MRC Laboratory of Molecular Biology

EMDA tutorial

Rangana Warshamanage

9.10.2021

ranganaw@mrc-lmb.cam.ac.uk

EMDA

- I/O and format conversion
- Likelihood-based methods (overlay, magnification, bestmap etc…)
- Validation metrics (RCC, FCC, FSC, OCC)
- General purpose tools

Obtaining EMDA

- Open source (MPL-2.0 license)
- Included in Python Package Index (PyPi)

```
pip install emda
```

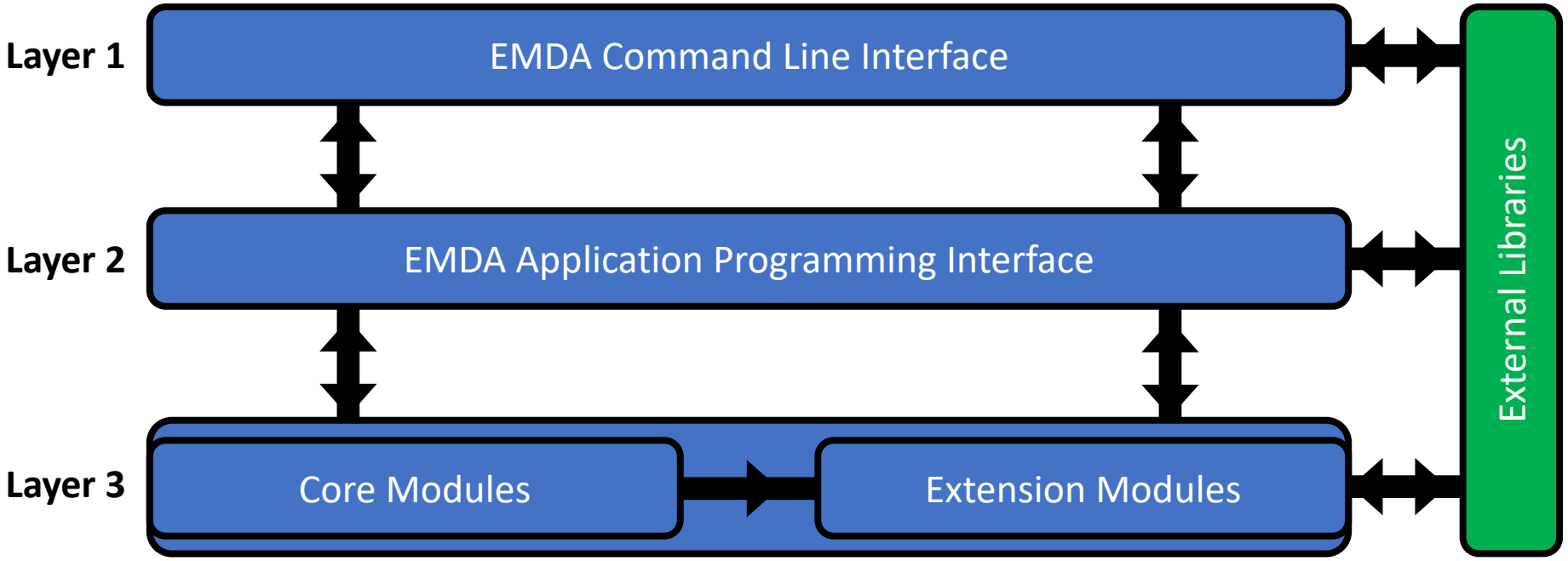
- Part of CCP-EM suit

Dependencies

- **General python packages** – Numpy, Scipy, Pandas, Matplotlib
- **Strucutral Biology specific python packages** – mrcfile, gemmi, servalcat, proshade
- CCP4(REFMAC) for model refinement

EMDA

Electron Microscopy Data Analytical toolkit



Command-Line Interface

Example:

```
>>> emda mapmask --map map1.mrc --knl 5 --prb 0.99 --itr 3 --res 15
```

Application Programming Interface

```
import emda.emda_methods as em

uc, arr, origin = em.get_data('map.mrc')
mapmask = em.mapmask(uc=uc,
                     arr=arr,
                     kern=5,
                     resol=15,
                     filter="butterworth",
                     order=1,
                     prob=0.99,
                     itr=3,
                     orig=None,
                     )
```

EMDA version and examples

- Examples described in this tutorial are the ones those presented in EMDA paper.
- EMDA version used in this tutorial is – emda-1.1.4

This tutorial has 3 parts

- Part1 – Local correlation in real space
- Part2 – Map superposition
- Part3 – Magnification refinement

Part 1: Local correlation calculation

Example 1: Detecting map-model differences

The Goal is to use local correlation in real space to detect discrepancies between map and the model.
In this example we use EMD-5623 map and 3j9i model.

Step 1: Download maps and model

Note that you won't be able to find half data for this entry in EMDB, but you will find them @ https://www.emdataresource.org/EMD-5623

Make sure you have EMD-5623-half-1.map.gz, EMD-5623-half-2.map.gz, emd_5623.map.gz and 3j9i.cif in your working directory. Now unzip all maps.

```
3j9i.cif
EMD-5623-half-1.map
EMD-5623-half-2.map
emd_5623.map
```

Step 2: refine the model

Since we're calculating local correlation between the map and the model, we need to refine the model w.r.t the map.
Note that we use the fullmap (average of half maps) as the map in the refinement because we want to get the correct distribution of atomic B values. If we use the primary map as the refinement target, we cannot be sure that we get the correct distribution because the sharpening affects the B values. On the other hand, since the fullmap is blurred, the whole distribution is shifted, but the shape of the distribution is not affected.

We can use servalcat to do the refinement.

```
$ servalcat refine_spa --model ../3j9i.cif --resolution 3.3 --halfmaps
../EMD-5623-half-1.map ../EMD-5623-half-2.map --ncycle 10
```

Make sure you have refined.pdb in your working directory. You may look at the refined.pdb in COOT.

Also, you may open primary map, halfmaps and refined model in COOT or Chimera, just to make sure that all maps and model are properly aligned.
Note that EMDA does not optimized map-model fit during local correlation calculation. It assumes that all maps and models are properly aligned beforehand.

You may use other ways to refine the model w.r.t the fullmap.

Step 3: Calculating local correlation

Now, we're ready to calculate local correlation in EMDA.
To do that, copy (or make a symbolic link) the refined.pdb into the directory where you have halfmaps.

```
$ emda rcc --h1 EMD-5623-half-1.map --h2 EMD-5623-half-2.map --mdl refined.pdb
--res 3.3 --nrm --knl 4
```

This command does several things.

1. First, it computes real space local correlation map using normalized and weighted half maps (normalization and weighting are done in Fourier space).
2. Also, it outputs the normalized and weighted fullmap that will be used in the subsequent map-model local correlation calculation.
3. Computes the map from the model (refined.pdb) using Mott-Bethe formula to the given resolution.
4. Apply weights and perform normalization on the calculated map
5. Finally, computes real space local correlation map between the normalized and weighted fullmap and the calculated map.

Output of this operation

Normalized and weighted maps:

- bin_normalized_fullmap.mrc, bin_normalized_halfmap1.mrc, bin_normalized_halfmap1.mrc, bin_normalized_modelmap.mrc

Local correlation maps:

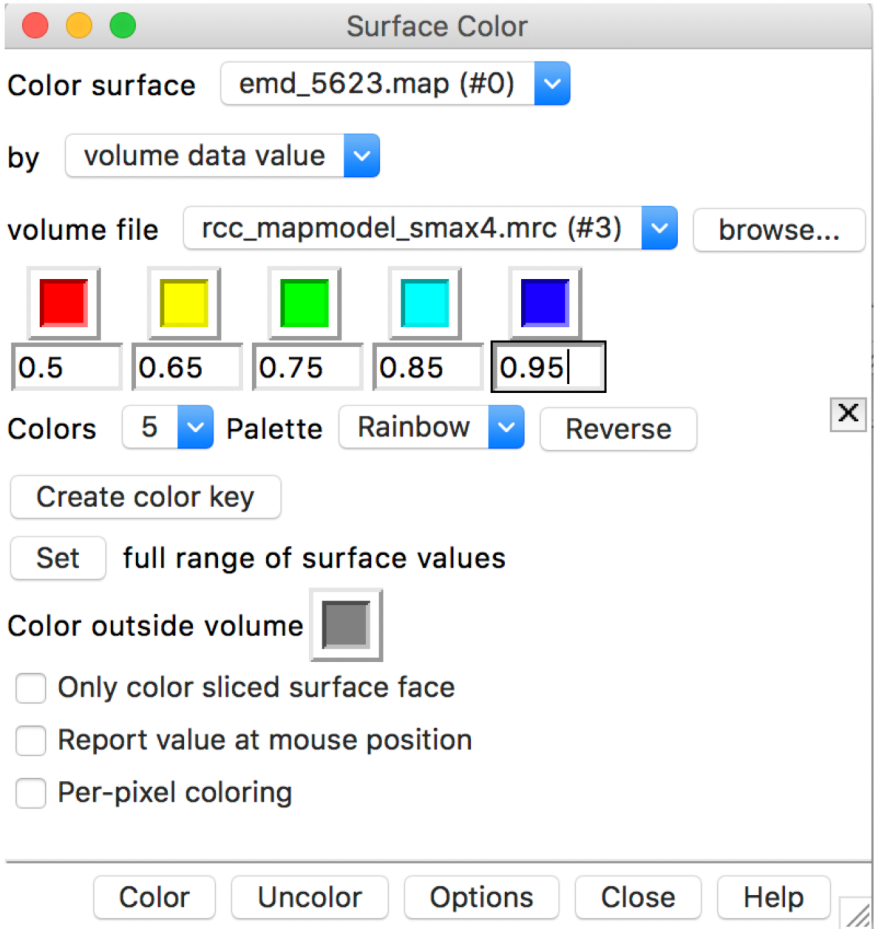
- rcc_halfmap_smax4.mrc, rcc_fullmap_smax4.mrc, rcc_fullmap_star_smax4.mrc, rcc_mapmodel_smax4.mrc

Step 4: Analyzing correlation maps

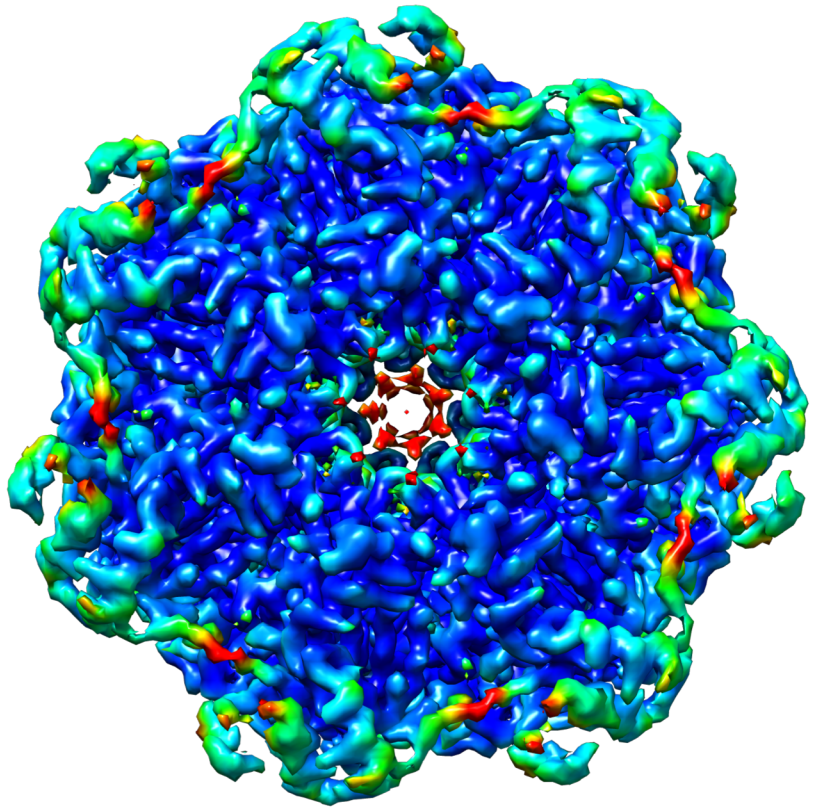
A convenient way to look at a correlation map is to use it as a map to colour another map.
In this case, we use rcc_fullmap_star_smax4.mrc and rcc_mapmodel_smax4.mrc to colour the primary map.
rcc_fullmap_star_smax4.mrc is the map with $\sqrt{CC_{full}}$ correlation
rcc_mapmodel_smax4.mrc is the local correlation map between fullmap and the model-based map (using normalized and weighted densities)

To colour the primary map by correlation maps,

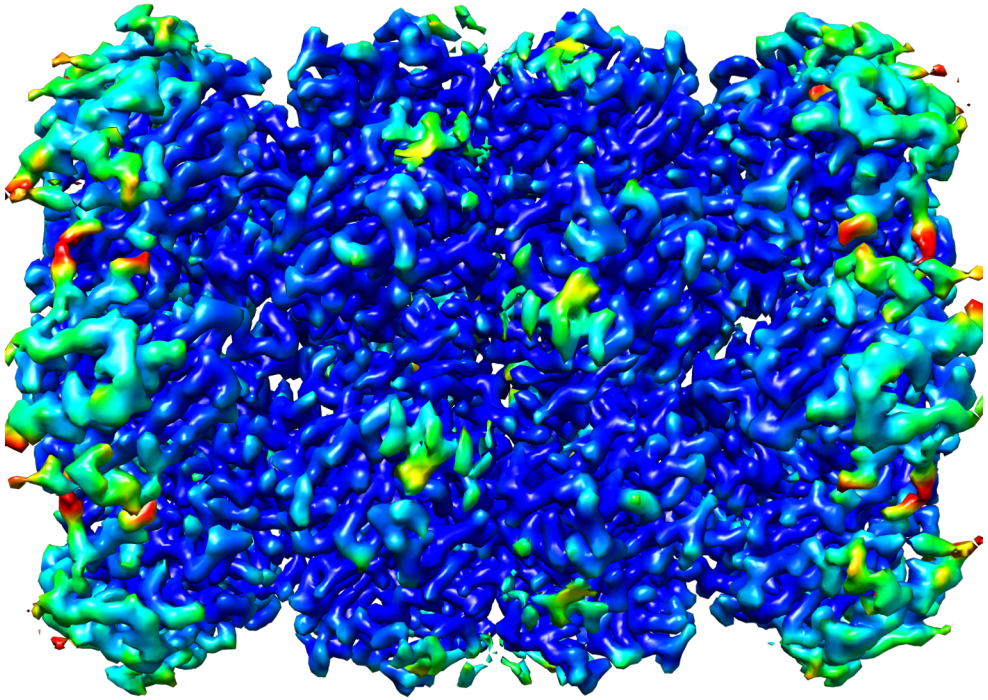
1. open primary map (emd_5623.map),
   rcc_fullmap_star_smax4.mrc and
   rcc_mapmodel_smax4.mrc in Chimera (you may use ChimeaX, as well)
2. In Volume Viewer – Tools > Surface Color
   Color surface – emd_5623.map
   By – volume data value
   Volume file – rcc_mapmodel_smax4.mrc
3. Then go to Options and click on Set button.
   That will automatically choose values for colour contours.
   But we would like to set those values by hand.
   So, give following values (0.5, 0.65, 0.75, 0.85, 0.95),
   and press Color.



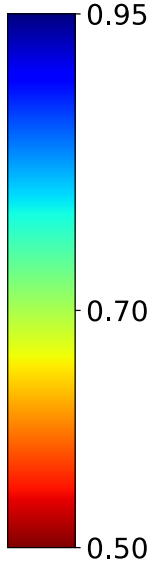
EMD-5623 primary map coloured by map-model local correlation



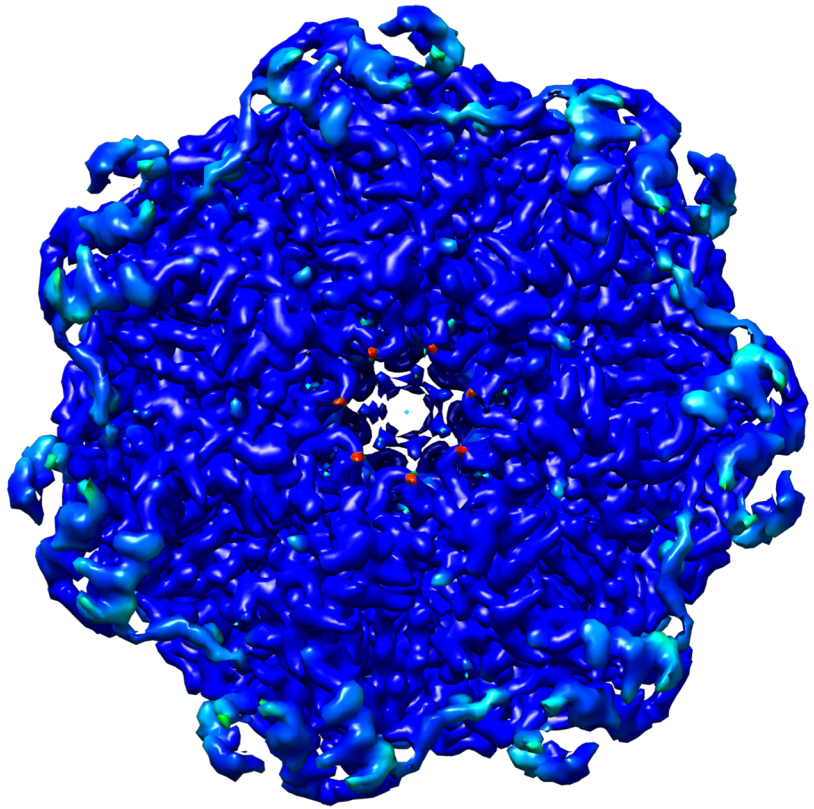
Top view



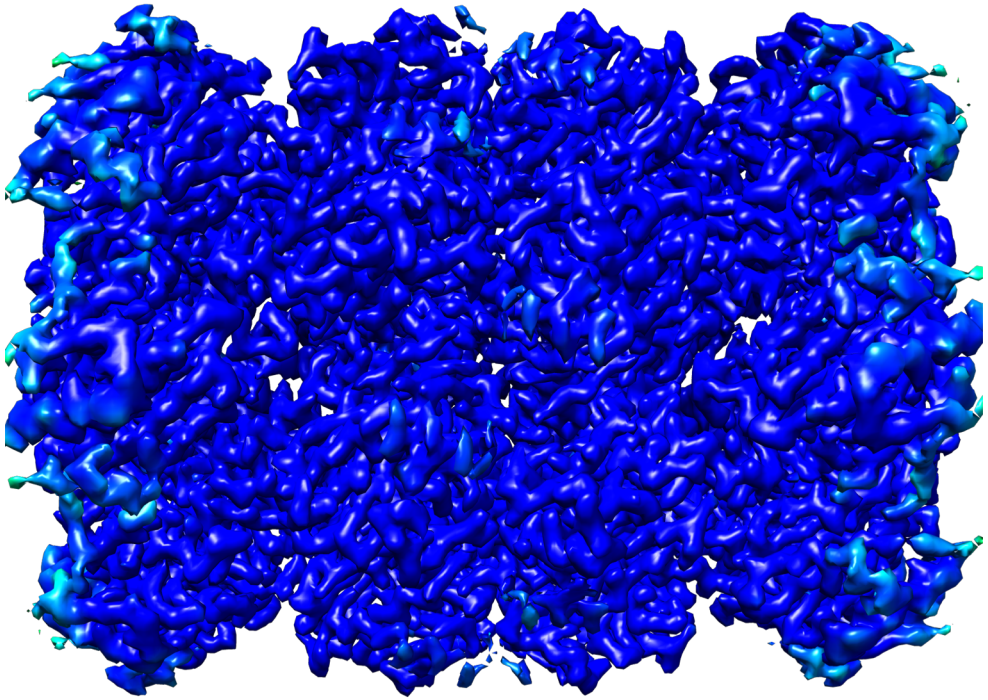
Side view



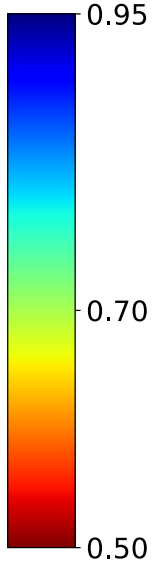
EMD-5623 primary map coloured by $\sqrt{CC_{full}}$ local correlation



Top view

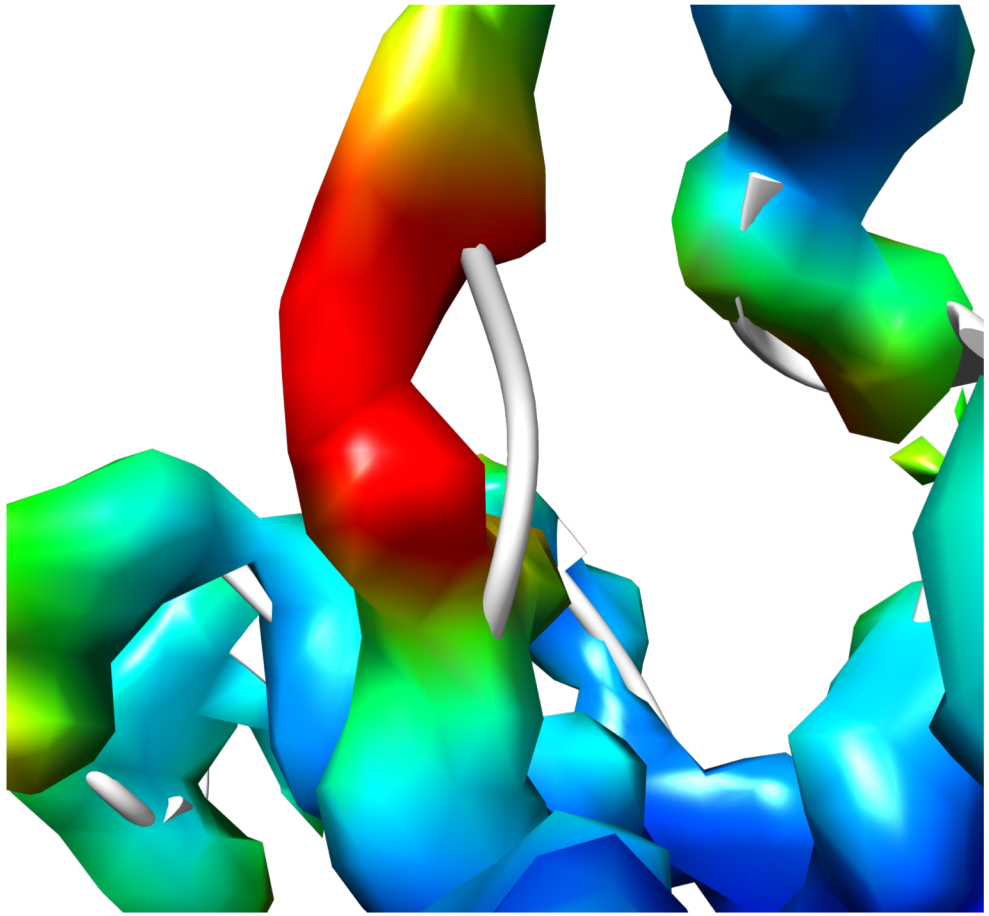


Side view

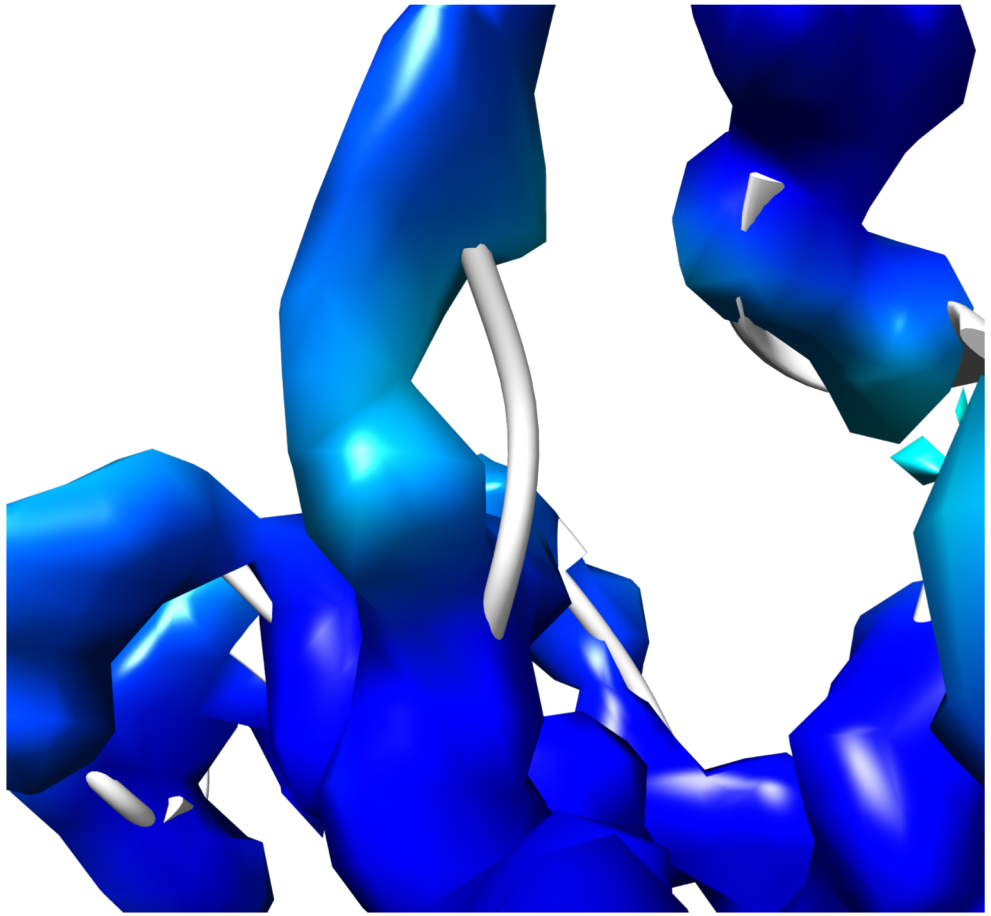


Some regions of interest (compared with model)

Lys52-Val54 of chain U

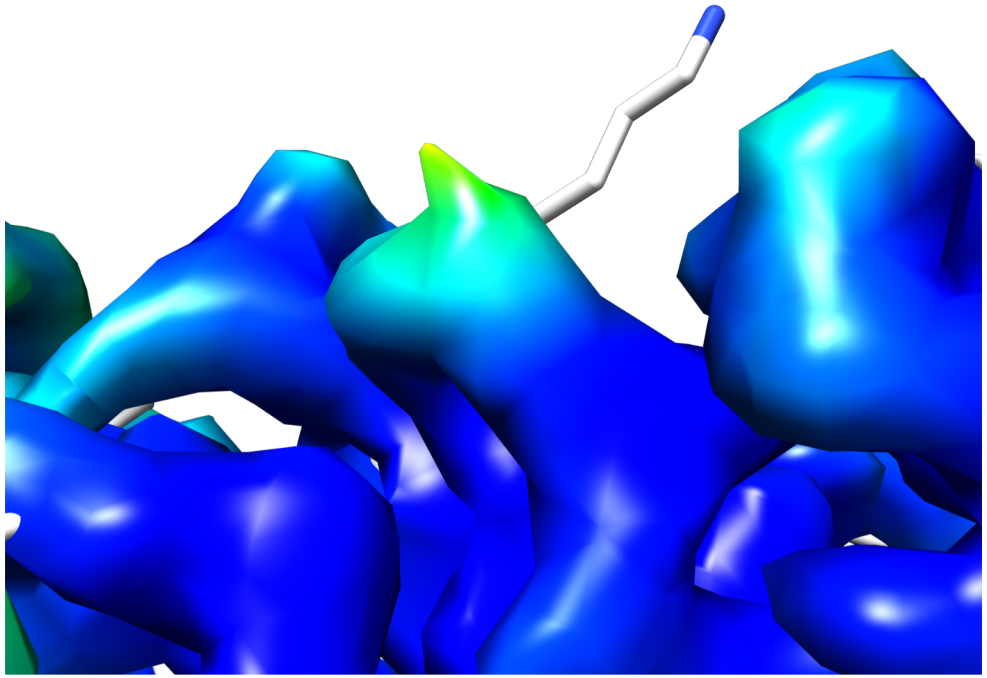


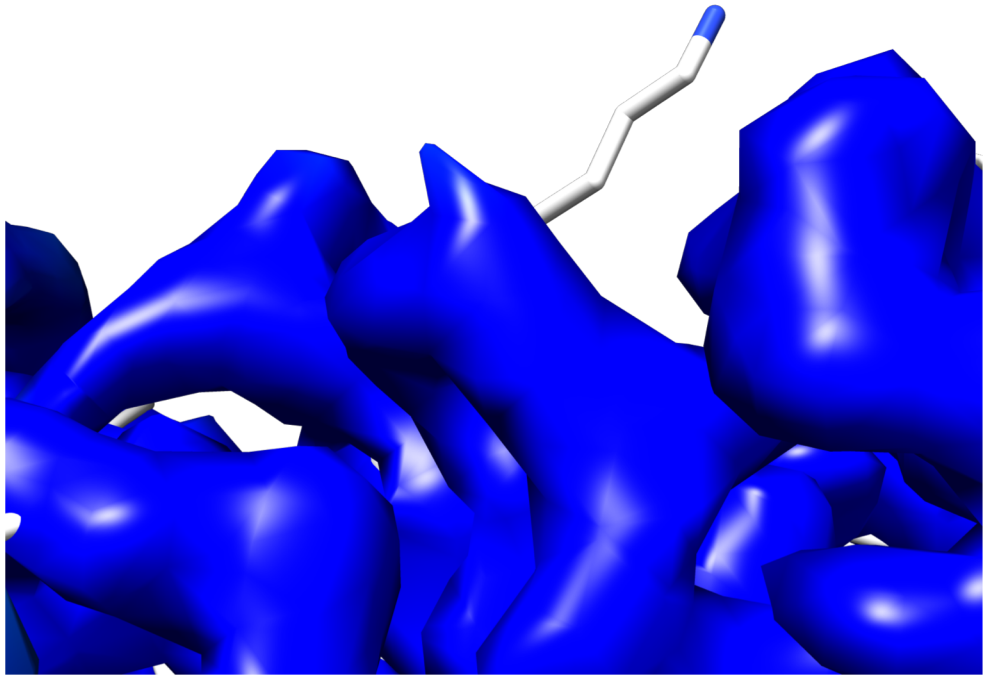
Map-model CC

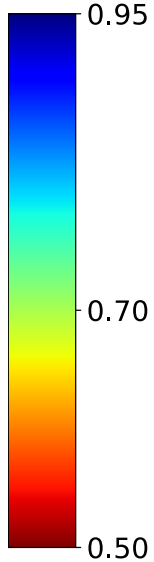


$\sqrt{CC_{full}}$

Lys9 of chain N







Map-model CC

$\sqrt{CC_{full}}$

Example 2: Detecting unmodeled density

In this example, we use real space local correlation to detect unmodeled density in EMD-11203 map.

Step 1: Download data

```
$ emda fetch --emd 11203 --all
Successfully created the directory /Users/ranganaw/MRC/REFMAC/tutorial_data/emda_rcc/EMD-
11203/
claimed resol  2.6
Mask Id:  emd_11203_msk_1.map
Half1id:  emd_11203_half_map_2.map.gz
Half2id:  emd_11203_half_map_1.map.gz
Fetched  11203
```

Step 2:
Refine the model 6zge that you just downloaded against the fullmap using servalcat (as you did in example 1)

```
$ servalcat refine_spa --model 6zge.cif --resolution 2.6 --halfmaps
emd_11203_half_map_1.map emd_11203_half_map_2.map --ncycle 10
```

Step 3: Calculate real space local correlation maps using EMD-11203 halfmaps and the refined.pdb

```
$ emda rcc --h1 emd_11203_half1.map --h2 emd_11203_half2.map --mdl refined.pdb --res 2.6 --knl 3 --nrm
```

Note: This calculation may take some time. Be patient ☺

Output of this operation

Normalized and weighted maps:

- bin_normalized_fullmap.mrc, bin_normalized_halfmap1.mrc, bin_normalized_halfmap1.mrc, bin_normalized_modelmap.mrc

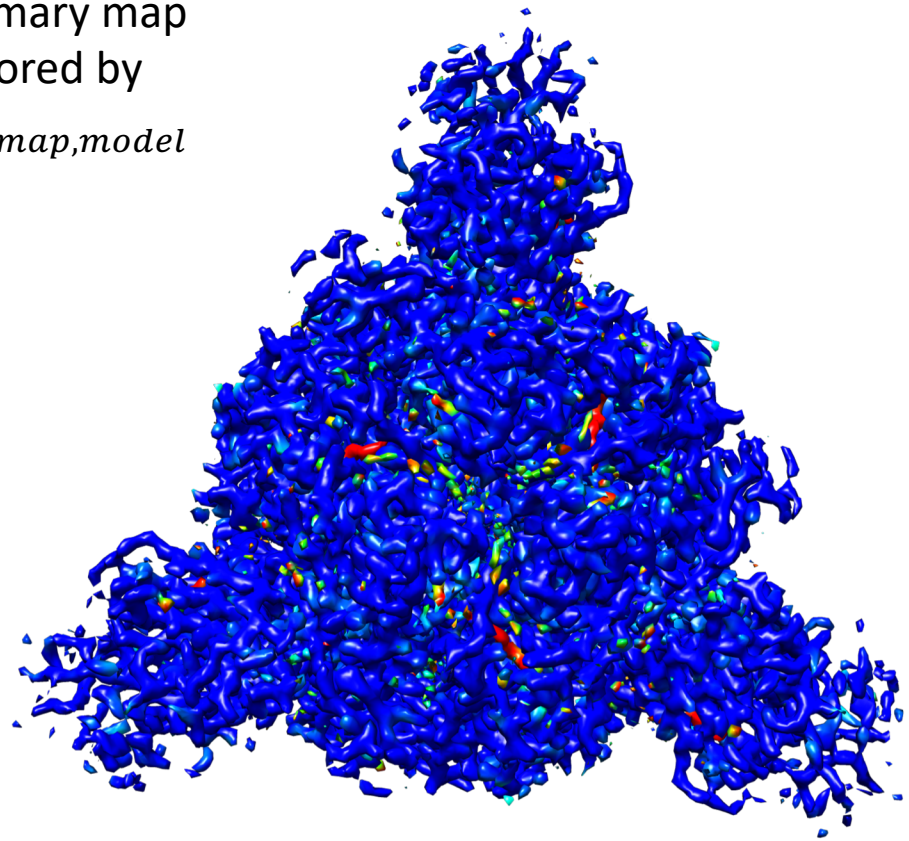
Local correlation maps:

- rcc_halfmap_smax3.mrc, rcc_fullmap_smax3.mrc, rcc_fullmap_star_smax3.mrc, rcc_mapmodel_smax3.mrc

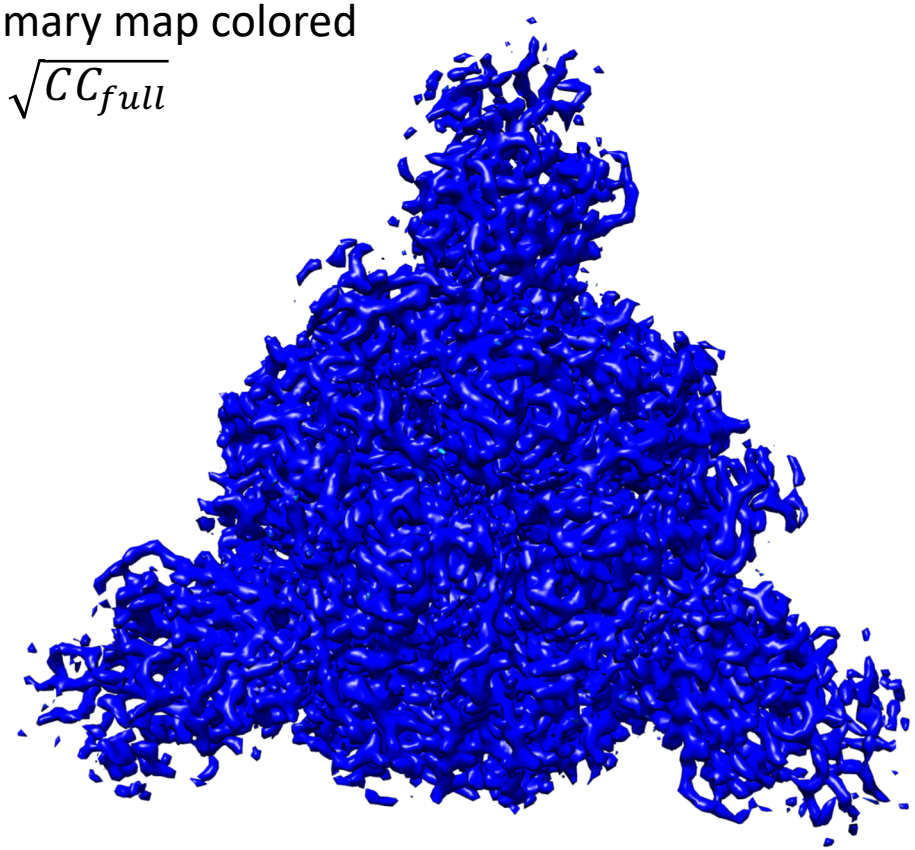
Step 4: Analyzing correlation maps.

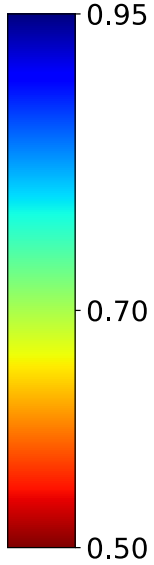
As before, let's colour the primary map by rcc_fullmap_star_smax3.mrc and rcc_mapmodel_smax3.mrc. Use the same procedure that explained in example 1.

Primary map colored by $CC_{map,model}$

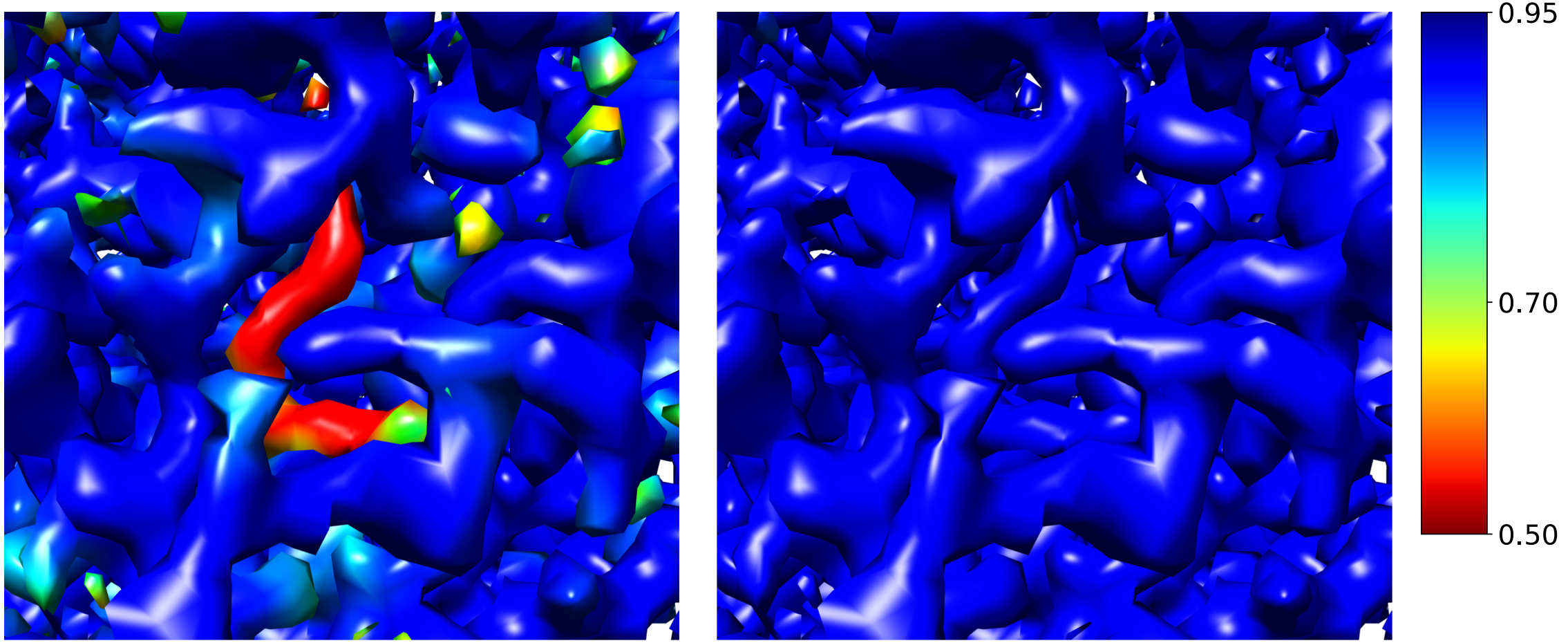


Primary map colored by $\sqrt{CC_{full}}$





Magnified view of an unmodeled density



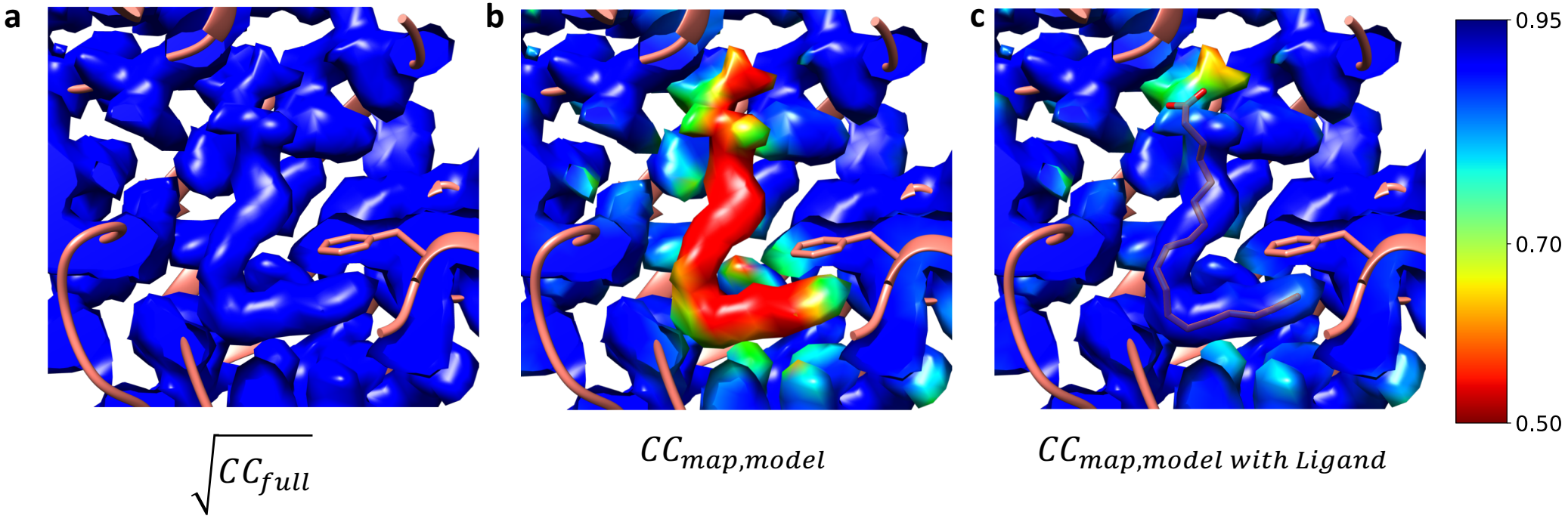
Primary map colored by $CC_{map,model}$

Primary map colored by $\sqrt{CC_{full}}$

No correspondence in the model for this density

Step 5: Use homology model 6z5b to account for this density as Linoleic acid.

1. use Chimera to fit the homology model to EMD-11203 map
2. refine coordinates and B values using servalcat (REFMAC)
3. Recalculate map-model local correlation using newly refined model.



a, b - unmodelled density in the primary map colored by $\sqrt{CC_{full}}$ and $CC_{map,model}$, respectively.

c – same density coloured by the $CC_{map,model}$ after the ligand has been fitted (using homology model 6zb5).

Step 6: Atomic correlation values

Atomic correlation values can be obtained from correlation maps by interpolation.

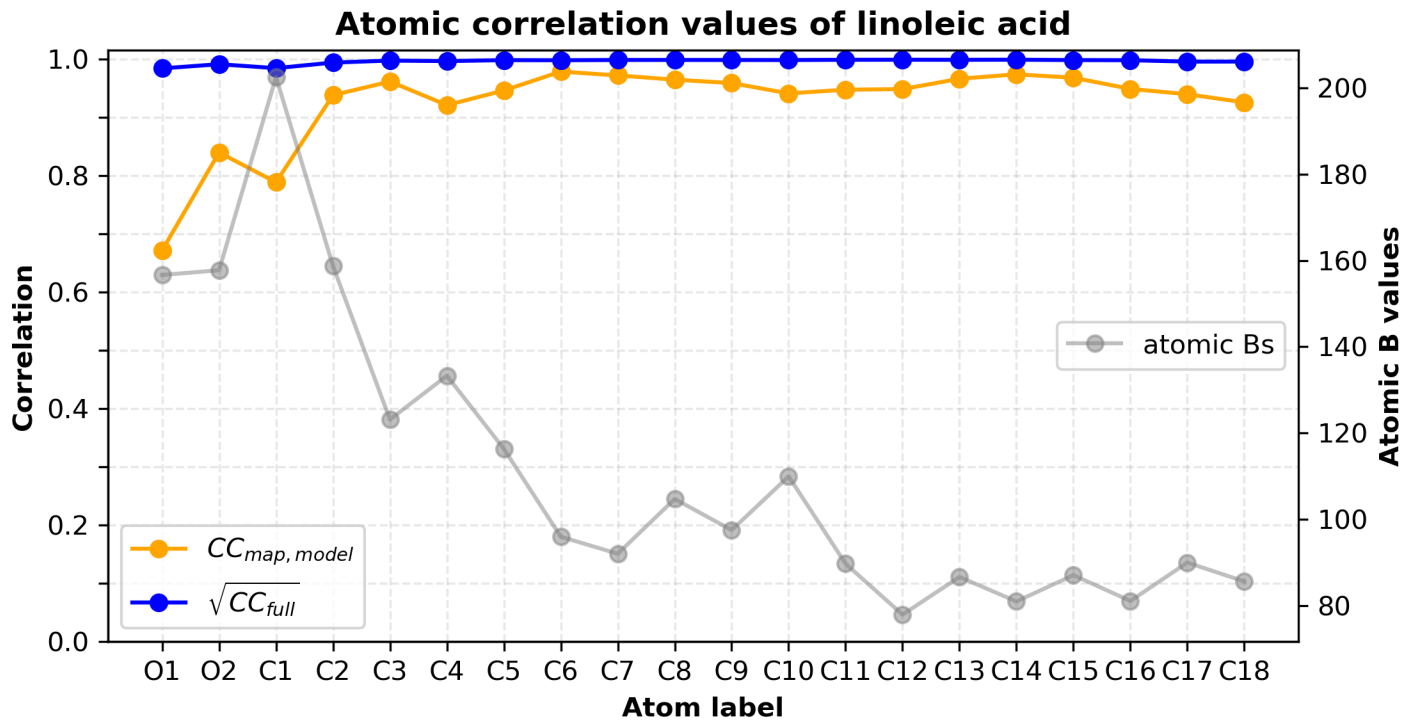
To help you with this, I have put a script at
https://gitlab.com/ccpem/emda/-/blob/master/tests/interpolate_at_atompos.py

Download this script to your working directory (where you have correlation maps) and run as

```
$ python interpolate_at_atompos.py rcc_fullmap_star_smax3.mrc
rcc_mapmodel_smax3.mrc refined_withLA.pdb
```

Note that your `rcc_mapmodel_smax3.mrc` is the correlation map you generated with the model with linoleic acid.

Atomic correlation values interpolated



Part 2: Map superposition

Goal

- To estimate the relative transformation between two selected receptor binding domains (RBDs) of two structures of SARS-CoV-2 spike proteins.
- The structures being compared are EMD-21997 and emd-21999.

Procedure involves following steps

- We need to estimate the global transformation between the two structures.
- Bring both structures into same coordinate frame.
- Carve out the desired RBDs from each map using a mask
- Estimate the relative transformation between the two RBDs.

Estimate the global transformation between the two structures

- We need to estimate the transformation between the two structures using densities.
- Then we need to apply the same transformation on the coordinates because we need to create a mask for RDB in each map for the next step.
- In this analysis, we choose emd-21997 map as static and emd-21999 as moving. The latter is brought on to former by rotating and translating.

EMDA command for estimating transformation:

```
$ emda overlay --map emd_21997.map emd_21999.map \
                --modellist 6x2a.cif
```

Overlay – calls the overlay functionality
Args:

--map

list of maps for overlay. The first map in the list is taken static.

--modellist

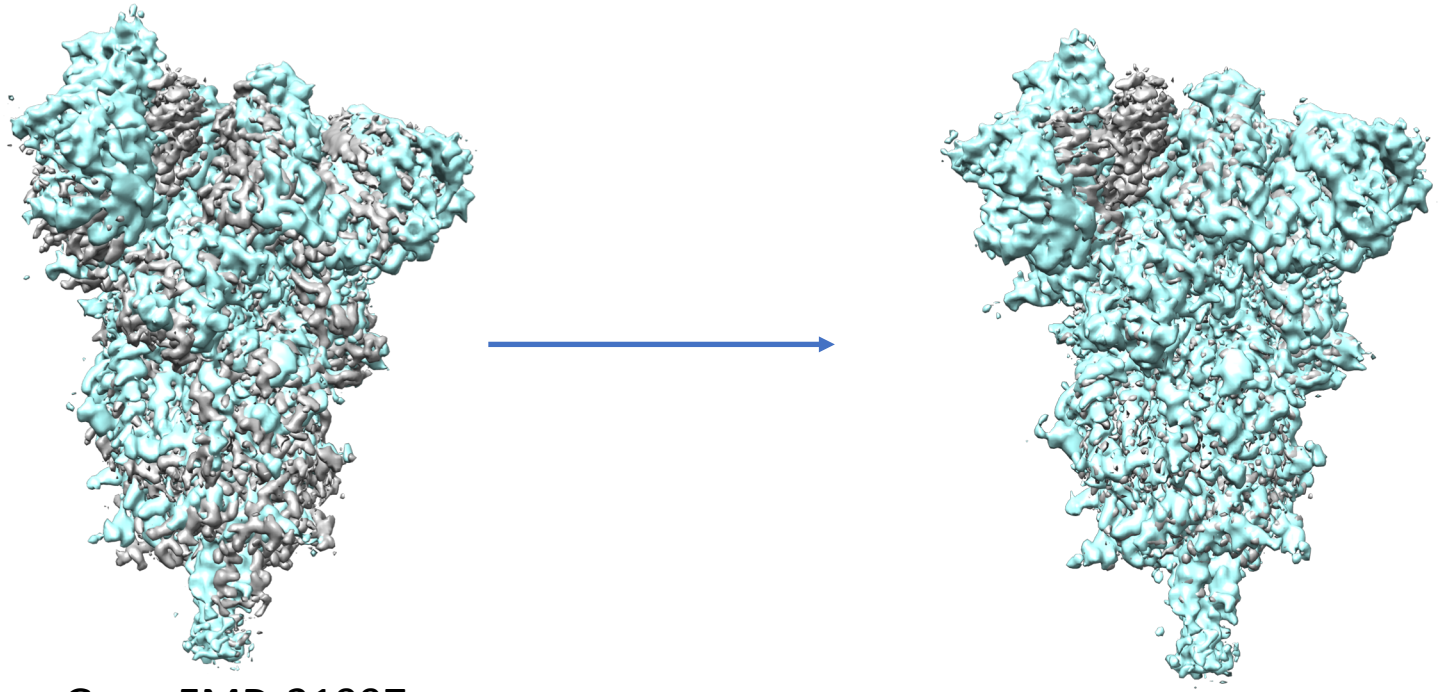
this list contains corresponding models to transform by the estimated transformation. In this case, we are interested in applying the transformation estimated using densities to coordinates. 6x2a is the atomic model corresponding to emd-21999 map.

This command outputs
static_map.mrc
fitted_map_1.mrc
emda_transformed_modellist_0.cif

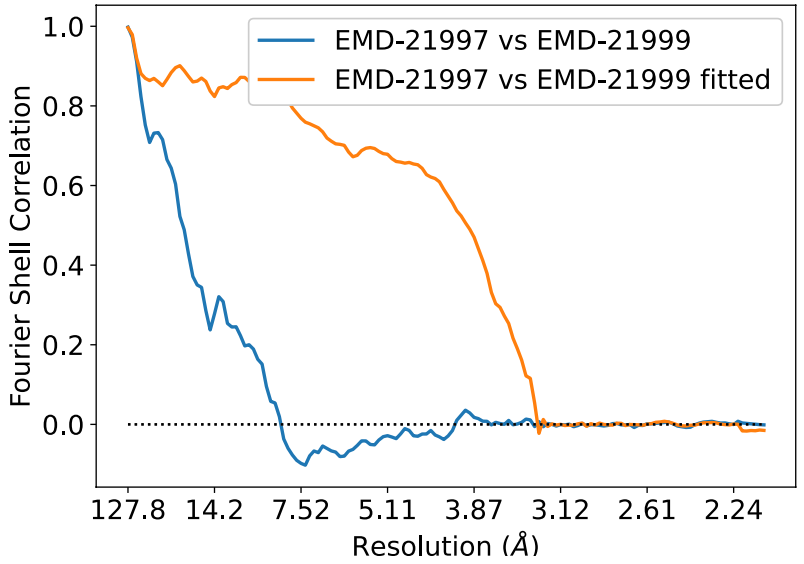
Rename the for clarity
static_map.mrc → static-21997.mrc
fitted_map_1.mrc → fitted-21999-on-21997.mrc
emda_transformed_modellist_0.cif → transformed-6x2a

Overlay optimization results

```
Fitting resolution:  3.4540544  (A)
Cycle#     Fval    Rot(deg)    Trans(A)    avg(FSC)

<truncated>

 19 2327865.7380 8.3500 4.1406 0.6213
 20 2327697.8927 8.3500 4.1413 0.6213
```



Gray: EMD-21997
Cyan: EMD-21999



Extracting RDBs in masks

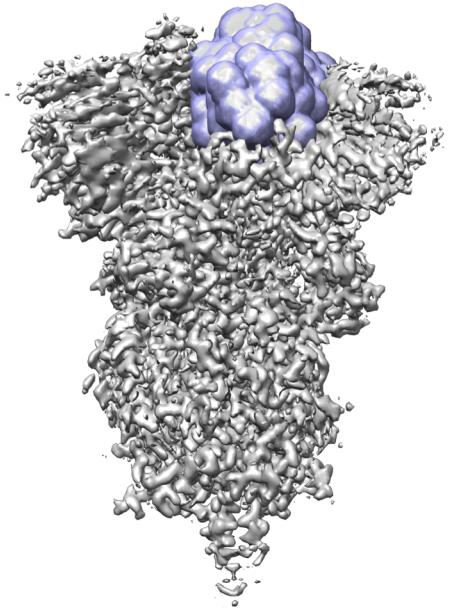
- To extract RDBs we need masks. We can generate masks from coordinates in EMDA.
- To do this, we first need to find the model corresponding to the desired RBD. For now, we edit coordinate files 6x29 and transformed-6x2a in a text editor (other options: Chimera or Pymol).
- Selected portions
  - Arg328-Ser530 of chain C of 6x29.cif → 6x29_closed_RBD.pdb
  - Phe329-Ser530 of chain C of transformed-6x2a.cif → 6x2a_closed_RBD.pdb
- Then, generate masks using edited coordinate files.
  - Instructions are in next slide.

```
$ emda modelmask --mdl 6x29_closed_RBD.pdb --map static_21997.mrc
```

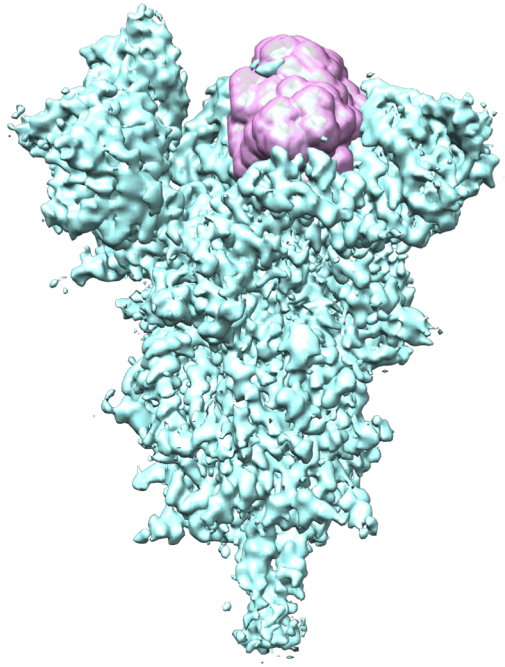
Please rename emda_atomic_mask.mrc to **emda_atomic_mask_6x29_closed_RBD.mrc**

```
$ emda modelmask --mdl 6x2a_closed_RBD.pdb --map static_21997.mrc
```

Please rename emda_atomic_mask.mrc to **emda_atomic_mask_6x2a_closed_RBD.mrc**



EMD-21997



Fitted EMD-21999

Overlay domains selected in masks

- You can give the fitted maps with corresponding masks to EMDA overlay.
- EMDA will perform the overlay and output fitted domains.
- Also, remember to include atomic model for the domain. EMDA will apply the same transformation on coordinates.
- EMDA command:

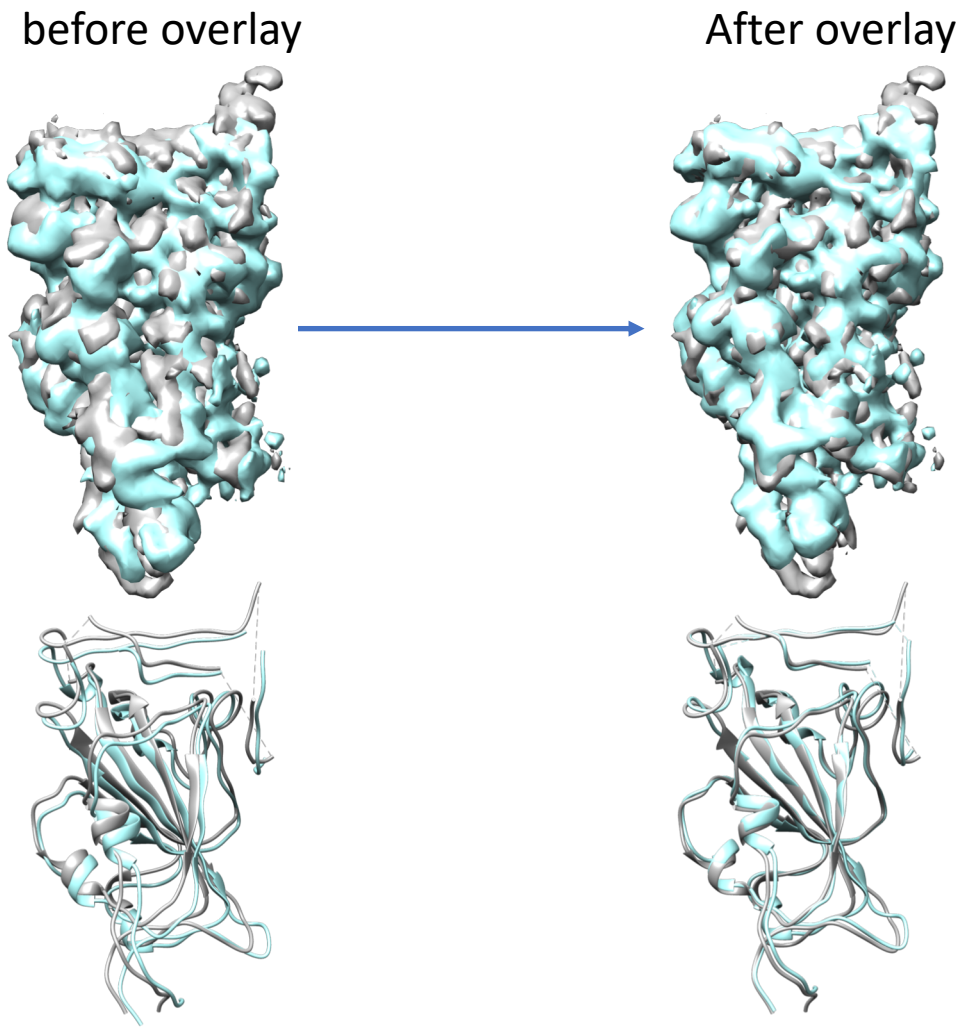
```
$ emda overlay --map static-21997.mrc fitted-21999-on-21997.mrc \
               --msk emda_atomic_mask_6x29_closed_RBD.mrc emda_atomic_mask_6x2a_cloased_RBD.mrc  \
               --modellist 6x2a_closed_RBD.pdb
```

Outputs:

static_map.mrc - RBD extracted from the static-21997.mrc
fitted_map_1.mrc - fitted RBD of the fitted-21999-on-21997.mrc
emda_transformed_modellist_0.cif - transformed coordinates of RBD 6x2a_closed_RBD.pdb

Domain overlay

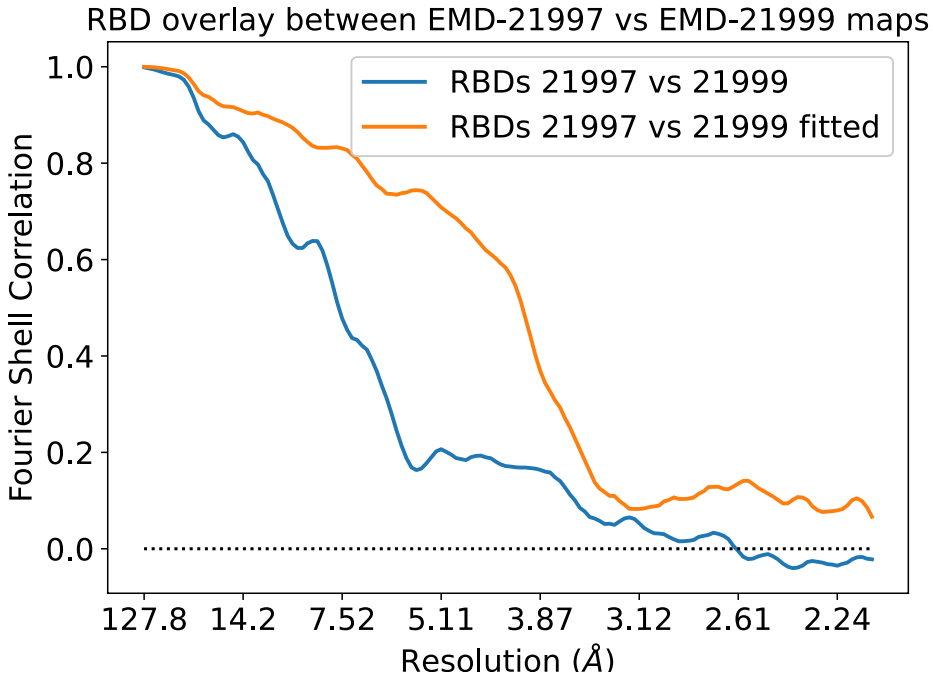
Gray: extracted RBD of EMD-21997
Cyan: extracted RBD of EMD-21999



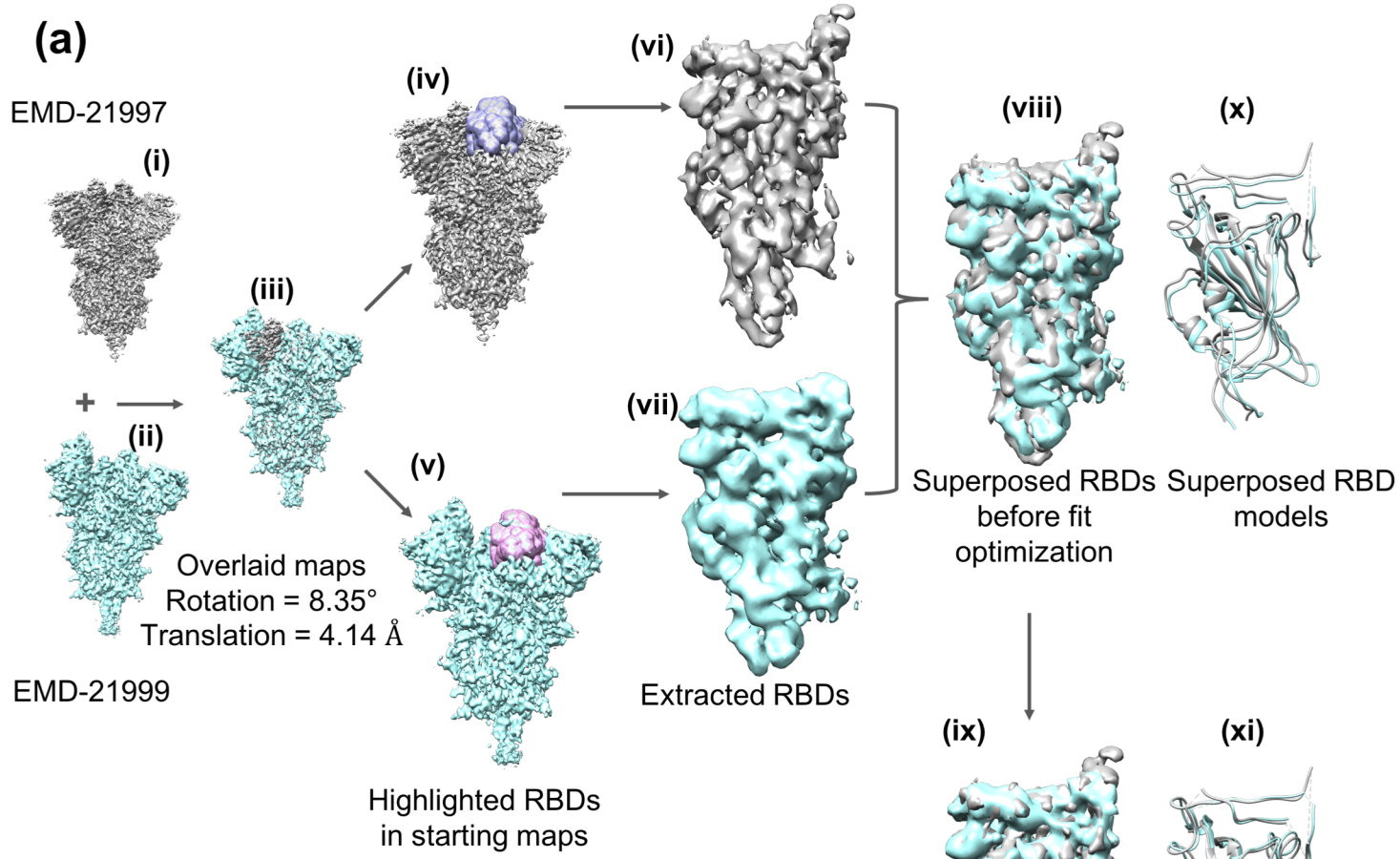
Fitting resolution: 3.4540544 (A)
Cycle# Fval Rot(deg) Trans(A) avg(FSC)

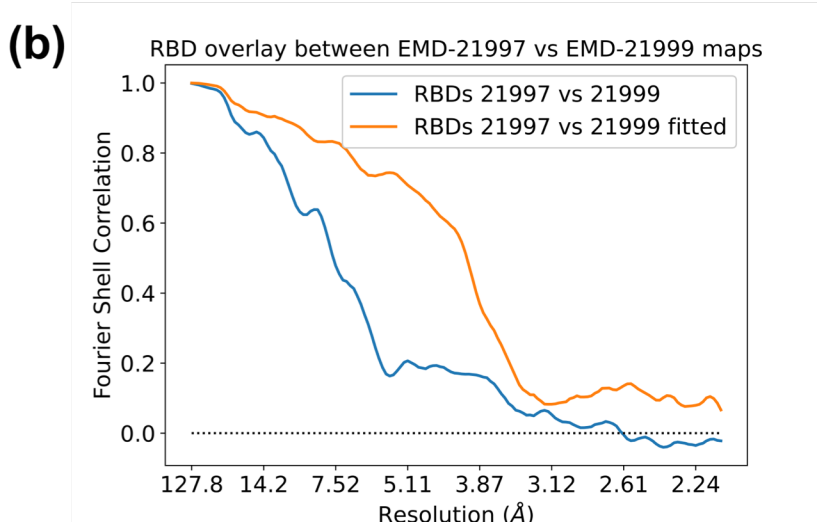
<truncated>

12 4196269.0887 3.3842 1.7620 0.5583
13 4196269.0007 3.3842 1.7620 0.5583



Full scheme of overlay





Part 3: Magnification refinement

Goal is to check the magnification of EMD-7770 and EMD-10574 cryo-EM maps w.r.t crystallography reference model 3dyp

Beta-galactosidase

Reference:

3dyp (X-ray model, 1.75 A)
(Juers et al. 2009)

Target maps:

1. EMD-7770 (EM map, 1.9 A)
(Bartesargi et al. 2018)

2. EMD-10574 (EM map, 2.2 A)
(Saur et al. 2020)

Step 1: Download data

```
$ emda fetch --emd 7770 --all
Successfully created the directory /Users/ranganaw/MRC/REFMAC/tutorial_data/EMD-7770/
claimed resol  1.9
Mask Id:  emd_7770_msk_1.map
Half1id:  emd_7770_half_map_1.map.gz
Half2id:  emd_7770_half_map_2.map.gz
Fetched  7770
```

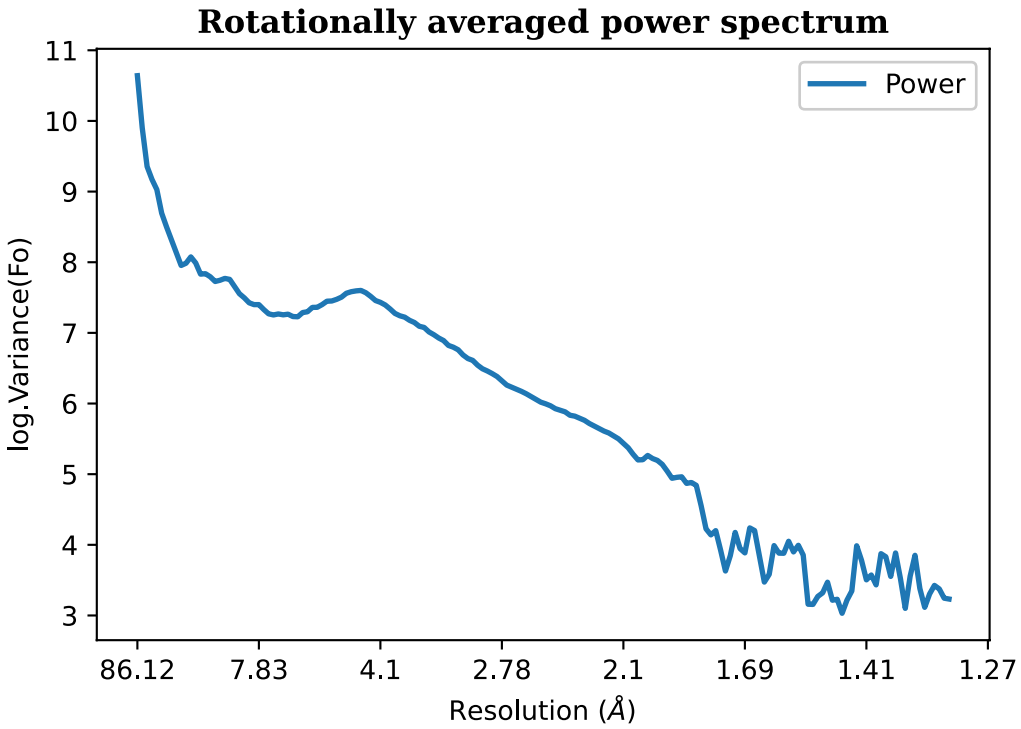
```
$ emda fetch --emd 10574 --all
Successfully created the directory /Users/ranganaw/MRC/REFMAC/tutorial_data/EMD-10574/
claimed resol  2.2
Mask Id:  None
Half1id:  emd_10574_half_map_2.map.gz
Half2id:  emd_10574_half_map_1.map.gz
Fetched  10574
```

Download **3dyp** model from RCSB – https://www.rcsb.org/structure/3DYP

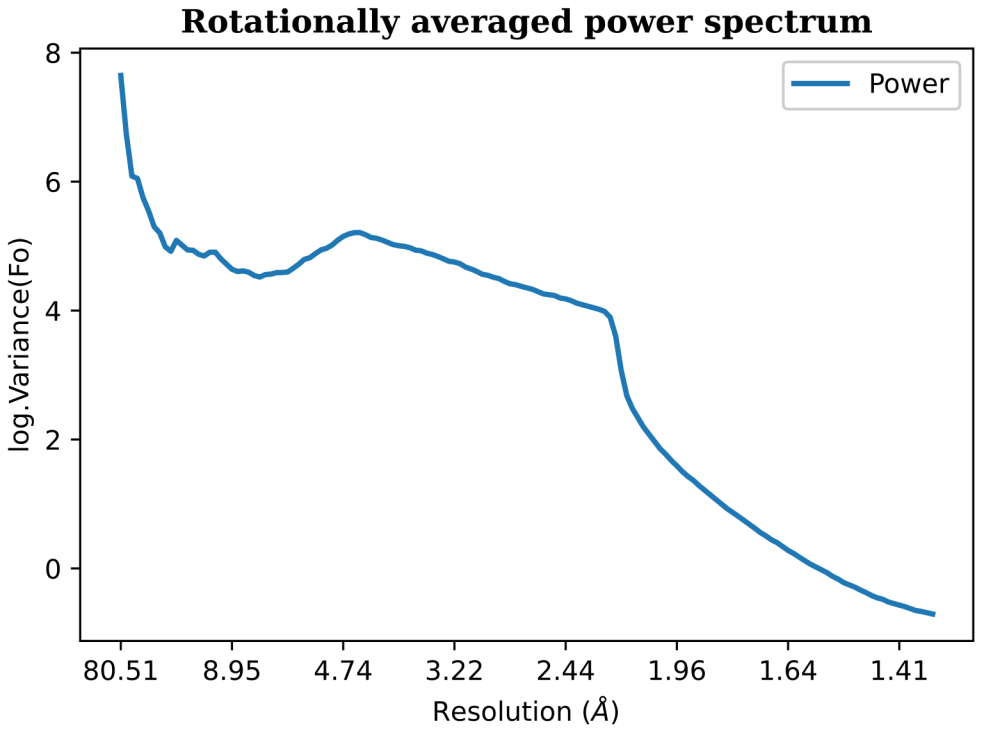
Step 2: Analyze EM maps emd_7770.map and emd_10574.map

Calculate their power spectra

```
$ emda power --map emd_7770.map
```



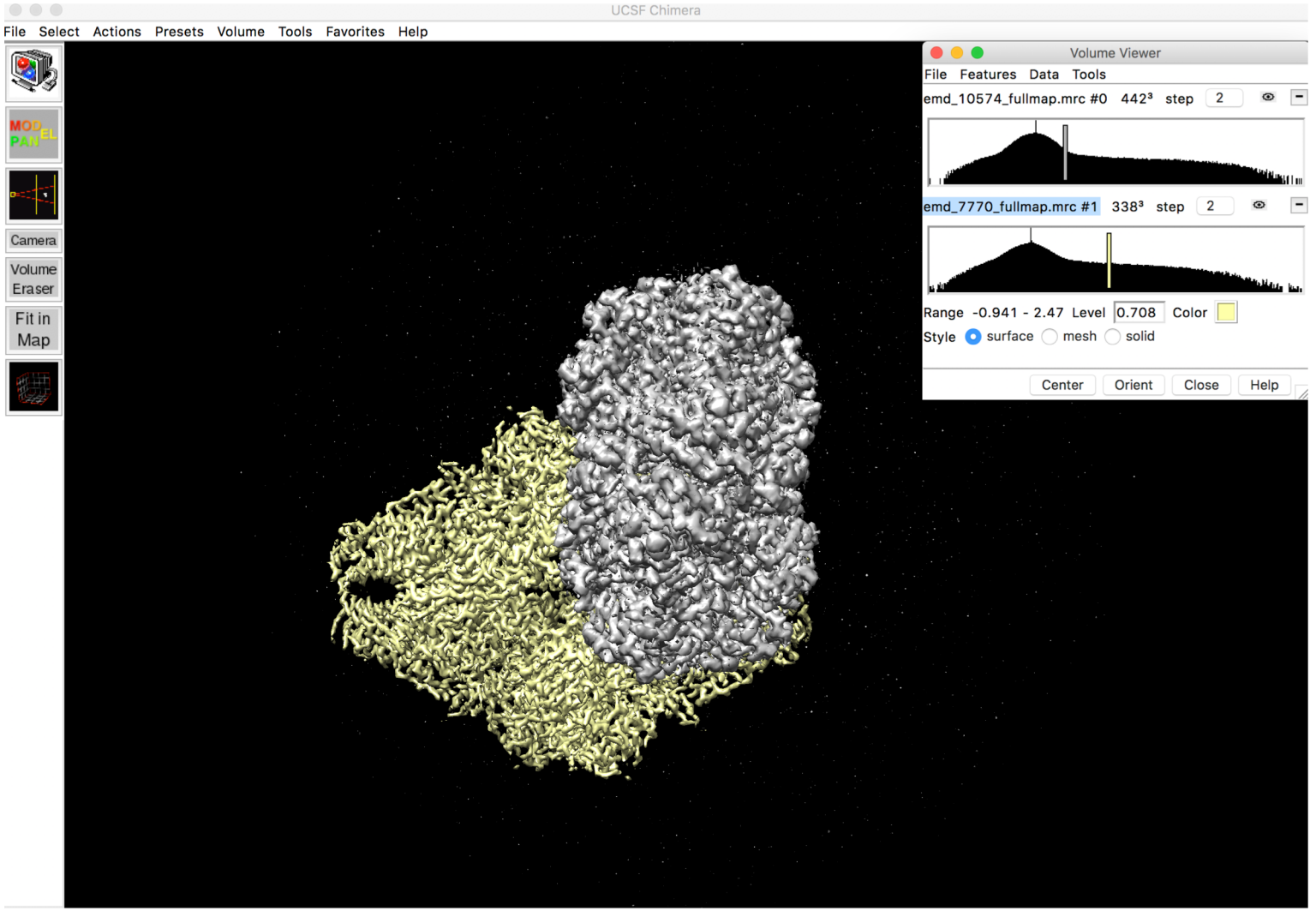

```
$ emda power --map emd_10574.map
```



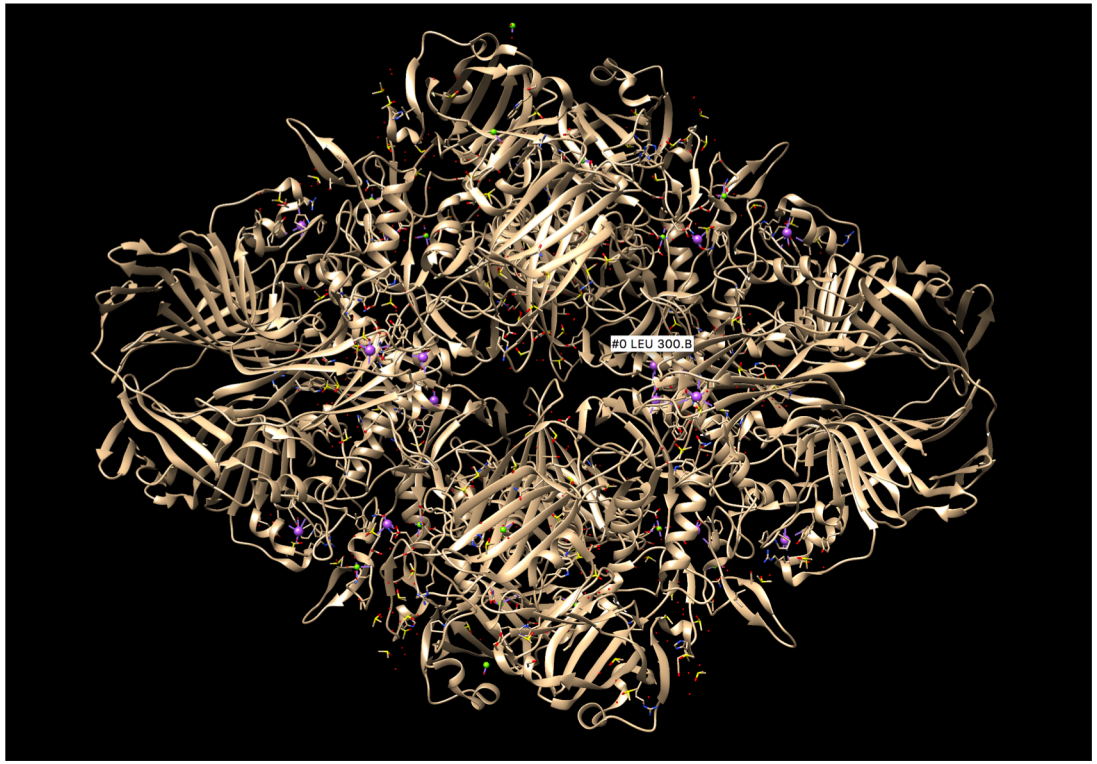
Since EMD-10574 map is filtered, let's use fullmaps for both entries.

To get the combined map (average of both half maps)

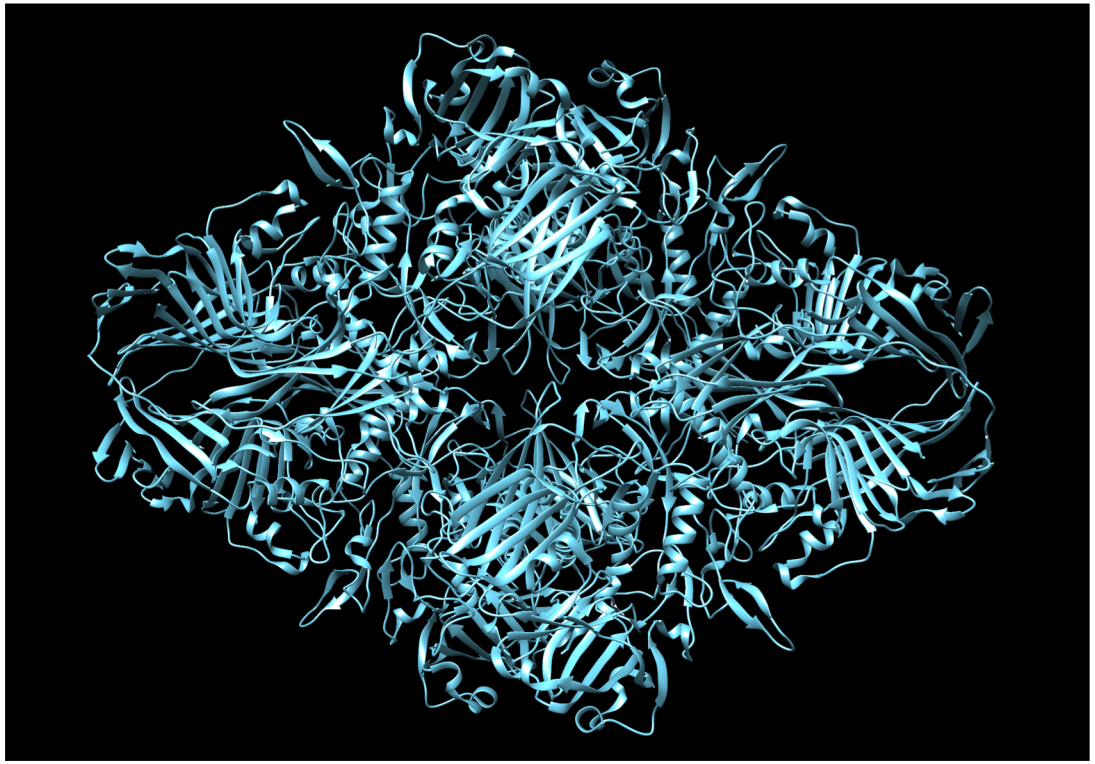
```
$ emda half2full --h1 emd_7770_half1.map --h2 emd_7770_half2.map --out emd_7770_fullmap.mrc
$ emda half2full --h1 emd_10574_half1.map --h2 emd_10574_half2.map --out emd_10574_fullmap.mrc
```



We need just the polymer from our crystallography model. So, remove all non-protein atoms from the 3dyp model. This can be done easily in a text editor.

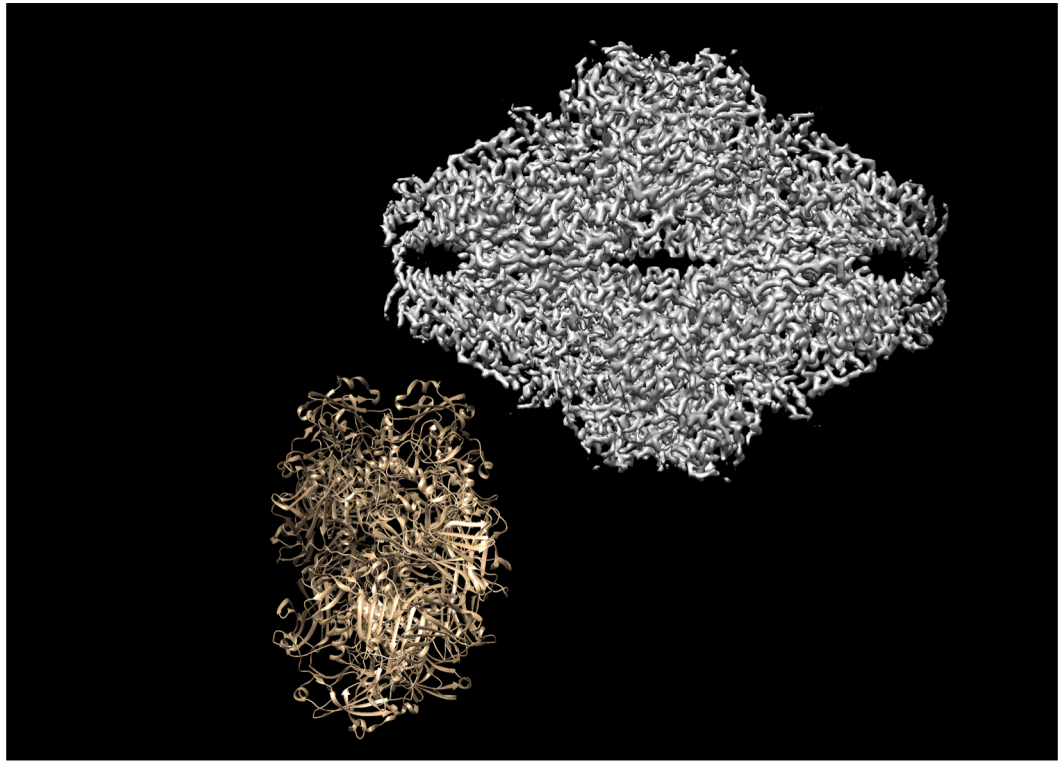


3dyp with all atoms

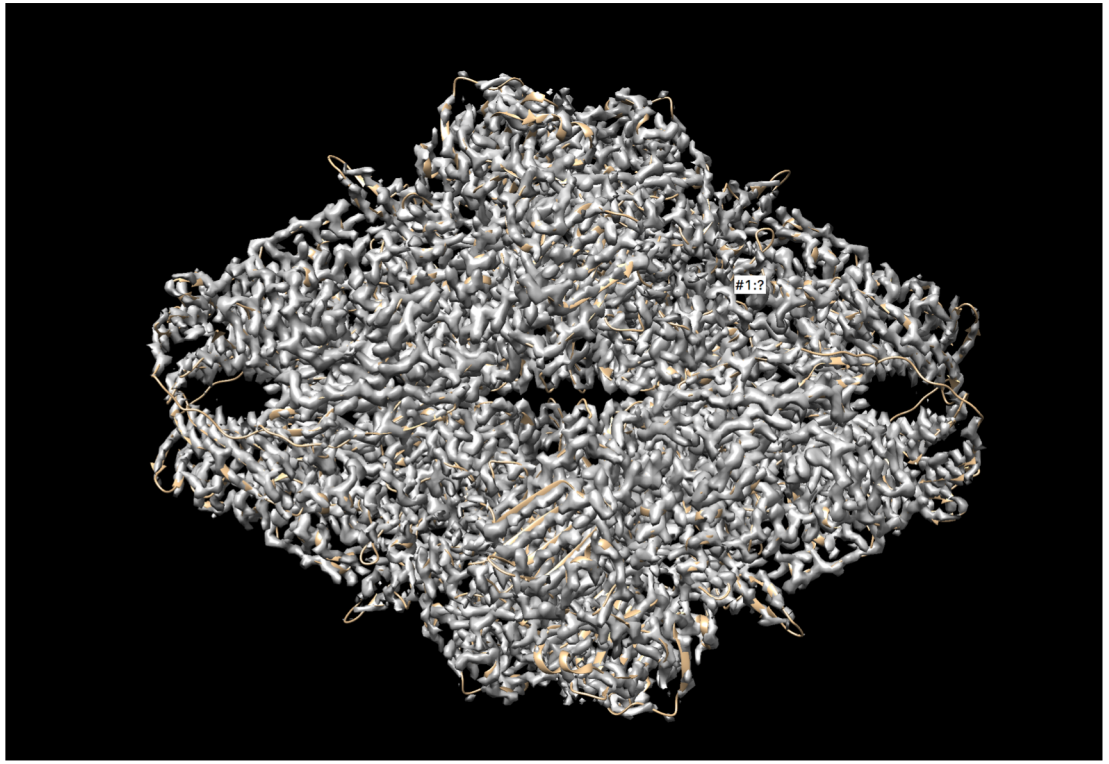


3dyp-polymer after removing all non-protein atoms

Step 3: Fit crystallography model to emd_7770_fullmap
This can be done in Chimera using ”Fit in Map” option or using Molrep



Model shown (in brown) is the 3dyp model after removing non-protein atoms. Emd_7770 map is shown in grey.



After fitting model into map.

Save the fitted model w.r.t EM map as reference_model.pdb

Step 4: Calculate the map from the reference model up to 1.9 A.

Use the same sampling and cell as of emd_7770_fullmap.mrc

To get information about emd_7770_fullmap.mrc

```
$ emda info --map EMD-7770/emd_7770_fullmap.mrc
EMD-7770/emd_7770_fullmap.mrc (338, 338, 338) [215.30601501 215.30601501 215.30601501]
Unit cell:  [215.30601501 215.30601501 215.30601501  90.          90.
  90.        ]
Sampling:  (338, 338, 338)
Pixel size:  0.637
Origin:  [0, 0, 0]
```

Now, let's calculate the model-based map

```
$ emda model2map --mdl reference_model.pdb --res 1.9 --dim 338 338 338 --cel 215.30601501
215.30601501 215.30601501 90 90 90
```

model2map uses Gemmi to generate map from the model.
In a moment modelmap_gm.mrc will be available in the working directory.
Rename it to reference_map.mrc with

```
$ mv modelmap_gm.mrc reference_map.mrc
```

Your reference map is now ready to use in magnification refinement.

Step 5: Resampling target maps on the reference map

Since the reference_map.mrc was generated on the emd_7770_fullmap.mrc both have the same sampling and pixel sizes.

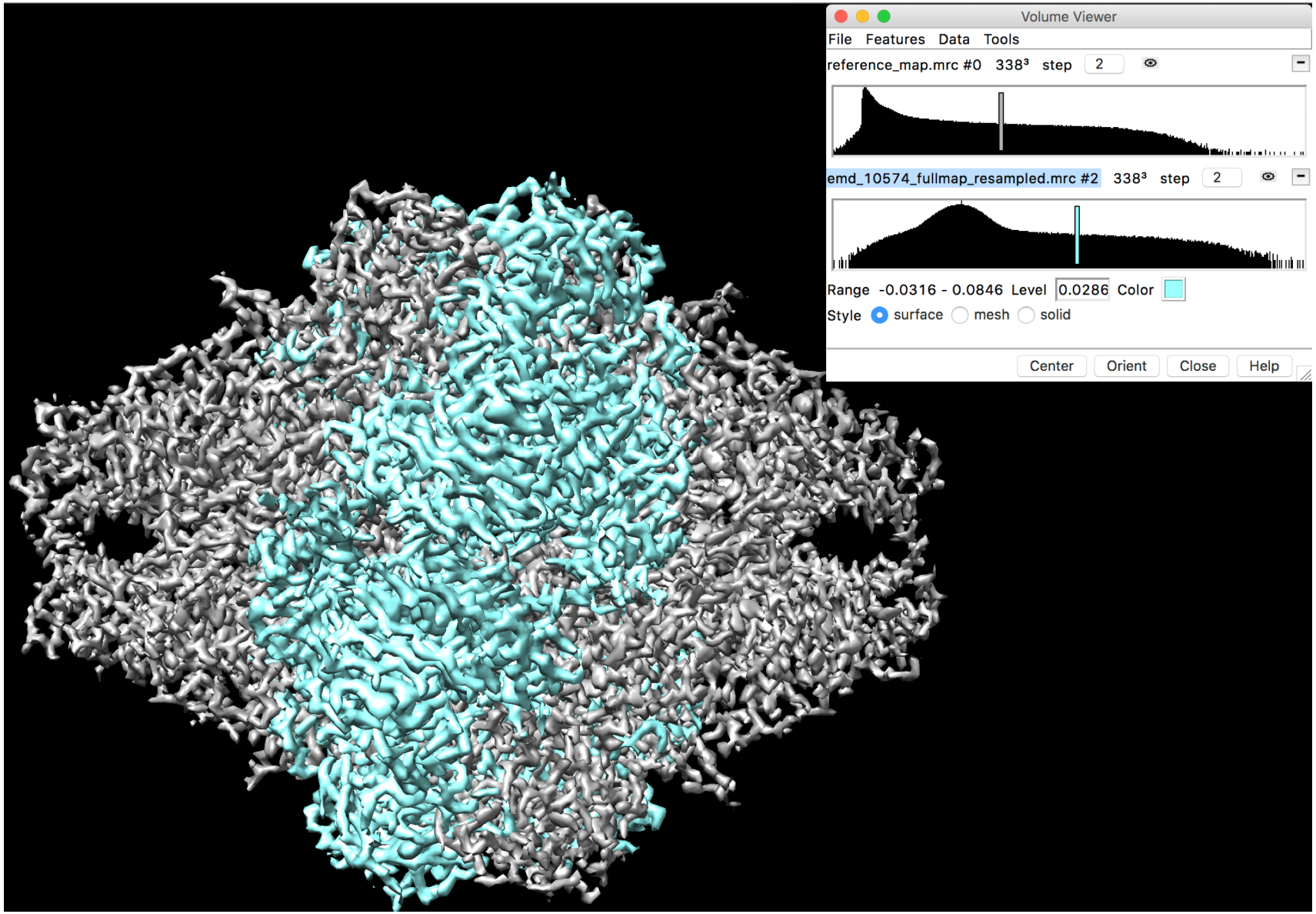
But, emd_10574_fullmap.mrc has different sampling and thus different pixel sizes.
We need to make sure all maps have the same sampling and pixel size.

Now, let's resample emd_10574_fullmap.mrc on reference_map.mrc

```
$ emda resamplemap2map --map1 reference_map.mrc --map2 EMD-10574/emd_10574_fullmap.mrc --out
emd_10574_fullmap_resampled.mrc
reference_map.mrc (338, 338, 338) [215.30601501 215.30601501 215.30601501]
EMD-10574/emd_10574_fullmap.mrc (442, 442, 442) [300.55999756 300.55999756 300.55999756]
Current pixel size:  [0.68, 0.68, 0.68]
Target pixel size:  [0.637, 0.637, 0.637]
arr.shape:  (442, 442, 442)
Resizing in Fourier space and transforming back
[-30 -30 -30]
upsampling...
cropping image...
```

Find emd_10574_fullmap_resampled.mrc in the working directory

Let’s have a look at emd_7770_fullmap.mrc and emd_10574_fullmap_resampled.mrc in Chimera



Note that these two maps are almost perpendicular to each other.

Step 6: aligning emd_10574_fullmap_resampled.mrc onto reference_map.mrc

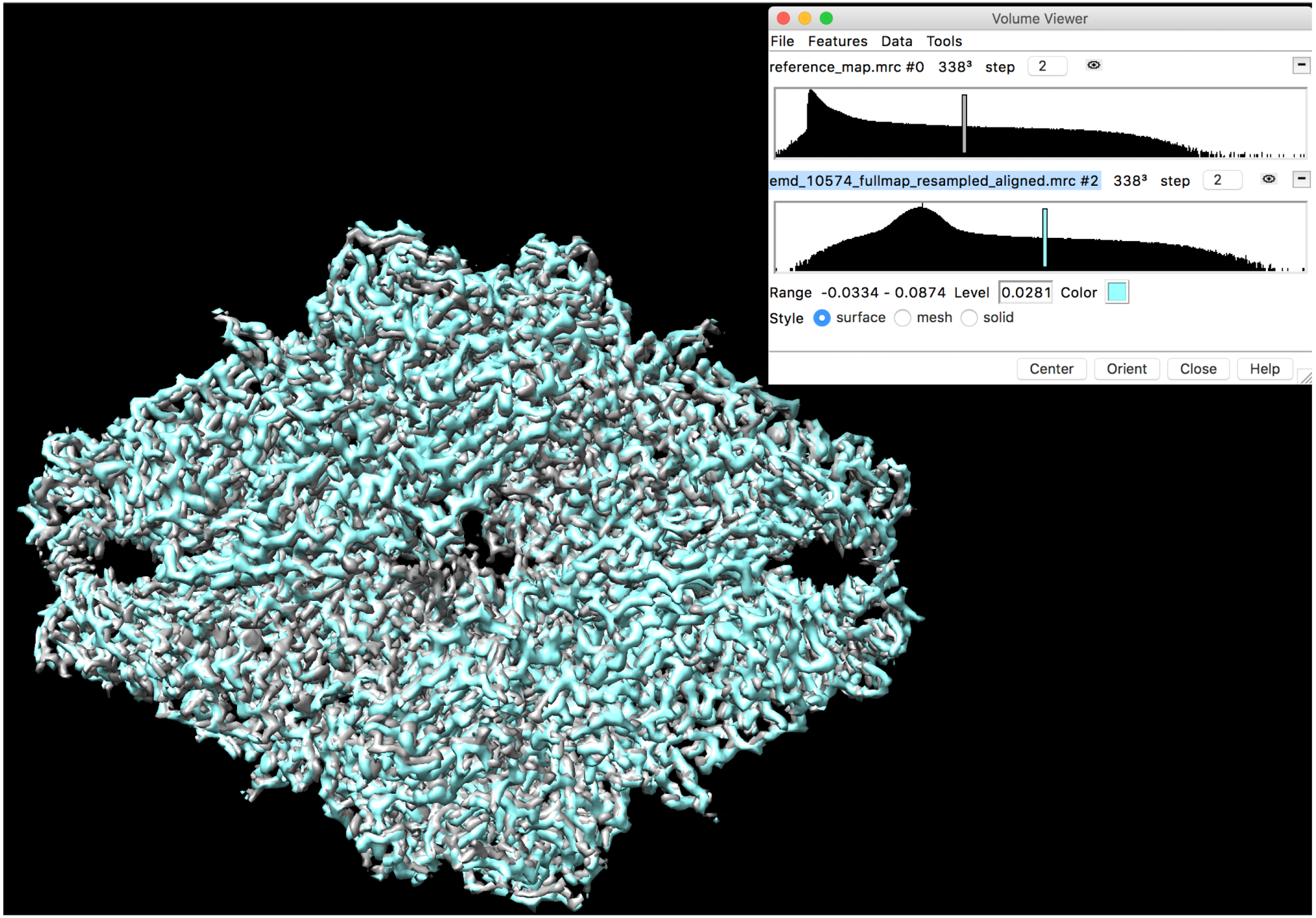
The idea is to get their orientations approximately correct. During magnification refinement, their orientations are further optimized.

There are many ways to align those two maps.

1. Use Chimera – keep the reference map static and bring the other map on to it.
2. Use Molrep to align them.
3. we can use a PCA analysis on their variance-covariance matrices and match the principle components.
   helper script https://gitlab.com/ccpem/emda/-/blob/master/tests/pca_for_fitting.py
   You can run this script as

   $python pca_for_fitting.py reference_map.mrc emd_10574_fullmap_resampled.mrc

   This will output pca_aligned.map. Please rename it to emd_10574_fullmap_resampled_aligned.mrc



As you see, they are pretty well aligned.
You started to see the magnification difference.

Now, let's refine magnifications.

Step 7: Magnification refinement

Refining magnification of emd_7770_fullmap.mrc against the reference_map.mrc

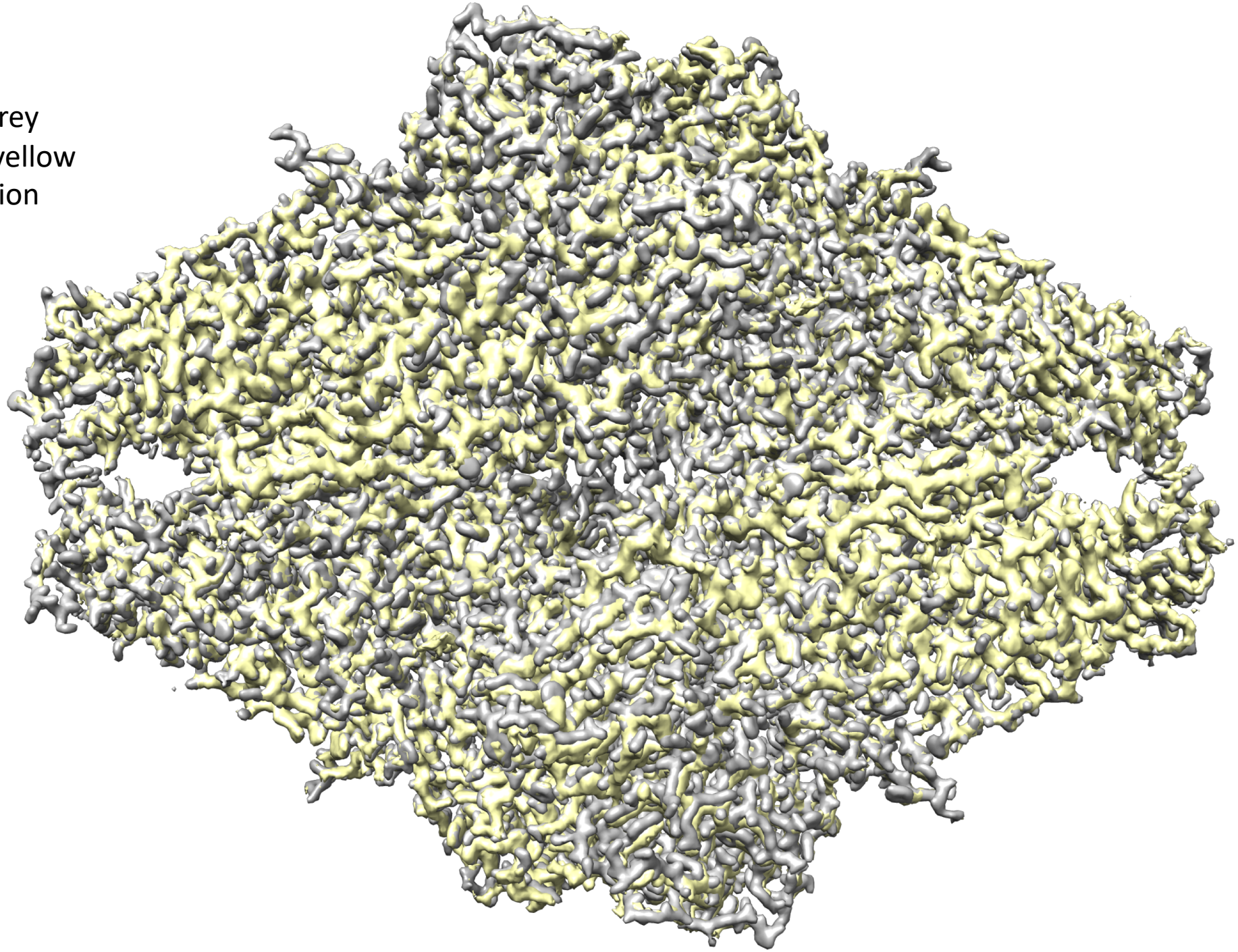
```
$ emda magref --map EMD-7770/emd_7770_fullmap.mrc --ref reference_map.mrc

<truncated>
Optimising overlay.....Done


Optimising map magnification...


ifit    cycle#    func
val.   magnification
    1     0 2377414.3486 1.0000
    1     1 2375491.7329 1.0027
    1     2 2375491.7310 1.0027
magnification error (%):  0.268
Magnification corrected map
emda_magcorretedmap_1.mrc was written.
```

Refining magnification of emd_10574_fullmap_resampled_align.mrc against the reference_map.mrc

```
$ emda magref --map emd_10574_fullmap_resampled_aligned.mrc --ref reference_map.mrc


<truncated>
Optimising overlay.....Done


Optimising map magnification...


ifit    cycle#    func
val.   magnification
    1     0 1613534.2744 1.0000
    1     1 1549638.7018 1.0171
    1     2 1549638.6560 1.0171
    1     3 1549638.6560 1.0171
magnification error (%):  1.708
```

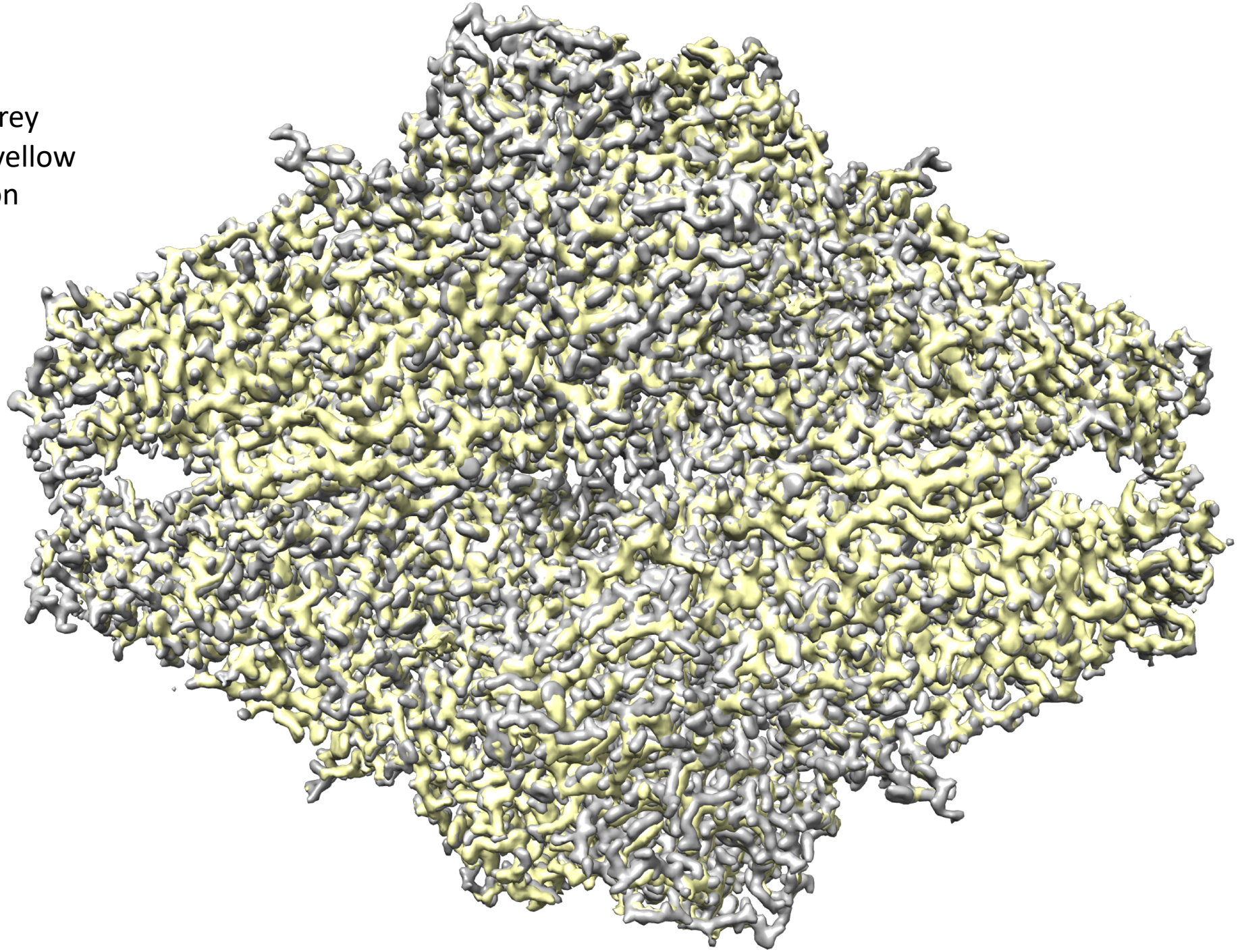
EMD-7770

Reference in grey
Emd_7770 in yellow
Before correction



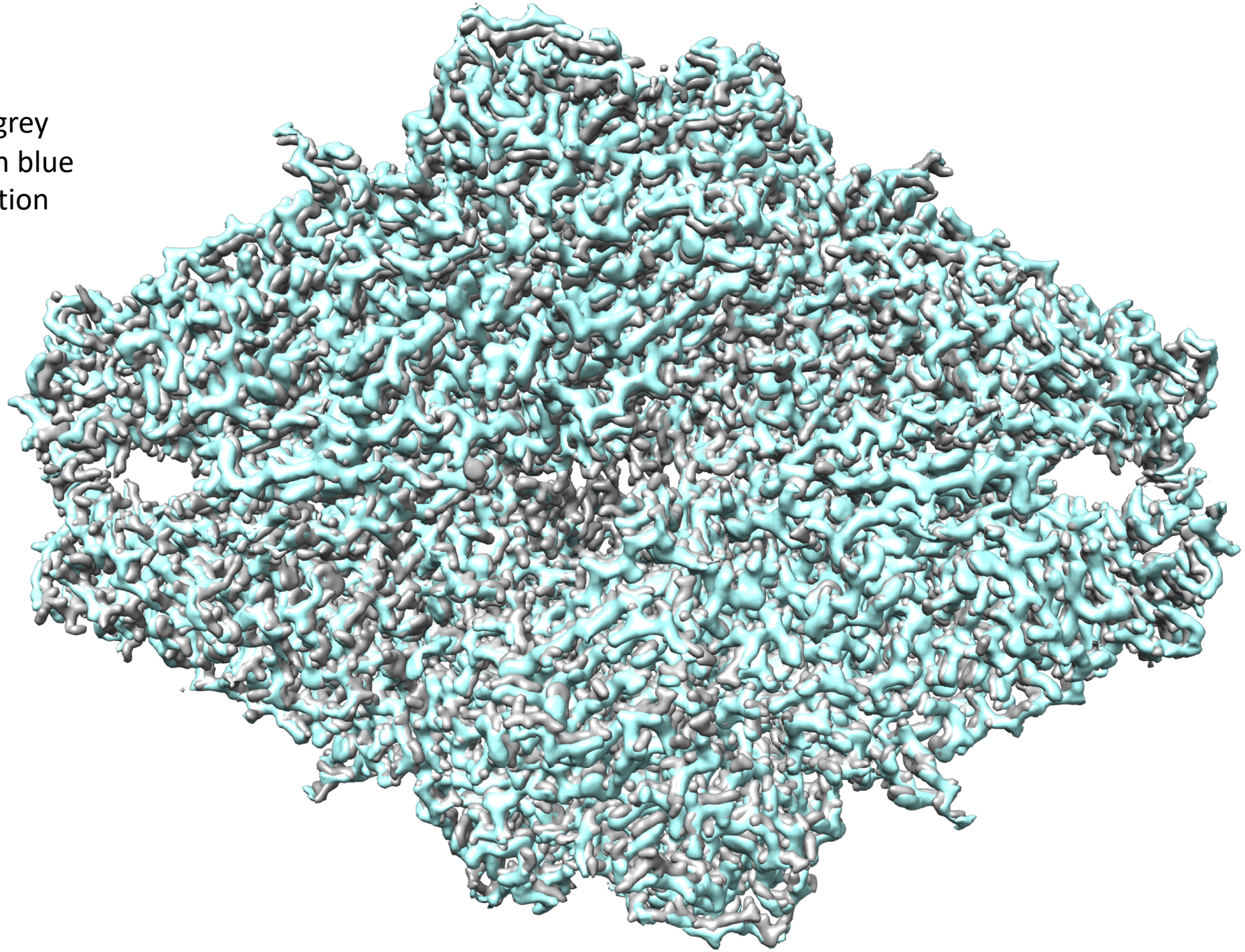
EMD-7770

Reference in grey
Emd_7770 in yellow
After correction



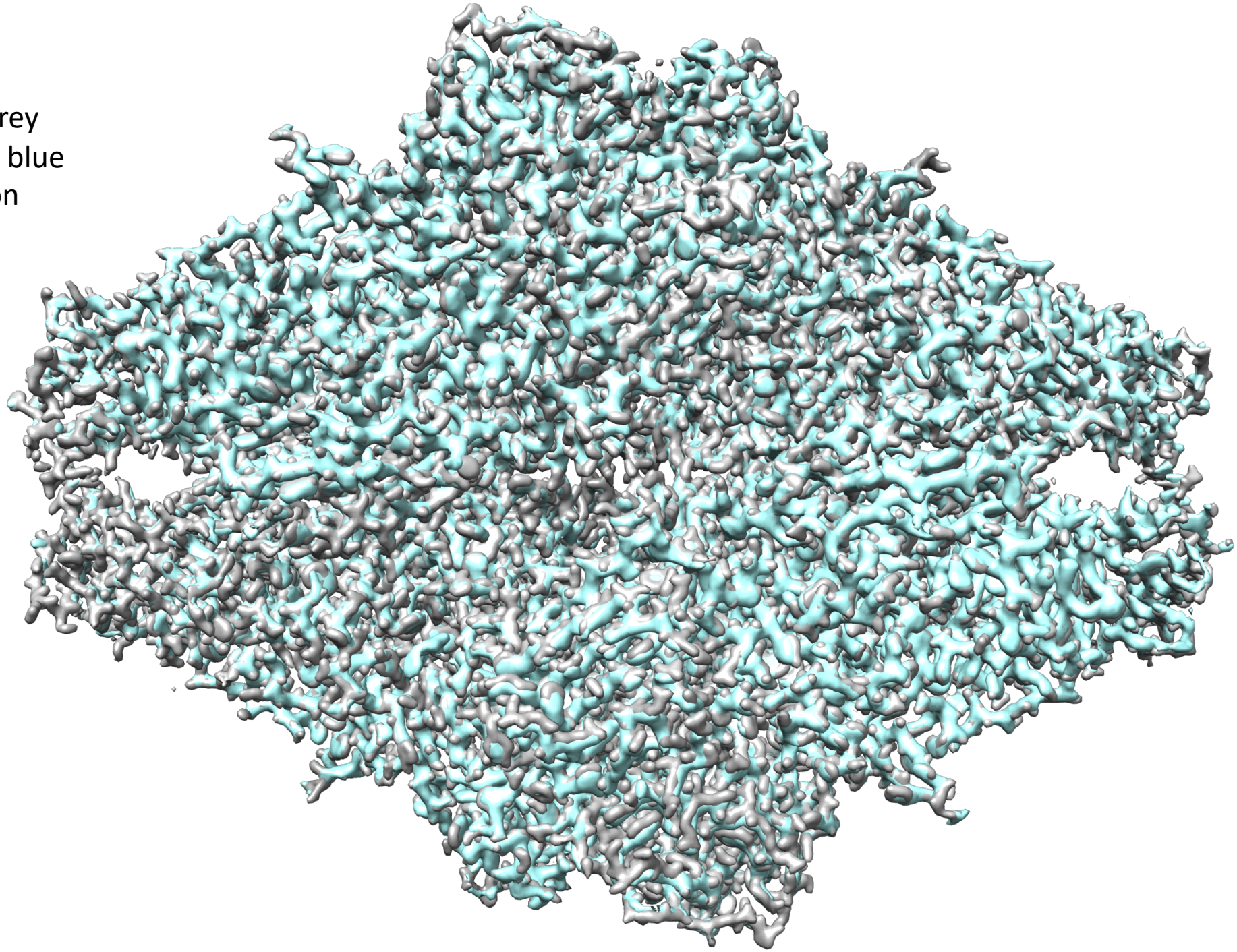
EMD-10574

Reference in grey
Emd_10574 in blue
Before correction

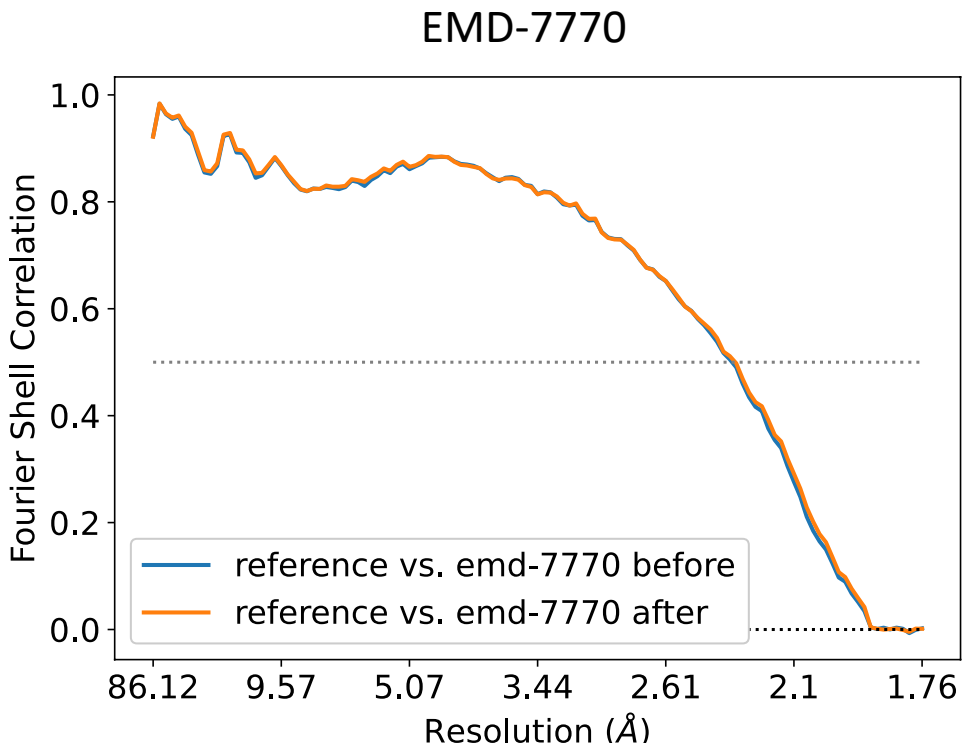


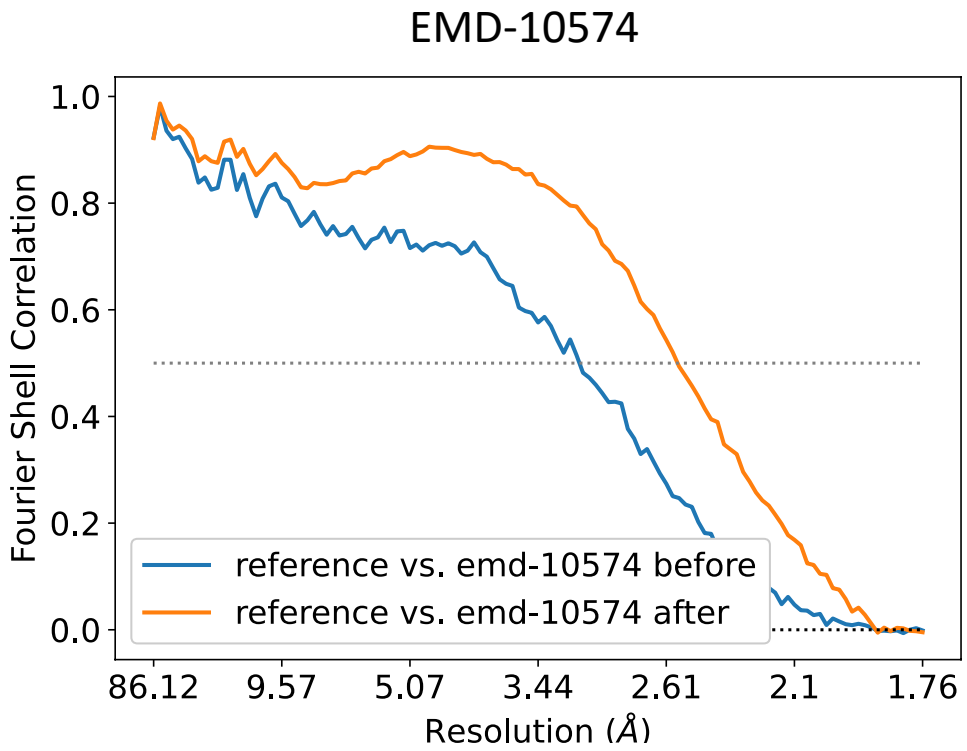
EMD-10574

Reference in grey
Emd_10574 in blue
After correction



FSC between reference and those maps before and after correction





Helper code to calculate FSCs is given at script https://gitlab.com/ccpem/emda/-/blob/master/tests/fsc_nmaps.py

Run as:

```
$ python fsc_nmaps2.py reference.mrc \
        emd_7770_fullmap.mrc \
        emd_7770_on_xtallography-reference_emda_magcorretedmap_1.mrc
```

Fit of monomers in beta-galactosidase

- Goal:
- To estimate the movement of one monomer of EMD-7770 relative to the corresponding monomer in the crystallography reference.
- This estimation is done using magnification adjusted EMD-7770 map relative to the reference map.

Step 1.
Extract monomer units in model generated masks.
To generate masks, select chain B from model 6cvm and save coordinates e.g. 6cvm_chain1.pdb

Similarly, extract the corresponding chain from the reference_model.pdb (aligned 3dyp).
Then save that coordinates e.q. ref-chain1.pdb

To generate the masks from those selected chains,

$emda modelmask --mdl 6cvm_chain1.pdb --map reference_map.mrc
Save emda_atomic_mask.mrc → emda_atomic_mask_emd-7770-chain1.mrc

$emda modelmask --mdl reference_model.pdb --map reference_map.mrc
Save emda_atomic_mask.mrc → emda_atomic_mask_ref-chain1.mrc

Step 2
Overlay of monomers

```
$emda overlay --map reference_map.mrc emd_7770_emda_magcorretedmap_1.mrc \
              --msk emda_atomic_mask_ref-chain1.mrc emda_atomic_mask_emd-7770-chain1.mrc
```