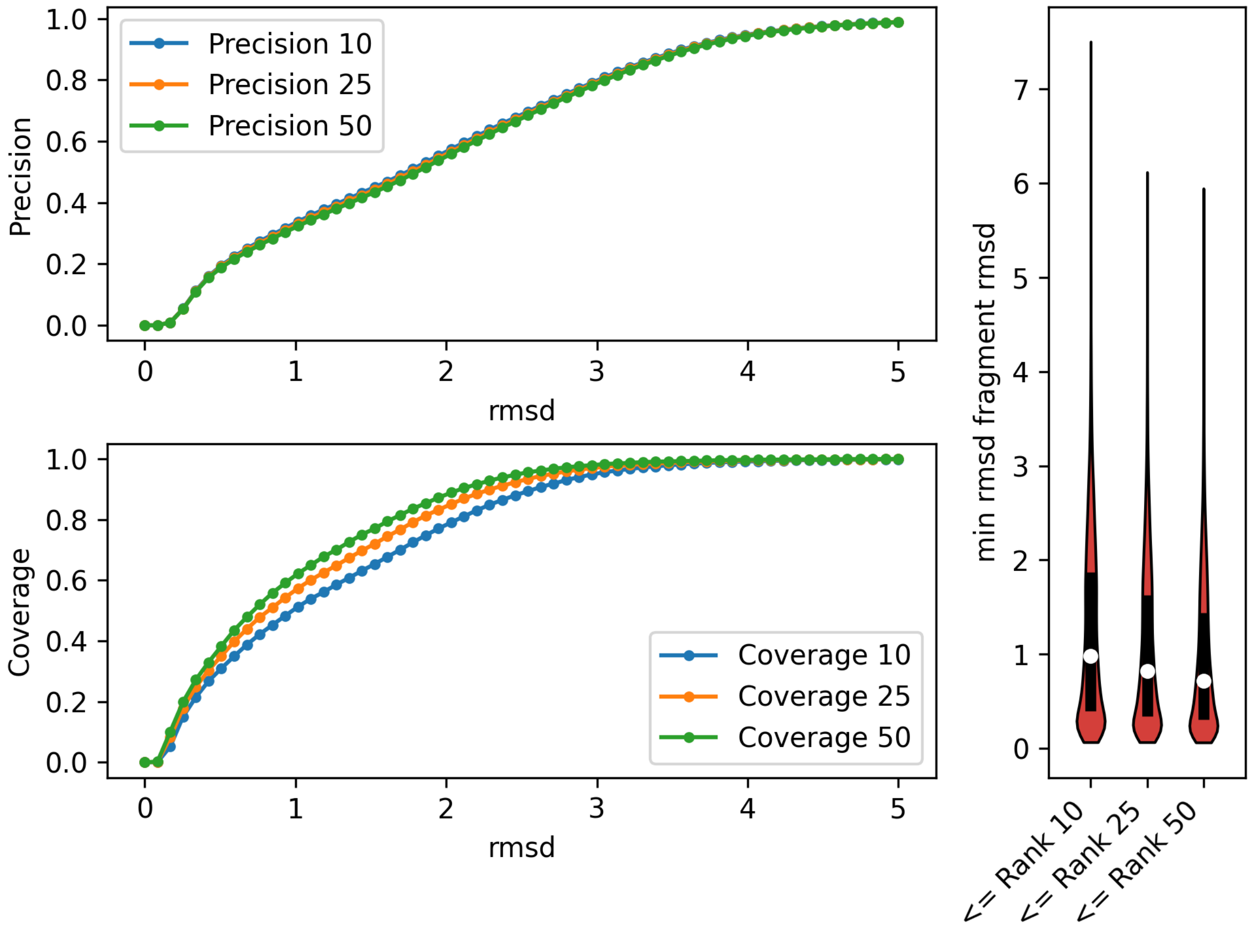
Scientific test: make_fragments

FAILURES

None

RESULTS



AUTHOR AND DATE

Author: Daniel Farrell

Email: danpf@uw.edu

PI: Frank DiMaio

@DD-MM-YYYY 30-06-2019

PURPOSE OF THE TEST

This tests how fragments are made and the results of fragment picking

What does the benchmark test and why?

This test benchmarks the whole 'make fragments' pipeline which includes:

```
- Running blast
- Running psipred
- Building the nr database
- running rosetta fragment picker
```

Why test the whole pipeline instead of just fragment_picker?

```
- Most improvements to the pipeline (unpublished so far) have mainly been
    - Improved databases
    - Improved SS prediction
    - Alternative methods of PSSM generation
    - Addition of additional outside information
- Database related questions often come up, this will give us a baseline t
    - The NR database is unversioned -- This gives us, and the community a
- The make-fragments pipeline relys on a lot of non-rosetta downloads, and
```

therefore we can come up a sort of pros and cons list for whole pipeline vs rosetta part only:

```
- Pros for whole pipeline:
    - Scientifically -- fragment picking is the sum of the whole, and test
    - Most community documentation is based on 'make_fragments.pl' and it
    - Because most improvements have been database/outside input improveme
    - When someone inevitably updates the pipeline, they will be able to e
- Cons against the whole pipeline:
    - It takes a very long time to run
    - It is multithreaded (which is incompatible with CONDOR)
    - Because outside programs are run -- we are scientifically checking t
```

In the end, there is no right or wrong answer, however, due to a signifigant number of forum, and email questions relating to the install

of fragment picking to local machines + the hopes that someone will be able to easily update the 2 perl scripts to something more manageable

(like python) I have chosen to benchmark the whole pipeline.

BENCHMARK DATASET

How many proteins are in the set?

From casp12: 51 proteins

From casp13: 14 proteins

Total: 65 proteins

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

casp12 and casp13

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

They are pulled pdbs from the casp website and manually renamed by me (Daniel Farrell). The set has the format:

```
{
        "casp_12": {
                "target_id": {
                        "pdb_text": "pdb_text...",
                        "target_sequence": "MYSEQVENCE..."
                }
        }
        ...
```

PROTOCOL

State and briefly describe the protocol.

The input file is a sequence, we take the sequence and run 'make_fragments.pl' on it. This tests how the most general/public facing fragment picking protocol works. The perl script runs and we get 3mers and 9mers from it. The 9mers are used for testing purposes and are compared against the native pdb if a full 9 residue match exists. those results are curated and plotted.

make_fragments.pl works briefly by:

```
- run `psiblast` for pssm generation
- run `psipred` for ss-prediction
- run `psiblast` for homolog detection
- run sparksX for phi psi and solvent accessibility predictions
- run `fragment_picker` to finally pick fragments
```

The protocol isn't ideal (multiple psiblast runs), however this is what most people use to pick fragments (if not robetta). The main differences between this and robetta are alternative methods of ss-prediction, and robetta has integrated the hhsuite into their fragment picking methods.

I unfortunately am unaware of the accuracy of the picker without sparksX or psiblast etc. This will simply be a baseline due to the fact that this is essentially a snapshot of the protocol from 5-7 years ago, and most of the people that built it have moved on.

Is there a publication that describes the protocol?

Gront, D., Kulp, D. W., Vernon, R. M., Strauss, C. E. M. & Baker, D. Generalized Fragment Picking in Rosetta : Design , Protocols and Applications. 6, (2011).

How many CPU hours does this benchmark take approximately?

First the nr database has to be built (~20 cores x 1-2 days). Then each total amount of running `make\_fragments.pl` takes about 1 (or 2-3) cores 3-10 hrs. For a total of ~2000 hrs. Different `psiblast` calls can take up to 3 cores (for unknown reasons) but this should work.

PERFORMANCE METRICS

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

I plotted `coverage` and `precision` based on many fragment picking papers such as:

```
de Oliveira SH, Shi J, Deane CM. Building a better fragment library for de
```

`precision` is defined as proportion of good fragments in the library (at various cutoff to define 'good')

`coverage` is defined as the percentage of target residues represented by at least one good fragment in the library

```
coverages = []
for every rmsd_cutoff in 0->5 (60 bins):
        total = 0
        cov_count = 0
        For every residue ( aka fragment set ):
                for every fragment_cutoff in [10, 25, 50]:
                        total += 1
                        if the minimum rmsd in the fragments from 0->cutof
                                cov_count += 1
        coveraged.append(cov_count/total)
```

I also plotted violin plots to show overall distributions of all plotted fragments at different cutoffs (most rosetta protocols use the top 25 fragments, but we pick 200).

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

We can set a pass/fail based on the median and quartile ranges of the top 25 fragments + looking at the deviations from run to run. With an identical database the files that we make are deterministic so we can use this test to determine the changes in all aspects of fragment-picking.

How were any cutoffs defined?

By the current results.

KEY RESULTS

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

The baseline is the solved crystal/nmr structures from casp

Describe outliers in the dataset.

There are no outliers.

DEFINITIONS AND COMMENTS

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

```
- Ask Sergey Lyskov to the stored `nr` database.  We pegged it to `May 25
- There are some executables (sparksX, psiblast, psipred) that we should p
- Most of the improvements in fragment picking over the last few years hav
```

LIMITATIONS

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

This benchmark is specifically crafted to test our capabilities at getting near-rmsd fragments. For some protocols having near-native fragments may not be enough (ie abinitio). Additionally we are limited to the older version of make fragments as the new one has been integrated into robetta and is not yet ready for public release.

How could the benchmark be improved?

```
- Newer fragment picker pipeline from robetta
- multithreaded fragment\_picker
- A non-casp dataset
- less blast/psiblast runs
```

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