**BioProtIS IVS tutorial**

the use of pipeline that included Modeller, AlphaFold, Gromacs, Fpocket, and AutoDock Vina

**Authors:**

Graziela Sória Virgens

Júlia Oliveira

Maria Izadora Oliveira Cardoso

João Alfredo Teodoro

Danilo Trabuco Amaral1

1Centro de Ciências Naturais e Humanas, Universidade Federal do ABC (UFABC), Santo André, São Paulo, Brasil.

email danilo.trabuco@ufabc.edu.br

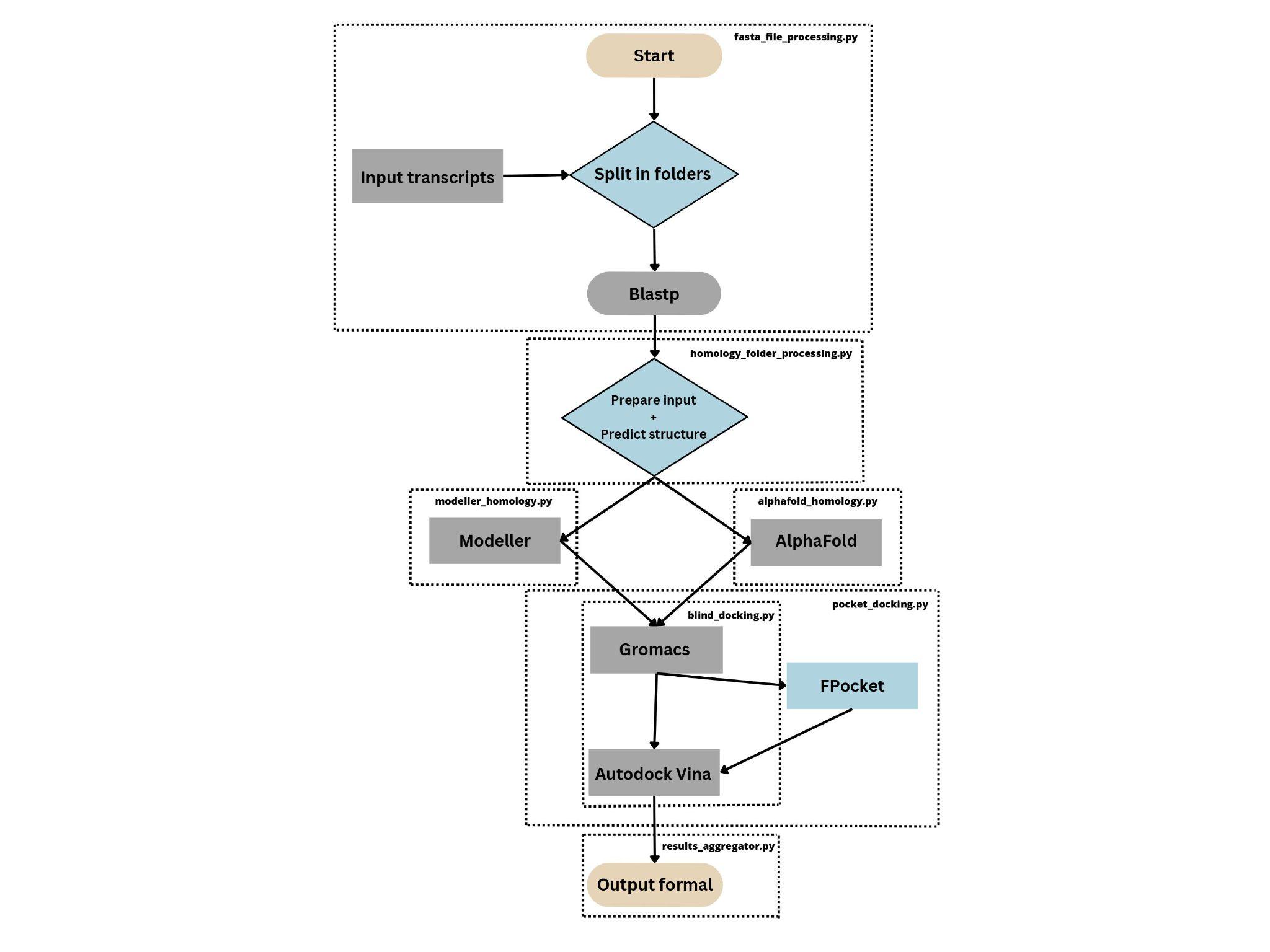
December, 2023

In any scientific field, it is imperative to validate experimental methodologies before embarking on actual production work. Therefore, in the initial use of *in silico* approaches, you will acquire biological knowledge to address the questions we aim to answer. Additionally, you will undergo training to assess the program's suitability for studying the binding mode of specific ligands (docking assessment).

The primary objective of this tutorial is to introduce an automated method for Inverted Virtual Screening (IVS), which may enhance molecular docking analyses and improve the replicability of the experimental bound conformation of a ligand within its target macromolecule during various analyses. For this tutorial, we will utilize the linux OS, human proteomic (<https://www.uniprot.org/proteomes/UP000005640>; receptors) and urea (CID: 1176; <https://pubchem.ncbi.nlm.nih.gov/compound/1176>; ligand) substrates, a common substrates present in human organism. The dataset was downloaded on December, 4th 2023. The pipeline applied the Modeller v. v.10.4 (Webb and Sali, 2017) and the blind docking approach.

This tutorial is designed as an introduction to the BioProtIS pipeline for genomic and transcriptomic datasets.

**IVS Flowchart**



**Flowchart depicting the complete BioProtIV pipeline.**

**Overall Steps:**

1. Download the human proteome data from: (<https://www.uniprot.org/proteomes/UP000005640>).
2. Retrieve the structure of the urea substrate from the .srf archive (PubChem CID: 1176)
3. Obtain the BioPROTIS package from the GitHub repository (<https://github.com/BBMDO/BioProtIS/tree/main>).
4. Update the analysis location for your data directly within the scripts.
5. Execute the pipelines in the sequence outlined in the above flowchart for protein modeling and docking.
6. Evaluate the results of the docking analysis.

**Installiation of all required sofwtares**

* Python 3.10 or higher
  + packages: `os`,`shutil`,`subprocess,`csv`,`matplotlib.pyplot,`re`,`openai`, and `argparse`.
* Python 2
* Modeller (<https://www.gnu.org/software/parallel/>)\*\*
* AlphaFold (<https://github.com/google-deepmind/alphafold.git>)\*\*
* BLAST (sudo apt-get install ncbi-blast+)
* GROMACS (sudo apt-get install gromacs)
* Seqtk (sudo apt-get install seqtk)
* FPOCKET (<https://github.com/Discngine/fpocket.git> or via conda)
* Autodock Vina (<http://vina.scripps.edu/>)
* #batch\_download.sh
  + wget <https://www.rcsb.org/scripts/batch_download.sh>
  + chmod +x batch\_download.sh
* PDB local database
  + wget <https://ftp.wwpdb.org/pub/pdb/derived_data/pdb_seqres.txt.gz>
  + gunzip pdb\_seqres.txt.gz
  + sed -i 's/ /\_/g' pdb\_seqres.txt
  + mkdir db
  + makeblastdb -in pdb\_seqres.txt -dbtype prot -out db/db
* BioProtIS Installation:

no installation is necessary

git clone <https://github.com/BBMDO/BioProtIS.git>

cd BioProtIS

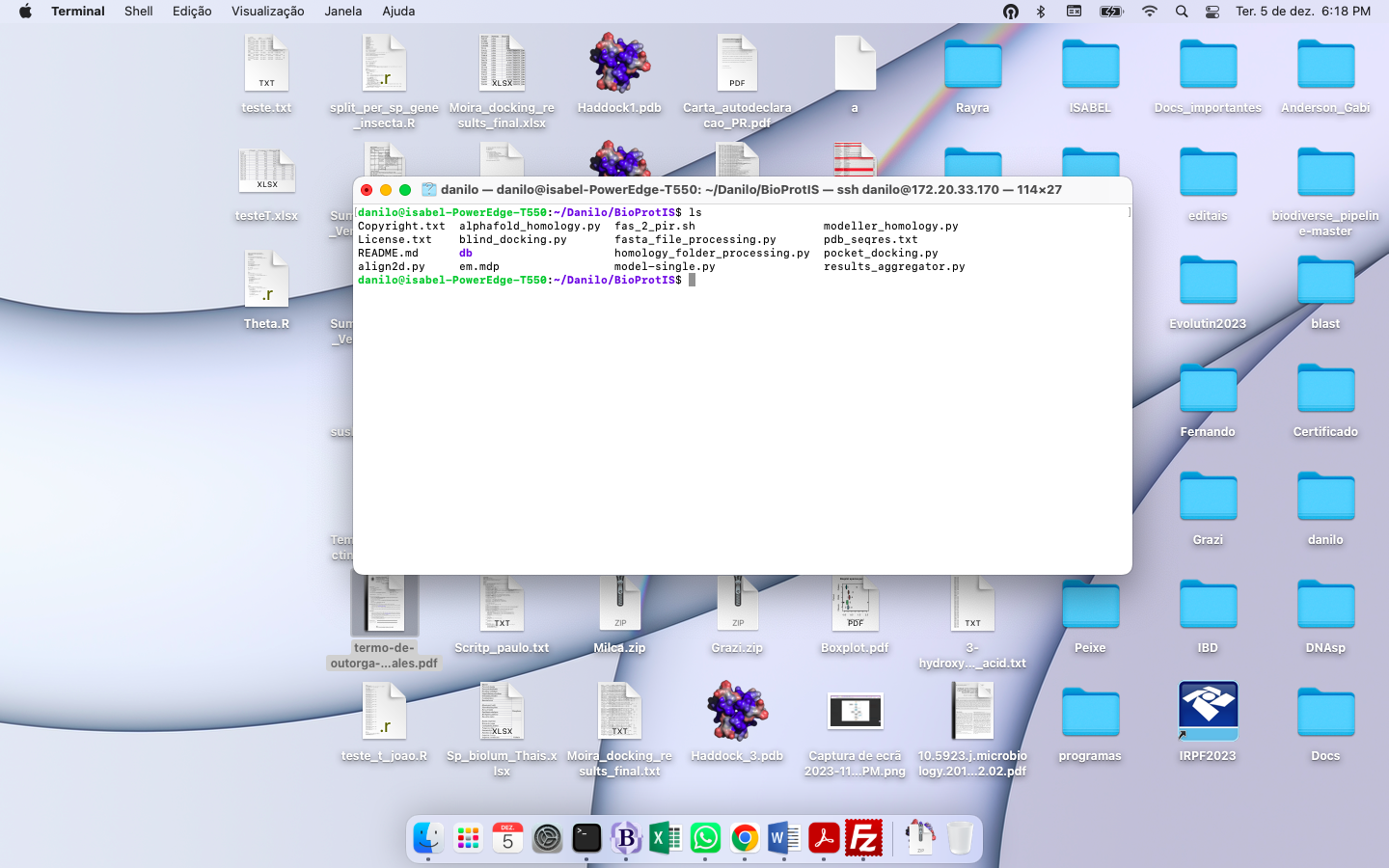
chmod +x \* #(executable permission)

\*\* Here, you have the option to install either both protein modeling software or just one.

**Step 1. Filtering data and preparing the ligand**

1. Proceeding to protein data filtering and protein modeling:

1.1. Access the BioProtIS/ directory. Download human protein using the link <https://www.uniprot.org/proteomes/UP000005640>. Download the PDB database, or if you have already done so, proceed to the folder moving it to the current directory.



**Shell screen displaying the folders and files required for molecular docking analysis.**

1.2. Update the script path folder to the location of your analysis. You can accomplish this by using the ***nano*** command to modify the path.

1.3. Download the urea substrate from PubChem using the CID: 1176. To convert the substrate from .srf file to .pdbqt file, using the command bellow:

1.4. Execute the following command to initiate the initial analysis:

python3 fasta\_file\_processing.py protein.faa

Note 1: Before proceeding to step 2, please ensure that you have downloaded the 'batch\_download.sh' script within the BioProtIS directory.

Note 2: It is possible to use more than one ligand at a time.

**Step 2. Download PDB templates and prepare the structures for Modeller**

At that point, we will download the PDB template structure and the protein sequence (FASTA file), reformatting the file to be accepted by Modeller.

1. Download the pdb templates and sequences:

1.1. In the BioProtIS/ directory, execute the following command to initiate the initial analysis:

python3 fasta\_file\_processing.py -f Seqs/Homology

P.S.: The choice of the Seqs/Homology directory is due to the modeling approach using a template. For Alphafold, disregard this step.

**Step 3a. Protein structure modelling using Modeller**

After preparing all inputs for Modeller, we are finally ready to begin the modeling process. At this stage, we use the default scripts provided by the Modeller tutorial. You are welcome to modify the scripts according to your specific needs.

3a. Structure Modeling:

3a.1. In the BioProtIS/ directory, execute the following command to initiate modeling:

python3 modeller\_homology.py

P.S.: If you intend to use AlphaFold, proceed to the next step.

**Step 3b. Protein structure modelling using AlphaFolder**

AlphaFolder uses its on database, thus, ay attention if you correctly instal it and downloaded its database

3b. Structure Modeling (AlphaFold:

3b.1. In the BioProtIS/ directory, execute the following command to initiate the initial analysis:

python3 alphafold\_homology.py

**Step 4a. Blind docking using Autodock Vina**

After the modeling step, we are able to perform the docking step. For that, we used the Gromacs software to prepare each of our receptors. Then, one at a time, we conducted the docking analysis using the substrate(s).

4a. Blind Docking:

3a.1. In the BioProtIS/ directory, execute the following command to initiate modeling:

python3 blind\_docking.py

P.S.: If you intend to define and utilize the active site, please use the script provided in step 4b.

**Step 4b. Active site docking using Fpocket and Autodock Vina**

After the modeling step, we are able to perform the docking step. For that, we used the Gromacs software to prepare each of our receptors. Then, one at a time, we conducted the docking analysis using the substrate(s). At this step, prior to the docking process, we employed the Fpocket software to identify putative active sites by searching for cavities and pockets. These were then classified based on criteria such as volume, surface area, and interaction energy.

4b. Active Site Docking:

4b. In the BioProtIS/ directory, execute the following command to initiate modeling:

python3 pocket\_docking.py

**Step 5. Merge the results**

The molecular docking process has concluded, generating new outputs within each directory, namely the "poses.pdbqt" files containing substrate poses, and the "docking\_results.txt" file. Nevertheless, the results pertaining to energy, structural validation, poses, and annotation are currently dispersed across various directories. To amalgamate these findings into a cohesive table, the provided command was employed. The resultant file, named "docking\_results\_final.txt", consolidates data for all substrates used by each protein from the transcriptome or genome dataset, and it will be created within the Docking/ directory.

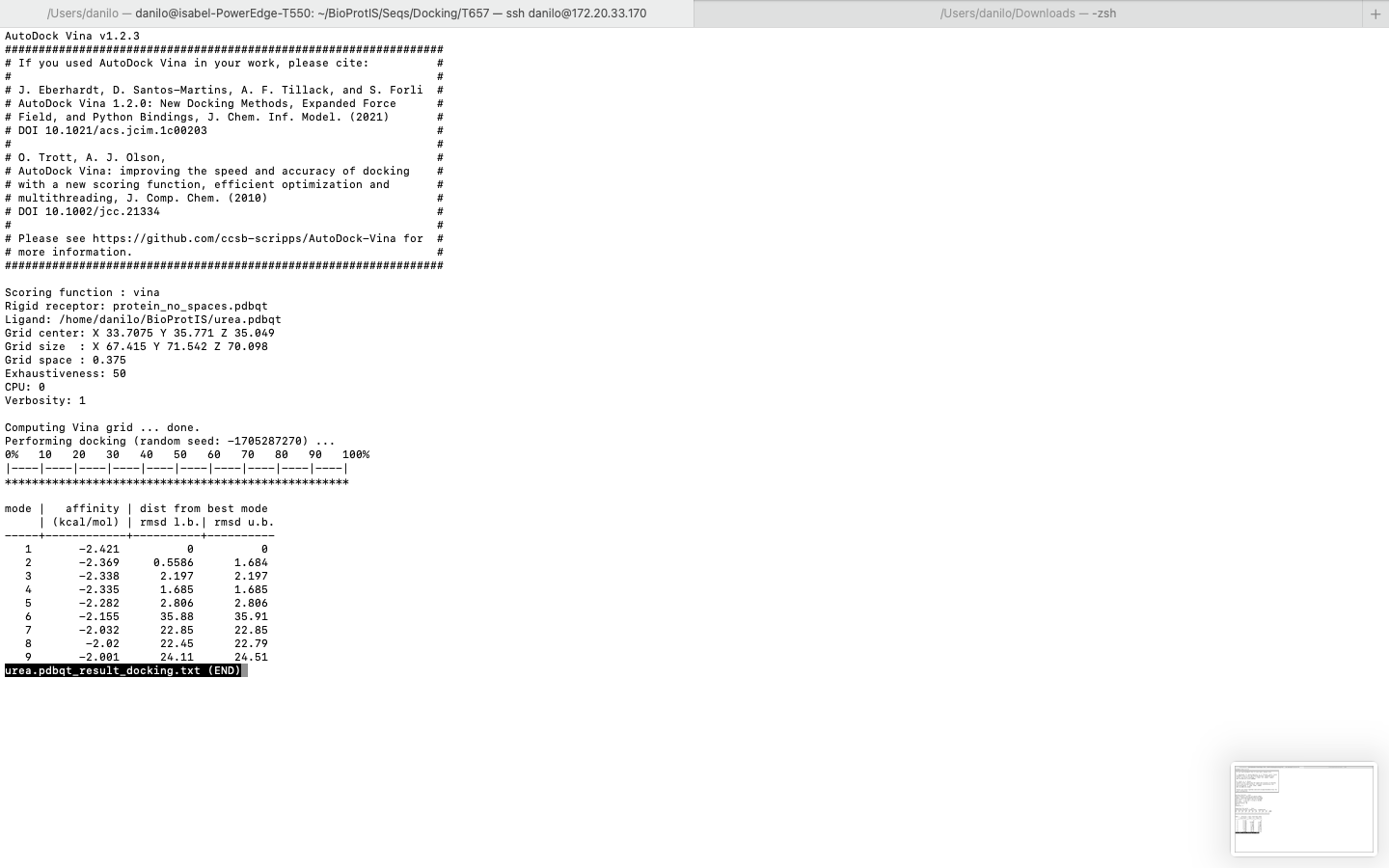
5. Merge the results:

5. In the BioProtIS/ directory, execute the following command:

python3 results\_aggregator.py

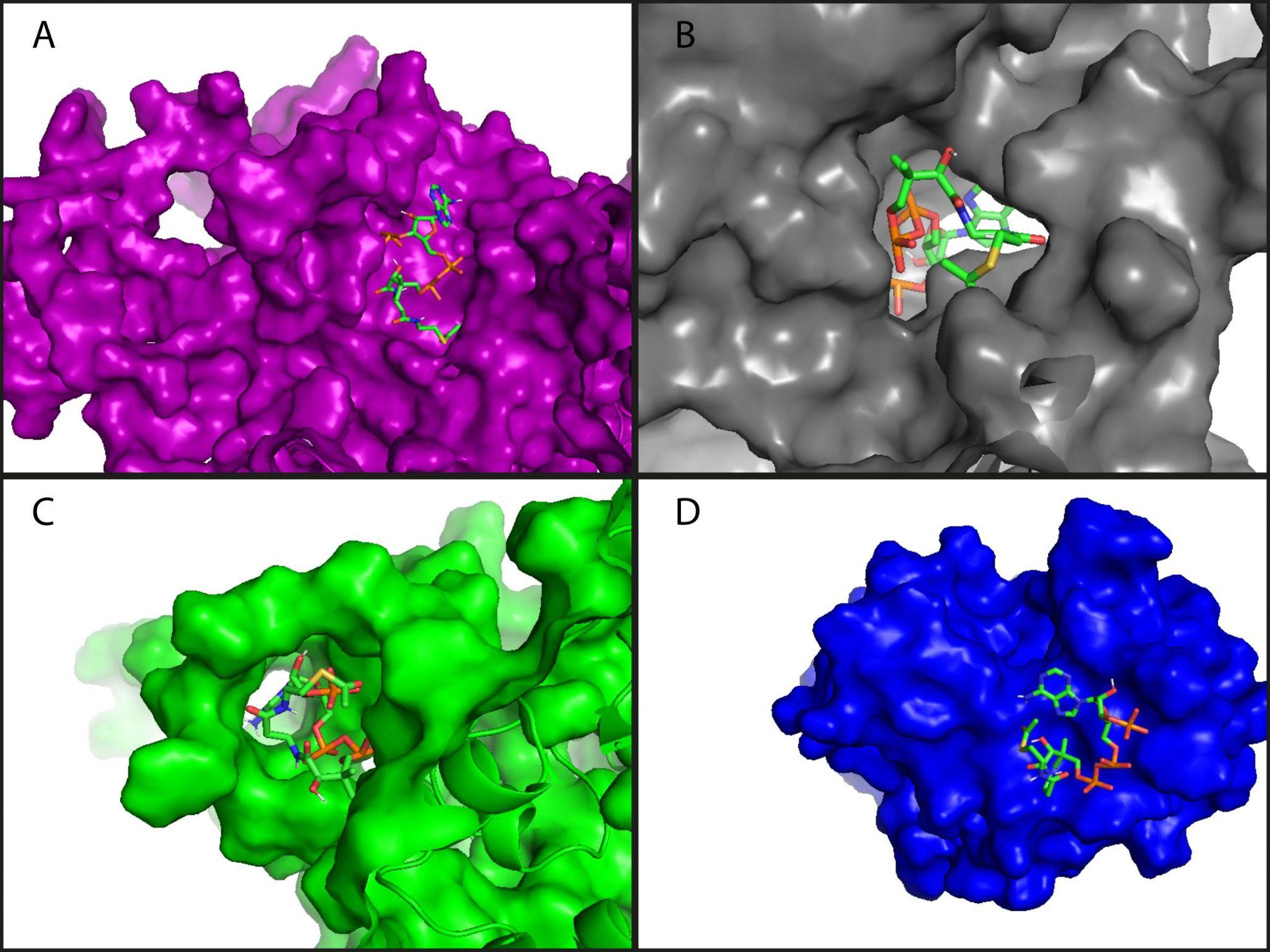
**Step 6. Evaluate the ligand energy and ligand poses**

As described earlier, within each folder in the Docking/ directory, a file named "docking\_result.txt" (depicted below) was generated, containing the binding energy results for each studied protein. This file presents the docking steps performed in AutoDock Vina, along with energy values and RMSD (Root Mean Square Deviation) values. Consequently, researchers can assess individual responses and analyze the analysis conditions.



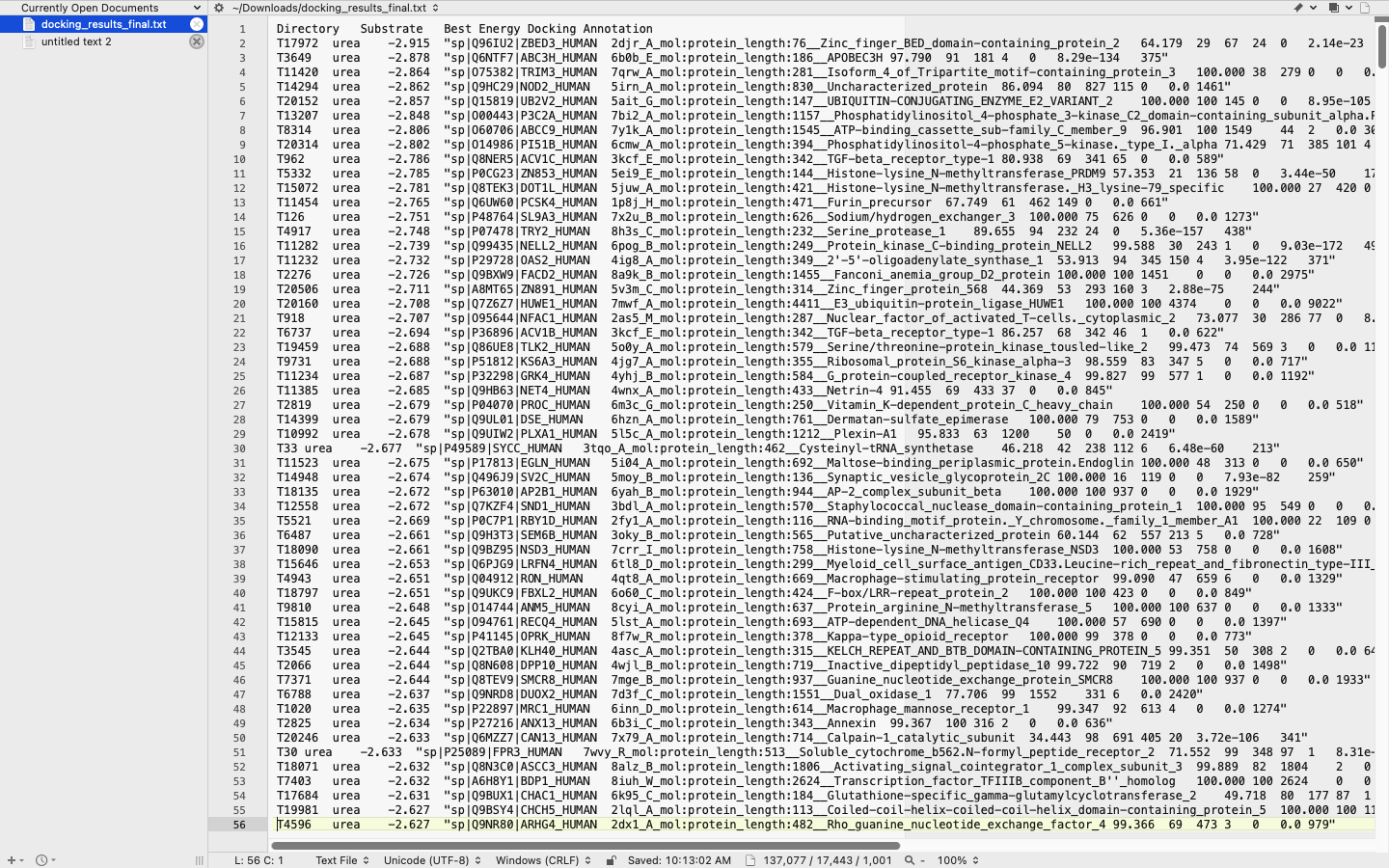
**Shell screen displaying the results of molecular docking analysis.**

Furthermore, this folder contains the "poses.pdbqt" file, which includes the structures and orientations for each evaluated substrate. Using programs like PyMol (Delano, 2002; depicted below), it is possible to evaluate the connections and positions obtained for each model. Remember, relying solely on binding energy is a flawed way to assess your results. Always check the model evaluation values and poses.

****

**Example of poses: Molecular docking poses for the four distinct transcripts of *Cereus jamacaru*, all docked to Acetyl-CoA. (a) 6WSG, (b) 5JRH, (c) 3XV0, and (d) 5FBQ.**

Finally, in the Docking/ directory, a file named "docking\_results\_final.txt" (sample format provided) will be created. This file summarizes in a single final table the energy results for all evaluated proteins.



**Screen displaying the results of molecular docking and annotation, consolidating all studied proteins.**

From these output files, the researcher can make informed decisions for the upcoming stages of their research.

To assist at any stage, all intermediate and final outputs of the above analysis using human genome data are available in the "example" folder.

**References**

DeLano, W. L. (2002). Pymol: An open-source molecular graphics tool. *CCP4 Newsl. Protein Crystallogr*, *40*(1), 82-92.

Webb, B., & Sali, A. (2017). Protein structure modeling with MODELLER. *Functional Genomics: Methods and Protocols*, 39-54.