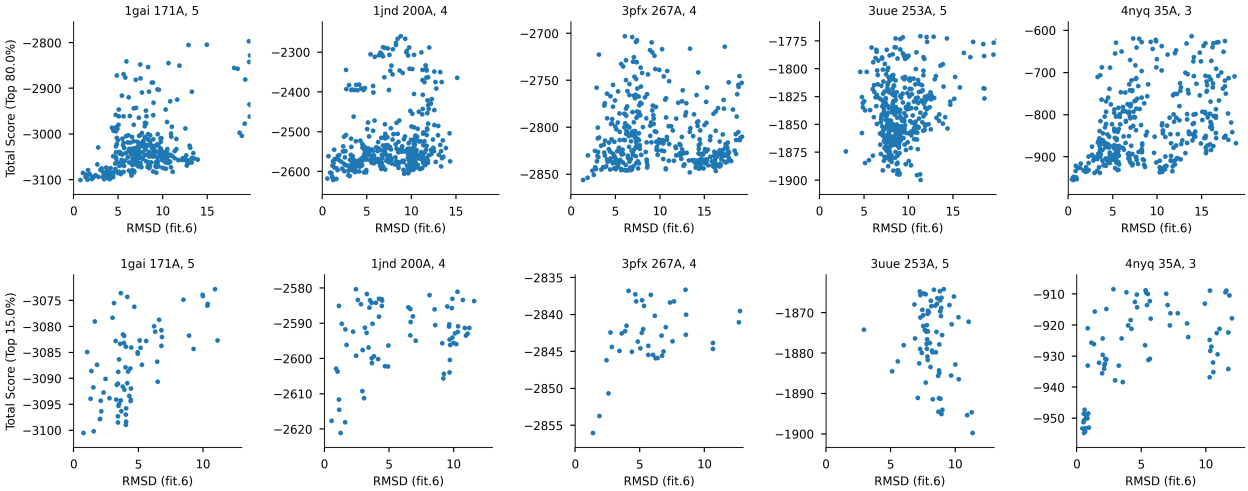
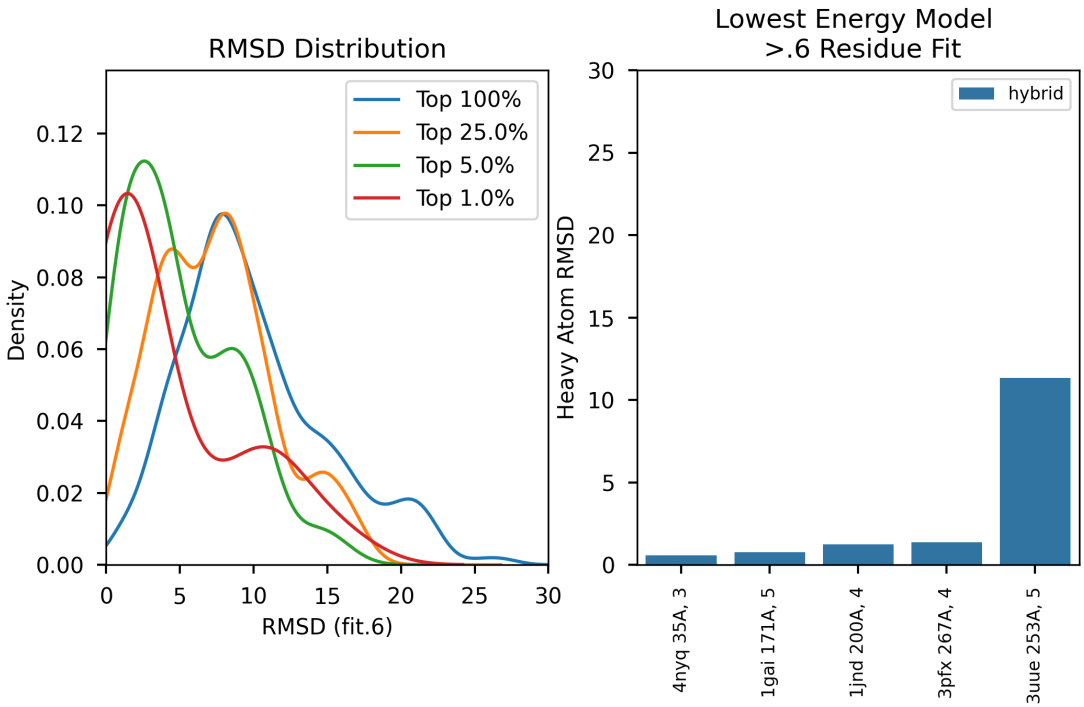
Scientific test: glycan_structure_prediction

FAILURES

RESULTS





AUTHOR

Jared Adolf-Bryfogle (jadolfbr@gmail.com) 9/5/2019

PURPOSE OF THE TEST

This test ensures that the GlycanTreeModeler retains its scientific performance. This application is intended to predict the structures of glycan trees on a protein surface

What does the benchmark test and why?

The benchmark tests 5 input structures from our 26 total used for (Adolf-Bryfogle, Labonte et. al, 2021).

The benchmark uses Symmetry to represent the original crystal environment of the inputs, as well as crystal densities.

Densities are used to determine the fit of each residue into the actual density - which is ultimately used to calculate RMSDs.

The XML script includes a number of SimpleMetrics that we use to determine performance.

Currently, the test is setup in a way to determine sampling of near-native structures.

The plots help to determine the overall performance of the benchmark.

BENCHMARK DATASET

How many proteins are in the set?

- Five proteins, each test modeling a single glycan tree. These all had density under 100 mb, so we can actually have them in the repository.

What dataset are you using? Is it published? If yes, please add a citation.

- The dataset will be published in the paper (Adolf-Bryfogle, Labonte et. al 2021).

What are the input files? How were the they created?

- The inputs are refined PDBs. Each input was relaxed in parallel for 10 structures a piece.

- Relax was done in the presence of the actual crystal density built from phenix.maps, using the fast_elec_dens score term.

PROTOCOL

State and briefly describe the protocol.

The protocol starts with randomizing all glycan backbone torsions and then modeling the glycan tree from the roots out to the leaves in layers.

Besides layer 1, all other residues start as virtual - and as the tree is built up, these residues are un-virtualized.

A layer is defined as the number of residues to the root, and this allows us to model the correct residues together.

Each build cycle has a GlycanSampler run internally, which is essentially a WeightedSampler made up of different moves.

These moves consist of:

- random purturbations (of small, medium, large moves)

- packing of glycans and protein residue neighbors

- conformer sampling based on a new bioinformatic analysis of the PDB

- sampling based on the sugar_bb energy term as probabilities

- minimization on a random residue and it's glycan children

- shear moves

Is there a publication that describes the protocol?

Some of the original sampling can be found in the RosettaCarbohydrates publication, listed in citations.

The core of the protocol is currently being benchmarked, with a paper to come in a few months.

How many CPU hours does this benchmark take approximately?

Estimates run at about 950 CPU hours.

PERFORMANCE METRICS

What are the performance metrics used and why were they chosen?

Except for 3UUE, which is not considered for a pass/fail, the following must be true for this test to pass:

- 1JND and 4NYQ must have at least one model < 1.0 A

- 1GAI and 3PFX must have at least one model < 5.0 A

These were determined based on a a similar benchmark for the paper. If we had more computational power, these cutoffs would be

much more rigorous. 3UUE has the least rigorous cutoffs and lacks a good score funnel with REF2015

How do you define a pass/fail for this test?

Failure of any of the above.

How were any cutoffs defined?

Arbitrarily, like so much else in Rosetta. These are based on the performance of Rosetta in predicting the crystal structure

of these proteins during spring 2019.

KEY RESULTS

What is the baseline to compare things to - experimental data or a previous Rosetta protocol?

Past iterations of this test.

Describe outliers in the dataset.

Note that 3uue is the worst performing glycan, and its cutoffs are not high. It is the only one that does not have a good funnel.

DEFINITIONS AND COMMENTS

State anything you think is important for someone else to replicate your results.

N/A.

LIMITATIONS

What are the limitations of the benchmark? Consider dataset, quality measures, protocol etc.

Computational power. 1k nstruct is usually the bare minimum (for these benchmarks, we have had to reduce that to 500 as well).

For a particular input, we recommend 5-10k.

How could the benchmark be improved?

More processing power. Cutoffs for score vs RMSD

What goals should be hit to make this a "good" benchmark?

Improved cutoffs once the full benchmarking (for the main glycan modeling paper) is complete.

Increased nstruct once new nodes come online.