Coot/ChimeraX Cryo-EM tutorial

1. Description of initial map and model

For this tutorial, we will start with EMD-34705 (PMID: 38057552), a 2.8 Å reconstruction of the human sodium-dependent glucose transporter SGLT2 in complex with a single TM auxiliary protein, MAP17, bound to sodium and a small molecule inhibitor, dapagliflozin.

Starting from an AlphaFold model of SGLT2, we will fit, trim and refine SGLT2, place and build MAP17, and fit and refine both the small molecule ligand and the sodium ion.

2. Required software

- Coot 0.9.8.96. This is the version distributed with CCP4, which we also need for ligand generation:

```
https://www.ccp4.ac.uk/download
```

Note: You can also complete this tutorial using Coot 1.1.20, but some of the steps may need modification due to changes in the interface.

- ChimeraX 1.11.1 or nightly build:

```
https://www.cgl.ucsf.edu/chimerax/download.html
```

- Phenix (not essential, but needed to generate restraints for metal coordination):

```
https://www.phenix-online.org/download
```

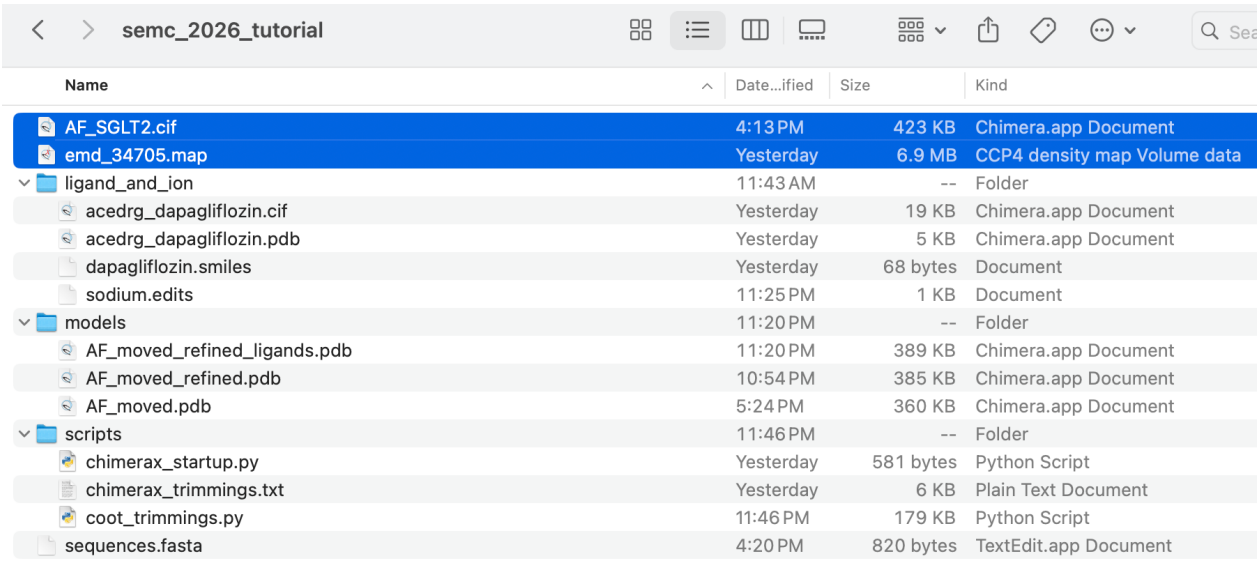
3. Getting Started

3.1 Download the required files from the following link:

semc_2026_tutorial.zip

An archive of the files you'll need for this tutorial should download automatically.

Unpack it and move it to your preferred working directory. The contents are as follows (the initial model and map are highlighted):

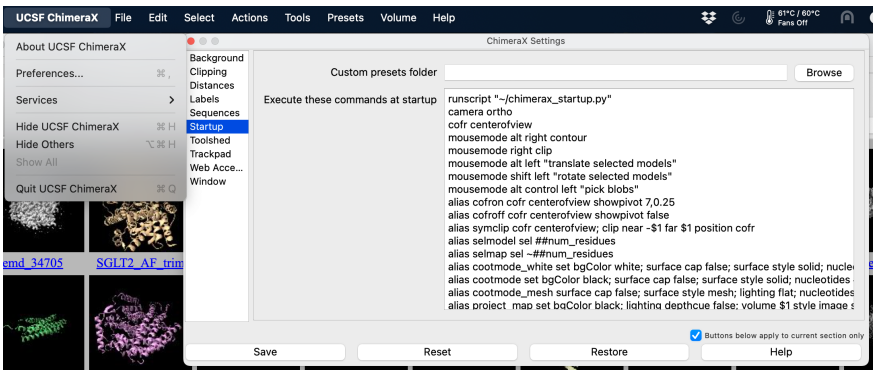


We will now install some custom scripts to enhance the functionality of ChimeraX and Coot for EM fitting.

3.2 Installing custom scripts.

chimerax-trimmings (ChimeraX)

To install chimerax-trimmings, copy the contents of `chimerax_trimmings.txt` to the "Startup" section of Preferences:



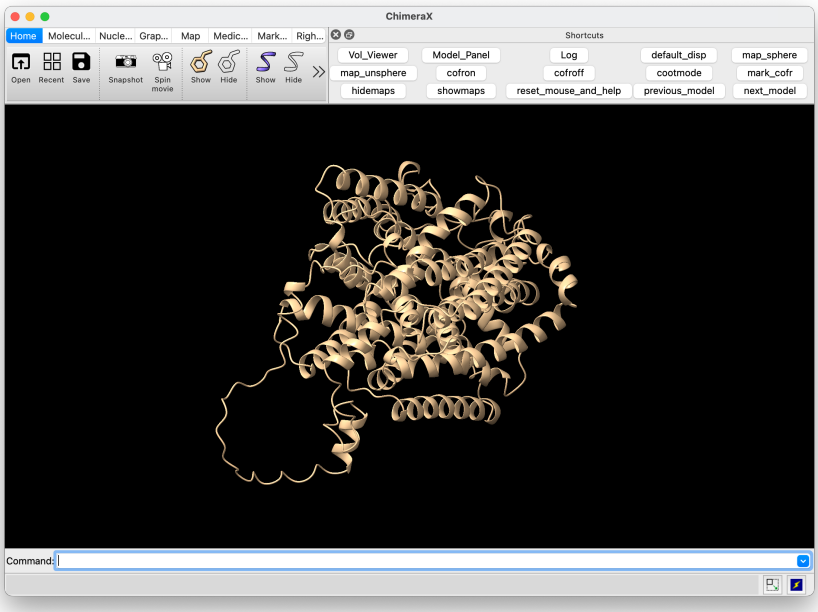
And copy `chimerax_startup.py` to your home directory, e.g.:

```
cp ./scripts/chimerax_startup.py ~/
```

Now, when you restart ChimeraX, you should have an additional panel of shortcuts, as well as additional alias commands. See here for details:

https://github.com/olibclarke/chimerax-trimmings

It should look like this:



coot-trimmings (Coot 0.9)

To install coot-trimmings, copy the coot_trimmings.py script from the "scripts" subdirectory to the hidden coot-preferences directory in your home directory, e.g.:

```
cp ./scripts/coot_trimmings.py ~/.coot-preferences/
```

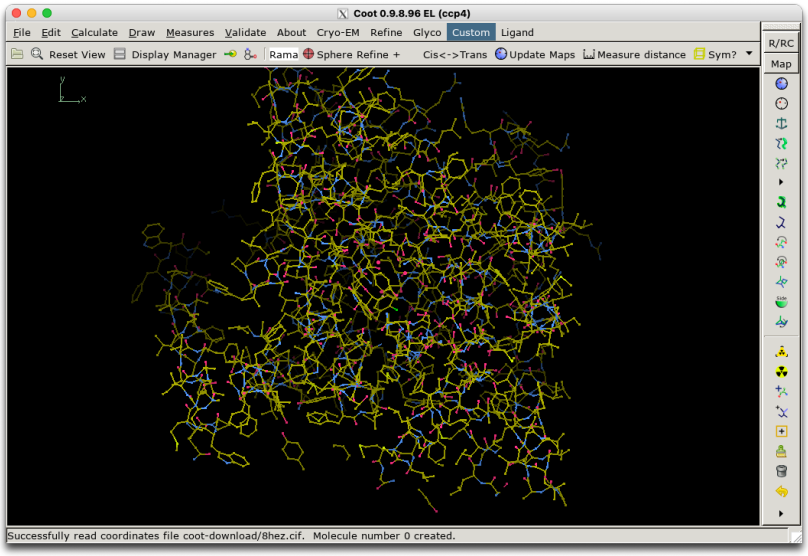
If you do not already have a coot preferences directory, you will need to create one, e.g.:

```
mkdir ~/.coot-preferences/
```

When you restart Coot, you should see a new "Custom" menu, and have a set of extra keybindings and custom functions. See here for details:

https://github.com/olibclarke/coot-trimmings

It should look like this:



Note 1: Do not start Coot from within the coot-preferences directory; it will crash.

Note 2: I have disabled scroll-wheel adjustment of map contour, as I prefer keyboard adjustment (+/-) when using a trackpad; if you want to change it back, edit `set_scroll_by_wheel_mouse(0)` *to* `set_scroll_by_wheel_mouse(1)` *in the* `coot_trimmings.py` *script.*

3.3 Sequences

SGLT2 (1-672):
MEEHTEAGSAPEMGAQKALIDNPADILVIAAYFLLVIGVGLWSMCRTNRGTVGGYFLAGRSMVWWP
VGASLFASNIGSGHFVGLAGTGAASGLAVAGFEWNALFVVLLLGWLFAPVYLTAGVITMPQYLRKRF
GGRRIRLYLSVLSLFLYIFTKISVDMFSGAVFIQQALGWNIYASVIALLGITMIYTVTGGLAALMYTDTVQ
TFVILGGACILMGYAFHEVGGYSGLFDKYLGAATSLTVSEDPAVGNISSFCYRPRPDSYHLLRHPVTGD
LPWPALLLGLTIVSGWYWCSDQVIVQRCLAGKSLTHIKAGCILCGYLKLTPMFLMVMPGMISRILYPD
EVACVVPEVARRVCGTEVGCSNIAYPRLVVKLMPNGLRGLMLAVMLAALMSSLASIFNSSSTLFTMDI
YTRLRPRAGDRELLLVGRLWVVFIVVVSVAWLPVVQAAQGGQLFDYIQAVSSYLAPPVSAVFVLALFV
PRVNEQGAFWGLIGGLLMGLARLIPEFSFGSGSCVQPSACPAFLCGVHYLYFAIVLFFCSGLLTLTVSL
CTAPIPRKHLHRLVFSLRHSKEEREDLDADEQQGSSLPVQNGCPESAMEMNEPQAPAPSLFRQCLL
WFCGMSRGGVGSPPPLTQEEAAAAARRLEDISEDPSWARVVNLNALLMMAVAVFLWGFYA

Note: I have mutated one of the cysteines (Cys-351) to an alanine here, due to a bug pertaining to disulfide refinement in Coot 0.9.8.96; this bug is not present in Coot 1.1.20.

MAP17:

MSALSLLILGLLTAVPPASCQQGLGNLQ**PWMQGLIAVAVFLVLVAIAFAVN**HFWCQEEPEPAHMILT
VGNKADGVLVGTDGRYSSMAASFRSSEHENAYENVPEEEGKVRSTPM

(Predicted TM helix in **bold**)

4. Rigid body fitting and trimming of SGLT2 (ChimeraX)

I have generated a model of SGLT2 using AF3; we will now move it, fit it, and trim the portions that are not well resolved in the density map.

Basic navigation/model manipulation in ChimeraX (with chimera-trimmings):

Rotate view: Click-drag

Translate view: Option-click-drag
(*Note: will translate selected model if there is an active selection!*)

Select atom/surface: Ctrl-click

Add atom/surface to selection: Shift-Ctrl-click

Expand/contract atom selection: Up/down arrows

Rotate selected model: Shift-option-click-drag

Translate selected model (x-y): Option-click drag

Translate selected model (z): Option-click-Ctrl-drag
(*Note: Order is important! Must press Ctrl after clicking, and z-translation only happens with vertical mouse motion*)

Pick/color blob: Ctrl-Alt-click

Adjust map threshold: Option-right-click-drag

Move both clip planes: Shift-right-click-drag

Move front clip plane: Right-click-drag

Function keys (customizable!):

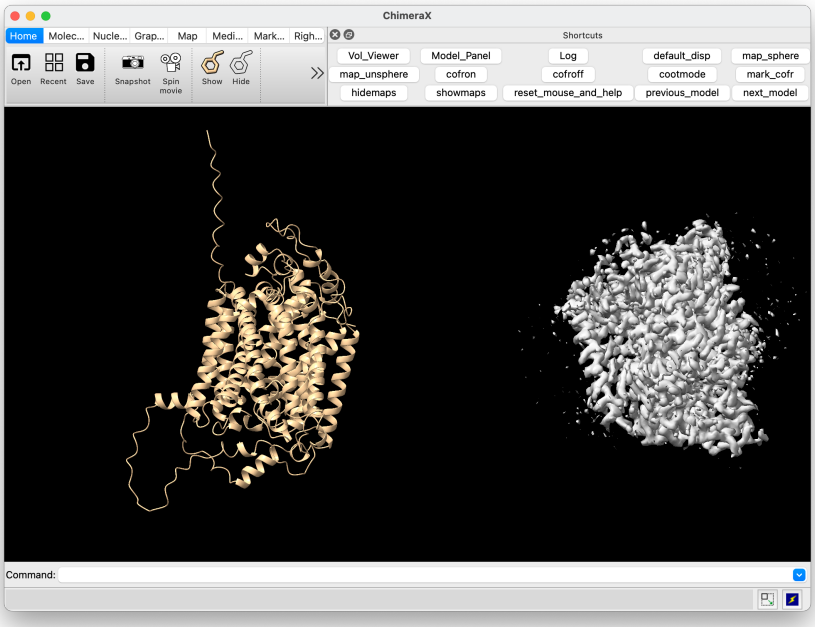
F1 - previous model

F2 - next model

F3 - center selection

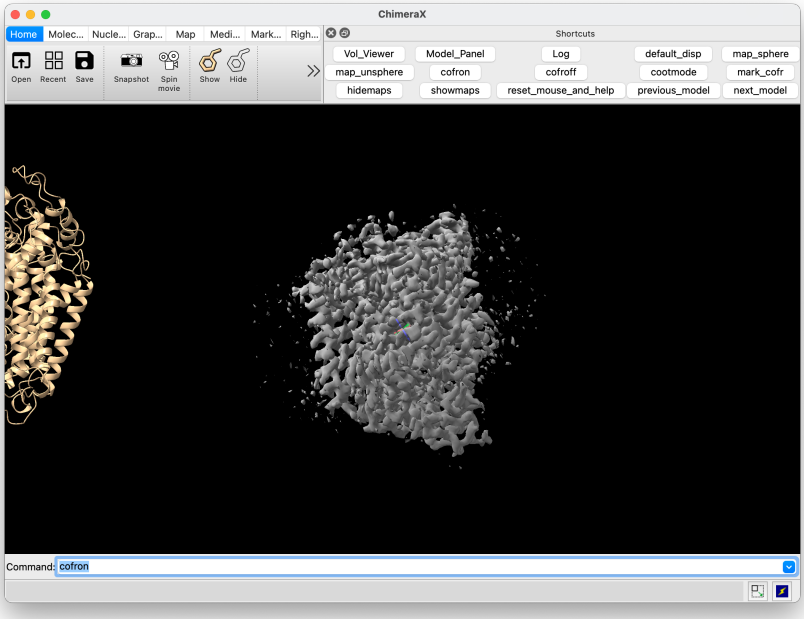
F4 - view all

4.1 Open the starting model and map in ChimeraX. You should see something like this:

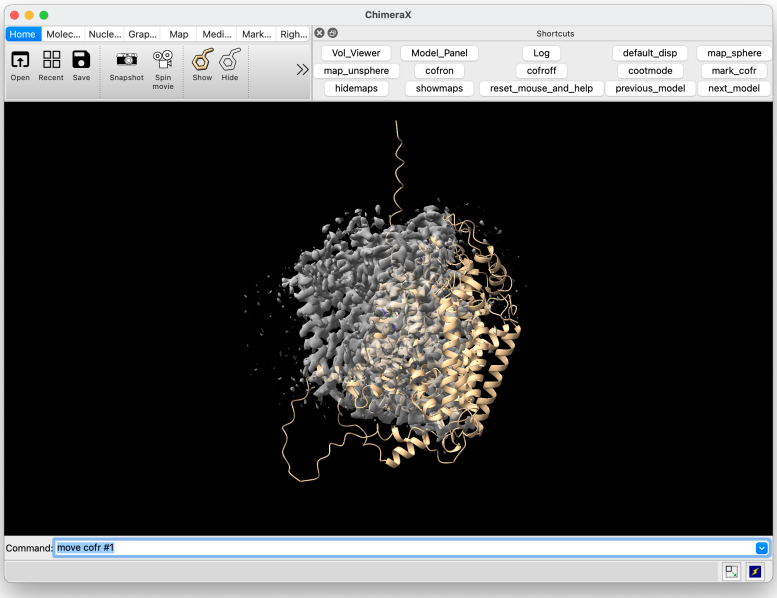


4.2 Let's make the map transparent. Enter `transparency #* 50` at the command line, and hit return.

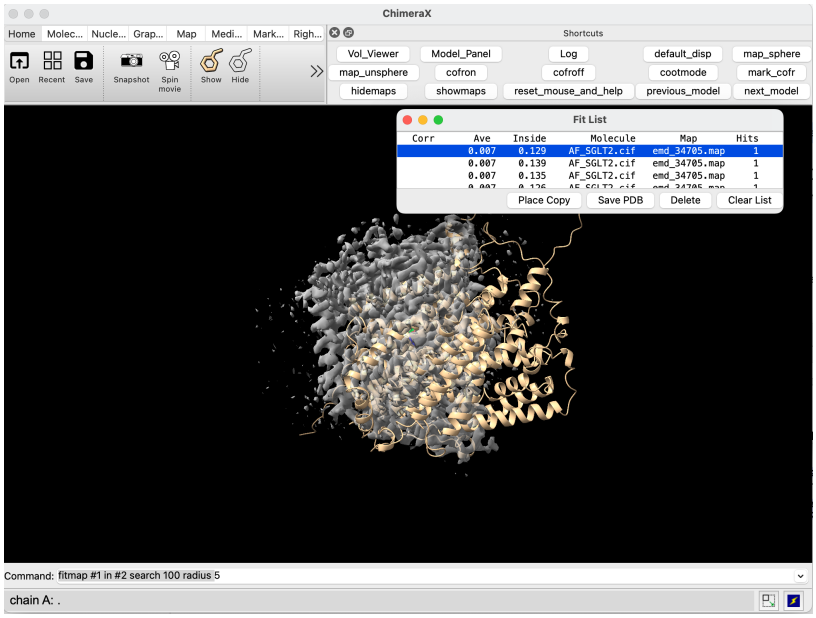
4.3 Enter the alias `cofron` on the command line (or press the corresponding shortcut button). You should now see a marker for the center of rotation; move it to the center of the density map:



4.4 Move the model to the current center of rotation with: `move cofr #1` (assuming the model ID of your starting model is #1; check in the Model Panel):

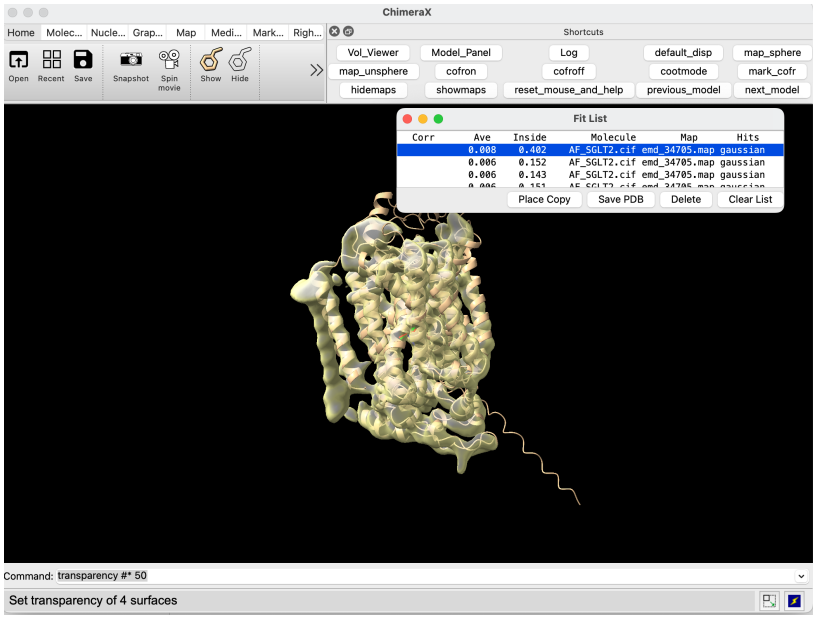


4.5 Fit the model to the map using global search: `fitmap #1 in #2 search 100 radius 5`. Hmm, that doesn't look too good...



4.6 Let's try filtering the map: `volume gaussian #2 sdev 2`, and then repeat the command from 4.5 (`fitmap #1 in #3 search 100 radius 5`) - much better!

Gaussian filtering of the map increases the radius of convergence of global search with fitmap, making it easier to find the correct solution.

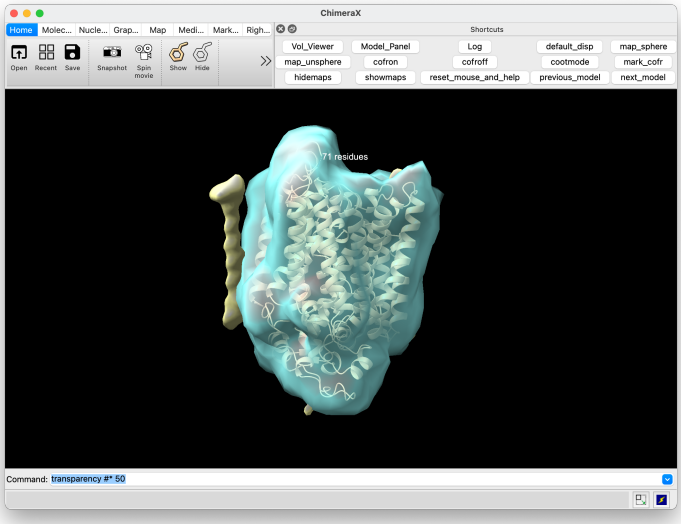


Note: Steps 4.1-4.6 are demonstrated in the following video:
semc_tutorial_video_1.mov

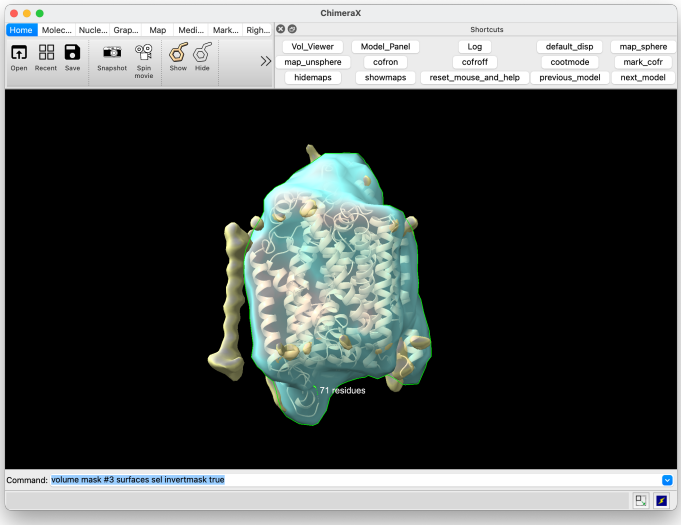
4.7 Masking the map by the model: In this instance, we are not going to fit the second component, as we will build it later in Coot. However, if we wanted to, it would be challenging in the raw map, as there are many different possible solutions which would clash with the existing

model. To deal with this, we can mask the map by the model as follows, so we can fit a second component to the residual density:

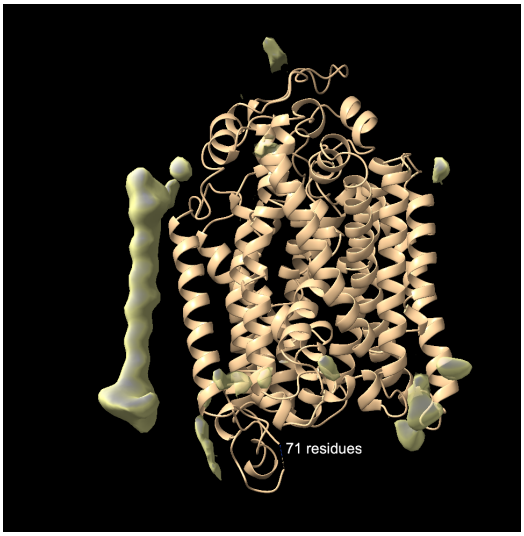
1. Make a map from the model - `molmap #1 12` and make it transparent, `transparency #* 50`:



2. Adjust the threshold so that the model-map is enclosing most of the density we want to remove, while not clipping too much of the density we want to keep. Select the surface (Ctrl-click).



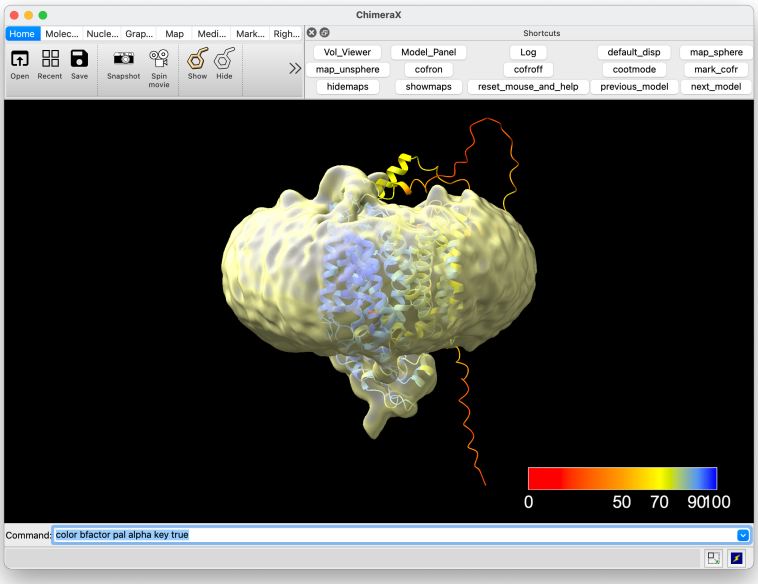
3. Mask the map we want to fit to by this surface: `volume mask #3 surfaces sel invertmask true`



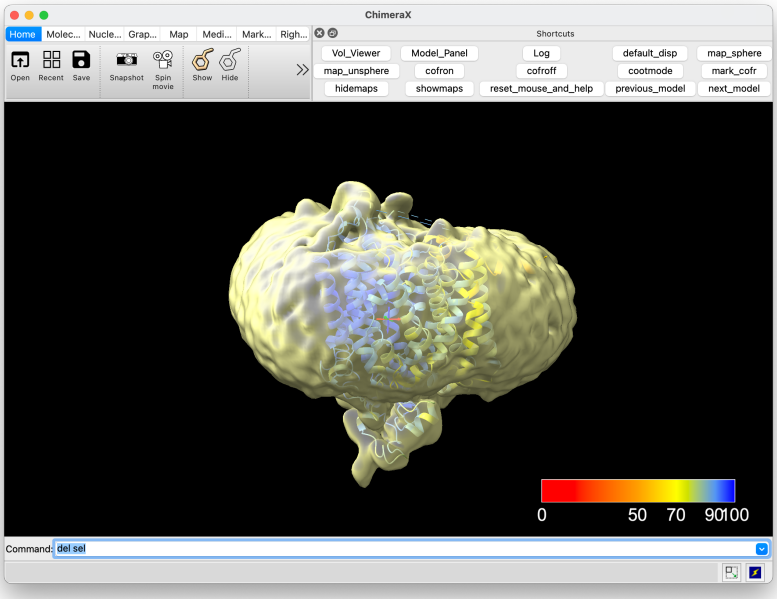
4. This masked map can now be used for global fitting, without the SGLT2 density interfering.

Note: step 4.7 is demonstrated in the following video:
semc_tutorial_video_2.mov

4.8 Let's identify parts of the model that are not well resolved in the density - mostly low pLDTT regions, which we can color with `color bfactor pal alpha key true`:



4.9 Now, let's delete them - for each segment, select the starting and ending residues (Ctrl-click and Shift-Ctrl-Click), then enter `selbetween` to select the residues in between, followed by `del sel` to delete the selected residues:

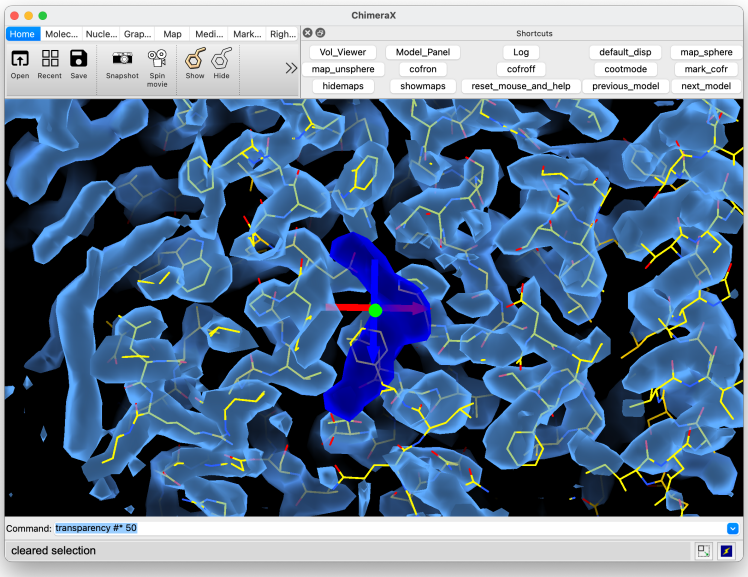


Note: Steps 4.8-4.9 are demonstrated in the following video:
semc_tutorial_video_3.mov

4.10 Rigid body fit the model to the original map (`fitmap #1 in #2`), close the filtered map, and inspect the results (use the "cootmode" shortcut and the `ca_and_sidechains` alias).

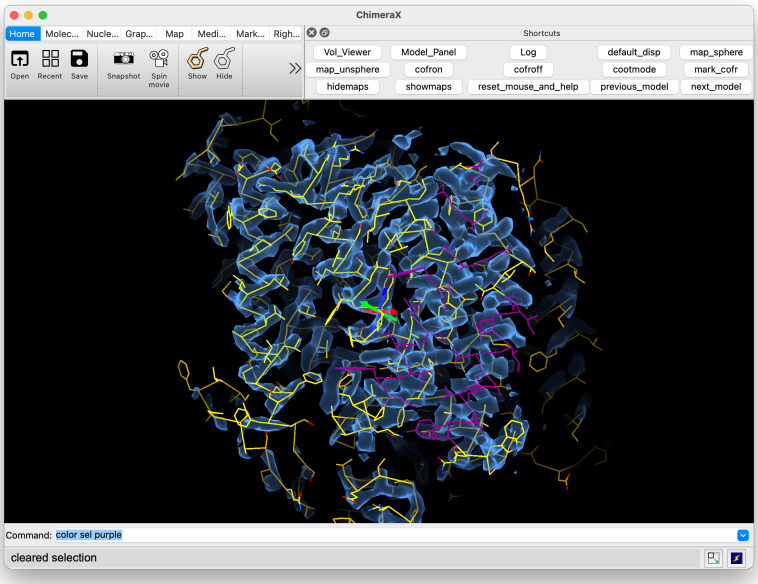
It may help to clip the scene around the center of rotation (`symclip 5`). Try to locate the ligand density (Hint: it is near F453). Ctrl-Alt-click on it to highlight the blob:

Note: For a guide to mouse modes, click the "reset_mouse_and_help" shortcut.



Notice that a number of sidechains are incorrectly placed, and a couple of helices are out of

register with the density (model here represented using `ca_and_sidechains #1`):



Time to fix that, which will require flexible fitting. Let's save these coordinates (my version is saved as `AF_moved.pdb` in the "models" subdirectory) and load them up in Coot.

Note: Step 4.10 is demonstrated in the following video:
semc_tutorial_video_4.mov

5. Flexible fitting and model completion (Coot 0.9)

5.1 Getting started, navigation and keyboard shortcuts

Open Coot and load the map and model we have just saved. The display status and appearance can be adjusted in Display Manager, or try the following `coot-trimmings` shortcuts:

"]" & "[": cycle representation mode of the active model forward/back;

"/": Toggle display of all models on/off;

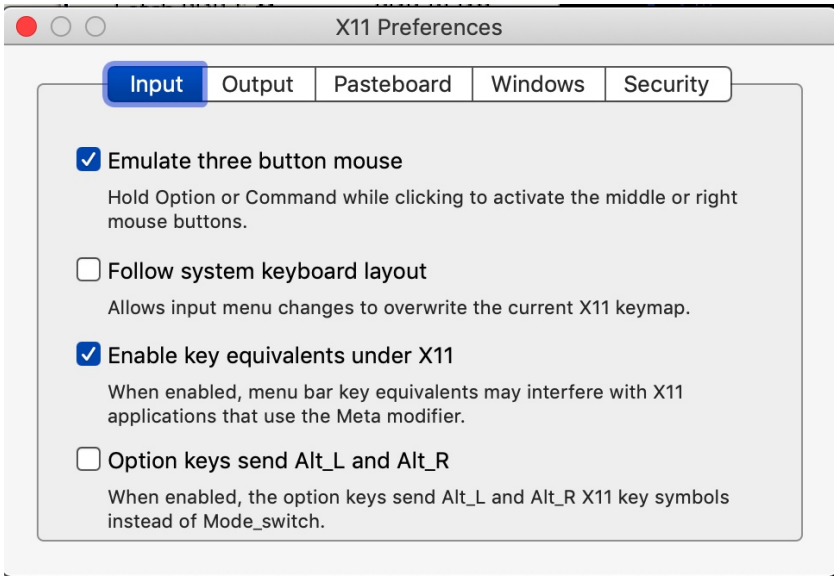
"?": Cycle display of each model (if multiple are loaded);

"`": Toggle all maps on/off

"~": Cycle display of each map (if multiple are loaded)

"Q": to quick-save active PDB (it will automatically make a backup of the existing file)

If you are using a Mac with a trackpad, make sure to set "Emulate three button mouse" in XQuartz input preferences:



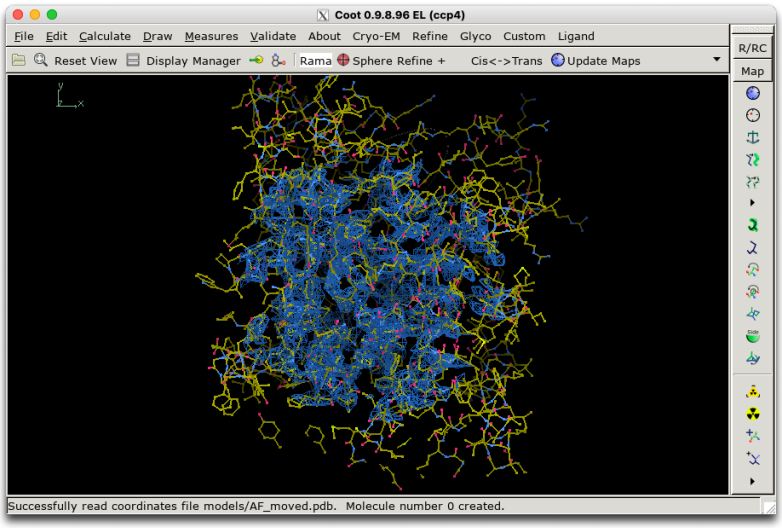
Navigation:

- **Ctrl-click-drag** to translate view
- **Click-drag** to rotate view
- **Right-click-drag** to zoom in/out
- **Middle-click (or option-click)** on atom to center
- **Ctrl-G** to quick-go to atom (box appears; type chain ID and residue number and hit enter)
- **Shift-click** to label atom/residue

View:

- **Ctrl-right-click-drag** *up/down* to translate slab in/out of plane
- **Ctrl-right-click-drag** *left/right* to change thickness of slab.
- **Edit...Map Color** to change map color
- **Edit... Map Properties** to change map radius

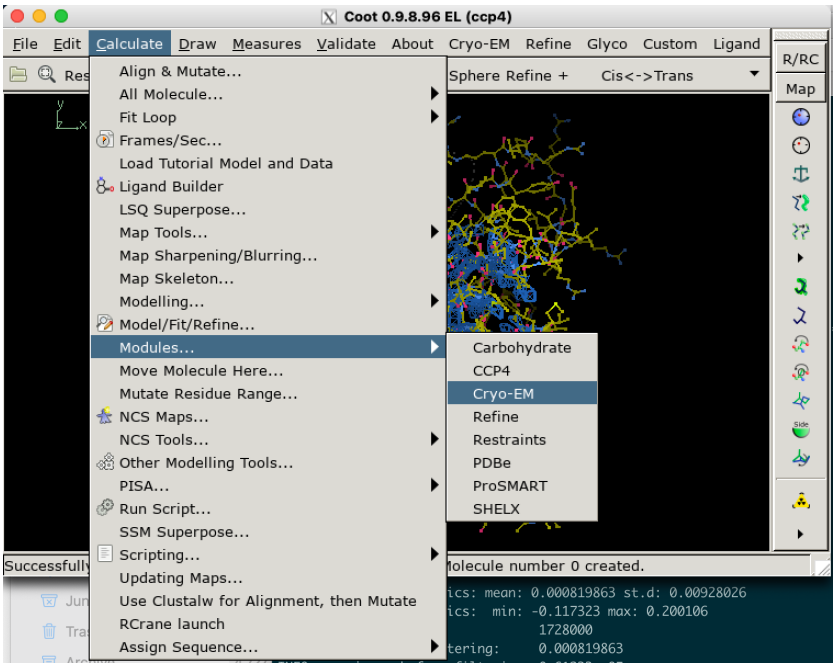
5.2 Loading the model and map: To load the model and map from the command line, you can type: `coot models/AF_moved.pdb --map emd_34705.map`. You should see something like this:



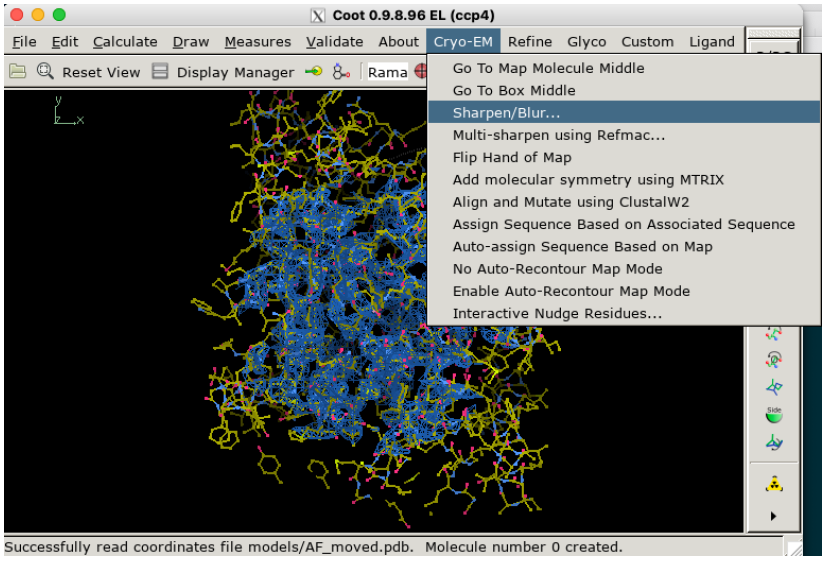
5.3 Map sharpening and resampling

The original map was sampled at ~1.1Å per pixel. This is fairly coarse for the resolution of 2.8 Å. Often best visual results are seen sampling at <1/5 of the resolution - which in this case, corresponds to 2x resampling.

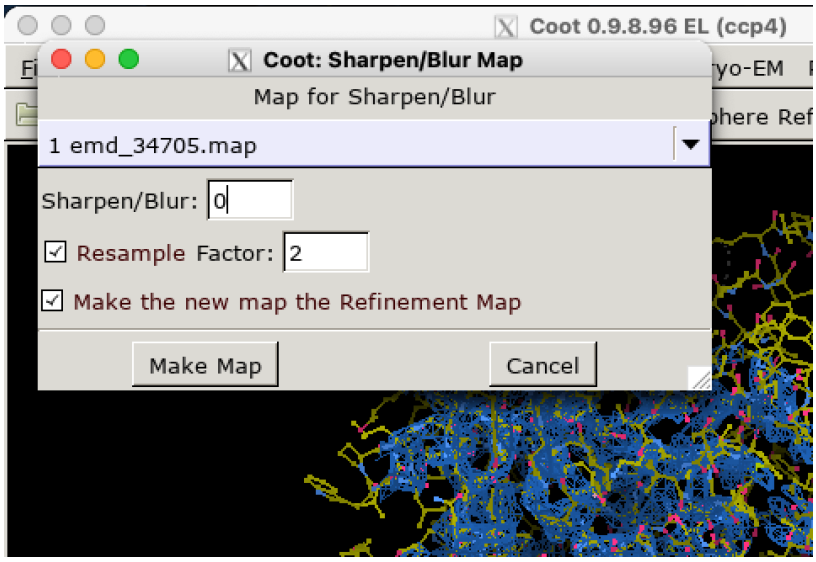
If the Cryo-EM module is not already loaded, load it:



Then click "Sharpen/Blur" in the Cryo-EM menu:

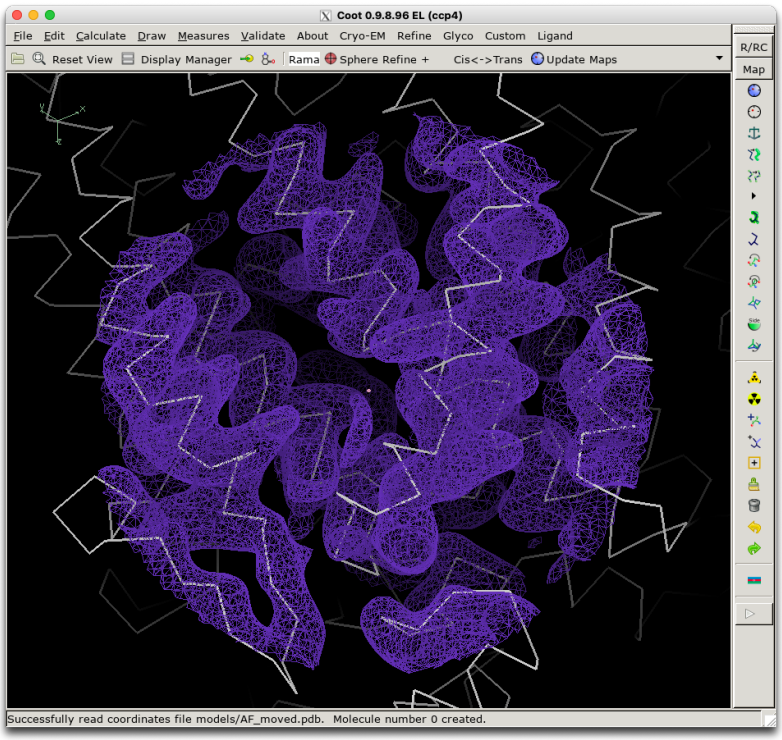


Now, resample 2X, without adjusting the B-factor:

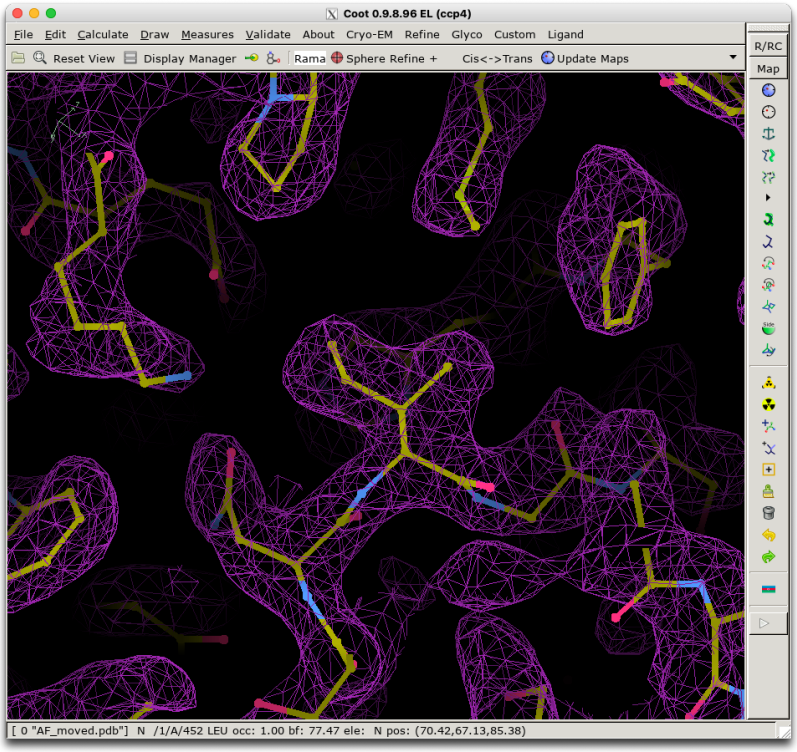


You can now close the original map from the Display Manager.

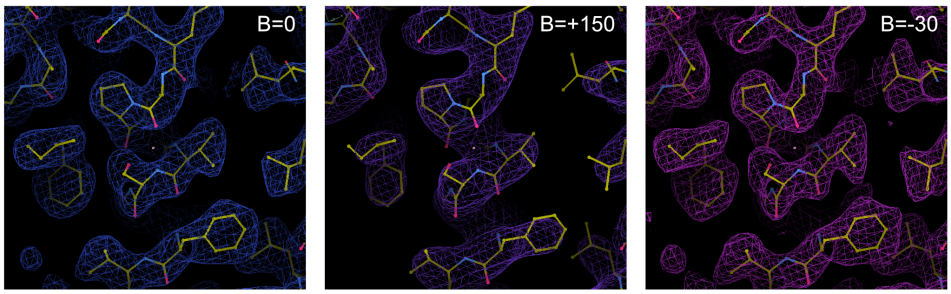
It is often useful to have a map with less sharpening - it will have less detail, but improved connectivity, and as we saw in ChimeraX, the radius of convergence for refinement of large regions will be larger. So let's make one - run Sharpen/Blur again, but this time apply no resampling, but a Blur factor of 150:



And now let's make one with a sharpening factor of -30:



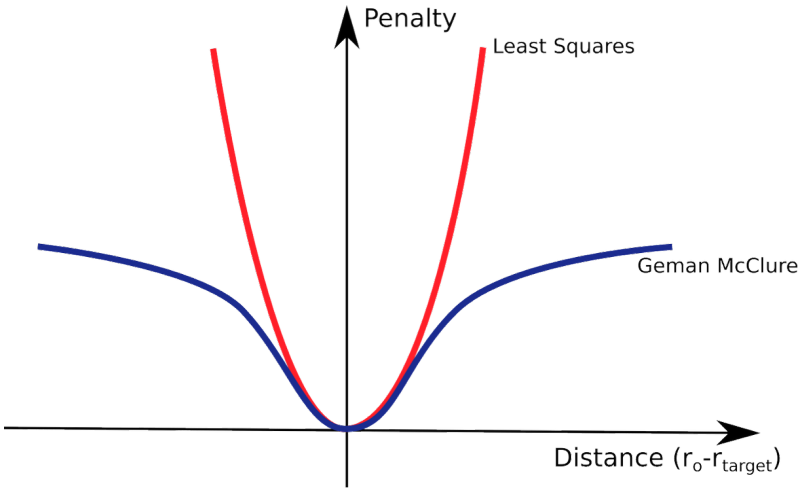
Here we see more detail, but also more noise. Compare the three maps (Blur150, 0 and Sharp30) in conjunction with the atomic model. Are some more informative than others? Does it vary by region?



Note: Step 5.3 is demonstrated in the following video:
semc_tutorial_video_5.mov

5.4 Reference restraints

AlphaFold models almost always have excellent local geometry, even if the overall conformation may not quite be a match for the sample at hand. We can take advantage of this feature by applying Geman-McClure (GM) reference restraints - restraining *local* distances in the structure to tend to remain close to their starting values, while allowing them to diverge if the information provided by the map is sufficiently persuasive:



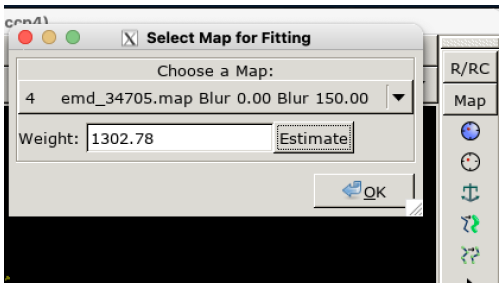
You can tune GM restraints using the "alpha" parameter. alpha=0 is equivalent to least squares; the higher the alpha, the more robust to true outliers.

To add reference restraints:

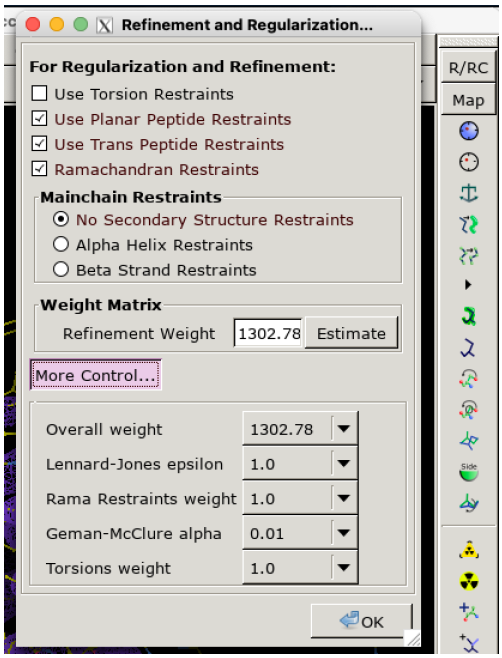
1. Launch the restraints module (*Calculate...Modules...Restraints*)
2. Click *Restraints...Generate Self Restraints 3.7 for Chain*. Restraints should now be visible as gray lines.
3. Undisplay them (*Restraints...Undisplay Extra Restraints*)

5.5 Setting the refinement map and Weights

1. Set the map to refine against by clicking "Map" in the right-hand toolbar. A dialog will appear:



2. Select the Blur150 map from the drop-down, and automatically estimate the refinement weight. This overall weighting factor sets the balance between the influence of geometry and the map. At 0, it is effectively geometric optimization, with no influence of the map; at higher values, the map will have progressively stronger effects.
3. If you want more control, click the "R/RC" button in the righthand toolbar, then click "More Control":



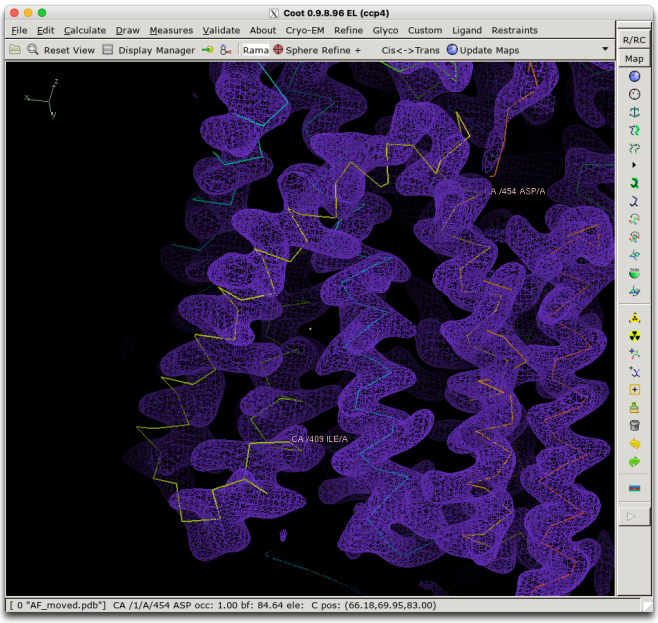
4. This is where you can experiment with the GM-alpha parameter mentioned in the previous section.

Note: Step 5.5 is demonstrated in the following video:

semc_tutorial_video_6.mov

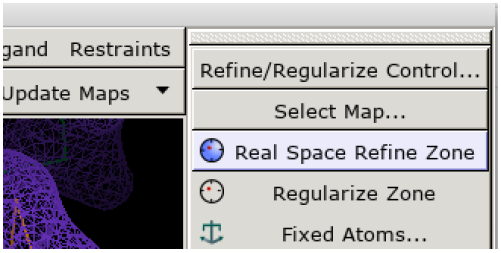
5.6 Real space refine zone

Consider this region, between ~409 and 454:

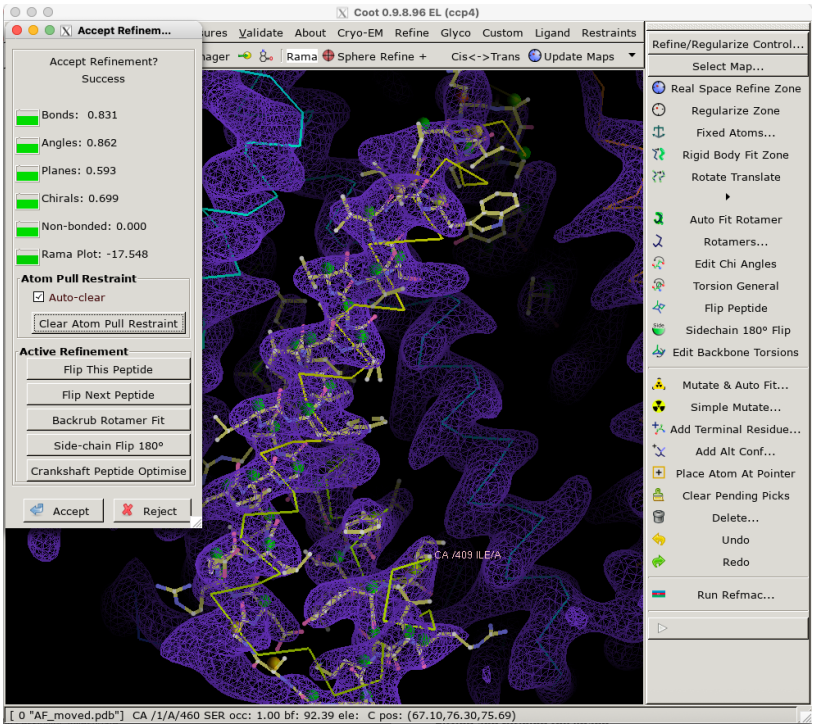


There is an obvious mismatch between the density and the model over a long region. Let's try to fix it up:

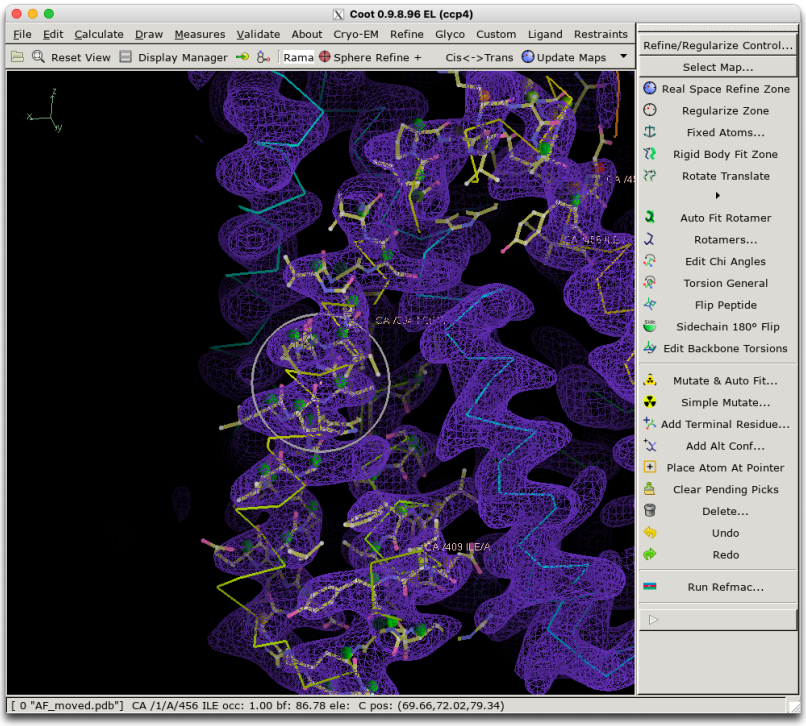
1. Click "Real space refine zone" (the blue ball in the righthand tool bar):



2. Click two atoms, with a generous boundary either side of the misfit region. E.g. residues 394 and 460. If you need to reorient after clicking the first atom in order to find the second, hold down Ctrl while click-dragging. Things will start moving. Wait until they stop:

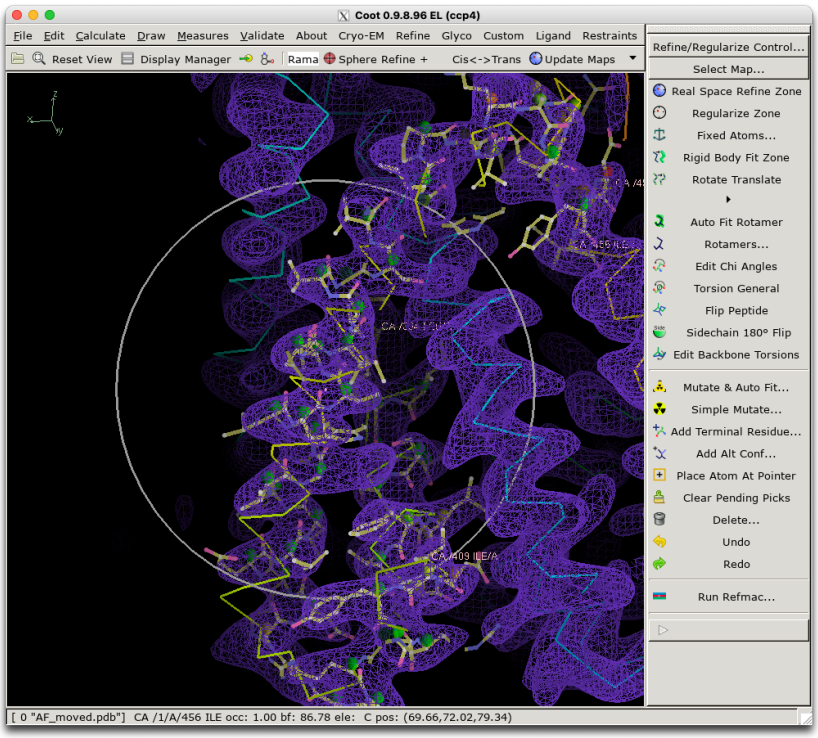


3. Does this look right? No - although the main chain is now in the density, the helix is out of register - look at the tryptophans. How can we fix it?
4. Click and drag on one of the tryptophans - notice that when we drag, only the atoms immediately attached to the point of contact are directly affected, and if we drag too far, the local geometry of the helix is disrupted - not what we want! We would like to drag a larger region - enter the concept of proportional editing:

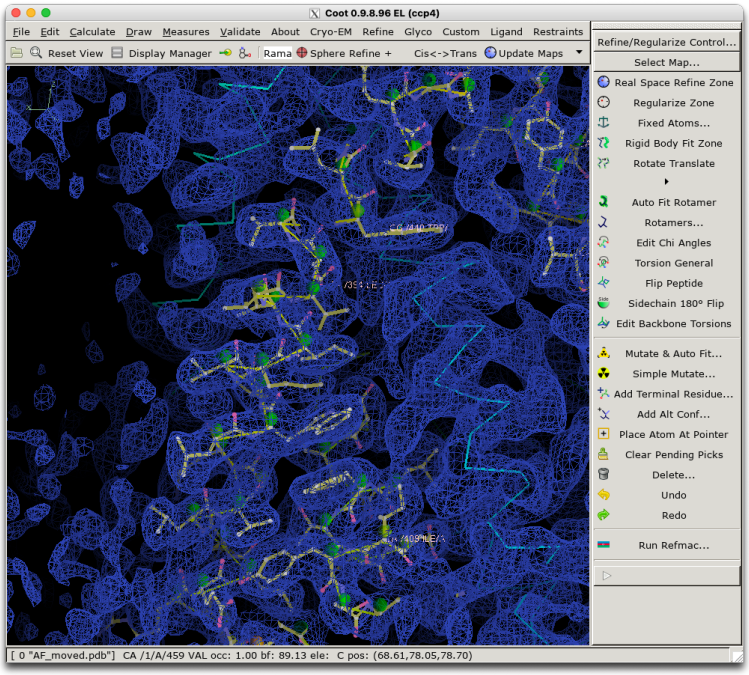


5. Hold down Ctrl during refinement, and a white circle should appear, as above. Ctrl-scroll, and the circle should change size, controlling the effective "sphere of influence" of dragged

refinement. Adjust it to about this size, and then try dragged refinement again:



6. Using a combination of proportional editing with different radii and careful dragged refinement, you should be able to fix this region. *Try experimenting with the GM alpha value too - how does increasing it affect the initial refinement? Does one of the three maps work better than the others for the initial refinement? How about for final tweaking?*

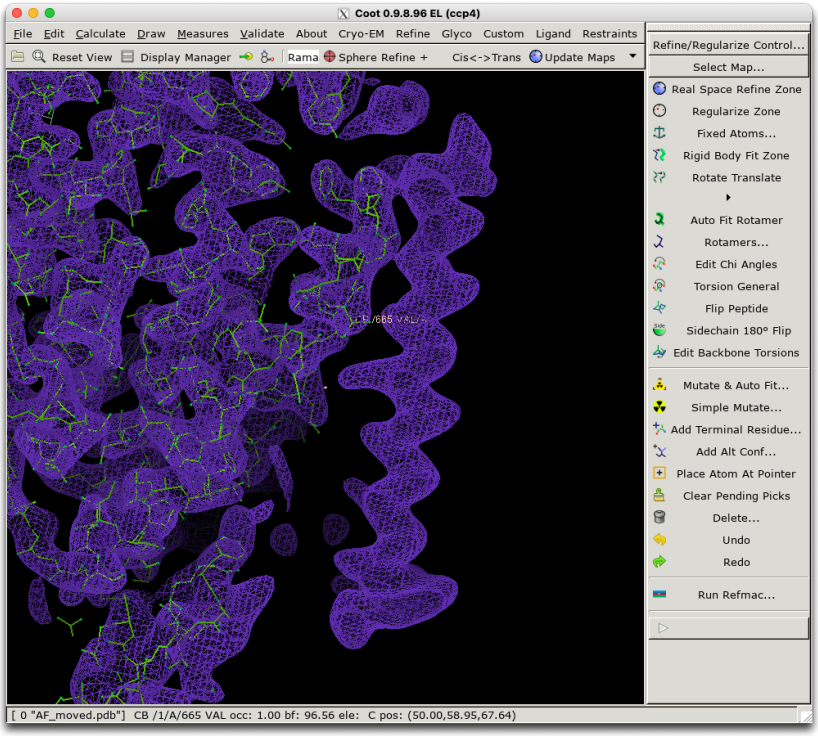


7. There is an adjacent helix (Use Ctrl-G to go to residue 186 to find it) that is similarly misfit. Try fixing that using what you have learned so far.
8. Now, try refining the whole molecule at once (Blur150 map, GM-alpha=1), using Refine...All-Atom Refine.

Note: Step 5.6 (with the exception of all atom refinement) is demonstrated in the following video: semc_tutorial_video_7.mov

5.7 Placing a helix

There is an unassigned, helical density near residue 665:

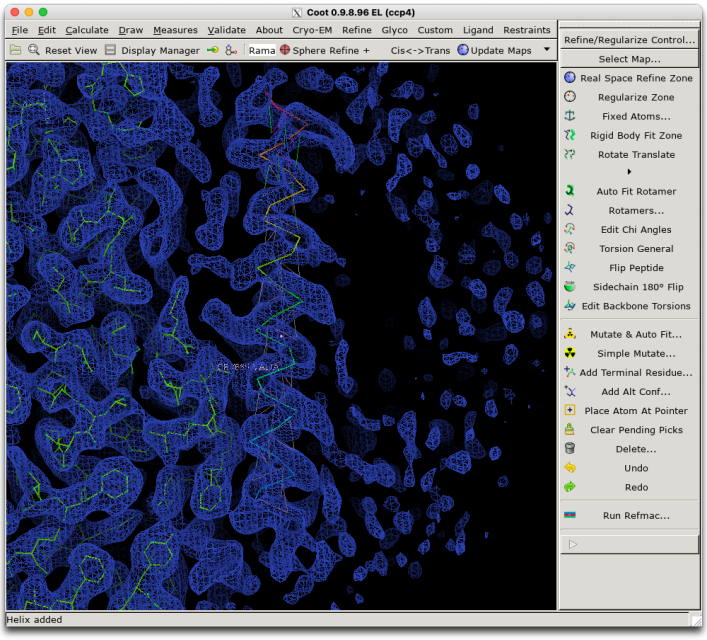


We know this corresponds to MAP17 - but for the sake of argument, let's say we weren't sure, that MAP17 was one candidate of several. How would we build it in and confirm our assignment? Remember, this is the sequence of MAP17:

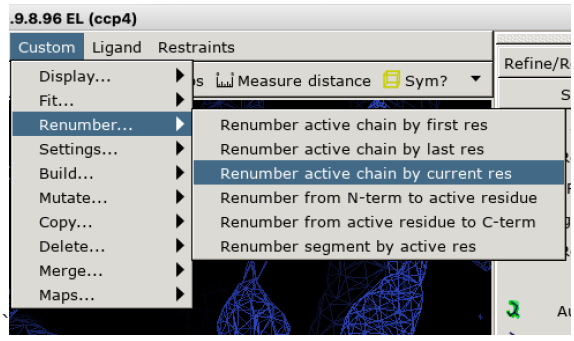
MSALSLLILGLLTAVPPASCQQGLGNLQ**PWMQGLIAVAVFLVLVAIAFAVN**HFWCQEEPEPAHMILTVGNKADC

Where the **bold** region is a putative TM helix. The density we see is in the TM region, so a reasonable starting point is to start building it as the TM helix of MAP17:

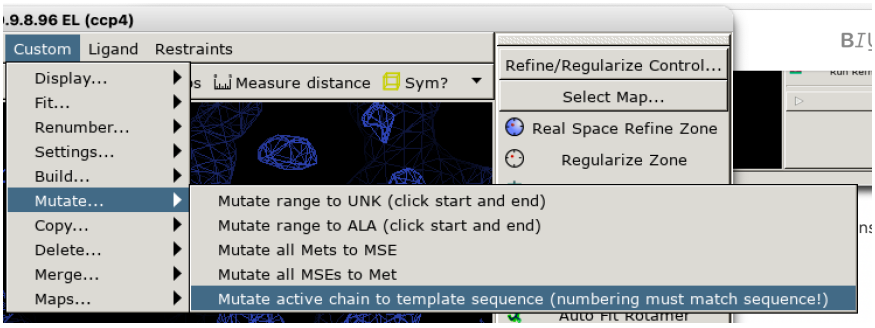
1. First, inspect the helical density. Based on what you know about helices (Xmas trees with branches pointing towards the N-terminus), which way do you think it points?
2. Navigate to the midpoint of the helical density. Press "h" (you can also launch this from "Calculate... Other Modeling Tools"). A helix should appear:

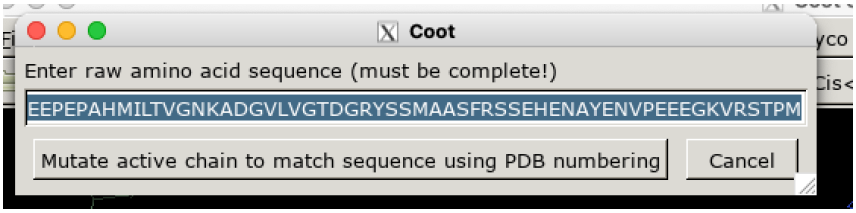


3. Inspect the density again, this time focusing on the sidechain densities, and consider the pattern of bulky aromatic residues in the putative TM sequence:
   P**W**MQGLIAVAV**F**LVLVAIA**F**AVN
   Can you identify a likely candidate for the first Trp (W30)? If you extend this hypothesis further along the chain, does it hold?
4. Center on this residue and mutate it (**Shift-M**) to W. Adjust the rotamer (**Shift-R**) to match the density.
5. Renumber this residue to 30:

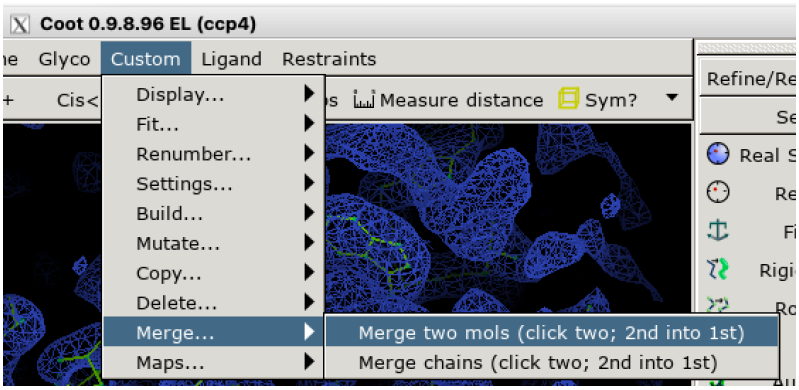


6. Mutate the helix to the sequence of MAP17, pasting in the *complete* MAP17 sequence:





7. Check each residue stub, and fill in the matching sidechain, adjusting rotamers and real-space refining as needed. Do you think this fit is correct? Why?
8. Try deleting a portion of the helix, and then rebuilding using "Add terminal residue" (on toolbar, or "y" shortcut).
9. Merge your final helix into the main model:

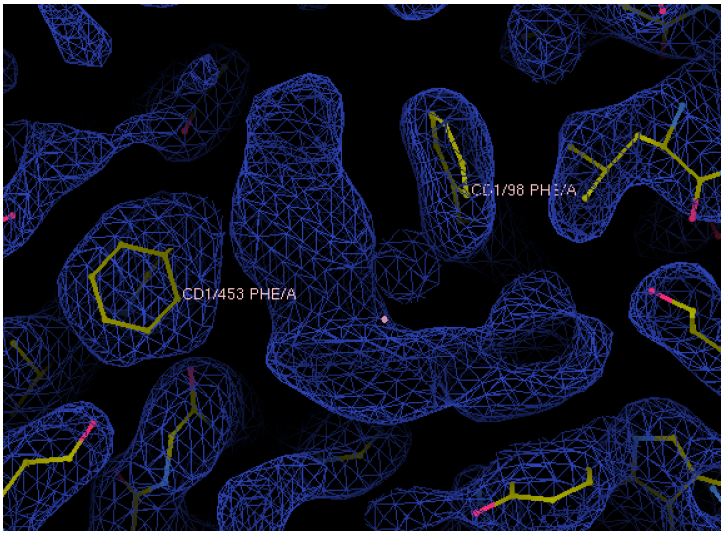


10. My model at this stage is saved as `AF_moved_refined.pdb` in the models subdirectory.

Note: Step 5.7 is demonstrated in the following video:
semc_tutorial_video_8.mov

5.8 Fitting and refining the ligand

We have some unknown density sandwiched between two phenylalanines, which may correspond to the inhibitor, dapagliflozin:



How do we build the ligand, and refine it into the density?

1. First, we need to generate coordinates and matching restraints for the ligand. We can do this using AceDRG, included in CCP4, from the command line:

   ```
   acedrg -i dapafligozin.smiles  -o acedrg_dapafligozin
   ```

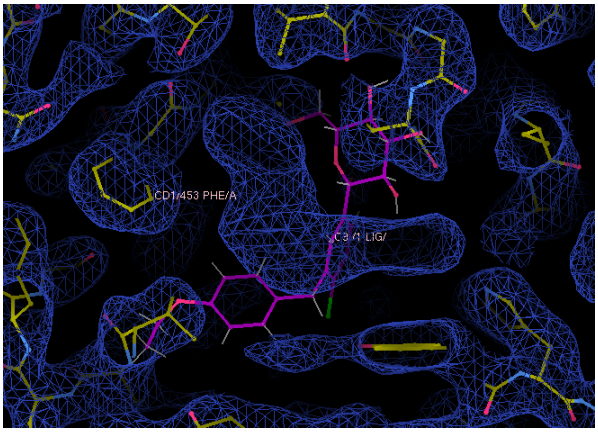
   Where the "smiles" file specifies the connectivity and stereochemistry of the ligand, in SMILES code:

   ```
   CCOc1ccc(Cc2cc([C@@H]3O[C@H](CO)[C@@H](O)[C@H](O)[C@H]3O)ccc2Cl)cc1
   ```

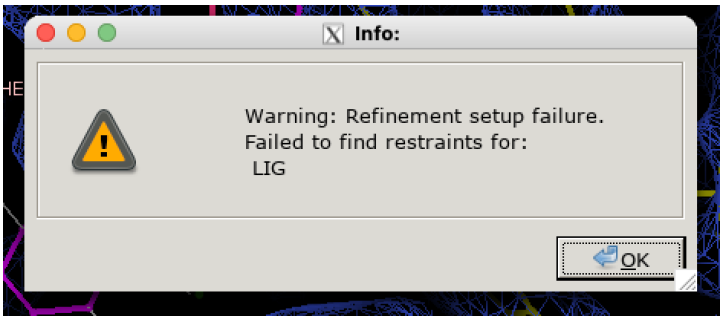
   We can also perform the same operation in phenix, using `phenix.elbow`:

   ```
   phenix.elbow smiles="CCOc1ccc(Cc2cc([C@@H]3O[C@H](CO)[C@@H](O)[C@H](O)
   [C@H]3O)ccc2Cl)cc1" --output="dapafligozin"
   ```

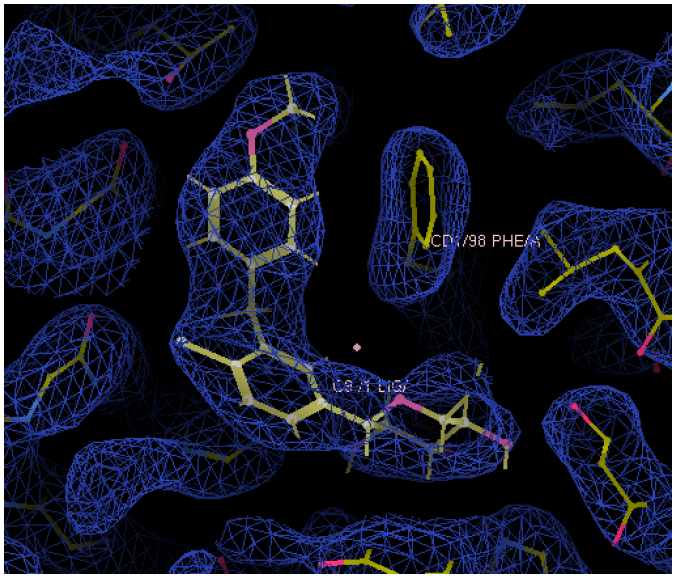
   Both commands should generate a PDB file, containing the coordinates, and a CIF dictionary, containing the definition of target bond distances, angles and connectivity. Pregenerated outputs from AceDRG are included in the "ligand_and_ion" folder.
2. Load the PDB, opening as a coordinate file and selecting "Recenter Molecule Here", so that it is centered on the ligand density.

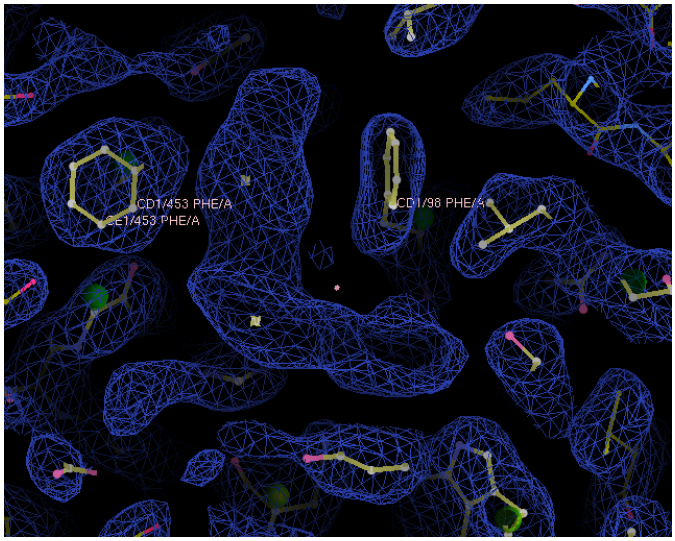
3. It is centered in the density, but not fit well at all. Try refining it:

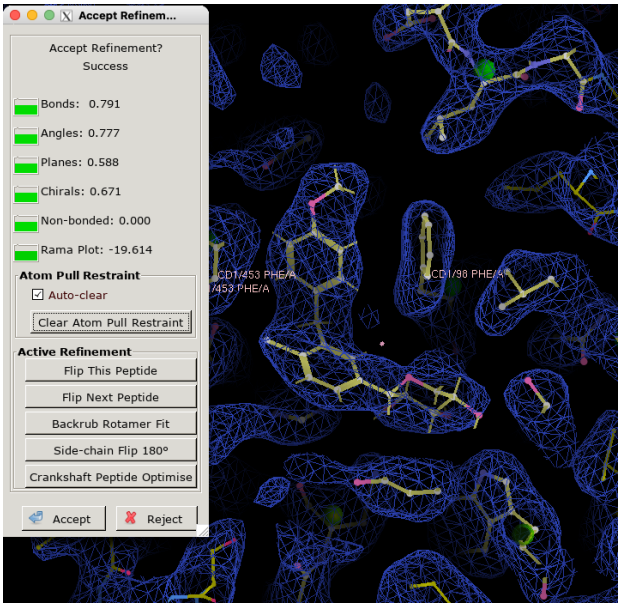
4. Whoops! We need restraints. Let's load a CIF dictionary (File...Import CIF dictionary). Make sure it is assigned to the correct molecule!
5. Now try refining again - using dragged refinement, match the shape and orientation of the ligand to the density. That's better!



6. Now merge it into the main molecule (Custom...Merge... Merge two mols). Try refining again... Uh oh:

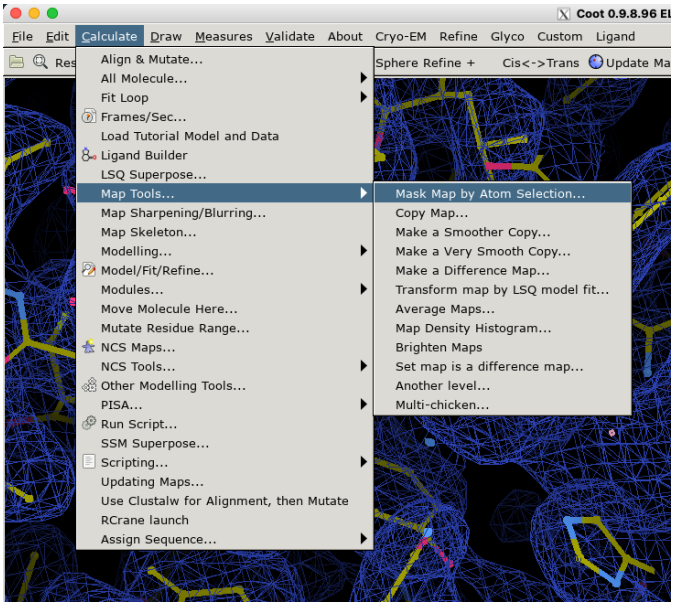


7. Our ligand exploded, because we didn't associate the CIF dictionary with the new model of which it it now a part. Cancel, re-import the CIF dictionary (for the merged model molecule), and re-refine:

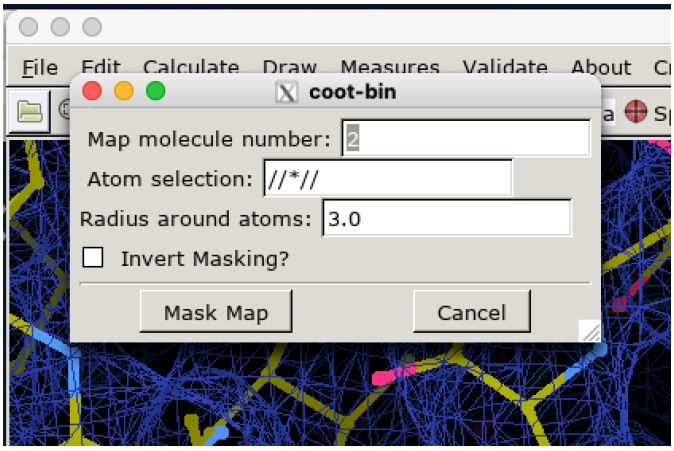


Much better!

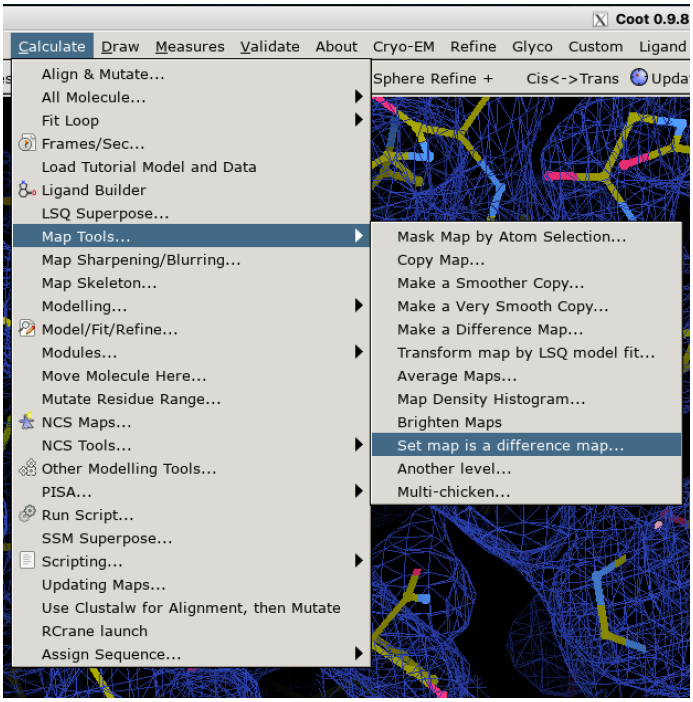
8. It may be useful in some circumstances to generate a pseudo-difference map to fit the ligand (to prevent the ligand floating into protein density. To try this, delete the ligand, then try *Calculate...Map Tools...Mask Map by Atom Selection:*



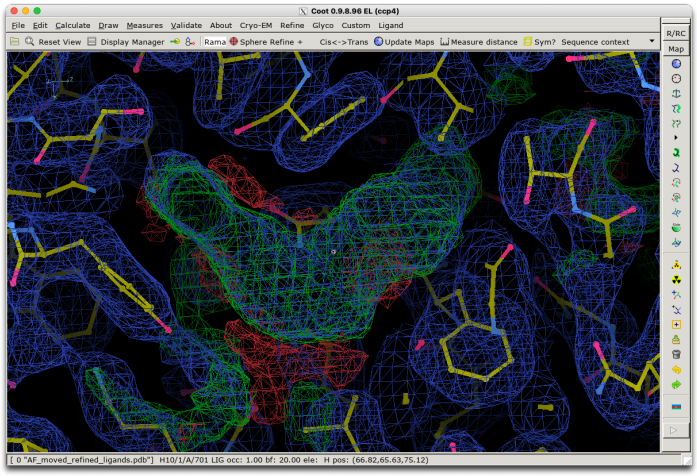
9. Select the model, and a dialog should appear (if you can't find it, it is probably behind the main window) - enter these parameters and hit Mask Map:



10. Run *Calculate...Map Tools... Set map is a difference map*, and select the map we just created:



11. You should now have clear positive "difference map" density for the ligand. Try refitting the ligand to it using the steps described earlier (*Note: Make sure to set the masked map as your refinement map first!*)

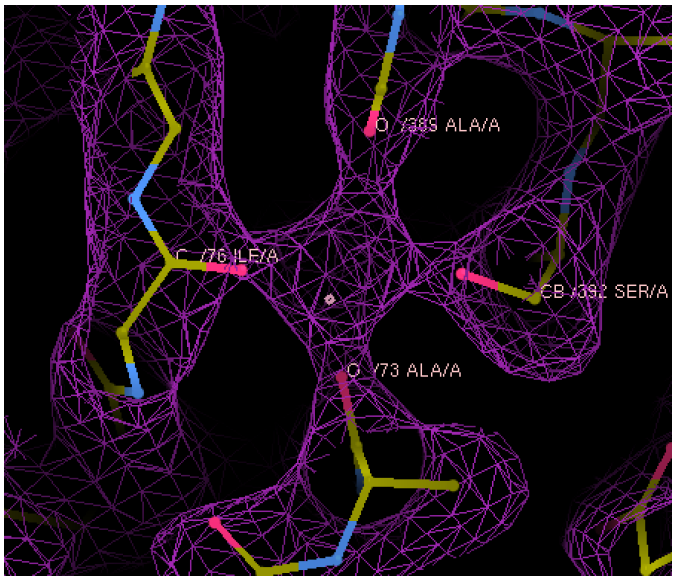


Note: Step 5.8 (with the exception of the generation of the masked map) is demonstrated in the following video:

semc_tutorial_video_9.mov

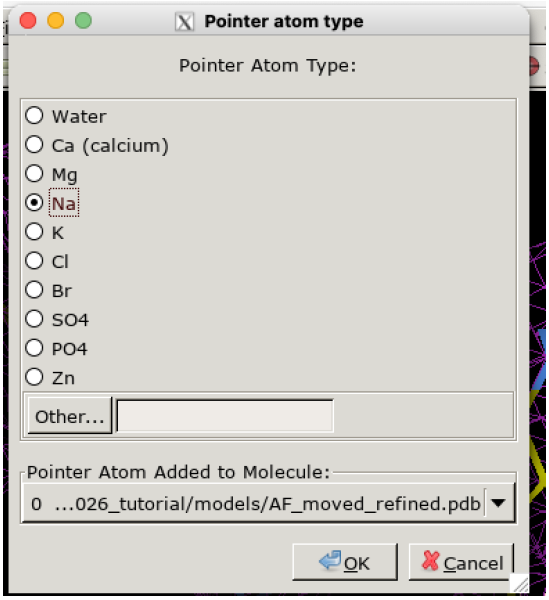
5.9 Placing a sodium ion and generating restraints

Consider this putative sodium ion binding site near S392:

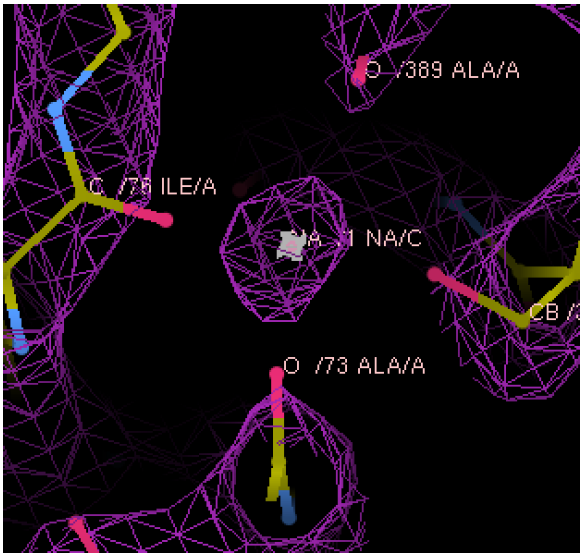


How can we place and refine a sodium ion here?

1. Navigate to the center of the density peak, and use "Place Atom At Pointer" (righthand toolbar) to place a sodium ion in the density peak:



2. We now have our sodium ion, but if we refine it, it will become distorted - Coot is not aware of the metal coordination. This model is saved as `AF_moved_refined_ligands.pdb` in the models subdirectory.



3. To fix that, we will use phenix.metal_coordination:

```
phenix.metal_coordination AF_moved_refined_ligands.pdb
```

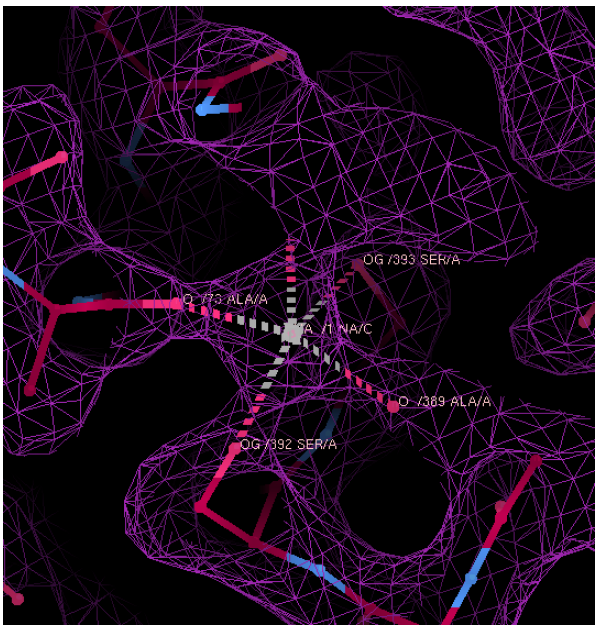
   This will generate a set of distance restraints for phenix refinement, in the `elbow.edits` file. If we want to also restrain angles, we can add the `output_angles=true` flag.

4. Check the elbow.edits file carefully! If any restraints are present that don't make chemical sense (e.g. restraints between Na and C), delete them. A copy of the restraints is included in the "ligand_and_ion" subdirectory, as `sodium.edits`.

5. Refine in phenix, e.g.:

```
phenix.real_space_refine models/AF_moved_refined_ligands.pdb emd_34705.map
ligand_and_ion/sodium.edits ligand_and_ion/acedrg_dapagliflozin.cif
resolution=2.8
```

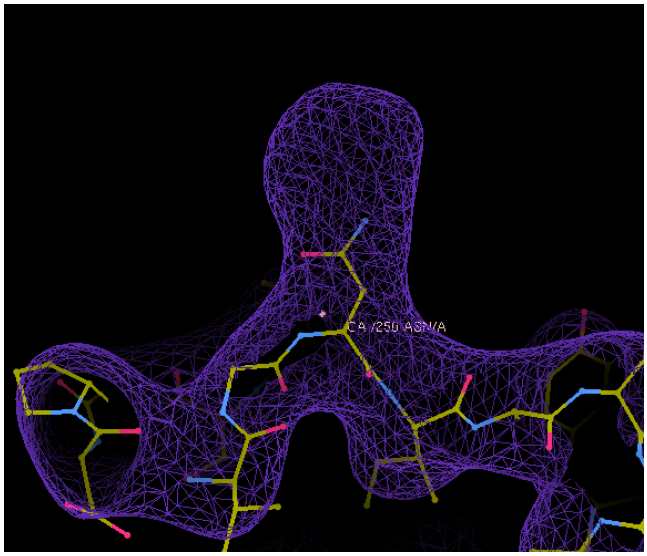
6. Re-load into Coot and try re-refining the sodium ion. What do you notice?



Note: Step 5.9 (up until phenix refinement) is demonstrated in the following video: semc_tutorial_video_10.mov

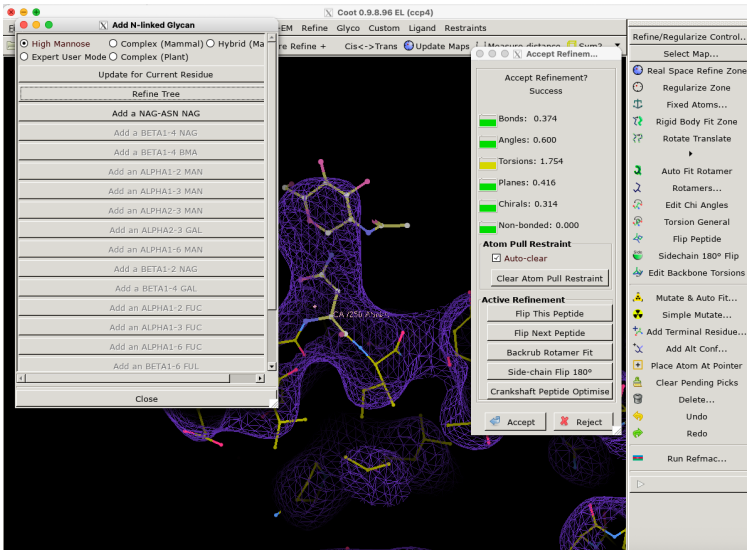
5.10 Building an N-linked glycan

Consider N250 (in the B150 map):

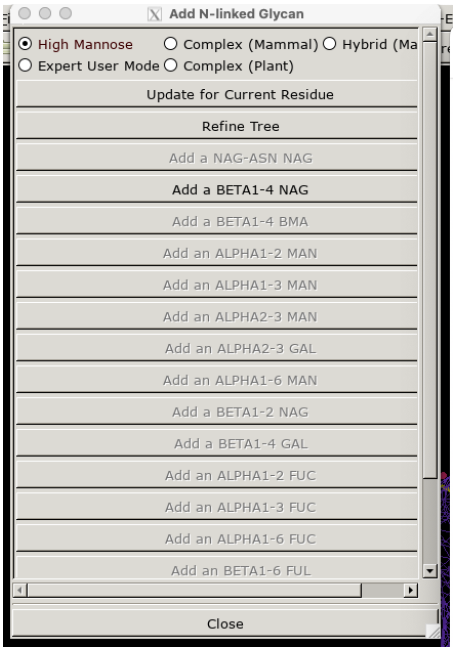


Let's say we had prior knowledge that this residue was a likely glycosylation site. How would we add and refine the glycan?

1. If it is not already loaded, load the Glyco module (*Calculate...Modules...Carbohydrate*).
2. Add a glycan and Refine Tree (*Glyco...N-linked glycan addition*):



3. Recenter on the just-added NAG residue, and launch the glycan builder again. You will notice that there are new options to extend the chain - we won't use them here, due to lack of additional density:



Note: Step 5.10 is demonstrated in the following video:
semc_tutorial_video_11.mov

5.11 Validation

1. Inspect the Ramachandran plot, Rotamer Analysis and Density fit analysis graphs. Can you find any outliers? Can you fix them?
2. Have a look at Unmodelled Blobds (try thresholding at 5 sigma). Are there any? What do you think they are?
3. Can you find any waters? Try adding them ("w" shortcut) and refine into density using Sphere refine.
4. Inspect the chain residue by residue. Did you pick up any errors that were not apparent from the validation graphs? Fix themn using real space refinement.