Supporting information for:

Pattern Based Detection of Potentially Druggable Binding Sites by Ligand Screening

Uttam Pal and Nakul Chandra Maiti*

Structural Biology & Bioinformatics Division, CSIR - Indian Institute of Chemical Biology, 4, Raja S.C. Mullick Road, Kolkata 700032, India

*Corresponding author: Nakul C. Maiti. E-mail: ncmaiti@iicb.res.in; Telephone: +91-3324995940.

Detailed Methodology

Centering the ligand and protein. The protein *.pdb and ligand *.mol2 files were opened in the PyMOL molecular viewer. Using the `centroid` command in PyMOL, all the molecules were placed in the origin i.e. the center of the mass of all the molecules were translated to `(0,0,0)` coordinate. The centroid script is not installed by default in PyMOL. However, it can be obtained from the PyMOL script repository (link: https://github.com/Pymol-Scripts/Pymol-script-repo/raw/master/centroid.py Retrieved 08/13/17). The following command shall place everything in the center.

```
centroid all, center=1
```

When `center=1` is not specified, `centroid` command only prints the center of the mass of all the molecules. If the `center=1` is specified, it will place the centers of mass of the molecules at the origin.

Placing the protein 40Å away. In order to get a better resolution in the pattern diagrams, the protein molecule is need to be placed a distance away from the ligands. In our studies we placed the protein 40Å away form the origin along the x axis. Using the `translate` command (inbuilt in PyMOL) the protein coordinate was translated to the stipulated position. The command is as follows:

```
translate [40,0,0], <name of the target protein>
```

Now, it is required to save the altered coordinates of all the molecules. The files can be saved from the “File→Save Molecules…” option in PyMOL.

Docking. The the docking was carried out using standard blind docking protocol in AutoDock 4.2 in Linux environment. ga_run parameter was set to 100. Grid box was defined large enough to encompass the whole protein. It will generate 100 binding conformation of each ligand. Grid and docking parameter files for each of the ligands are required to be generated using AutoDockTools interface following the default docking manual. Finally, docking run can be automated for all the ligands in the library using the following bash script:

```
#! /bin/bash

for  f in *.gpf; do
     b=`basename $f  .gpf`
     echo Processing grid $b
     autogrid4 -p $f -l $b.glg
     echo docking $b
     autodock4 -p $b.dpf -l $b.dlg
done
```

Alternatively, PyRx interface can be used for automated screening of a large number of ligands. PyRx can be obtained freely from http://pyrx.sourceforge.net/.

The *.dlg file, which is generated by AutoDock 4.2, prints the reference RMSDs, which are the deviation of the bound conformers from the initial input structure, along with their binding energy. A part of the *.dlg file listing the RMSDs and binding energies are shown as follows:

Rank	Sub-Rank	Run	Binding Energy	Cluster RMSD	Reference RMSD	Grep Pattern
1	1	83	-5.67	0.00	33.39	RANKING
1	2	135	-4.99	0.88	33.00	RANKING
2	1	157	-5.35	0.00	21.15	RANKING
2	2	127	-4.60	0.66	21.40	RANKING
3	1	154	-5.09	0.00	13.91	RANKING
3	2	179	-4.69	0.89	13.55	RANKING

Reference RMSD in this RMSD table indicates the structural deviation from the input orientation of the ligand. The usefulness of AutoDock 4.2 for this study lies in the fact that AutoDock 4.2 generates these reference RMSDs by default. However, if some other docking program is used for the docking, which does not gives the reference RMSDs of the ligands, the same can be calculated by `compute_rms_between_conformations.py` python script. This script can be found in the Utilities folder in in the AutoDockTools installation directory. The RMSD and binding energy information given in the columns 4 and 6 in this RMSD table of the *.dlg file are required to be extracted for the pattern generation. In order to extract these information from multiple *.dlg files simultaneously, the following bash script was used.

```
#! /bin/bash

for  f in *.dlg; do
     b=`basename $f  .dlg`
     grep 'RANKING' $f > $b.out
     awk '{print $4,$6}' $b.out > $b.csv
     rm $b.out
done
```

Pattern generation. Now, RMSD and corresponding energy values for a bound ligand was obtained. The data was then imported to Wolfram Mathematica 11 as follows:

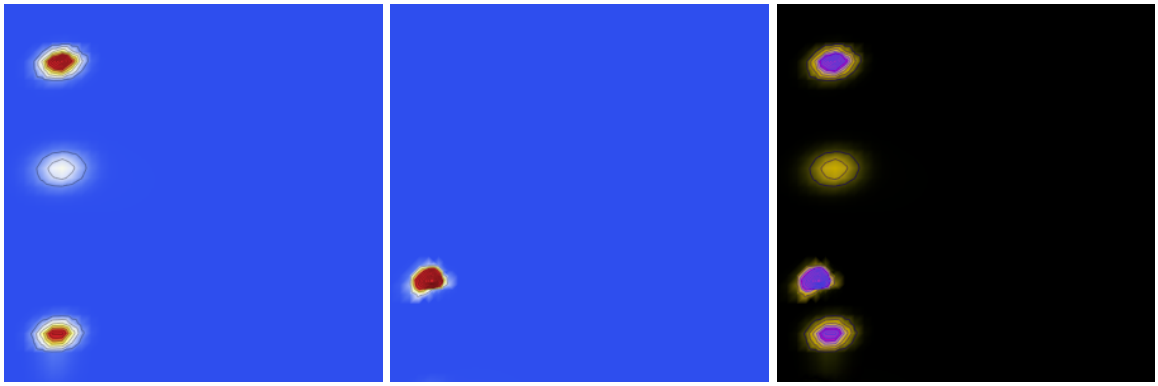
```
data1=Import[</path/to/the/file.csv>, "Table"]
```

Then, the data was plotted as smooth density histogram using the following command in Wolfram Mathematica 11.

```
pattern1 = SmoothDensityHistogram[data1,
     PlotRange -> {{-12,-4},{20,70}}
     PlotTheme -> "Minimal",
     ColorFunction -> (ColorData["TemperatureMap"][Rescale[#, {0, .
1}]] &),]
```

`PlotRange` parameters needs to be determined from the minimum and maximum values of the binding energies and RMSDs, respectively. `Min[data1[[All,1]]]` gives the minimum of the first column of `data1`, which is the binding energy. Similarly, `Max[data1[[All,1]]]` gives the maximum binding energy. Wolfram Mathematica was used in this study, however, gnuplot utility, which is freely available for Linux can also be used to generate the density patterns. Moreover, due to its command line feature, gnuplot can be used to automate the pattern generation for all the ligands in the library. Another freely available alternative is iPython (https://ipython.org/) or jupyter (https://jupyter.org/).

Interpretation of the patterns. The control experiment for the ligand that binds to the active site of the target (e.g, ligand ISO-1 binding with the target protein human MIF) produced the density patterns as shown below (left figure). Whereas, an unknown ligand (epoxyazadiradione) from the library produced the pattern as in the middle figure below. Comparison of these two patterns can be done using the `ImageDifference[pattern1,pattern2]` command in Mathematica as shown in the right figure.



Mismatch of the patterns is an indicator of the presence of a potentially druggable allosteric binding site in the target protein.