Molecular docking with Python in Jupyter Notebooks: Towards the development of accessible docking procedures

Lee Schoneman, Dr. Paul Craig

Template

In order to maintain easy readability of the notebooks, all notebooks follow a set of guidelines to present information. The introduction of each notebook contains the following sections:

- Purpose
  - Target audience
  - Brief overview
  - Notebook series summary
  - Stepwise summary for notebook
  - Acknowledgements (if applicable)
- Table of libraries used

1 Basil Docking V0.1 - Docking Preparation

1.1 Purpose

Target Audience
Undergraduate chemistry/biochemistry students and, in general, people that have little to no knowledge of protein-ligand docking and would like to understand the general process of docking a ligand to a protein receptor.

Brief Overview
Molecular docking is a computational method used to predict where molecules are able to bind to a protein receptor and what interactions exist between the molecule (from now on, refered to as "ligand") and the receptor. It is a popular technique utilized in drug discovery and design, as when creating new drugs and testing existing drugs aginst new receptors, it is useful to determine the likelihood of binding prior to screening as it can be used to eliminate molecules that are unlikely to bind to the receptor. This significantly reduces the potential cost and time needed to test the efficacy of a set of possible ligands.

The general steps to perform molecular docking, assuming the ligand and receptor are ready to be docked, include the generation of potential ligand binding poses and the scoring of each generated pose (which predicts how strongly the ligand binds to the receptor, with a more negative score corresponding to a stronger bond). To dock a ligand to a protein, both the receptor and the ligand/s need to be "sanitized"; which includes making sure bonds and protonation states are as they would be in an organism. The receptor and ligand/s also need to be converted into the correct file formats depending on which docking engine is utilized. With all of these steps needed for preparation alone, introducing a (more)

This notebook series encompasses

1. **The preparation needed prior to docking (protein and ligand sanitation, ensuring files are in readable formats, and finding possible binding pockets)**
2. The process of docking ligand/s to a protein receptor using two docking engines (VINA and SMINA) and visualizing/analyzing the outputs
3. Further data collection and manipulation
4. Utilizing machine learning to determine key residues (on the protein) and functional groups (on the ligand) responsible for protein-ligand binding

Stepwise summary for this notebook (docking preparation, notebook 1 out of 4)

- Get PDB file from the protein data bank and separate the protein and ligand into different files
- Import additional ligands (if desired)
- Prepare and separate ligands into their own MOL2 and PDBQT files
- Find possible binding pockets in protein
- View protein and ligand/s

The methods utilized by this notebook are based off of Angel J. Ruiz-Moreno's Jupyter-Dock notebooks, which can be found on their GitHub account AngelRuizMoreno

Ruiz-Moreno A.J. Jupyter Dock: Molecular Docking integrated in Jupyter Notebooks. https://doi.org/10.5281/zenodo.5514956

Methods for sanitizing the protein PDBQT file was adapted from Jessica Nash's iqb-2024 repository, which was used in the IQB 2024 workshop - Python for Molecular Docking, and can be found on her GitHub account janash.

1.2 Table of Libraries Used

1.2.1 Operations, variable creation, and variable manipulation

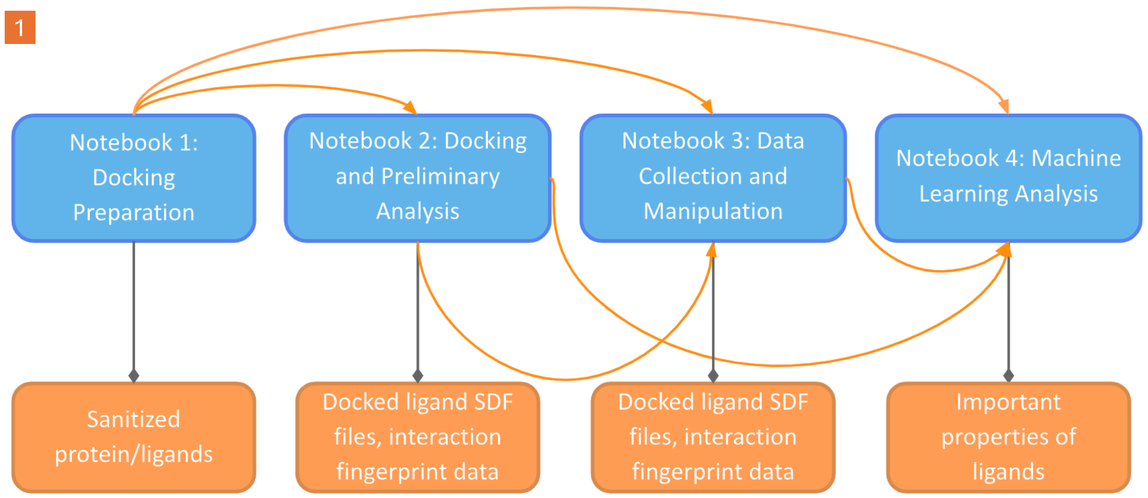
Module (Submodule/s)	Abbreviation	Role	Citation
numpy	np	perform mathematical operations, fix NaN values in dataframe outputs, and get docking box values from MDAnalysis	Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2. (Publisher link).
pandas	pd	organize data in an easy-to-read format and allow for the exporting of data as a .csv file	The pandas development team. (2024). pandas-dev/pandas: Pandas (v2.2.3). Zenodo. https://doi.org/10.5281/zenodo.13819579
re	n/a	regular expression; find and pull specific strings of characters depending on need, allow for easy naming and variable creation	Van Rossum, G. (2020). The Python Library Reference, release 3.8.2. Python Software Foundation.
os	n/a	allow for interaction with computer operating system, including the reading and writing of files	Van Rossum, G. (2020). The Python Library Reference, release 3.8.2. Python Software Foundation.
sys	n/a	manipulate python runtime environment	Van Rossum, G. (2020). The Python Library Reference, release 3.8.2. Python Software Foundation.
glob	n/a	pull files of interest, specifically for blind docking	Van Rossum, G. (2020). The Python Library Reference, release 3.8.2. Python Software Foundation.
warnings	n/a	filter warnings	Van Rossum, G. (2020). The Python Library Reference, release 3.8.2. Python Software Foundation.

Example of the notebook template followed by the series

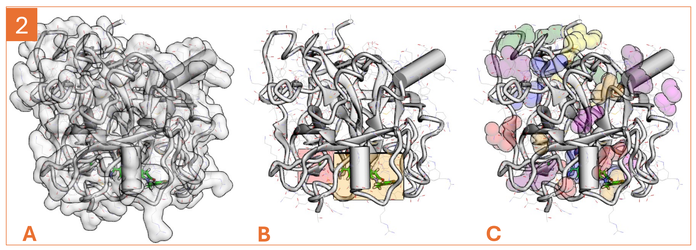
Abstract

Molecular docking is a computational technique used to predict ligand binding for a given receptor and is regarded as an attractive method to_use in drug design due to its relatively low computational and monetary cost. However, molecular docking programs tend not to be accessible to novice users. To increase general access to molecular docking, basil_dock utilizes a series of easy-to-use Jupyter notebooks that do not assume user familiarity with molecular docking procedures and concepts, requiring little command line usage and software installation. The series includes four notebooks that were created to reflect the different steps in the molecular docking process shown in Figure 1. This supports novice users' flexibility and customization in exploring docking procedures and systems, as well as teaching users the basis behind molecular docking without having to leave the environment to obtain information and materials from other applications.

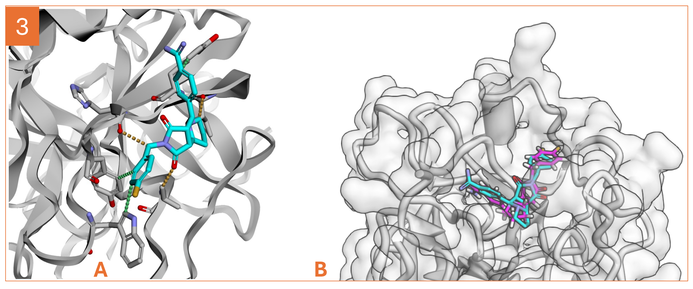
1



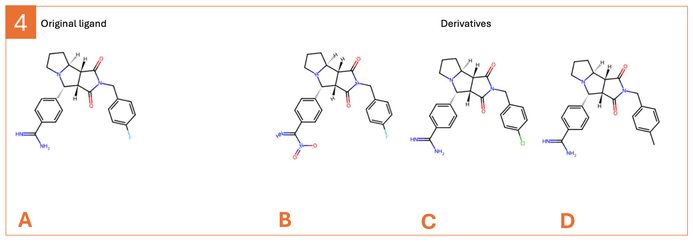
2



3



4



Looking Forward

Adding additional customization options for the creation of ligand derivatives is being worked on. Currently, only one functional group substitution can be made in the generation of related ligands. While users can get around this by selecting the derived ligand for further manipulation, supporting the substitution of multiple functional groups at once would increase efficiency of derivative binding.

Support to allow for the determination of which ligand functional groups and which protein residues are the most important in complex formation is also in progress and will hopefully be implemented soon.

Link to open-source repository on GitHub
https://github.com/leesch27/basil_dock.git

cav_id	drug_score	volume	nb_asph	inter_chain	apol_asph_proportion	mean_asph_solv_acc	mean_loc_hyd_dens	flex	hydrophobicity_score
1	0.4498	706.1058	100	0	0.48	0.4976	32.375	0.2117	34.65
2	0.0022	191.1717	35	0	0.0	0.3682	0.0	0.2693	-6.75
3	0.0148	215.9488	34	0	0.3529	0.3718	11.0	0.1495	23.3846

Table 1. Example of the fpocket output to find potential druggable pockets

Frame	Score	UNL1	UNL1	UNL1	UNL1	UNL1	UNL1
		GLY69.H	GLY69.H	GLY69.H	GLY69.H	GLY69.H	GLY69.H
		VdWConta	Functional group involved (VdWContact)0	Residue type(VdWContact)0	Distance (VdWContact)0	Index 1 (Ligand) (VdWContact)0	Index 3 (Protein) (VdWContact)0
1	-3.355		0	0	0.0	0	0
2	-2.808		0	1	2.9496014985078900	2	506
3	-2.466		0	0	0.0	0	0

Table 2. Example of the docking output containing interaction data

	filename_hydrogens	num_of_atoms	num_of_heavy_atoms	num_of_C_atoms	H_donors	H_acceptors	mol_refractivity	rotatable_bonds	polar_surface_area
0	fluoxetine-R	40	22	17	1	2	79.79870000000000	7	21.260000000000000
1	fluoxetine-S	40	22	17	1	2	79.79870000000000	7	21.260000000000000
2	tylenol	20	11	8	2	2	42.41050000000000	3	49.33
3	aspirin	21	13	9	1	4	44.71030000000000	3	63.60000000000000

Table 3. Example of the training data used in machine learning analysis

A	Importance
H_acceptors	0.195640
H_donors	0.155883
num_of_O_atoms	0.152726
log_P	0.119156
molecular_weight	0.113236
num_of_heavy_atoms	0.076509

B	Importance
log_P	0.143474
H_donors	0.127599
num_of_O_atoms	0.125464
H_acceptors	0.115905
mol_refractivity	0.085782
num_of_heavy_atoms	0.077370
molecular_weight	0.071143

C	Importance
log_P	0.168959
polar_surface_area	0.109958
num_of_O_atoms	0.095080
mol_refractivity	0.090682
molecular_weight	0.080909
H_acceptors	0.078048
H_donors	0.071359
num_of_heavy_atoms	0.068783
num_of_atoms	0.047248

*Table 4. Example of the feature importances using different rules on the same dataset (**A.** Lipinski's rule of 5, **B**. Lipinski's rule and the Ghose filter, **C.** Veber's Rule)*

Notebook 1: Docking Preparation

Retrieving Desired Protein and Ligands

Given a PDB ID an an input, a protein (and any ligands that may be in complex with it) is able to be downloaded onto the user's computer as either a PDB or MMCIF file.

- If multiple ligands are present in the initial PDB ID, ligands will be separated and a MOL2 file for each ligand will be generated.

Additional ligands can be acquired by:

- Searching the Chemical Component Dictionary by chemical name, type, ID, or brand name or similar formulas or structural similarities
- Using a local file found on the computer being used
- Generating a ligand using a SMILES string.

Protein and Ligand Sanitization

The protein as well as all ligands provided will be sanitized to make sure all structures are biologically relevant.

- Hydrogens will be added to all structures
- All bonds and conformations are checked

Sanitized structure information will be saved in PDBQT format for both the protein and ligands.

Determine Possible Binding Pockets

The center and size of each ligand is calculated in order to allow for site specific docking. For allow for blind docking, potential druggable protein pockets are able to be identified using fpocket (Table 1).

Visualization

Both the receptor and ligands can be visualized in a three–dimensional space using py3Dmol, with the ability to view surface topology, docking boxes, and druggable pockets identified by fpocket (Figure 2 A-C).

Notebook 2: Docking and Preliminary Analysis

Docking

Docking can be performed using either the vina or smina docking engine with the number of poses tested for each run and the exhaustiveness is set to five by default.

- Site-specific docking using the center and size of the ligand
- Blind docking using locations of potential protein pockets

After docking, an SDF file containing the atomic coordinates, bonds, molecular properties, and binding energy for each ligand and each binding pose is created.

Analysis of Docking Output

For each pose, interaction fingerprints are generated, providing the distance from the ligand to the protein, atoms involved with the interaction along with the specific ligand functional groups and protein residue types involved, and the interaction type (Table 2).

Visualization

The output of docking can be compared with the position and conformation of the input (Figure 3A). Interactions between the docked ligand and the receptor are able to be visualized (Figure 3B).

Notebook 3: Data Collection and Manipulation

Ligand Manipulation

For a ligand with a given functional group, a different functional group is able to substitute the original in order to create derivatives or related ligands (Figure 4 A-D). The functional group that is able to be added is dependent on the functional group that is to be replaced; for example, a primary amine is able to be replaced by a nitro or methyl group but is unable to be replaced by a halogen.

Dock, Analyze, and Visualize Derivatives

As with the second notebook, docking of ligands can be performed against the receptor using either the vina or smina docking engine. For each pose, interaction fingerprints are generated, providing the distance from the ligand to the protein, atoms involved with the interaction along with the specific ligand functional groups and protein residue types involved, and the interaction type. Visualization can be performed similarly to the second notebook.

Notebook 4: Machine Learning Analysis

Data Preparation and Calculations

A csv file containing various ligand properties and ability to be orally bioactive is provided to be used as data for determining which properties correlate to higher bioavailability and therefore which drugs can be taken by mouth (Table 3). Given a set of ligands, these properties can be calculated for ligands being tested by the user and the likelihood of bioavailability by mouth can be determined using a random forest model using one of the following rules: Lipinski's rule of 5 (Table 4A), Lipinski's rule and the Ghose filter (Table 4B), or Veber's rule (Table 4C).

Acknowledgements

The Rochester Institute of Technology, College of Science, Gosnell School of Life Sciences
Angel Ruiz Moreno, developer of Jupyter Dock: Molecular Docking integrated in Jupyter Notebooks (Github: AngelRuizMoreno)
Jessica Nash of MOLSSI, developer of iqb-2024 repository used in the IQB 2024 workshop - Python for Molecular Docking (Github: janash)
Support for this project was provided by NSF 2142033.

Sources

B. E. Granger and F. Pérez, "Jupyter: Thinking and Storytelling With Code and Data," in *Computing in Science & Engineering*, vol. 23, no. 2, pp. 7-14, 1 March-April 2021, https://doi.org/10.1109/MCSE.2021.3059263.
H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The Protein Data Bank (2000) *Nucleic Acids Research* 28: 235-242 https://doi.org/10.1093/nar/28.1.235