

DiscoverBMB, April 12-15, 2025, Chicago, IL

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

Authors: Lee Schoneman, Paul A. Craig

Molecular docking is a computational technique used to predict ligand binding potential, conformation, and location 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. Most docking programs require at least a basic knowledge of command line and computer programming to install and configure the program. Additionally, tutorials for the most commonly used programs tend to be inflexible, requiring a specific molecule or set of molecules to be bound to a specific receptor, and need the installation and usage of other programs or websites to download and prepare structures.

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: (1) the preparation of ligand and protein files prior to docking, (2) the docking of ligands to a protein receptor, (3) analyzing the resulting data and determining how different functional groups in the ligand can affect protein-ligand binding, and (4) identifying essential locations for binding within the ligand and protein. The notebooks enable 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.

The first version of basil\_dock allows users to choose from receptors uploaded to the Protein Data Bank and to add additional ligands as desired. Users can then select between the Vina and Smina docking engines and change ligand functional groups to see how the substitution of atom groups affects binding affinity and ligand conformation. The data can then be analyzed to determine residues in the receptor and atom groups in the ligand that are likely to be integral to forming the ligand-protein complex and to discern which ligands are likely to be orally bioactive based on Lipinski's Rule of Five.

From this work, a package of python scripts has been created to streamline the generating, splitting, and writing of ligand files, greatly reducing the number of errors arising from attempting to split a comprehensive ligand file manually. Libraries used in basil\_dock include Vina, Smina, RDKit, openbabel, and MDAnalysis. While the package has been designed based off the needs of basil\_dock, it has been created to be extensible.

Support for this project was provided by NSF 2142033.

