**Ligand Function Prediction Guide**

**1/19/2023**

**Author: Gregory L. Szwabowski**

**Contact: gszwabowski@gmail.com**

The following guide outlines the steps required to predict a ligand's function based on its experimentally determined/docked binding mode in complex with a GPCR structure. Files necessary for this tutorial are located in a .zip file available at:

<https://github.com/gszwabowski/guides/blob/master/ligand_fxn_prediction_tutorial_files.zip>

.zip File Explanation

* svl-scripts-for-data-extraction/: directory containing .svl scripts necessary for interaction energy/type extraction from docked ligand-GPCR complexes
  + *get\_topscored\_pose\_by\_mseq.svl*: used if you wish to predict ligand function for only top scoring poses
  + *loopnumber.svl*: used to index entries in output docking database
  + *create\_indexing\_mdb.svl*: used to generate a database denoting start, end, and TM x.50 residue positions for each GPCR structure serving as a docking target
  + *dockdb\_to\_lf\_input*.*svl*: used to extract interaction energies/types at each residue position for each ligand-receptor complex entry in a database of docking results. This script relies on the database generated with *create\_indexing\_mdb.svl* to correctly identify which GPCR residues correspond with certain Ballesteros-Weinstein [1] indexed residue positions
  + *extraneous\_scripts/*: contains extra scripts created during development of this project that may be useful
* *LFP\_classifier.py*: python script used to classify the text file containing interaction energies/types that is created using *dockdb\_to\_lf\_input.svl*
* *LFP\_label\_encoder.pkl*: label encoder that is loaded with joblib
* *LFP\_rf\_model.pkl*: random forest model for ligand function prediction that is loaded with joblib

Prerequisites

* Python libraries: sklearn >= v0.24.2, pandas, numpy, joblib
* Molecular Operating Environment (MOE) for ligand docking/running SVL scripts necessary for extraction of interaction energies/types from a ligand-GPCR complex
  + .mdb file containing 5 docked poses per ligand is ideal to start with

Tutorial

|  |  |  |
| --- | --- | --- |
| **Step** | **Description** | **Figure** |
| 1 | Note: if you wish to make predictions based on multiple docked poses per ligand, skip to step 2. Otherwise, continue on.  After docking has concluded, the top scoring docked poses for each ligand must be extracted using the *get\_topscored\_pose\_by\_mseq.svl* script. First, open the script (located in the “scripts” folder) using the *Edit…* button and then save and load the script by clicking *SVL 🡪 Save and Load*. Set your CWD to a the directory containing docking results and ensure that databases containing docking results are the only .mdb files located in the directory. This script will work if only one or multiple .mdb files containing docking results are present in a directory. Use the command:  get\_topscored\_pose\_by\_mseq [prefix]  where prefix is the name you wish to suffix the database containing top scoring poses for each docked ligand in each database. Top scored poses for each docking database will be located in .mdb files titled ‘prefix\_topscored\_poses.mdb’. |  |
| 2 | Next, a field denoting the target needs to be added to each top scored pose database. For each ‘prefix\_topscored\_poses.mdb’ database, open it in the database viewer and click *Edit* 🡪 *New* 🡪 *Field*. Set the new field type to ‘char’, the Name to ‘Target’, and the Value to the name of your target. Click *OK* to create the field in the database. The field can then be manually filled for each entry.  Alternatively, *create\_target\_fields\_dir.svl* can be used to create this field if multiple targets are represented in your docking database. |  |
| 3 | Next, all top scoring poses databases must be imported into a single database. After setting your CWD to where you wish to aggregate the top scored poses from each database, click *File* 🡪 *New* 🡪 *Database…* and name the database ‘XXX\_topscored\_poses.mdb’, where ‘XXX’ is the name of the target that was docked into. In the new database’s viewer, click *File* 🡪 *Import…* and then click the + icon to select files for database import. For each database that was generated in step 1, select the database and then click *Add*. Once all databases generated in step 1 are added, click *OK and* then *OK* to import them to the ‘XXX\_topscored\_poses.mdb’ database. | *Database with top scored poses imported* |
| 4 | Next, each entry in the database will need to be numbered with the *loopnumber.svl* script. Save and load the *loopnumber.svl* script and use the command  loopnum [mdb]  where mdb is the ‘XXX\_topscored\_poses.mdb’ file. This script will create an index field that numbers each of the entries in the database of aggregated top scoring poses. |  |
| 5 | Next, save and load the *create\_indexing\_mdb.svl* script and use the command  create\_indexing\_mdb [mdb, prefix]  where mdb is the ‘XXX\_topscored\_poses.mdb’ file and prefix is the name of your target, entered as a token. This script will create a database titled ‘XXX\_tm\_indexing.mdb’ that contains the ‘Target’ and ‘index’ fields from ‘XXX\_topscored\_poses.mdb’ as well as fields for denoting the start, x.50, and end residue for each transmembrane helix within a GPCR. In addition, this script will create a .txt file that can be imported to Excel and manually filled in. |  |
| 6 | The .txt file generated in the previous step now needs to be imported into Excel. To do this, open Excel and then click *Open* 🡪 *Browse*. Navigate to where your .txt file is saved and then open your .txt file after setting the filetype filter to ‘Text Files’. In the Excel Text Import Wizard, be sure to click “My data has headers” and then click *Next*. In the next window, set the delimiter to ‘Comma’ and then click *Next*. Click *Finish* in the last window to finish importing your .txt file to Excel. |  |
| 7 | Once the empty TM indexing database is imported into Excel, the start, x.50, and end residue position numbers for each transmembrane domain in each unique target structure (denoted by a unique entry in the “Target” field of the database) need to be filled in.  The following process should be repeated for each unique target structure:  To get the start/end residues for a target structure, open the ‘XXX\_topscored\_poses.mdb’ in MOE and double click the entry containing the target structure you wish to analyze in its ‘Receptor’ field to open the target structure in MOE. Next, open the MOE sequence editor and toggle on secondary structure annotation and residue numbering using the buttons in the bottom right of the sequence editor. Next, start and end residue positions for each transmembrane helix in the target structure can be denoted in the TM indexing Excel spreadsheet. For each TM domain, residue numbers of the leftmost and rightmost helical residues should be entered into the “TMx\_start” and “TMx\_end” columns in the Excel spreadsheet for a given target structure. Using the example on the right, the start and end residue numbers for TM1 would be 28 and 58, respectively, while the start and end residue numbers for TM2 would be 65 and 92, respectively.  To get the x.50 residue positions for a target structure, navigate to <https://gpcrdb.org/> and click the ‘Sequence alignments’ link located under the ‘GPCRdb’ dropdown menu. Once this link loads, select your first target structure and then click *Next*. In the sequence segment selector, click *Full sequence* and then *Show alignment*. For each TM segment in the alignment, make a note of the residue labeled ‘.50’, as this residue will need to be identified within your target structure. For each TM domain in your target structure, identify the .50 residue and enter its residue number into Excel.  Once TM start/x.50/end residues are indexed for each target structure listed in the Excel spreadsheet, resave the spreadsheet as a .txt file. | TM start/end residue identification          TM x.50 residue identification |
| 8 | Using MOE, delete the original ‘XXX\_tm\_indexing.mdb’ database. Next, click *File* 🡪 *Open* and open the .txt file that was saved in the last step to import it to a MOE database that will denote TM start/x.50/end position residue numbers for each target structure. |  |
| 9 | Next, save and load the *dockdb\_to\_lf\_input.svl* script. At this point, the following files will be used to extract interaction energies and types that are used to predict ligand function:   * ‘XXX\_topscored\_poses.mdb’ * ‘XXX\_TM\_indexing.mdb’   For each docked pose in ‘XXX\_topscored\_poses.mdb’, the *dockdb\_to\_lf\_input.svl* script will load the ligand and receptor and use the ‘XXX\_TM\_indexing.mdb’ database as a reference to obtain TM start/x.50/end residues of the receptor structure. Prior to running the script, it is important to ensure that:   * The top scoring pose and TM indexing databases contain the same number of entries * The ‘Target’ and ‘index’ fields in both databases are exactly the same   To extract interaction energies and types using both databases, use the following command:  dockdb\_to\_LF\_input [topscored\_poses\_mdb, name\_prefix, tm\_idx\_mdb]  where topscored\_poses\_mdb is the ‘XXX\_topscored\_poses.mdb’ file, name\_prefix is the name of your target, and tm\_idx\_mdb is the ‘XXX\_TM\_indexing.mdb’ file. Once the script is done running, open the newly generated ‘XXX\_interaction\_energies.mdb’ file and save it as a .txt file using default settings. |  |
| 10 | Prior to making predictions of ligand function with the newly generated text file, extract *LFP\_classifier.py*, *LFP\_label\_encoder.pkl*, and *LFP\_rf\_model.pkl* to a directory. Additionally, copy the .txt file that was generated in the previous step to this directory. Next, navigate to this directory using your machine’s command prompt and use the following command to predict ligand function for each docked pose whose interaction energies/types are represented in the .txt file:  python LFP\_classifier.py <textfile>  Where <textfile> is the name of the .txt file that was generated in the previous step and copied into the current directory.  When running this script, you will be asked whether docking was performed with homology models. If yes, enter y. If no, enter n. The script will also ask for the number of docked poses per ligand, which determines majority predictions if multiple docked poses were generated per ligand. This should be entered as an integer.  Once the script runs, ligand function predictions will be written to a .csv file. |  |

REFERENCES

(1) Ballesteros, J. A.; Weinstein, H. Integrated Methods for the Construction of Three-Dimensional Models and Computational Probing of Structure-Function Relations in G Protein-Coupled Receptors; Methods in neurosciences; Elsevier, 1995; Vol. 25, pp 366–428.