Shc Inhibitors Docking Protocol and Results

February 10, 2021

1 Software

This section breifly describes the primary software used for conducting and preparing ligand docking simulations.

1.1 Rossetta

The Rosetta suite was used to preform the actual ligand docking simulations. Rosetta was downloaded and compiled to a remote cluster where all simulations were ran.

I often accessed the following resources when determining how to run Rosetta for ligand docking.

1. Rosetta ligand docking demo
2. Rosetta Ligand docking with flexible XML protocols
3. Rosetta Documentation

1.2 BioChemicalLibrary (BCL)

BCL was used for generating libraries of conformers (different possible conformations of a ligand) from one specific structure. These conformer libraries would then be given to Rosetta in order to simulate ligand flexibility.

1.3 Open Bable

Open Babel suite was used for one-off file conversions in cases where `sdf` files needed to be converted to `pdb` or similar operations.

1.4 PyMol

PyMol was used to to visualize the results of ligand docking experiments, create images and present results.

1.5 SLURM

All significant compute (namely docking experiments) where ran using the workload manager SLURM. Some programs will not work (RDBC) if other workload managers are used.

1.6 RDBC

Rosetta ligand docking batch job submission control and organizer is a Python command line program I created to help me submit organize and analysize large number of Rosetta docking simulations to the remote cluster's workload manager, SLURM.

2 General workflow

This section describes the general procedure for running docking simulations to access the binding of a specific ligand to a specific protein target.

2.1 Prepare protein structure

First, a ligand free protein structure was repacked using the `ligand_rpkmin.static.linuxgccrelease` program of the Rosetta suite. Repacking was repeated for 100 structures and the lowest energy structure is selected as the docking target.

2.2 Prepare ligand

First, a file that describes the ligand needs to be aquired. For `RTX ...` drugs these was just a matter of using the `sdf` files. If the simulation involved docking a ligand / peptide from an existing co-crystal structure Pymol was used to create a `pdb` file containing only the ligand / peptide. Open Bable was then used to convert the `pdb` file to `sdf` format.

Next, the ligand conformer library and Rosetta `params` files was generated for the to-be-docked ligand. This simulates ligand flexibility during docking. This was usually completed with this Python script which wraps the BCL `molecule:ConformerGenerator` program and the Rosetta `molfile_to_params.py` located in `main/source/scripts/python/public/` of a standard Rosetta installation.

2.3 Prepare RDBC files

I almost always used RDBC to actually run the docking simulations on the remote cluster. RDBC works by using templates of files that would be required for individual jobs and filling them out based on the command line arguments to run many jobs. One of the most important is the Rosetta XML docking protocol which determines exactly what Rosetta does during the simulations. For random docking experiments (where the ligand is positioned at a random position around the protein before docking) I used the random_docker_template.xml.

2.4 Submit jobs with an RDBC command

Once all nessecary files are created the Rosetta docking simulations where submitted to SLURM using RDBC. Below is an example command I used to for randomly docking the NPEYp peptide to the 1OY2 shc structure.

```
python3 ~/software/RDBC/rh.py -l ~/jobs/dock_random_NPEYp/ligand \
-p ~/jobs/dock_random_NPEYp/protein/Shc1-PTB_1OY2_0061.pdb \
-o ~/jobs/dock_random_NPEYp/results \
-e ~/software/rosetta_bin_linux_2020.08.61146_bundle/\
main/source/bin/rosetta_scripts.static.linuxgccrelease \
```

```
-i 2000 \
-op ~/software/RDBC/handler/xml_templates/random_docker.xml \
-b ~/jobs/dock_random_NPEYp/templates/NPEYp.sbatch \
-mi 10
```

- -l: Location of my ligand preparation files produced during the *prepare ligand* step. This included
  - NPEYp_conformers.pdb: Conformer structures generated with BCL.
  - NPEYp.params: Rosetta params file generated by molfile_to_params.py
  - NPEYp.pdb: PDB structure of the NPEYp ligand converted from a provided sdf file.
- -p: Path to target protein. Selected during the *prepare protein structure* step.
- -o: Output path. Where I want the results of simulations to be written to.
- -e: Path to Rosetta scripts exe.
- -i: Number of individual simulations each job should run. This is equivlanet to the number of poses Rosetta will produce.
- -op: Path to Rosetta XML protocol file. In this case I used one designed for random docking.
- -b: Path to template `batch` file. This is filled out for each individual job in order to submit many smaller jobs instead of one larger one. This avoids issues if the reasrouces you can use for one job are limited (as was my case).
- -mi: Tells RDBC to submit 10 copies of this job, which allows for simulating more poses when resources for individual jobs are constrained.

Then we just wait for our jobs to complete.

2.5 Aggregate run copies

For random docking that submits multible copies of the same job, I found it easer to aggregate the results into one large file that is easier to work with for plotting in programs like R.

If RDBC was used to submit such a job, it can also be used for aggregating the results using the `-mai` argument. If my jobs created by the example command had just finished they could be aggregated into one large results tab seperated file called `NPEYp.agg.tsv` with the command below.

```
RDBC/rh.py -o /home/ethollem/jobs/dock_random_NPEYp/results -mia NPEYp.agg.tsv
```

Each row (observation) in this file represented one completed docking simulation. Simulations can be uniquely identified by comparing both the `description` and the `iter_id columns`.

2.6 Add additional data to aggregate score file

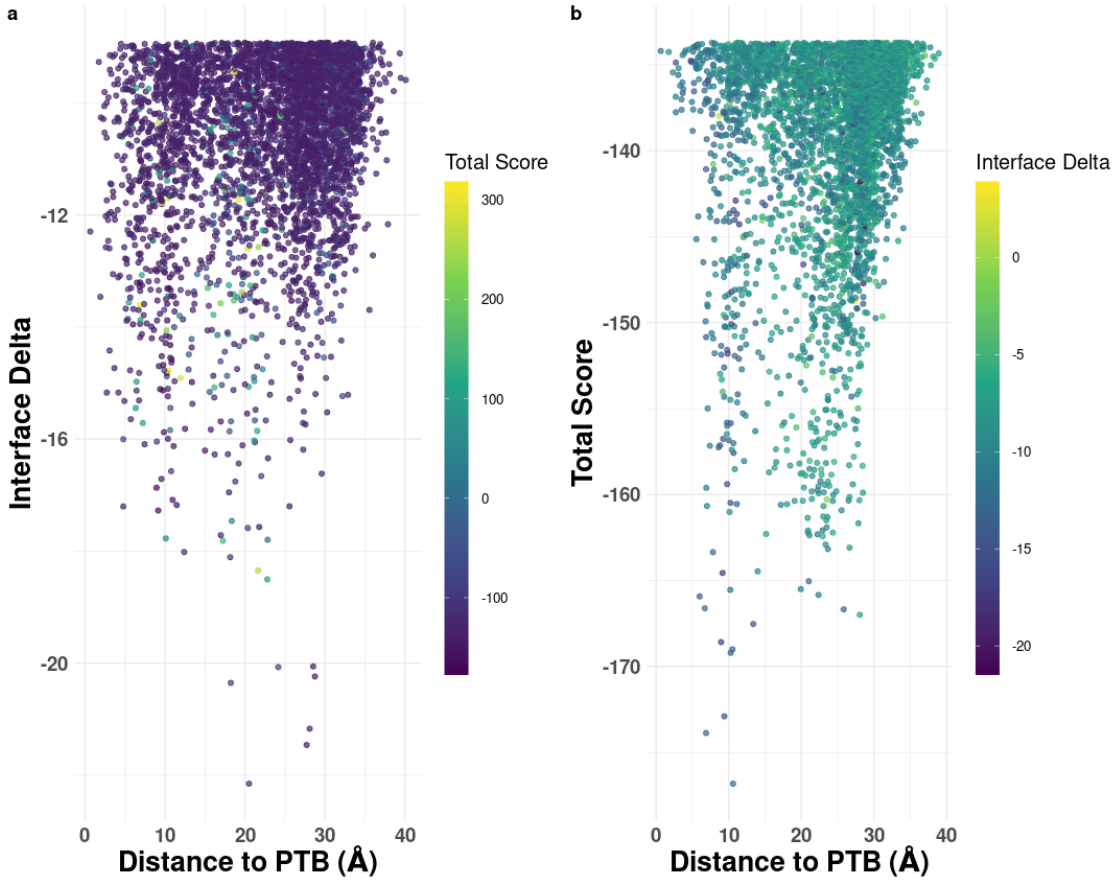
In addition to the metrics produced by Rosetta during the docking simulations we are also interested in having some additional data points listed below.

1. Average ligand position: An easy to calculate metric that describes the general location of the ligand in space. Will include three columns, one for each coordinate (x, y, z).
2. Path to pdb file: This is to make viewing the pose easier later on. This will be a column that contains the path to the pdb file representing the final results of the docking simulation. It should be noted that this path will be specific to one machine.
3. Distance to PTB domain. Using the average ligand position we can calculate an approximate euclidian distance to the PTB dommain. This value is unsigned.

These data points can be added to aggregated score files using the extend_agg_file.r R script, namely the `extend_agg_file` function contained therein. This function will produce a `RDS` file that can then be opened using the R function `readRDS`.

2.7 Optionally create "energy sink" plots

While not strictly required for the analysis you can use the energy_sink R script in order to create plots that compare distance to the PTB domain (or whatever location is specified when running extend_agg_file.r) to total score and interface delta X of all, or a subset of the simulations. An example plot selecting for the top 15% of results is shown below.



2.8 Identify best poses

With all the results of the docking simulations aggregated into one file, they can be reviewed using any program capable of reading large deliminated text files (I will be using R for the rest of this document).

Two primary metrics where used to access to docking quality of a specific pose.

1. `total_score`: A measure of the overall stability of the protein-ligand complex. Lower values indicate increased stability.
2. `interface_delta_X`: The difference in protein structure stability with and without the ligand in complex. Lower values indicate the ligand has a greater stabilizing effect on the protein structure upon binding and therefore potentially higher affinity.

In order to balance out the two metrics I also assessed poses using a combined metric which considered both interface delta X and total score. This is because I saw a non-zero number of cases where one metric would be very low and the other very high or vice versa.

$$s = 2\frac{s_t - min(s_t)}{max(s_t) - min(s_t)} - 1 + 2\frac{i_\Delta - min(i_\Delta)}{max(i_\Delta) - min(i_\Delta)} - 1$$

Where

- s_t = total score
- i_Δ = interface delta X

This normalizes both total score and interface delta x between -1 and 1 and then adds them together. Poses with both low interface delta X and total score will be ranked higher.

I haven't really seen a combined metric like this used very often, or at least not explicitly in the papers I looked through so I mainly used it as an indicator to access which primary metric might make more sense to use.

3 Results

This section describes the results of specific docking simulations and the PyMol session file which accompanies this document.

3.1 Docking RTX60933293

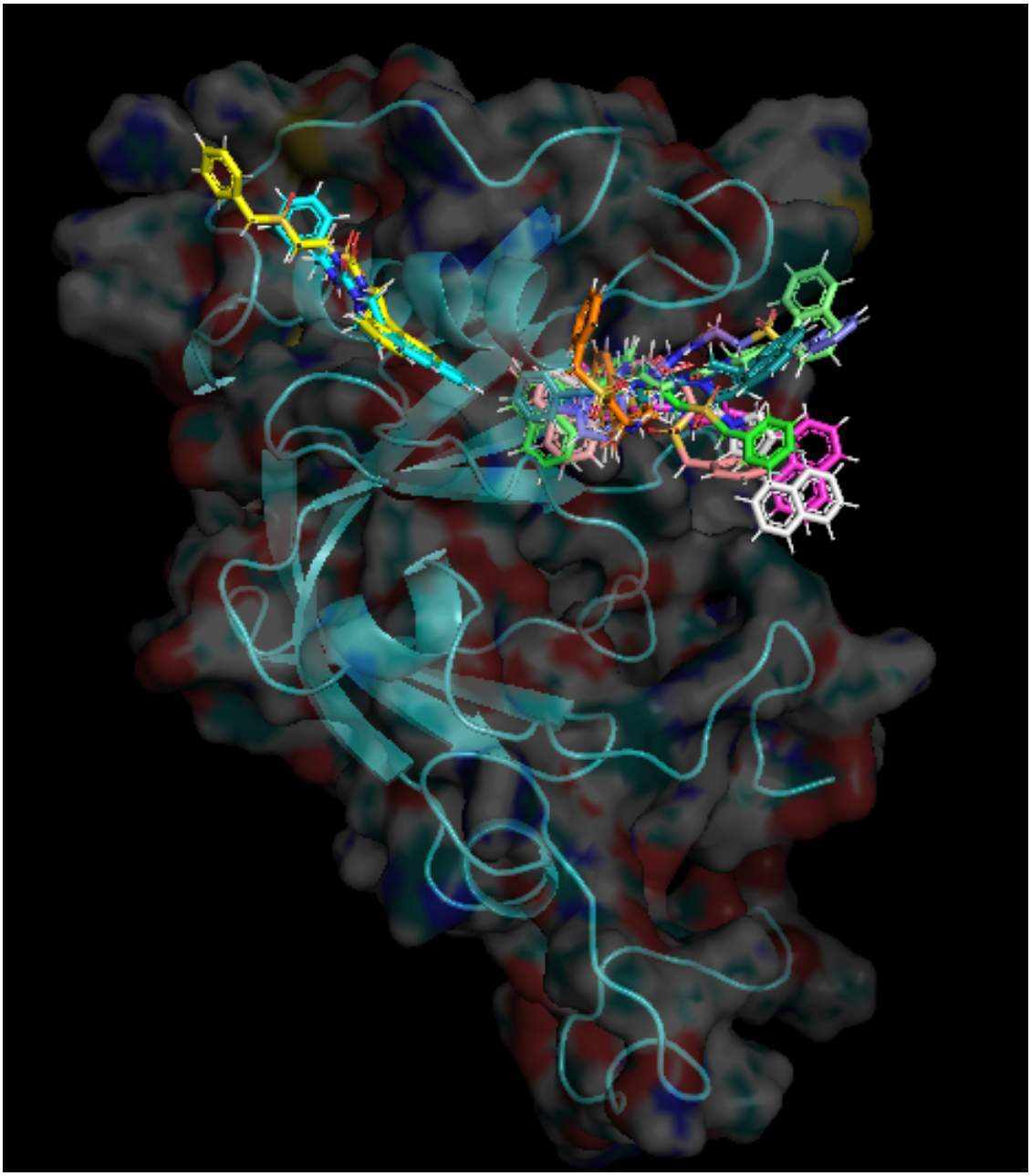
RTX60933293 was docked into a repacked version of `1OY2` (`Shc1-PTB_1OY2_0061.pdb`) using a library of 100 possible conformers and randomized starting positions. 27847 different poses were scored.

3.1.1 Best poses by total score

Filename	Total Score
Shc1-PTB_1OY2_0061_RTX60933293_3913.pdb	-176.816
Shc1-PTB_1OY2_0061_RTX60933293_2879.pdb	-173.871
Shc1-PTB_1OY2_0061_RTX60933293_0313.pdb	-172.901
Shc1-PTB_1OY2_0061_RTX60933293_3287.pdb	-169.206
Shc1-PTB_1OY2_0061_RTX60933293_3917.pdb	-169.001
Shc1-PTB_1OY2_0061_RTX60933293_2406.pdb	-168.58
Shc1-PTB_1OY2_0061_RTX60933293_3818.pdb	-167.529
Shc1-PTB_1OY2_0061_RTX60933293_1493.pdb	-166.992
Shc1-PTB_1OY2_0061_RTX60933293_0693.pdb	-166.676
Shc1-PTB_1OY2_0061_RTX60933293_1167.pdb	-166.612

Link to Pymol Session

Eight of the top ten ligands localize in a "hole" near residue 76 when ranked by total score.



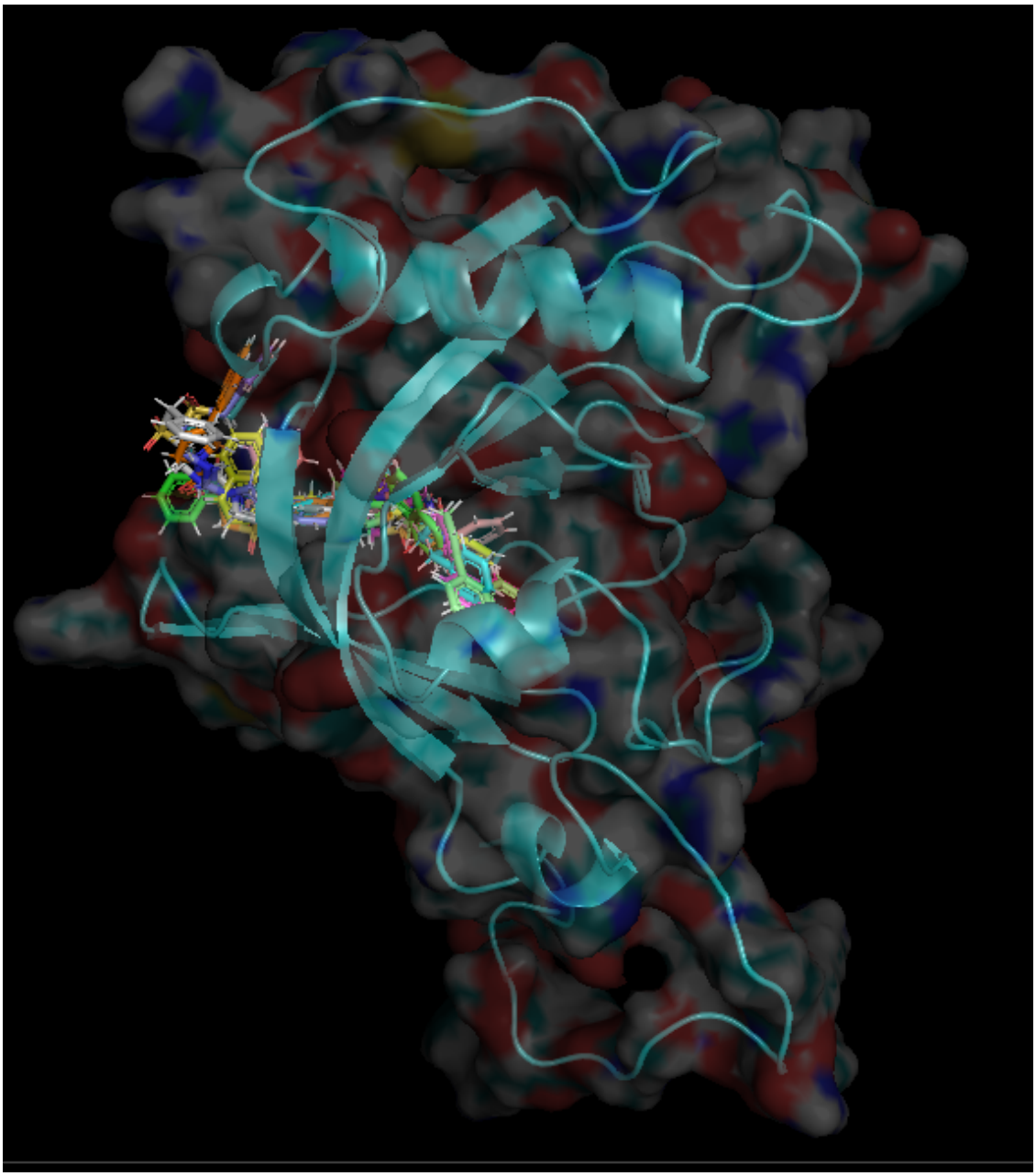
3.1.2 Best poses by interface delta X

Filename	Interface Delta X
Shc1-PTB_1OY2_0061_RTX60933293_0388.pdb	-22.15
Shc1-PTB_1OY2_0061_RTX60933293_3583.pdb	-21.459
Shc1-PTB_1OY2_0061_RTX60933293_1898.pdb	-21.169
Shc1-PTB_1OY2_0061_RTX60933293_2943.pdb	-20.351
Shc1-PTB_1OY2_0061_RTX60933293_2844.pdb	-20.236
Shc1-PTB_1OY2_0061_RTX60933293_0260.pdb	-20.069
Shc1-PTB_1OY2_0061_RTX60933293_1709.pdb	-20.056
Shc1-PTB_1OY2_0061_RTX60933293_0925.pdb	-18.502

Filename	Interface Delta X
Shc1-PTB_1OY2_0061_RTX60933293_1029.pdb	-18.345
Shc1-PTB_1OY2_0061_RTX60933293_0839.pdb	-18.107

Link to Pymol Session

When ranking poses by interface delta X, ligands cluster to a pocket on the other side of the protein, in closer proximity to the beta sheet.



3.2 RTX73145433

RTX73145433 was docked into a repacked version of `1OY2` (`Shc1-PTB_1OY2_0061.pdb`) using a library of 100 possible conformers and randomized starting positions. 11851 different poses were scored. The reduced number of poses was due to some jobs being failing to complete after running out of memory of the cluster.

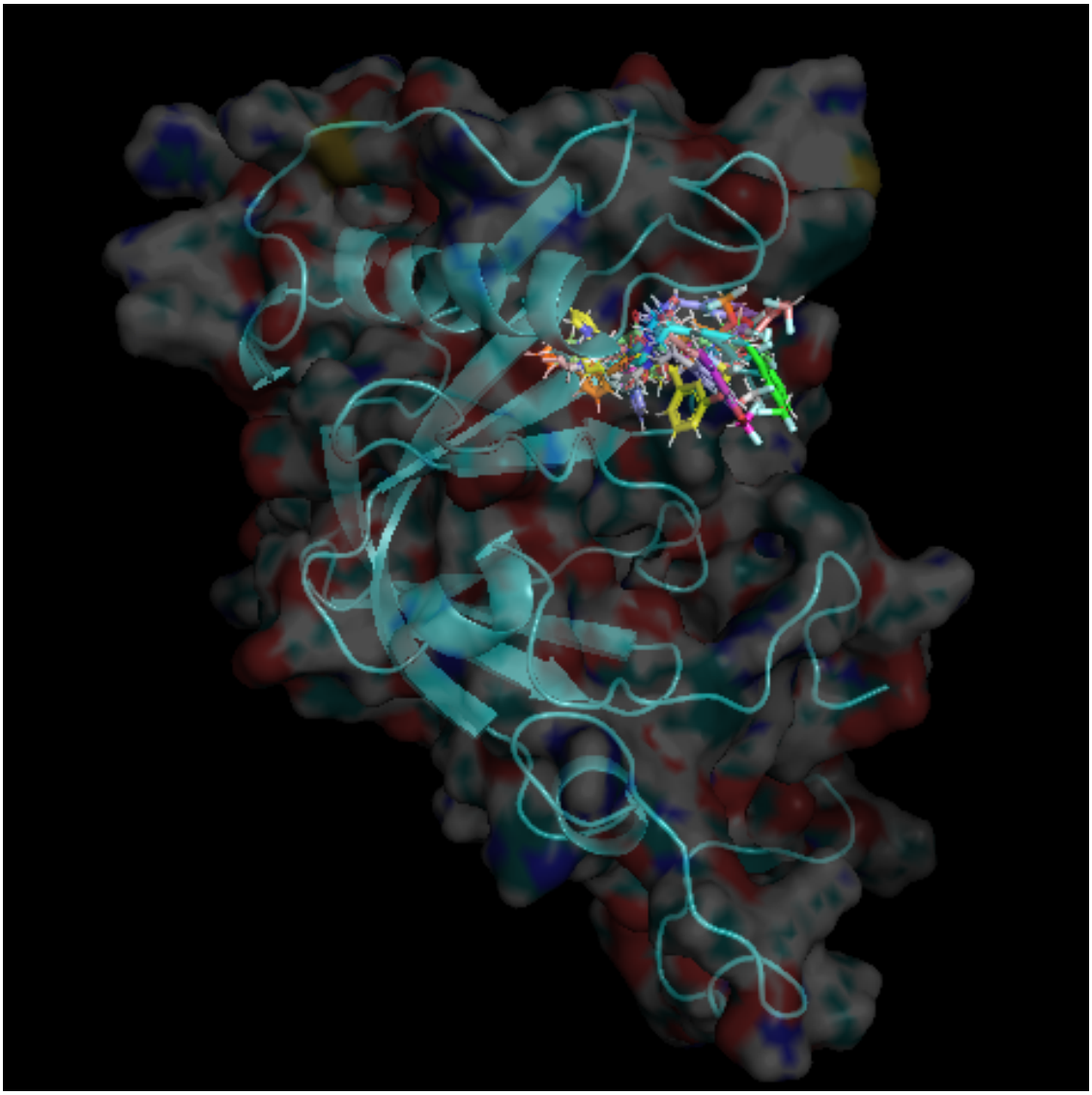
3.2.1 Best poses by total score

Filepath	Total Score
Shc1-PTB_1OY2_0061_RTX73145433_2004.pdb	-181.271
Shc1-PTB_1OY2_0061_RTX73145433_1670.pdb	-175.014
Shc1-PTB_1OY2_0061_RTX73145433_2468.pdb	-174.888
Shc1-PTB_1OY2_0061_RTX73145433_3451.pdb	-174.707
Shc1-PTB_1OY2_0061_RTX73145433_0529.pdb	-174.689
Shc1-PTB_1OY2_0061_RTX73145433_1889.pdb	-173.494
Shc1-PTB_1OY2_0061_RTX73145433_1610.pdb	-172.676
Shc1-PTB_1OY2_0061_RTX73145433_0417.pdb	-172.376
Shc1-PTB_1OY2_0061_RTX73145433_1412.pdb	-171.933
Shc1-PTB_1OY2_0061_RTX73145433_3579.pdb	-171.669

The csv version of this table is available from this link

Link to Pymol session

In general, when accessing ligands by total score, there is extremely strong preference for a "hole" hear residue 76.



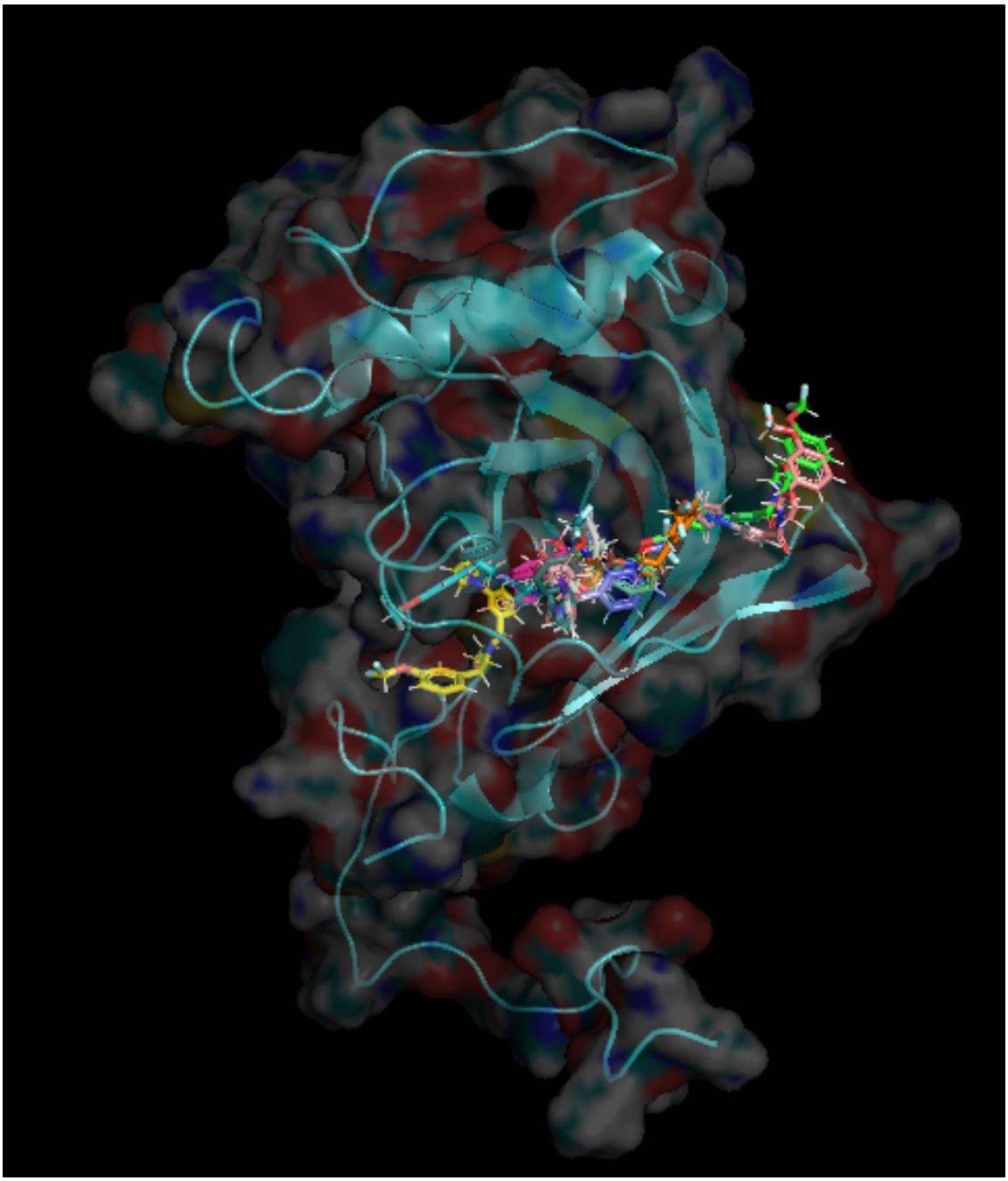
3.2.2 Best poses by interface delta x

Filename	Interface Delta X
Shc1-PTB_1OY2_0061_RTX73145433_2082.pdb	-24.12
Shc1-PTB_1OY2_0061_RTX73145433_2480.pdb	-23.709
Shc1-PTB_1OY2_0061_RTX73145433_2282.pdb	-23.628
Shc1-PTB_1OY2_0061_RTX73145433_2123.pdb	-23.275
Shc1-PTB_1OY2_0061_RTX73145433_3408.pdb	-23.015
Shc1-PTB_1OY2_0061_RTX73145433_0811.pdb	-22.881
Shc1-PTB_1OY2_0061_RTX73145433_1380.pdb	-22.648
Shc1-PTB_1OY2_0061_RTX73145433_2358.pdb	-22.411
Shc1-PTB_1OY2_0061_RTX73145433_2155.pdb	-22.266
Shc1-PTB_1OY2_0061_RTX73145433_0099.pdb	-22.186

The csv version of this table is available from this link

Link to Pymol session

When accessing the best poses by lowest interface delta X ligands tend to cluster on the opposite side of Shc (comapred to clustering as measured by total score)



However, some of these poses have very high total scores including some with positive values, while none of the best ligands as ranked by total score had a positive interface delta X.

When accessing using the combined metric, ligands generally look like those scored using only total score.