Using Coot for model building into EM maps: A tutorial

Alan Brown, Paul Emsley,
MRC Laboratory of Molecular Biology
Francis Crick Avenue
Cambridge, CB2 0QF

04-Sept-2015

Abstract

This tutorial provides an introduction to the use of Coot for EM maps. The test case for the tutorial is F420-reducing NiFe hydrogenase (Frh) {Allegretti:2014tz}. The tutorial should take no more than a couple of hours. If you find *Coot* useful for model building into EM maps, please cite our paper:

Brown A, Long F, Nicholls RA, Toots J, Emsley P, Murshudov G (2015) Tools for macromolecular model building and refinement into electron cryo-microscopy reconstructions. *Acta Cryst.* ***D****71, 136-153*

1. Getting prepared

1.1. Install CCP4 and Coot

The latest version of CCP4 including Coot (http://www.ccp4.ac.uk/) should be installed on your system. Our scripts rely on CCP4 programs being discoverable in your path, so make sure CCP4 is sourced in your shell configuration file, *e.g.* for .cshrc:

- **source /path/to/ccp4-6.5/setup-scripts/ccp4.setup-csh**

1.2. Download the EM scripts

Scripts for model building and refinement into EM maps should be downloaded from:

http://www2.mrc-lmb.cam.ac.uk/groups/murshudov/content/em_fitting/em_fitting.html

1.3. Download the test data

This tutorial uses the EM map and model for the F420-reducing NiFe hydrogenase, Frh {Allegretti:2014tz}. From http://www.rcsb.org/pdb/explore/explore.do?structureId=4CI0 download the EM map (emd_2513.map).

Download the Tutorial data:

http://www2.mrc-lmb.cam.ac.uk/Personal/pemsley/coot/files/Brown-Tutorial.zip

For reference, download the coordinates for FRH (4CI0.pdb)

2. Load the model into *Coot*

First, start *Coot*:

- **`coot`**

Load model_**`fragment-of-A.pdb`** (File>Open Coordinates). This is a section of chain A from Frh (PDB ID: 4CI0). By default, Coot displays molecules with all atoms. This view can be changed to just the backbone trace:

- **`Display Manager > Bonds (Colour by Atom) > C-alphas/Backbone`**

3. Convert the map to structure factors

Although it is not necessary to convert the map into structure factors (in the form of an mtz file) for model building it can improve the appearance of the map and help guide tracing of the protein.

As a test case, we are going to use chain A from Frh. A map fragment

- `mrc2mtz emd_2513_mask_A.map emd_2513_mask_A.mtz`

Open emd_2513_mask_A.mtz in Coot (File>Open MTZ). Note 'Auto Open MTZ' does not work for MTZ files from EM maps. Before the map is loaded, a 'Column Label Assignment' window will pop up. This allows you to select where the appropriate information is stored in the MTZ file. Click 'OK'.

Density should now appear around the model. However, it may not be displayed at a reasonable sigma level. Scroll the mouse so that protein features (such as side chains) become visible. During scrolling the map level is displayed at the top of the screen. Aim for ~24 rmsd.

4. Jiggle Fit

The model is too far outside the density for a simple rigid-body refinement to be suitable. Therefore, we need to apply a Jiggle-Fit to this fragment.

- **`JiggleFit > Jiggle-fit this molecule 7000`**

This will apply 7,000 different random rotations and translations. The top ten solutions are output to the terminal screen (stdout), e.g.:

```
------------------------- initial_score 528.2 ------------------------------
INFO:: Jiggle-fit: optimizing trial  0: prelim-score was 548.1 post-fit 516.7
INFO:: Jiggle-fit: optimizing trial  1: prelim-score was 532.6 post-fit 903.3 ***
INFO:: Jiggle-fit: optimizing trial  2: prelim-score was 530.3 post-fit 475.6
INFO:: Jiggle-fit: optimizing trial  3: prelim-score was 507.9 post-fit   807
INFO:: Jiggle-fit: optimizing trial  4: prelim-score was 497.3 post-fit  1504 ***
INFO:: Jiggle-fit: optimizing trial  5: prelim-score was 460.1 post-fit 489.5
INFO:: Jiggle-fit: optimizing trial  6: prelim-score was 451.7 post-fit 511.6
INFO:: Jiggle-fit: optimizing trial  7: prelim-score was 451.4 post-fit 589.4
INFO:: Jiggle-fit: optimizing trial  8: prelim-score was 450.6 post-fit 489.5
INFO:: Jiggle-fit: optimizing trial  9: prelim-score was 450.5 post-fit 418.4
```

Each new top score is highlighted (***) and the coordinates are updated to the final highest score. The fragment should now agree much better with the density.

5. Building helices in the unassigned density

This fragment does not describe all the density. In particular there are a number of helices that need to be built.

Pan the screen so that the center of the screen is position in the middle of one of the helices (a good example is a close to (160 126 100)):

- **`Calculate > Other Modelling Tools > Place Helix Here`**

This will automatically place two poly-alanine alpha helices with different orientations in the density, called ‘Helix’ and ‘Reverse Helix’. In some cases where the orientation is unambiguous, only one helix may be inserted. Look to see which Helix best describes the density, and delete the other one.

The fit can be improved by:

- **`JiggleFit > Jiggle-fit this Fragment 1000`**

At this resolution, JiggleFit should identify the correct orientation 100% of the time.

The length of the helix can be often over-predicted due to surrounding density. Delete residues outside the density using:

- **`Delete Item > Delete Zone`**

And then click on the first and last residues of the section you want to delete.

The majority of the unassigned density can be very quickly populated with 5 helices.

6. Assigning sequence to the helices

The next step is to assign sequence to these helices. This can be done through a number of ways:

1) assigning sequence directly from the sidechain density without reference to the protein’s amino acid sequence
2) based on connectivity to the domain we have already placed in the density
3) by matching the length of the helices with secondary structure predictions
4) Identifying distinctive sidechains and matching these to the protein’s amino acid sequence

In reality, all of these methods can be used simultaneously.

7. Connecting helices

Once the helices have been placed and sequence assigned they can be connected to the rest of the protein using Coot’s model building tools.

To help build the backbone, it can be useful to skeletonize the map.

- **Calculate > Map Skeleton > Prune sidechains > OK**

The skeletonisation level needs to be changed:

- **Edit > Skeleton Parameters > Skeletonization level > 0.1** electrons/Å^3

This skeleton can help C-alpha tracing:

- **Calculate > Other Modelling Tools > C-alpha Baton Mode**

Now convert to poly-alanine

- **Calculate > Other Modelling Tools > Ca Zone -> Mainchain**

Then click on the start and end positions of the C-alpha baton trace.

Hint: interesting loop at (162, 123, 75)