

Scientific test: ligand_docking

SAMPLING FAILURES

ligand: 0

ref2015: 0

SCORING FAILURES

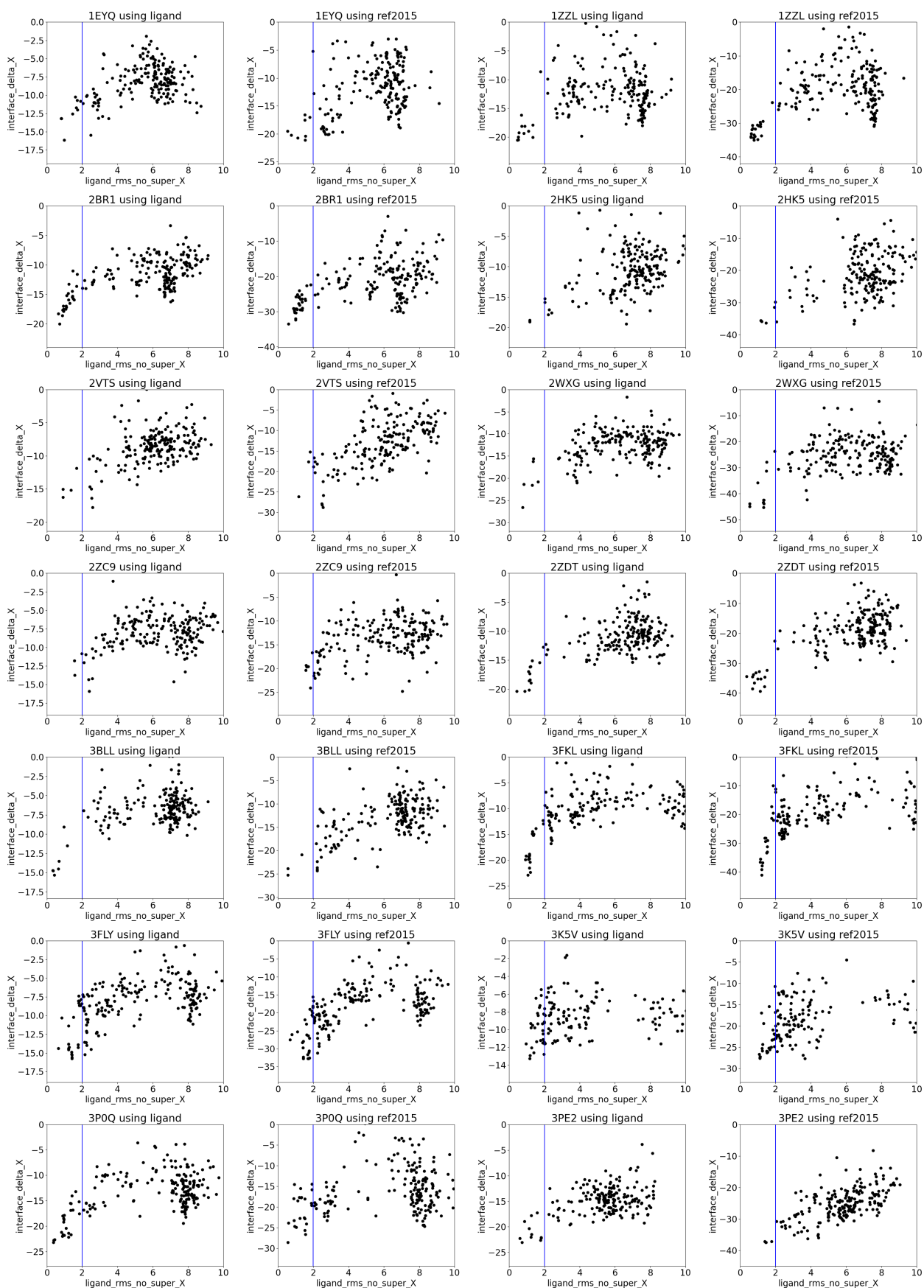
ligand: 18

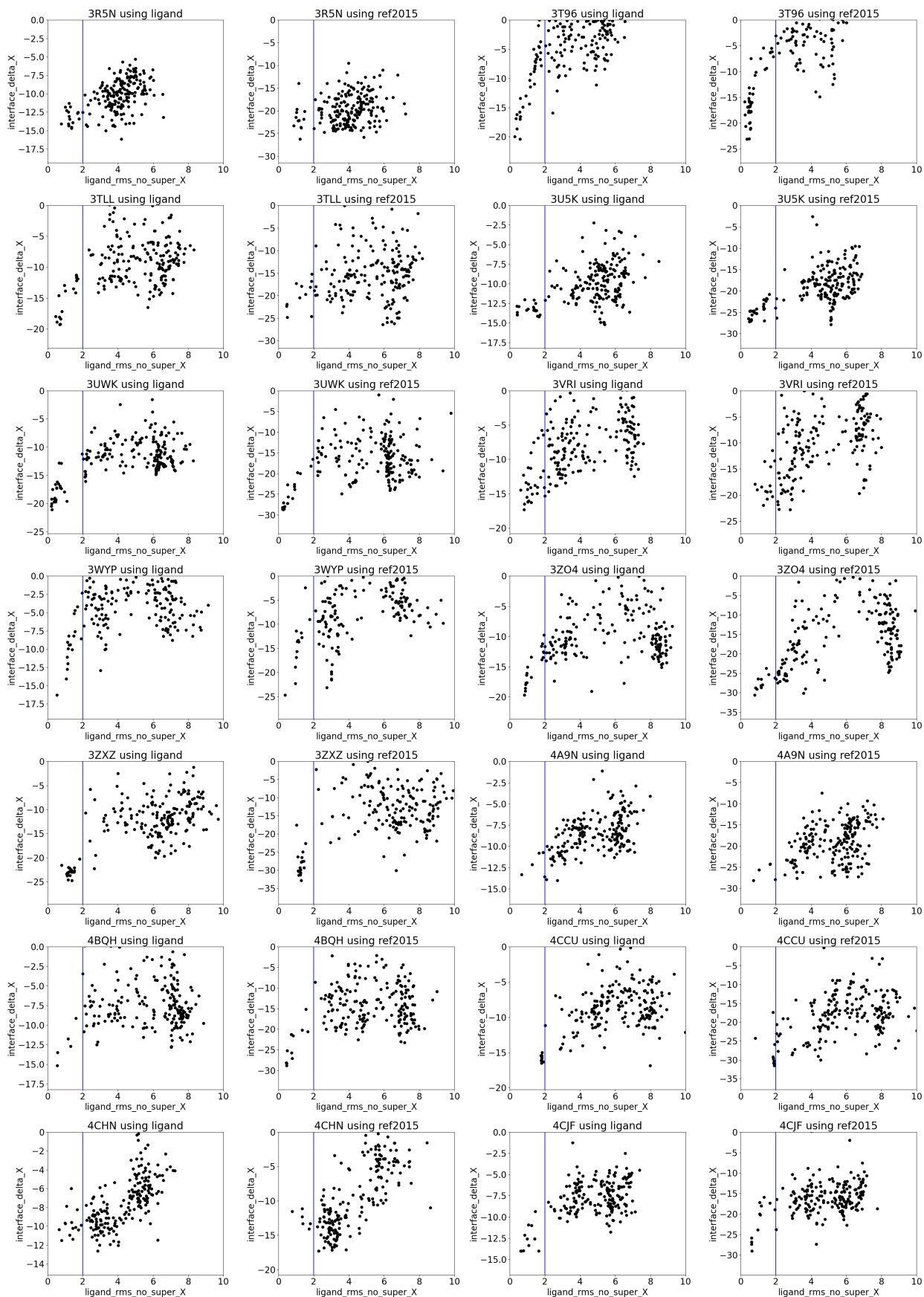
1EYQ 2VTS 2ZDT 3BLL 3FLY 3K5V 3U5K 3ZXZ 4CCU 4CJF 4FA2 4FPJ 4KTN 4MCC
4UAL 5CTU 5D7P 5FQV

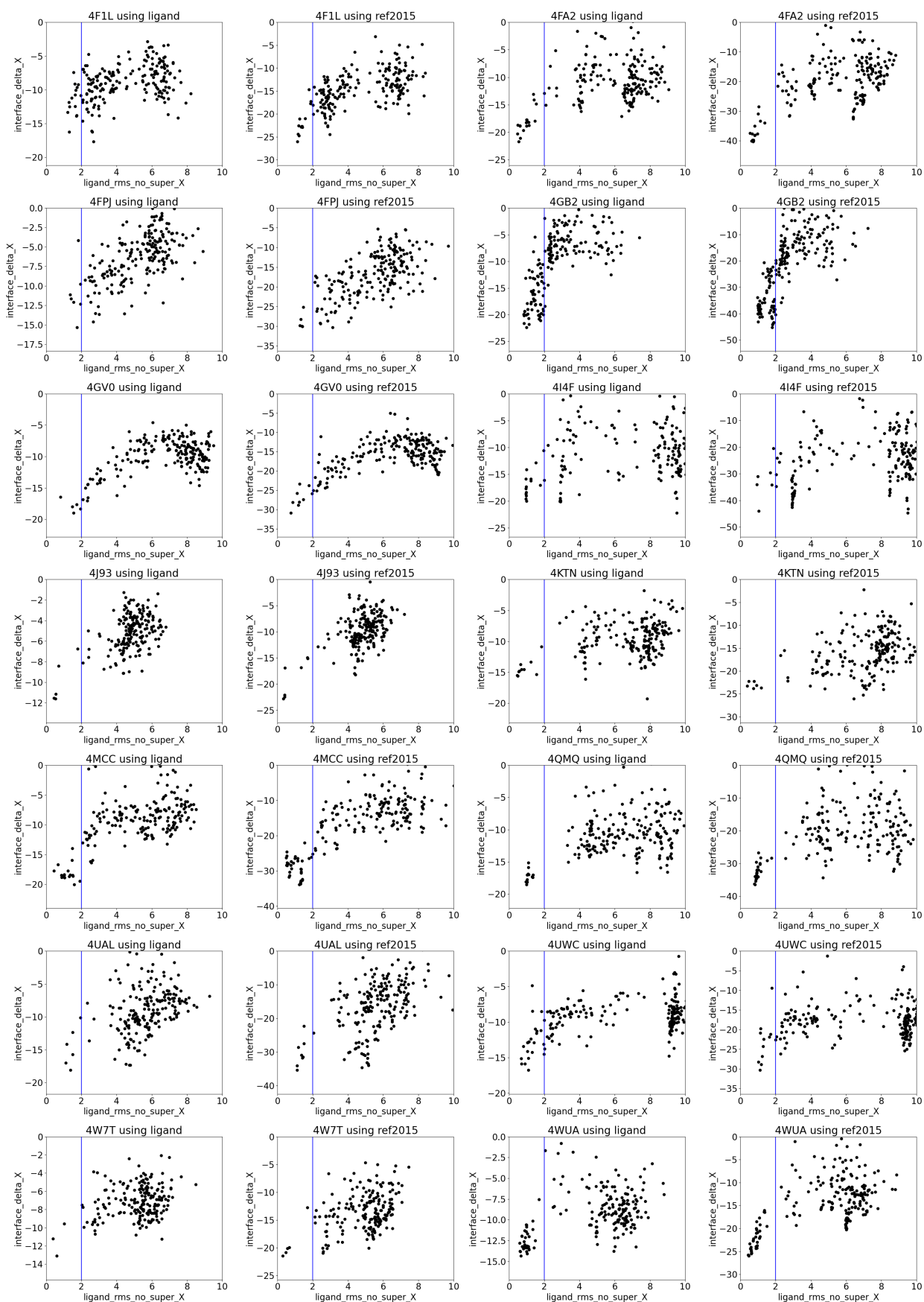
ref2015: 22

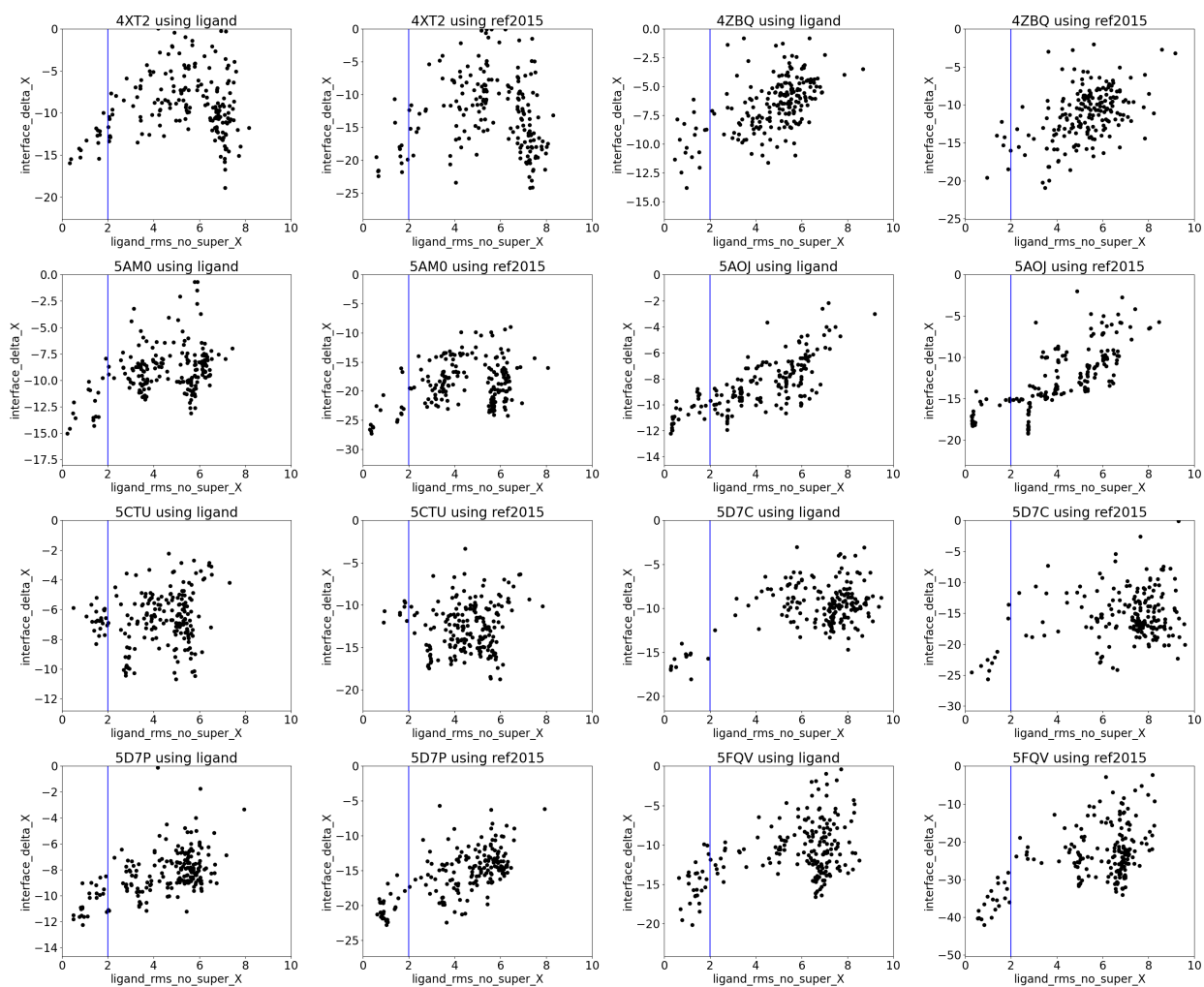
1EYQ 2VTS 3BLL 3FLY 3K5V 3PE2 3TLL 3U5K 3UWK 4BQH 4CCU 4CHN 4F1L 4FPJ 4I4F
4KTN 4UWC 5AM0 5CTU 5D7C 5D7P 5FQV

RESULTS









AUTHOR AND DATE

Adapted for the current benchmarking framework by Shannon Smith (shannon.t.smith.1@vanderbilt.edu; Meiler Lab), Feb 2019

PURPOSE OF THE TEST

This benchmark is meant to test how well we discriminate native small molecule binding orientations from decoys based on the interface score term by performing standard ligand docking experiments across 50 diverse protein-ligand complexes.

BENCHMARK DATASET

The dataset consists of 50 protein-ligand complexes extracted from the Diverse Platinum Dataset (Friedrich, N. et al. J. Chem. Inf. Model, 2017). In addition to the filters described thoroughly in Friedrich et al., the maximum resolution allowed was reduced to 2.0Å, drug-like ligands were chosen using standard Lipinski Rules (Lipinski, C.A., et al. Advanced Drug Delivery Reviews, 1997), and visually inspected for unrealistic orientations and crystallographic artefact. This dataset was also filtered to eliminate cofactors or multiple ligands within the binding site. The full list of protein-ligand complexes is given in the 1.submit.py file by PDB ID.

PROTOCOL

Structure preparation:

Proteins were extracted directly from the PDB according to their PDBIDs, underwent minimal cleaning to remove the ligand. Note that no prior relax or other structural manipulations were performed prior to beginning the docking run.

Ligand preparation:

Initial ligand structures were downloaded from the Protein Data Bank using the ligand ideal SDF. Note that this is not the same as the structure from the input cocrystal structure in order to minimize conformational bias towards a particular pre-generated pose. Ligand files were cleaned using OpenBabel (J. Cheminf. 2011, 3, 33), run through BCL Conformer Generator (Kothiwale, Meiler. J. Cheminform., 2015) and generated Rosetta-readable parameter files according to the following scripts:

BCL Conformer Generator

```
bcl.exe molecule:ConformerGenerator -rotamer_library cod -top_models 100 -  
ensemble_filenames NAME.sdf -conformers_single_file NAME_conformers.sdf -  
conformation_comparer 'Dihedral(method=Max)' 30 -max_iterations 1000
```

Rosetta Parameter File Generation:

```
/programs/x86_64-linux/rosetta/3.8/main/source/scripts/python/public/molfile_to_params.py -n  
$NAME -p $NAME --mm-as-virt --conformers-in-one-file NAME_conformers.sdf --chain X
```

This protocol currently uses 50 protein-ligand complexes, each containing the input file containing the cleaned protein + input ligand PDB (target_input.pdb), the native protein-ligand complex for RMSD calculations (target_native.pdb), the ligand params file (target_ligand.params) and the ligand conformer library file pointed to by the params file (target_ligand_conformers.pdb).

NOTE: the protein and ligand preparation steps were performed previously and are given as the input in the data/ directory. In other words, these steps are not performed each time this benchmark is run.

Each test takes ~8 CPU hours x 50 tests = ~400 CPU hours.

PERFORMANCE METRICS

The big question that this benchmark intends to test is how well we discriminate native versus non-native binding poses. A run is determined successful if there is an near-native (<2Å) structure within the top 1 percent of models based on the interface_delta_X score.

Sampling failure is defined as having no sub-2Å output structures.

Scoring failure is defined as not having a sub-2Å output structure within the top 10% ranked by interface score (this is a pretty large margin and may adjust accordingly).

A pass/fail is defined in the following way: Each score function has a number of allowed failures, i.e. for how many targets a scoring failure exists, out of 50 targets. The cutoffs for these were derived by looking at 10 runs, taking the maximum number of targets that fail, plus 2. The cutoffs for these are defined in the 9.finalize.py step. If any scorefunction has more targets failing than this cutoff, the entire test fails.

KEY RESULTS

This benchmark is meant to be a longitudinal test to determine how changes in the scorefunction impact small-molecule docking performance.

Performance varies greatly across different test cases, so I am not sure how to go about grading overall performance. This also makes it difficult to define a binary pass/fail criteria to the entire benchmark.

DEFINITIONS AND COMMENTS

There is a bit of noise in the runs, so you may need to compare several runs to get better sense of how different scorefunctions compare.

LIMITATIONS

The run-to-run variability is rather high, which might indicate that the benchmark could be improved by running several output structures for each input, rather than just the current one.

This benchmark currently utilizes the ligand scorefunction (an off-shoot of the score12 environment), as this is currently the best protocol for ligand docking in Rosetta. Could be updated for talaris2014, but performance has been shown to deteriorate in the newer scoring environments, notably ref2015 and later.