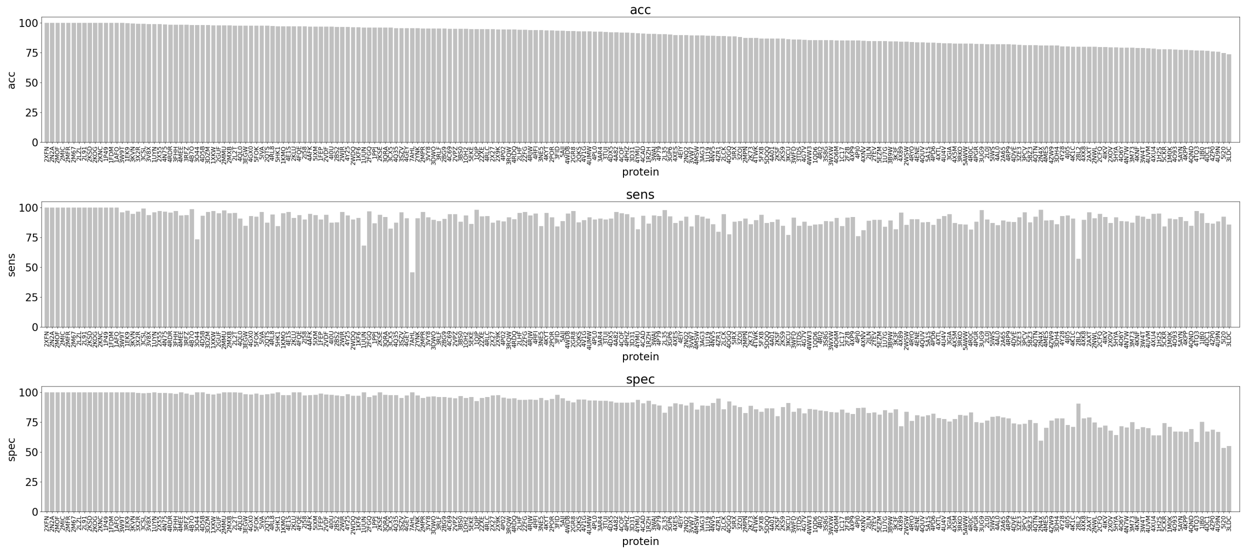
Scientific test: mp_lipid_acc

FAILURES

3WU2
4WW3

RESULTS



## AUTHOR AND DATE

The benchmark was set up by Julia Koehler Leman (julia.koehler.leman@gmail.com) in March 2019.

The PI is Richard Bonneau.

## PURPOSE OF THE TEST

The benchmark tests the quality of the concave hull algorithm that underlies the algorithm to classify lipid accessibility from structure. Identifying lipid exposed residues from a membrane protein structure differs from SASA (solvent-accessible surface area) algorithms in that pore-forming proteins would have pore-facing residues to be predicted as solvent-accessible, but wouldn't distinguish water vs. lipid as solvent. SASA algorithms would predict exposed residues as solvent-exposed, yet our mp_lipid_acc algorithm predicts residues that are purely exposed to lipid.

## BENCHMARK DATASET

We test the protocol on a diverse set of 223 membrane proteins. We wanted to include structures that are helical, beta-barrels, single TM span proteins, proteins with large pores, small pores, multiple pores (aquaporin-like), oval (like chloride channels) etc. How the dataset was created is described in detail in (Koehler Leman et. al, BMC BioInfo, 2017).

The algorithm was developed by testing a small set of proteins (also in the publication). Then, the algorithm was run on the benchmark set and the output was further manually curated via PyMol. For residues that are lipid-exposed, we set the B-factor to 50, for others, the B-factor is 0. The entire benchmark set was therefore hand-curated. This dataset is given in

Rosetta/main/tests/scientific/data/mp_lipid_acc/db_hand_curated

The input dataset is the set of PDBs in

Rosetta/main/tests/scientific/data/mp_lipid_acc/db_input

the output of which is compared against the hand-curated set.

## PROTOCOL

Note that computing lipid accessibility from structure is a pure geometric algorithm that does not consider any other score terms. The input is a membrane protein already transformed into membrane coordinates. The details of the algorithm are described in detail in (Koehler Leman et. al, BMC BioInfo, 2017) and the command lines are given in the supplement to the paper. Note that we are testing default parameters here, which are:

slice width 10 A

distance cutoff 10 A

shell radius 6 A

angle cutoff 65 deg

The runtime is very short and we only need to build a single model since the algorithm is deterministic. It takes about 30s per model x 223 proteins = 6690s which is <2h CPU time.

## PERFORMANCE METRICS

Accuracies are computed for each protein individually: between the hand-curated set and predicted output from mp_lipid_acc. The protein-accuracy set was sorted by accuracy from largest to smallest and the list of proteins was used as input into the algorithm. Now, if at any run the accuracies are not sorted from largest to smallest any more, we call this a failure. Note that the predictions are deterministic and there is no reliance on a Monte-Carlo algorithm. Therefore, if the accuracies have changed, the underlying concave hull algorithm must have changed. Further, we will flag when accuracies have changed for a small set of proteins, as it wouldn't be flagged as a failure when all accuracies would have gone up or down by an equal amount.

We also plot the sensitivities and specificities on the set of proteins.

## KEY RESULTS

## DEFINITIONS AND COMMENTS

## LIMITATIONS

The benchmark set is chosen well and incredibly diverse, I don't think large improvements would be expected if the benchmark set were picked 'better'. The algorithm works well but I am sure there is room for improvement. By developing the algorithm I have tested all-atoms vs. CA-only and didn't see a marked improvement, but the runtimes increased quite dramatically, so we stick to CA-atoms only. The algorithm could be improved somehow because we use relatively thick slices (10A thick) and extrapolate from that. Also, the membrane thickness should be a multiple of the slice width to avoid sparse data in the last slice. It would be nice if the slices could be smaller and the dependency to the membrane thickness could be removed. Another idea might be to use a 3D concave hull algorithm, which I am not quite sure whether and how well this would work, since we are looking at a 2D membrane plane.