Scientific test: Protein Data Bank diagnostic

AUTHOR AND DATE

Steven M Lewis, smlewi@gmail.com, Cyrus Biotechnology, Oct 2018

Sergey Lyskov, sergey.lyskov@gmail.com, Gray lab, Johns Hopkins University, Oct 2018

FAILURES

The PDB diagnostic was run on **194,622** PDB files. **189,899** files passed the test and **4,723** failed!

PURPOSE OF THE TEST

This test tries to load every PDB file in the PDB database and classifies the failures that occur. The command line below shows what was done; broadly all versions of this test examine load-time problems and more expensive versions (`-PDB_diagnostic::skip_pack_and_min false`) also check for errors during scoring, packing, and minimization.

"Hunting down these bugs is the most fun thing you can do on a Thursday morning" - Andy Watkins, probably.

An individual PDB passes or fails this test based on whether it errors out or completes the diagnostic. The test as a whole passes or fails based on a "reference results" system, like an expected result in a unit test. About 700 PDBs and 1200 CIFs fail at the time of this writing; the purpose of the test is to document the failures and watch for new ones, so PDBs failing in an expected manner does not constitute an overall test failure. The test will fail if PDBs pass or fail **UNEXPECTEDLY**, where the expectation is defined by the reference results (see below).

If you find that this page is telling you the test failed because there are **FEWER** errors: **GREAT**! You fixed some bugs! You can update the reference results following the instructions at the bottom of the page.

If you find that there are **MORE** failures than expected, especially non-timeout failures, consider it a warning that recent code changes may have introduced bugs into the PDB reading machinery. If you don't know what's going on: post to Slack or devel. **DO NOT** just update the reference results in this case.

If you want to know more about what this test does - pester Steven Lewis to write proper documentation.

Command line used:

```
/home/benchmark/w/rosetta-2.hpc-c/rosetta/commits/main/source/test/timelimit.py 32 /home/benchmark/w/rosetta-2.hpc-c/rosetta/commits/main/source/bin/PDB_diagnostic.default.linuxclangrelease -no_color -out:file:score_only /dev/null -jd2::delete_old_poses true -ignore_unrecognized_res false -load_PDB_components true -packing::pack_missing_sidechains false -packing::repack_only true -s {input_file} -ignore_zero_occupancy false -in:file:obey_ENDMDL true -PDB_diagnostic::skip_pack_and_min false -PDB_diagnostic::reading_only false
```

4,723 total PDBs failed with the following error codes:

```
2,087 unrecognized_residue
  467 zero_length_xyzVector
  349 fill_missing_atoms
  290 missing_disulfide_partner
  254 missing_bond
```

```
222 「rotlib_file」
196 「bad_patch」
144 「pseudobond_connection_change」
112 「multiple_disulfides」
104 「duplicate_atom_name」
101 「no_orient_atoms」
 81 「aa_difference」
 77 「prepro_cyclic_pep」
 67 「unknown」
 53 「unknown_atom_name」
 31 「reroot_disconnected」
 27 「zero_atom_restype」
 14 「nu_conformer」
 12 「alias_missing_atom」
 10 「unknown_hbond_acceptor」
  9 「no_usable_coords」
  5 「base_of_chi」
  5 「merge_with_next」
  4 「insufficient_mainchain」
  1 「no_hbond_deriv」
  1 「exceed_timeout」
```

Test marked as **FAILED** due to following errors:

```
PDB 6O6E was passing test before but now failed with 「unrecognized_residue」 error!
Its run-log could be found in 「6O6E」
PDB 6QMS was passing test before but now failed with 「unrecognized_residue」 error!
Its run-log could be found in 「6QMS」
PDB 6YD4 was passing test before but now failed with 「unrecognized_residue」 error!
Its run-log could be found in 「6YD4」
PDB 6YD7 was passing test before but now failed with 「unrecognized_residue」 error!
Its run-log could be found in 「6YD7」
PDB 6Z6Y was passing test before but now failed with 「unrecognized_residue」 error!
Its run-log could be found in 「6Z6Y」
```

NOTE: 4622 PDB's passed the tests but was not listed in reference results.

To update reference results please copy the files below into the main repository:
reference-results.full.new.json → 「main repository」 as tests/benchmark/tests/scientific/protein_data_bank_diagnostic/reference-results.full.json
blocklist.full.new.json → 「main repository」 as tests/benchmark/tests/scientific/protein_data_bank_diagnostic/blocklist.json (note the cif/pdb/fast mode is ignored: the blocklist is segfaults and huge PDBs and is shared)