

Bradley's Notes on Doing Random Stuff with PDB Structures

Bradley J. Hintze

November 8, 2016

Contents

| | | |
|----------|--|----------|
| 1 | Creating Maps and Kinemages for a Given PDB | 2 |
| 1.1 | bash script | 2 |
| 2 | Simple Refinement | 3 |
| 3 | DNA Stuff | 3 |
| 3.1 | Flip A Purine 180° | 3 |

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.kinemap`.

1. Fetch the PDB files (including SFs) from the PDB and create an SF mtz.
`$ phenix.fetch_pdb --mtz 1ORN`
2. Create kinemap.
`$ phenix.kinemap 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 which is expected to be a PDB code.  
# Note that PHENIX must be sourced to run this script.  
#  
# This script makes a directory with the name of the given PDB  
# code and downloads PDB data, creates a kinemap, and  
# calculates maps.  
  
mkdir $1  
cd $1  
phenix.fetch_pdb --mtz $1  
phenix.kinemap $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 10RN.pdb 10RN.mtz
```

To do 10 macrocycles (default = 3):

```
$ phenix.refine 10RN.pdb 10RN.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 10RN.pdb
```

This will create a cif file, called `10RN.ligands.cif`, with the ligand restraints required for refinement.

```
$ phenix.refine 10RN.pdb 10RN.mtz 10RN.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 flipping canonical DNA bases (if you can and want to fix this issue everyone would appreciate it). The following steps are how to get around 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 are 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`.