Bradley's Notes on Doing Random Stuff with PDB Structures

Bradley J. Hintze

November 16, 2016

Contents

1 Creating Maps and Kinemages for a Given PDB

This is a rather easy thing to do. We will use `phenix.fetch_pdb`, `phenix.maps` and `phenix.kinemage`.

1. Fetch the PDB files (including SFs) from the PDB and create an SF mtz.

   `$ phenix.fetch_pdb --mtz 1ORN`

2. Create kinemage.

   `$ phenix.kinemage 1ORN.pdb`

3. Calculate 2mF$_o$-DF$_c$ and mF$_o$-DF$_c$ maps as well as map coefficients (needed for `mmtbx.flipbase`).

   ```
   $ phenix.maps map.map_type=2mFo-DFc map.map_type=mFo-DFc
   map_coefficients.map_type=2mFo-DFc 1ORN.pdb 1ORN.mtz
   ```

1.1 bash script

Here it is in a bash script.

```
# $1 is the first argument witch is expected to be a PDB code.
# Note that PHENIX must be souced to run this script.
#
# This script makes a directory with the name of the given PDB
# code and downloads PDB data, creates a kinemage, and
    calculates maps.

mkdir $1
cd $1
phenix.fetch_pdb --mtz $1
phenix.kinemage $1.pdb
phenix.maps map.map_type=2mFo-DFc map.map_type=mFo-DFc
    map_coefficients.map_type=2mFo-DFc $1.pdb $1.mtz
cd ..
```

Save this as get_pdb.sh. To run :
`$ bash get_pdb.sh 1ORN`

2 Simple Refinement

There are lots of refinement options but here I will tell you just the basics. The mos basic refinement is ran as follows:

```
$ phenix.refine 1ORN.pdb 1ORN.mtz
```

To do 10 macrocycles (default = 3):

```
$ phenix.refine 1ORN.pdb 1ORN.mtz main.number_of_macro_cycles=10
```

If you get something like this:

```
    Number of atoms with unknown nonbonded energy type symbols:  11
```

it means that you have ligands that need restraints. To do this run `phenix.ready_set`

```
$ phenix.ready_set 1ORN.pdb
```

This will create a cif file, called `1ORN.ligands.cif`, with the ligand restraints required for refinement.

```
$ phenix.refine 1ORN.pdb 1ORN.mtz 1ORN.ligands.cif
main.number_of_macro_cycles=10
```

3 DNA Stuff

3.1 Flip A Purine 180°

There is a tool within cctbx that can do this and is called *mmtbx.flipbase*. It works by flipping the given purine by 180° about the glycosidic bond and then doing 3 cycles of real-space refinement to bring the purine into the optimal place in the density. This means that you actually need density for this tool to work reasonably – no wishful flipping. As of today (11/8/2016) `mmtbx.flipbase` cannot handle ligands that require cif restraint files in refinement. This is super annoying because `mmtbx.flipbase` only refines the given residues and you should only be flippind canonical DNA bases (if you can and want to fix this issue everyone would appreciate it). The following steps are how to get arround this issue. If you don't have ligands that require cif restraints then you can run `mmtbx.flipbase` on the unmodified PDB (Step 3).

1. Copy the original file.
   `$ cp 1ORN.pdb 1ORNNOHETS.pdb`
2. Delete the HETATM records for the ligands (waters and metals ar OK to keep).
3. Run `mmtbx.flipbase`. It will create 1ORNNOHETS_flipbase.pdb.
   `$ mmtbx.flipbase 1ORNNOHETS.pdb 1ORN_map_coeffs.mtz chain=B res_num=1`
4. Copy the original file.
   `$ cp 1ORN.pdb 1ORN_flip.pdb`
5. Replace the ATOM records for the purine of interest in 1ORN_flip.pdb with the ones in 1ORNNOHETS_flipbase.pdb.

4 bb/sc RSCC Kludge

This is a kludge to get around the fact that `phenix.real_space_correlation` cannot sepoarate backbone and sidechain yet. This is done by creating two pdbs, one with just the bb and the other just sc. We then run `phenix.real_space_correlation` on the two files. The following is for DNA only. It can be easily modified for protein of RNA by modifying the selection criteria. I will give this as a bash script but each line can be ran on the command line separately.

Here it is in a bash script.

```
# $1 is the first argument witch is expected to be a PDB file.
# $2 is the second argument witch is expected to be a SF mtz.
# Note that PHENIX must be souced to run this script.
#
# This script takes a PDB file and creates two files. one with
   just the DNA BB ("xxxxx_bb.pdb") and the other with the DNA
   BB removed ("xxxxx_sc.pdb"). Then phenix.
   real_space_correlation is ran on each file creating xxxxx_bb.
   rsc and xxxxx_sc.rsc.

echo "Keeping DNA bb"
```

```
phenix.pdbtools modify.keep="(name␣'␣P␣␣'␣or␣name␣'␣OP1'␣or␣name
    ␣'␣OP2'␣or␣name␣'␣O5\''␣or␣name␣'␣C5\''␣or␣name␣'␣C4\''␣or␣
    name␣'␣O4\''␣or␣name␣'␣C3\''␣or␣name␣'␣O3\''␣or␣name␣'␣C2\''␣
    or␣name␣'␣C1\'')" output.file_name="$1_bb.pdb" $1

echo "Removing␣DNA␣bb"
phenix.pdbtools modify.remove="(name␣'␣P␣␣'␣or␣name␣'␣OP1'␣or␣
    name␣'␣OP2'␣or␣name␣'␣O5\''␣or␣name␣'␣C5\''␣or␣name␣'␣C4\''␣
    or␣name␣'␣O4\''␣or␣name␣'␣C3\''␣or␣name␣'␣O3\''␣or␣name␣'␣C2
    \''␣or␣name␣'␣C1\'')" output.file_name="$1_sc.pdb" $1

echo "Running␣phenix.real_space_correlation"
phenix.real_space_correlation detail=residue $1_bb.pdb $2 >
    $1_bb.rsc
phenix.real_space_correlation detail=residue $1_sc.pdb $2 >
    $1_sc.rsc
```