutations over ligand recognition by using the SCORING module of AMDock. We have selected as a study case the human farnesyl-transferase (hFTase), the same protein we used in the first tutorial. However, we will focus now on substrate (farnesyl diphosphate, FPP) recognition. The hFTase is a prenyl-transferase. It transfers a 15-carbon isoprenoid (donated by FPP) to the cysteine of a C-terminal motif of substrate proteins. Prenyl-transferases are generally highly selective for their cognate isoprenoid diphosphate substrates. Thus, hFTase binds FPP while the human geranylgeranyl-transferase type I (hGGTase-I, another prenyl-transferase) binds geranylgeranyl diphosphate (GGPP). The 20-carbon GGPP binds to hFTase with nanomolar affinity (but as a competitive inhibitor), while the 15-carbon FPP is a poor substrate for hGGTase-I.

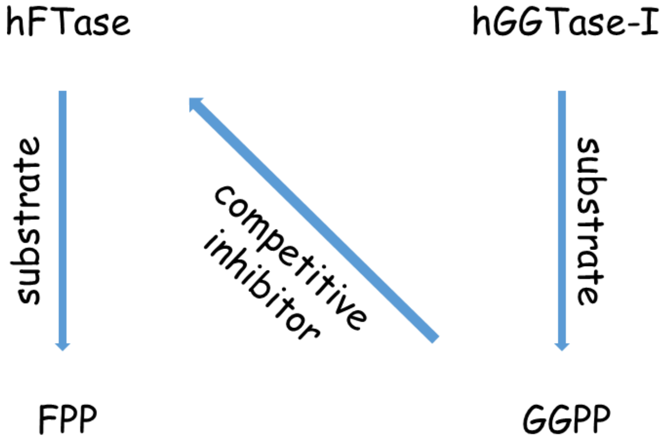


Figure 21. Diagram showing the relationship between enzymes and their cognates substrates and inhibitors.

Positioning GGPP-CaaX (GGPP + CaaX motif in proteins, where “a” is any amino acid) in the corresponding region of hFTase, leads to a steric clash with W102 and Y365 residues, suggesting that one or both of these residues may be key determinants of isoprenoid specificity. In elegant work, Terry *et al*.,1 re-engineered hFTase through site-directed mutagenesis to create an hFTase able to recognize GGPP substrate and geranylgeranylate its native CaaX substrates.

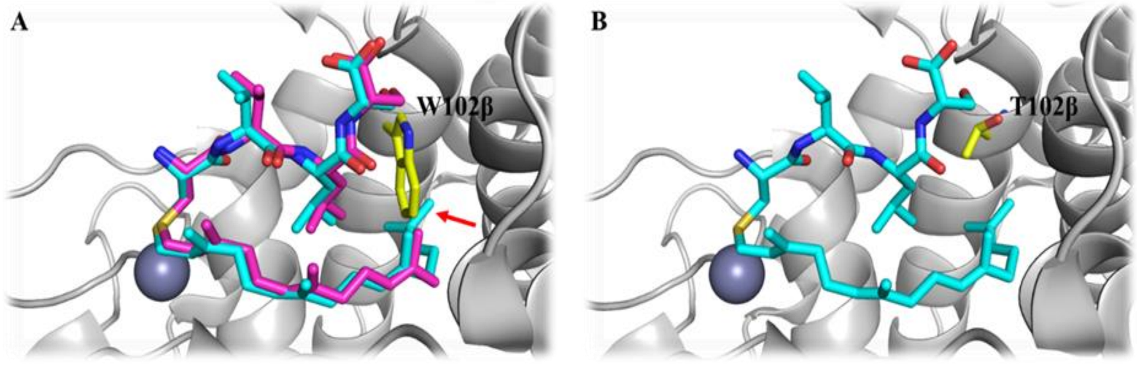


Figure 22. Farnesyl transferase ligand site. (**A**) Superposition of FPP-CVLS (magenta) and GGPP-CVLS (cyan), in wild-type hFTase active site. Red arrow indicates the steric clash between GGPP and W102β residue in the hFTase active site (**B**) GGPP-CVLS (cyan), in mutant hFTase W102βT active site. The mutation W102βT allows GGPP to fit into the active site of hFTase.

In this tutorial, we will try to predict the binding properties of both variants (the wild-type and the re-engineered) with AMDock SCORING module.

** Remember, the SCORING module will not generate a new complex from those of the ligand and receptor coordinates. Essentially, this module calculates the binding energy between two molecules belonging to a complex by using the scoring function of the selected docking engine. So, the ligands and receptors coordinates we will use as input were extracted from those of the complexes X-ray structures (PDB IDs: 2H6F and 2H6G respectively).*

Let’s begin with the wild-type first!

1. Open AMDock program
2. Select Autodock4Zn
3. Set the Project Name (default **Docking_Project**)
4. Set the Location for Project and push the button "**Create Project**"
5. Check that “**Scoring**” checkbox is selected
6. Choose the receptor file

   (**Installation_path/Doc/Tutorials/III_Scoring/hFTase.pdb**)
7. Choose the ligand file

   (**Installation_path/Doc/Tutorials/III_Scoring /FPP_CVLS.pdb**)
8. Press “**Prepare Input**” button
9. As we said above, this module will not generate a new complex; hence, it is not necessary to define a search space. Press “**Run**” button then.

The predicted binding constant for the farnesylated CaaX peptide is 28nM. Let’s continue with the mutant…

1. Open AMDock program
2. Select Autodock4Zn
3. Set the Project Name (default Docking_Project)

4. Set the Location for Project and push the button “**Create Project**”

5. Check that “**Scoring**” checkbox is selected

6. Choose the receptor file

(**Installation_path/Doc/Tutorials/III_Scoring /hFTase_W102T.pdb**)

7. Choose the ligand file

(**Installation_path/Doc/Tutorials/III_Scoring /GGPP_CVLS.pdb**)

8. Press “**Prepare Input**” button

9. Press “**Run**” button.

Here, the predicted binding constant for the geranylgeranylated CaaX peptide is 9.9nM which is, as expected, very similar to that of farnesylated CaaX peptide (28nM).

AMDock

Appendix

5. APPENDIX

5.1 Change Box specifications on AMDock PyMol plugin

Although several methods to determine the optimal search space have been implemented in AMDock, sometimes the user might want to modify the box specifications. To do so, a PyMOL plugin called “AMDock Box Builder” can be used. Once the user modifies one of the box specifications, AMDock will automatically activate the “user-defined box” method and the user will control the whole process.

* *It’s recommended to be cautious when modifying the box specifications from the “Automatic” method since AMDock will activate automatically the “user-defined box” method and no longer perform a docking run for every predicted binding site.*

Receptor (4uwh.pdb) and ligand (ligand.pdb) files are available in **Installation_path/Doc/Tutorials/IV_AMDock_Box_Builder** folder.

1. Open AMDock program
2. Select Autodock Vina
3. Set the Project Name (default **Docking_Project**)
4. Set the Location for Project and push the button "**Create Project**"
5. Check that “**Simple Docking**” checkbox is selected (It is checked by default)
6. Choose the receptor file

   (**Installation_path/Doc/Tutorials/IV_AMDock_Box_Builder/4uwh.pdb**)
7. Choose the ligand file

   (**Installation_path/Doc/Tutorials/IV_AMDock_Box_Builder/ligand.pdb**)

   * *Before going to the next step, go to the configuration tab and check “Ligand Protonation” with OpenBabel*
8. Press “**Prepare Input**” button

9. Then, for defining a search space, pick "**Center on Residue(s)**". A text box will appear where residues should be specified by following this notation:

CHAIN:RESIDUE1:NUMBER; CHAIN:RESIDUE2:NUMBER…

AMDock checks automatically if those residues exist in the receptor.

In this tutorial, the following residues were chosen:

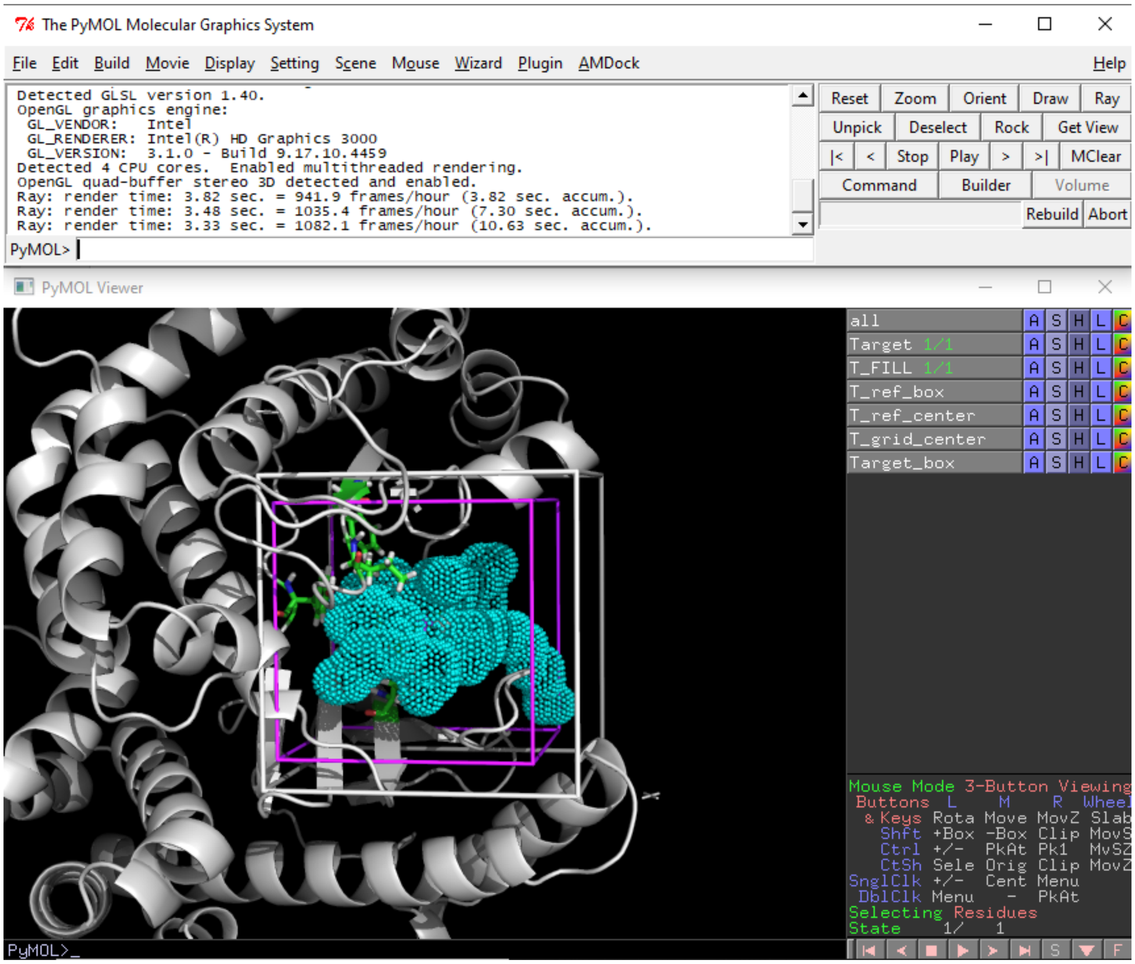
A:ILE:634; A:TYR:670; A:PHE:684; A:PHE:758; A:ILE:760

10. Press "**Define Search Space**" button

** Once the search space is defined, you can see the box by pressing the button "Show in PyMOL". In this case, the receptor is represented as cartoon, binding site residues in sticks and the object generated in AutoLigand as mesh representation.*

11. Find AMDock in the PyMOL menu bar and click "AMDock Box Builder"

12. In the "AMDock Box Builder" window, observe the initial coordinates of the box (20.1833; -13.8611; -23.2944) as well as the dimensions (19; 19; 19Å). Then, increase the size (**23; 23; 23Å**) and move it in the x-direction (**21.6000**; -13.8611; -23.2944) and you will see a new box (Target_box; white) along with the reference box (in magenta).

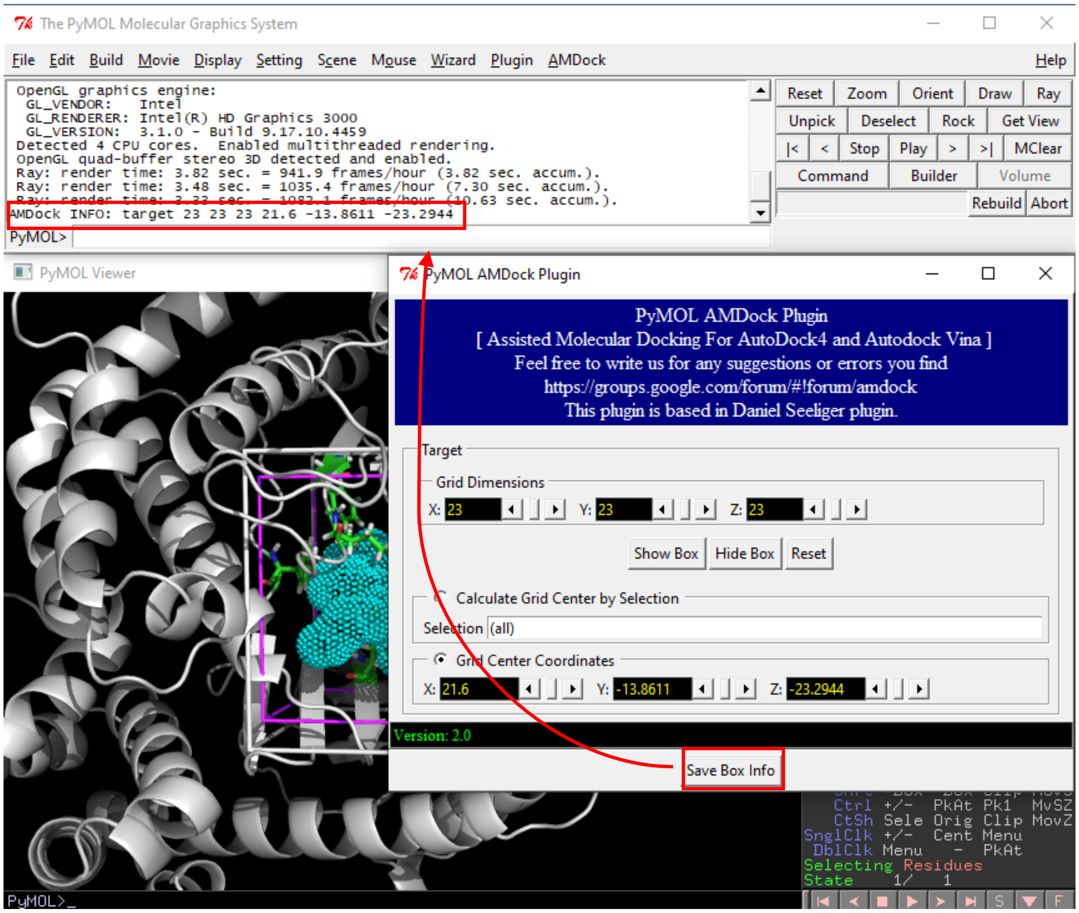


```
Detected GLSL version 1.40.
OpenGL graphics engine:
 GL_VENDOR:   Intel
 GL_RENDERER: Intel(R) HD Graphics 3000
 GL_VERSION:  3.1.0 - Build 9.17.10.4459
Detected 4 CPU cores.  Enabled multithreaded rendering.
OpenGL quad-buffer stereo 3D detected and enabled.
Ray: render time: 3.82 sec. = 941.9 frames/hour (3.82 sec. accum.).
Ray: render time: 3.48 sec. = 1035.4 frames/hour (7.30 sec. accum.).
Ray: render time: 3.33 sec. = 1082.1 frames/hour (10.63 sec. accum.).
PyMOL>
```

Figure 24. The specifications of the new box (Target_box (white)) from a predefined box (T_ref_box (magenta)) centered on Residue(s).

13. Click "Save Box Info" and check in the PyMOL command input area that the new specifications have been introduced successfully. Close PyMOL after that and check that the new specifications are written in the Search Space.

* *A similar procedure can be applied to modify the two search spaces when performing "off-target" docking*



```
 OpenGL graphics engine:
  GL_VENDOR:   Intel
  GL_RENDERER: Intel(R) HD Graphics 3000
  GL_VERSION:  3.1.0 - Build 9.17.10.4459
 Detected 4 CPU cores.  Enabled multithreaded rendering.
 OpenGL quad-buffer stereo 3D detected and enabled.
 Ray: render time: 3.82 sec. = 941.9 frames/hour (3.82 sec. accum.).
 Ray: render time: 3.48 sec. = 1035.4 frames/hour (7.30 sec. accum.).
 Ray: render time: 3.33 sec. = 1082.1 frames/hour (10.63 sec. accum.).
AMDock INFO: target 23 23 23 21.6 -13.8611 -23.2944
PyMOL>
```

PyMOL AMDock Plugin
[Assisted Molecular Docking For AutoDock4 and Autodock Vina]
Feel free to write us for any suggestions or errors you find
https://groups.google.com/forum/#!forum/amdock
This plugin is based in Daniel Seeliger plugin.

Target

Grid Dimensions

X: 23 Y: 23 Z: 23

○ Calculate Grid Center by Selection

Selection (all)

● Grid Center Coordinates

X: 21.6 Y: -13.8611 Z: -23.2944

Version: 2.0

Figure 25. Instructions for saving box information.

14. Press "**Define Search Space**" button and now proceed with the docking run

5.2 Making a high-quality image for publications

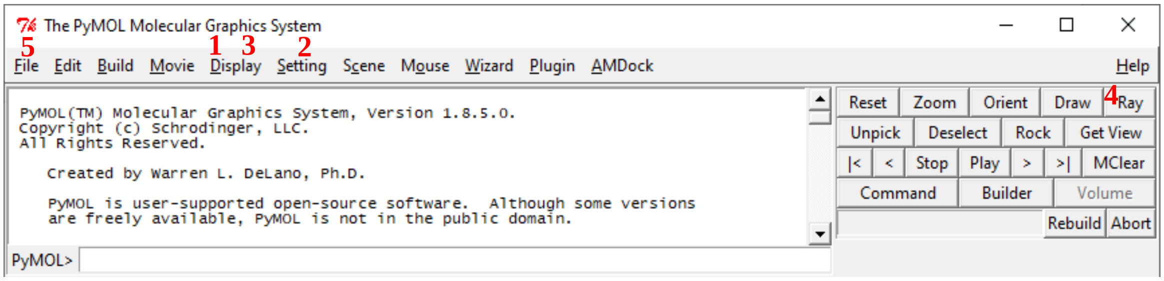
We tried to generate high-quality representations automatically. However, sometimes they won't be at the most informative angle. Usually, after a few rotations, it is possible to get better visualization. We keep a black background for better contrast in the representation. The following recommendations should help you in getting a high-quality image for publication in PyMOL.

1. *Change background color:* Display > Background > White
2. *Remove shadows:* Settings > Rendering > Shadows > None
3. *Maximum quality representation:* Display > Quality > Maximum Quality

4. *Rendering:* Press the Ray button in the upper left corner or type into the command line area:

```
ray 2400, 2400
```

5. *Saving:* File > Save Image As > PNG



```
PyMOL(TM) Molecular Graphics System, Version 1.8.5.0.
Copyright (c) Schrodinger, LLC.
All Rights Reserved.

   Created by Warren L. DeLano, Ph.D.

   PyMOL is user-supported open-source software.  Although some versions
   are freely available, PyMOL is not in the public domain.
```

Figure 26. Instructions for saving scene images.