Model building and validation for cryoEM

Oliver Clarke

(@OliBClarke)

COLUMBIA UNIVERSITY
MEDICAL CENTER

“Is my map buildable??”

An atomic model is a compact interpretation of the density map in light of prior knowledge (both specific and general).

- Aim is to build a model that is consistent with **both** the density map and everything we independently know about the structure & composition of the macromolecule of interest, both specifically and in terms of our general knowledge of protein structure and chemistry.

- At medium resolution (3-5 Å), this still requires manual building (yes, even if you start from an AlphaFold prediction…🙃). Even the best autobuilt model still requires manual inspection and correction in most cases. (generates many fragments which need inspection, correction, merging)

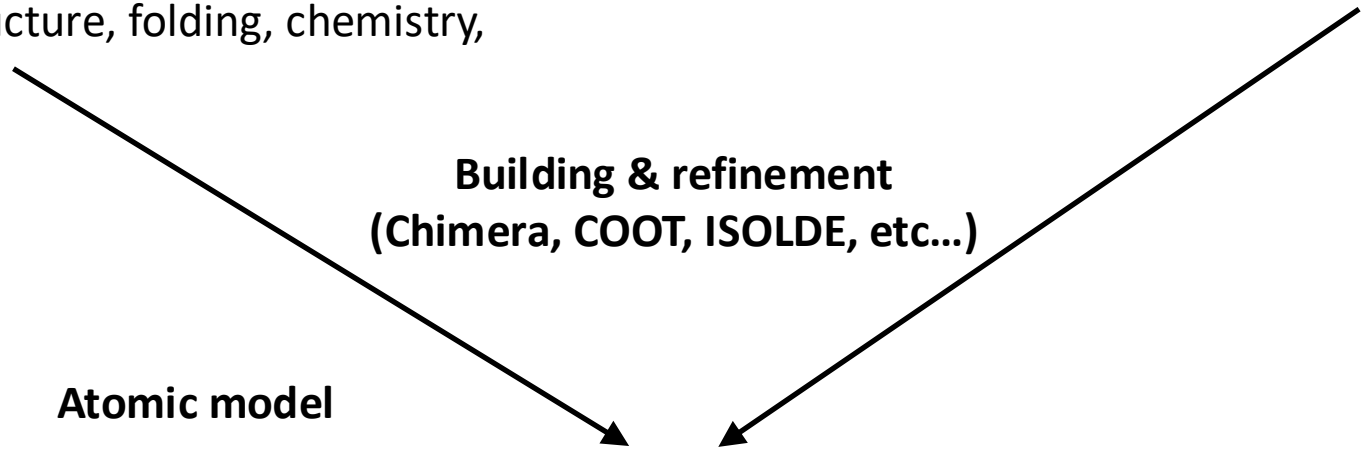
- Tradeoff between available prior knowledge and required resolution for atomic modelling – at the extremes, if a complete crystal structure is already available, 10Å data may be sufficient, while if no sequence/composition data is available even 3Å may not suffice.

Prior knowledge

- Protein sequence and derived info (secondary structure predictions, covariation/conservation, patterns of large/aromatic residues), disorder & contact prediction
- Crystal structures (+ homology & **ML-derived models – AlphaFold, Modelangelo**)
- Knowledge of protein structure, folding, chemistry, geometry.

Density map

- Resolution (+ local resolution, + map modification/sharpening, anisotropy)
- Patterns of large/small/absent sidechains
- Sharpening and density modification
- Conformational/compositional heterogeneity

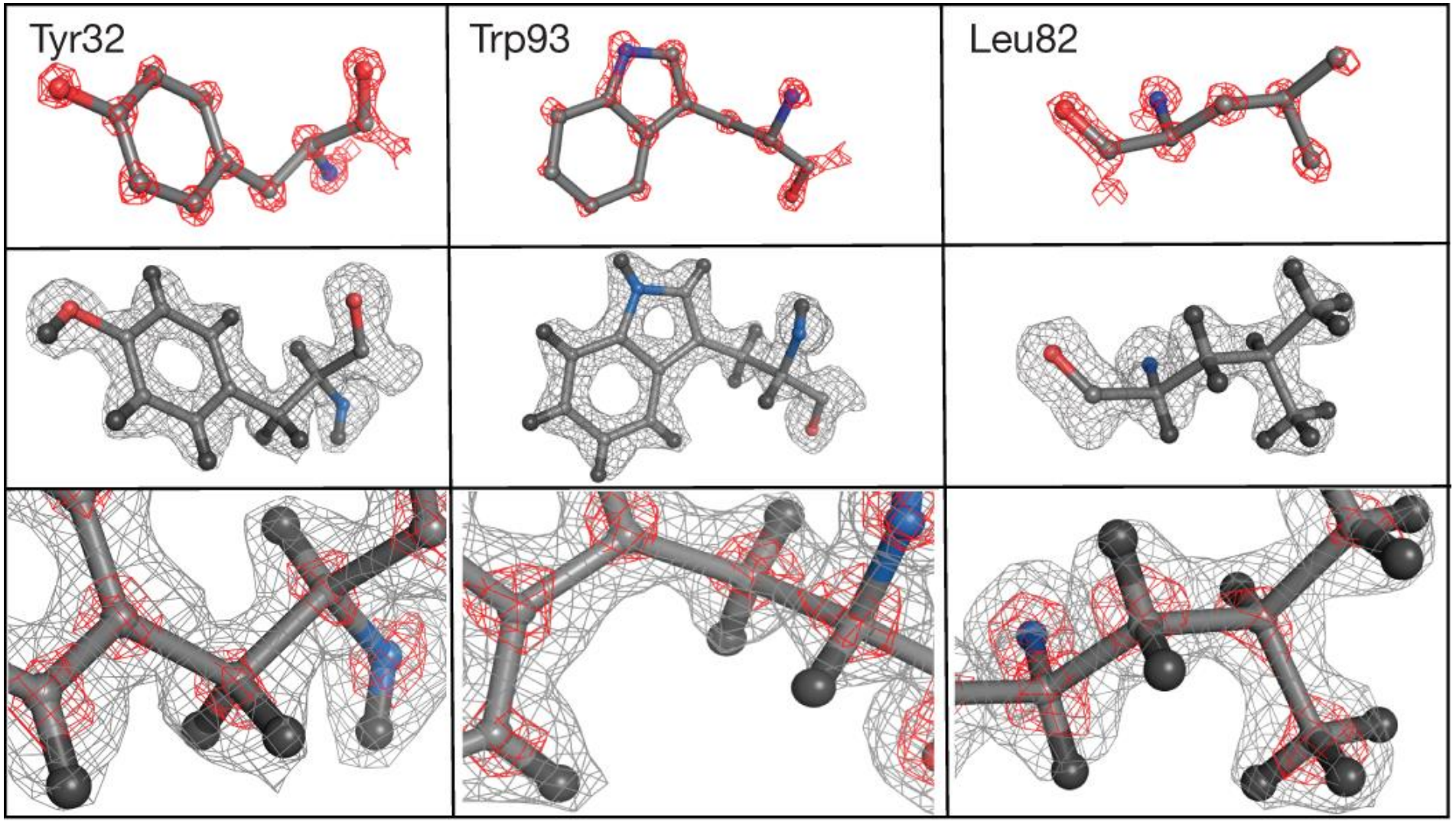


Building & refinement
(Chimera, COOT, ISOLDE, etc…)

Atomic model

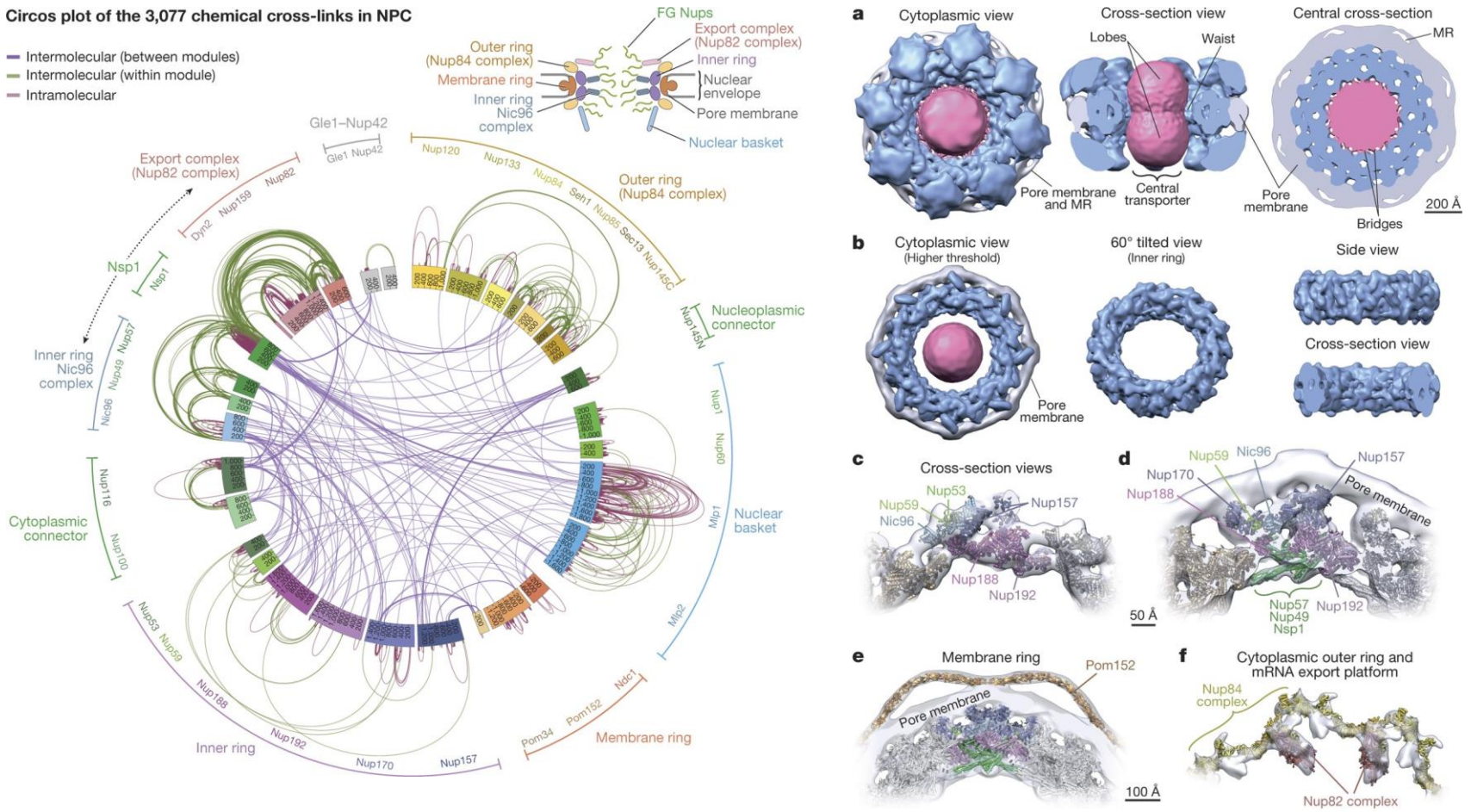
- If possible, unique model (or ensemble) that agrees with both density map and priors
- Otherwise (and per region), specify ambiguity (w/UNK residues and numbering or Ca only model)
- Validation not just (or even mostly) about overfitting.
- Identify, analyse, fix errors.
- Direction and register of sequence fit.
- Ligand identification/assignment.
- No model is, or ever will be perfect. That’s okay.

One extreme – at atomic resolution, the position of many atoms can be inferred without prior knowledge of the sequence



Yip, K.M., Fischer, N., Paknia, E. et al. Atomic-resolution protein structure determination by cryo-EM. Nature ***587,*** *157–161 (2020)*

At 20 Å (here using cryoET), an informative model can be generated by taking advantage of external information – crystal structures, connectivity from crosslinking & MS, even when de novo building is not possible.



Kim, S., Fernandez-Martinez, J., Nudelman, I. et al. Integrative structure and functional anatomy of a nuclear pore complex. Nature ***555,*** *475–482 (2018)*

Usually, we are somewhere in between the two – combining prior knowledge with inferences made from analyzing the density map.

To build a better/more reliable model, we can either get additional/better priors, or improve our density map (or part of it).

What do we mean by validation?

- Validation=Verification! Verifying what?
  - Are these particles my protein? (Picking & 2D classification)
  - Is my 3D reconstruction correct? (Overall shape, Symmetry, Details)
  - What species are present in my reconstruction? (Composition)
  - How detailed is my reconstruction? (Estimating resolution)
  - How does detail vary with direction and location? (Anisotropy & local resolution)
  - Is the reconstruction the best representation of the underlying data? (Sharpening & postprocessing)

General principles to keep in mind…

- Be skeptical of your own data – always consider the null hypothesis
  - E.g. particles ***not*** protein, not ***my*** protein, ligand ***not*** bound, etc…
- Test, test, & test again…
  - E.g. do different reconstruction/refinement algorithms give same result?
  - Confirm presence of ligands/subunits using difference maps, labels, mass spec..
- Trust your eyes!
  - If 3Å reconstruction has no sidechains, retrace your steps – postprocessing? Masking? Initial model?
  - Remember that many metrics (e.g. FSC) measure **self-consistency**, not accuracy/reality. No substitute (yet!) for human judgement and careful attention to detail.

Before you start – make sure your maps are appropriately sharpened and low pass filtered! (and consider whether building is justified or whether further improvement of the reconstruction is required first)

- **Often it is helpful to build using multiple maps**. Assuming 3-3.5Å global res, I would suggest using a map filtered to the global resolution, one filtered to the best local resolution, and one filtered to ~4-4.5 Å (to better visualize connectivity and mobile ligands/lipids).

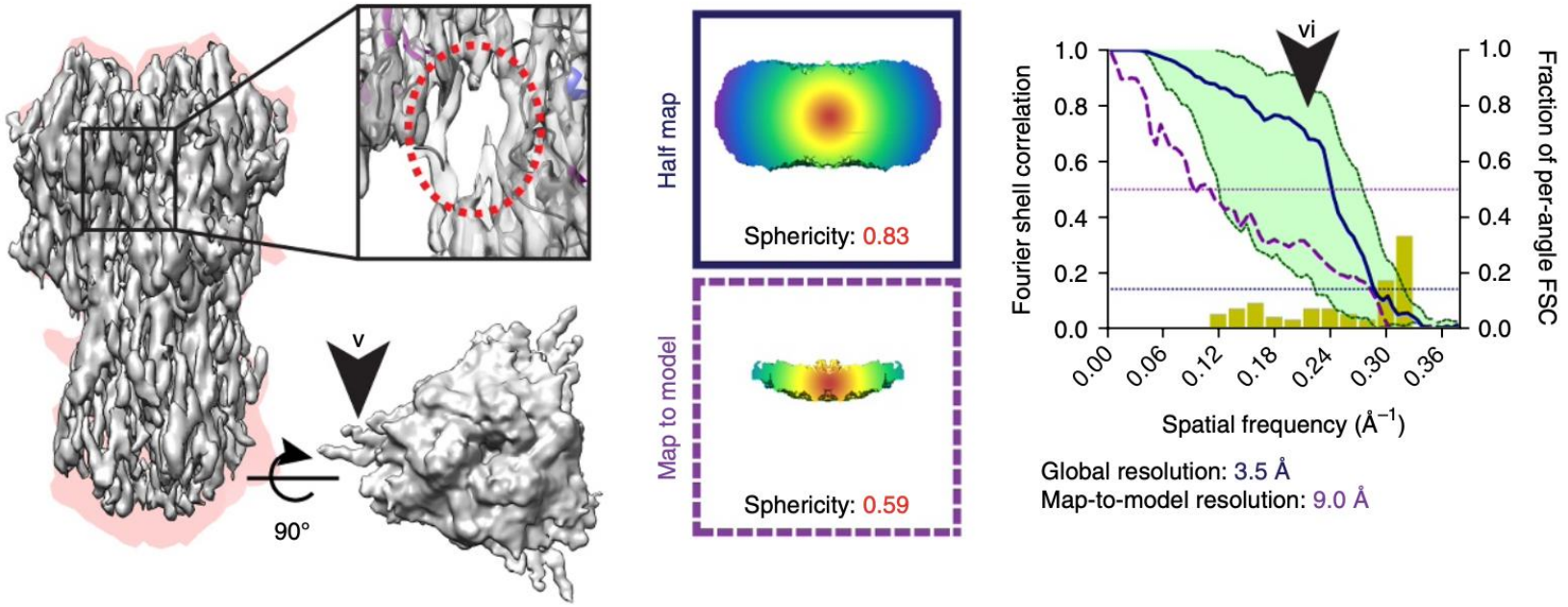
- Try both simple B-factor sharpening and the approach used by `phenix.resolve_cryo_em`, which incorporates anisotropy removal and statistical density modification. In cases of **severe** anisotropy, deepEMhancer can be useful to assist map interpretation (**approach with caution**).

- Also, if your map doesn't "look like" 4 Å, trust your eyes! If it is nominally 4Å and there are no sidechains visible, or your helices look "stretched", assess orientation bias (3D-FSC server: *https://3dfsc.salk.edu*), local resolution variation, and double check sharpening and masking parameters (are you *sure* you're looking at the sharpened map? Is the mask used for FSC calculation sensible?)

Why does resolution vary (directionally)?

- Preferred orientation:
  - Instead of adopting random orientations in the ice, particles adopt a limited subset (often due to association with the air-water interface).
  - Can address using additives, tilting, or (sometimes) classification, discarding preferred views.
  - Often additionally associated with particle dissociation/damage, so biochemical fixes favored.
- Conformational heterogeneity:
  - Hinge-like motions, with flexibility along a single axis, can cause map anisotropy
  - Can be corrected, in favorable cases, by local refinement (sometimes combined with classification).

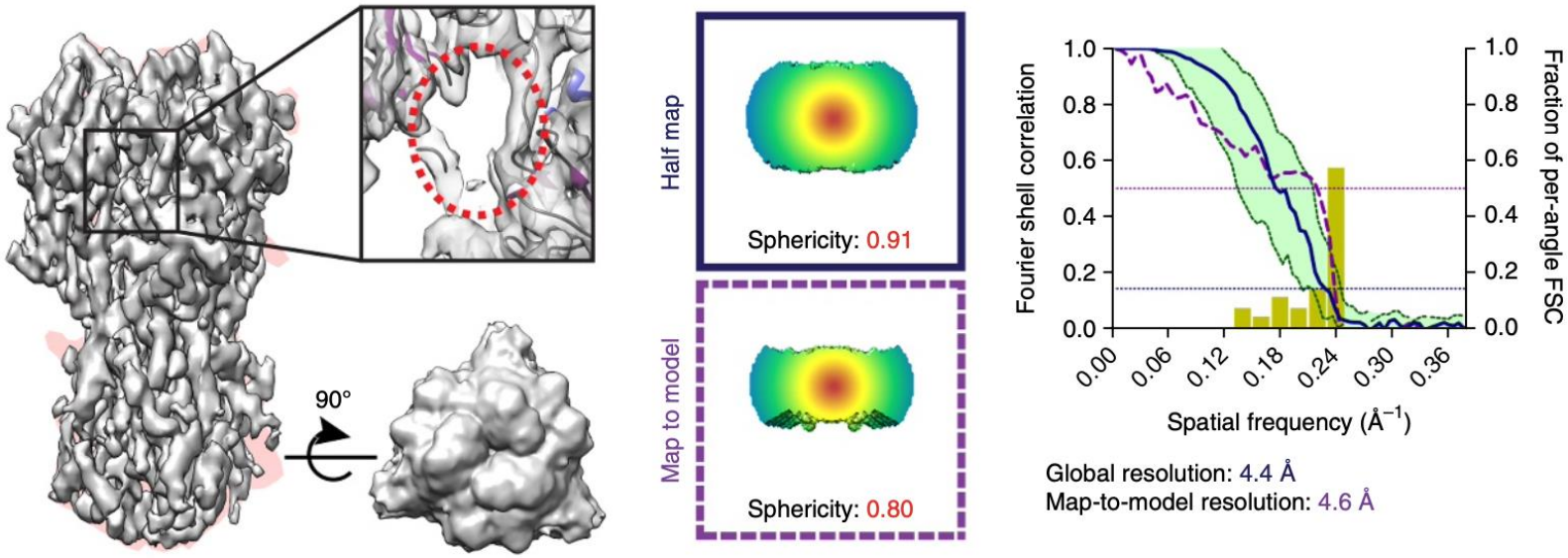
Preferred orientation



- Lack of side views results in distortion, stretching of map. Hinders interpretation
- Can diagnose using 3D-FSC (calculates equally spaced conical FSCs).
- Reduced sphericity, large spread in directional resolution can originate from preferred orientation
- Correct with additives (CHAPS, FOM) or collecting with a tilt.

Tan YZ et al., Nat Methods, 2017

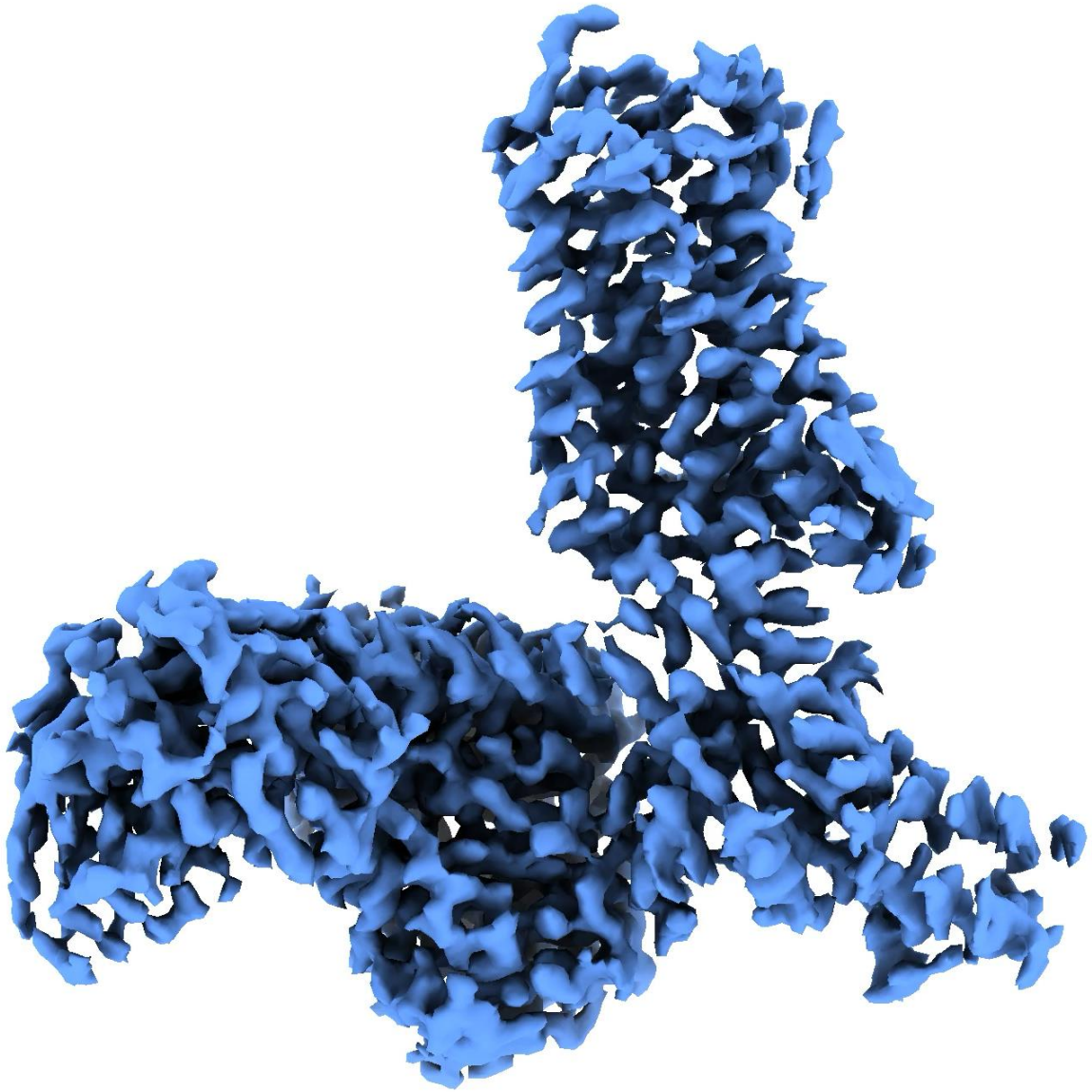
Preferred orientation



- Lack of side views results in distortion, stretching of map. Hinders interpretation
- Can diagnose using 3D-FSC (calculates equally spaced conical FSCs).
- Reduced sphericity, large spread in directional resolution can originate from preferred orientation
- Correct with additives (CHAPS, FOM) or **collecting with a tilt.**

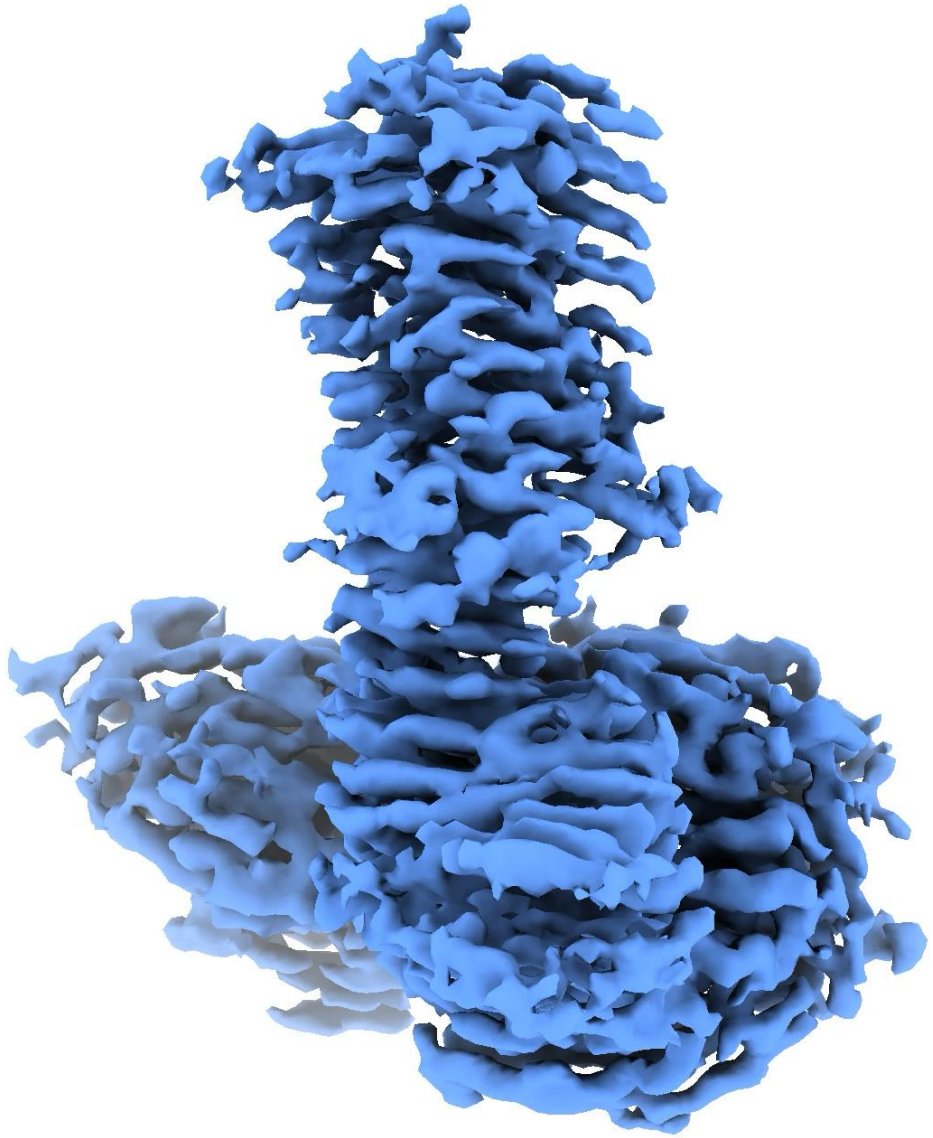
Tan YZ et al., Nat Methods, 2017

Example from the literature



Good direction – looks okay

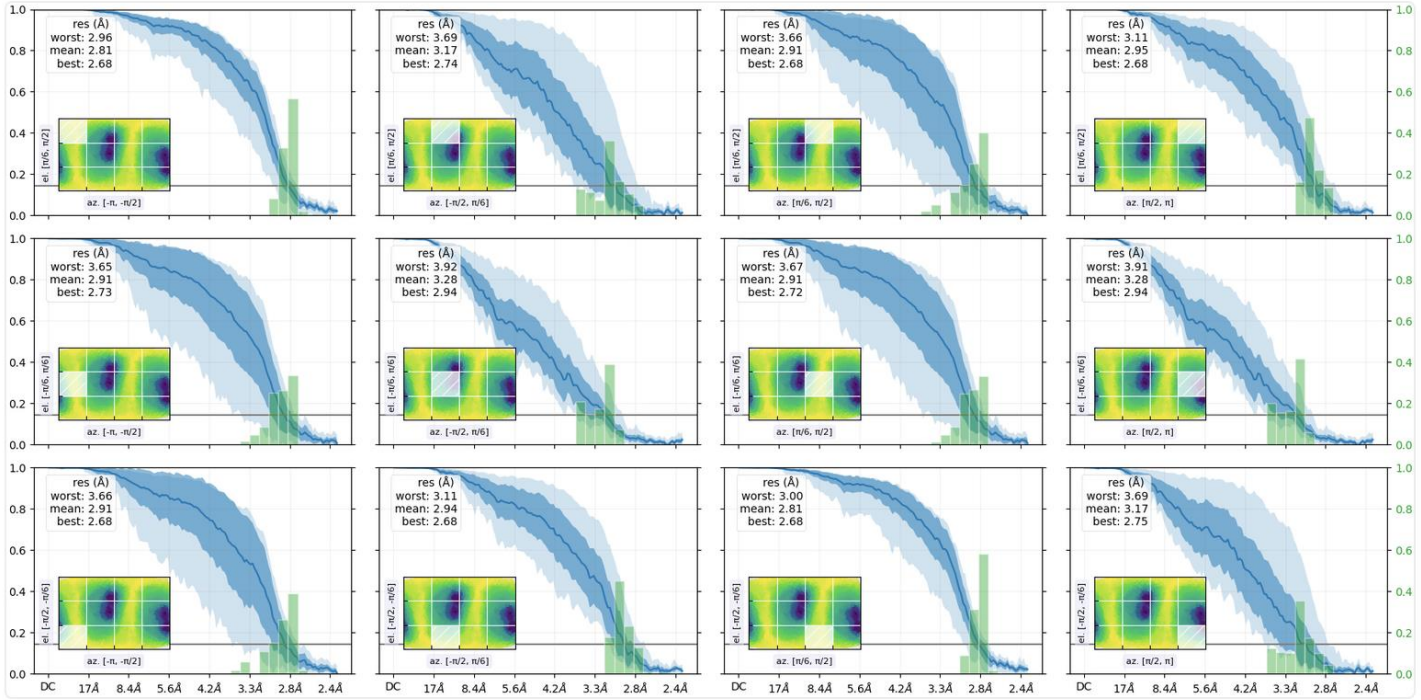
Example from the literature



Bad direction – ”ropy”, stretched, hard to interpret

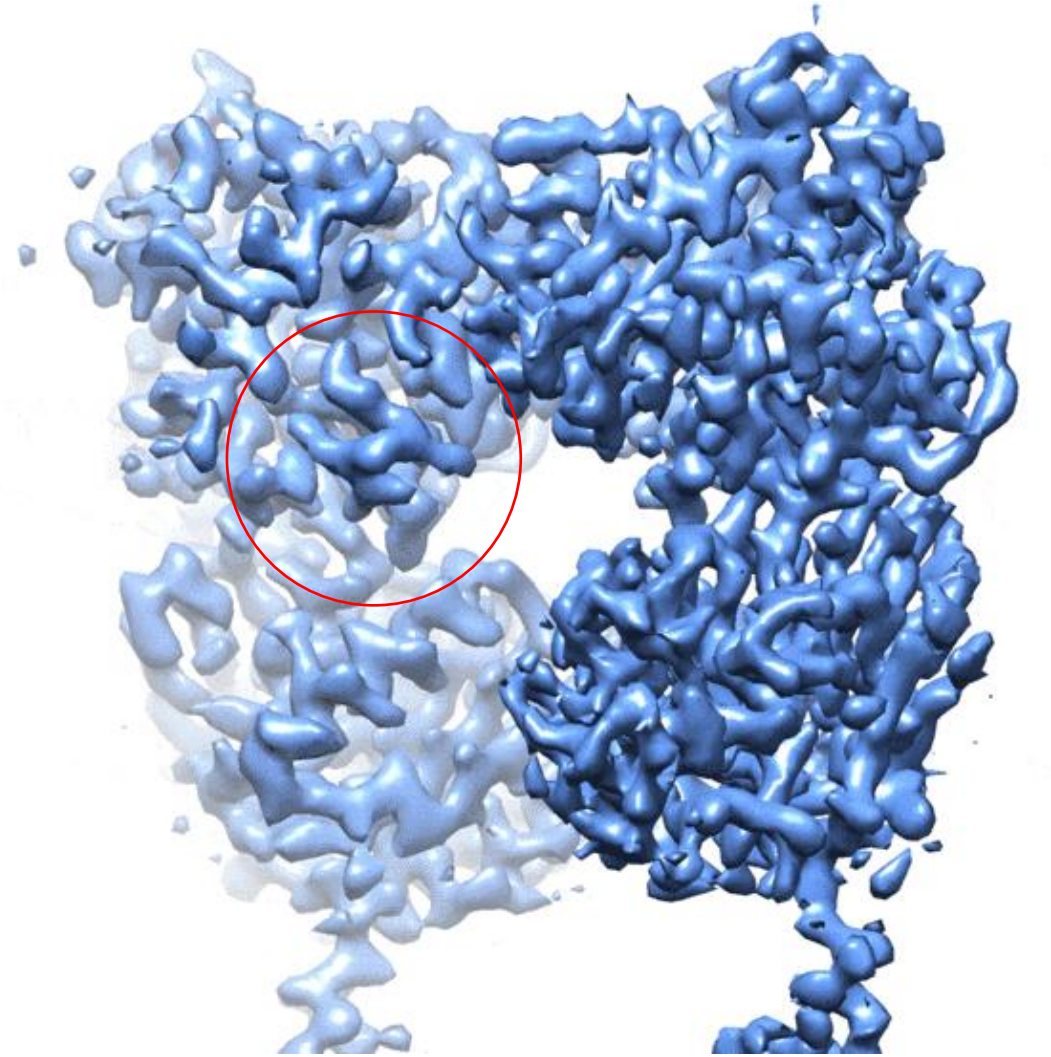
If using CryoSPARC – please use Orientation Diagnostics in v4.4+

- Provides several metrics for anisotropy in an easy to digest format, including 3D-FSC
- Conical FSC area ratio particularly useful for assessing anisotropy in my experience (doesn't rely on noisy FSCs dropping below arbitrary threshold)
- Nice graphical summary of conical FSCs gives easy to understand overview of directional resolution variation.

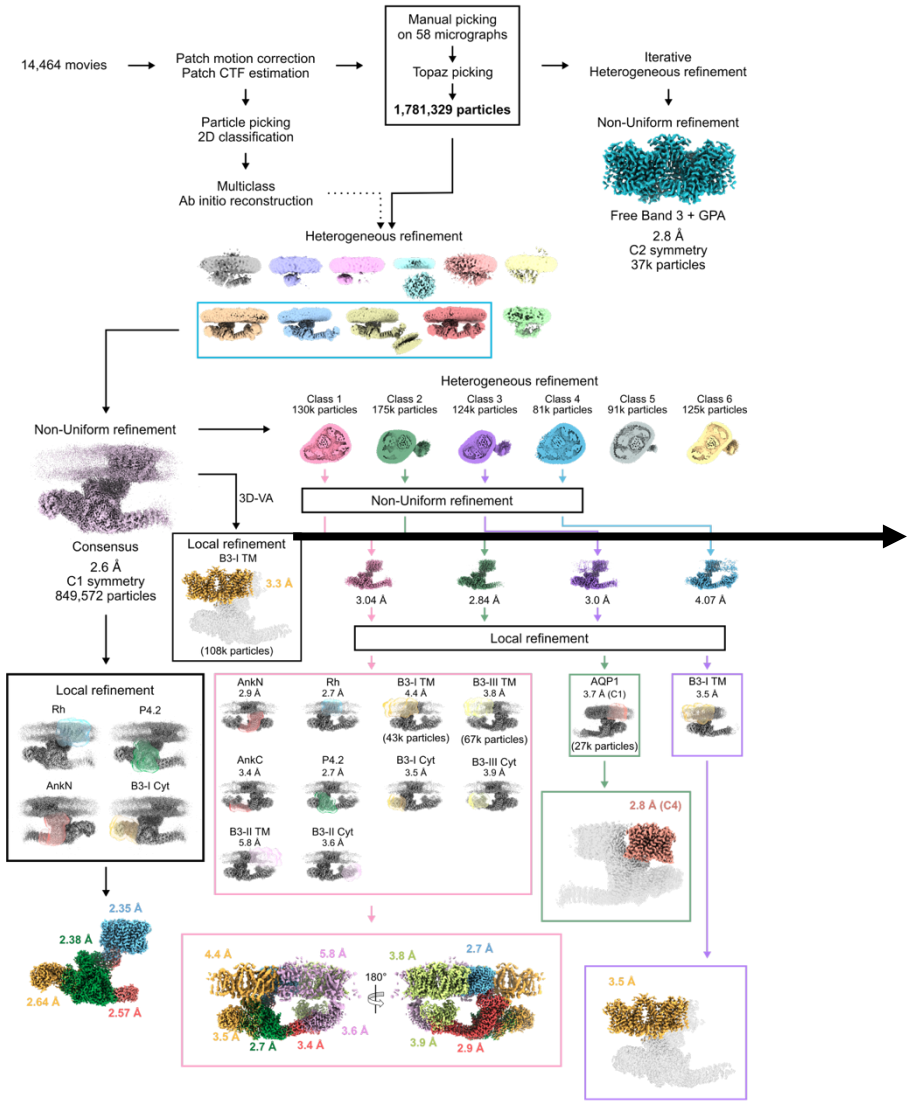


Example of map anisotropy mitigated by masked refinement

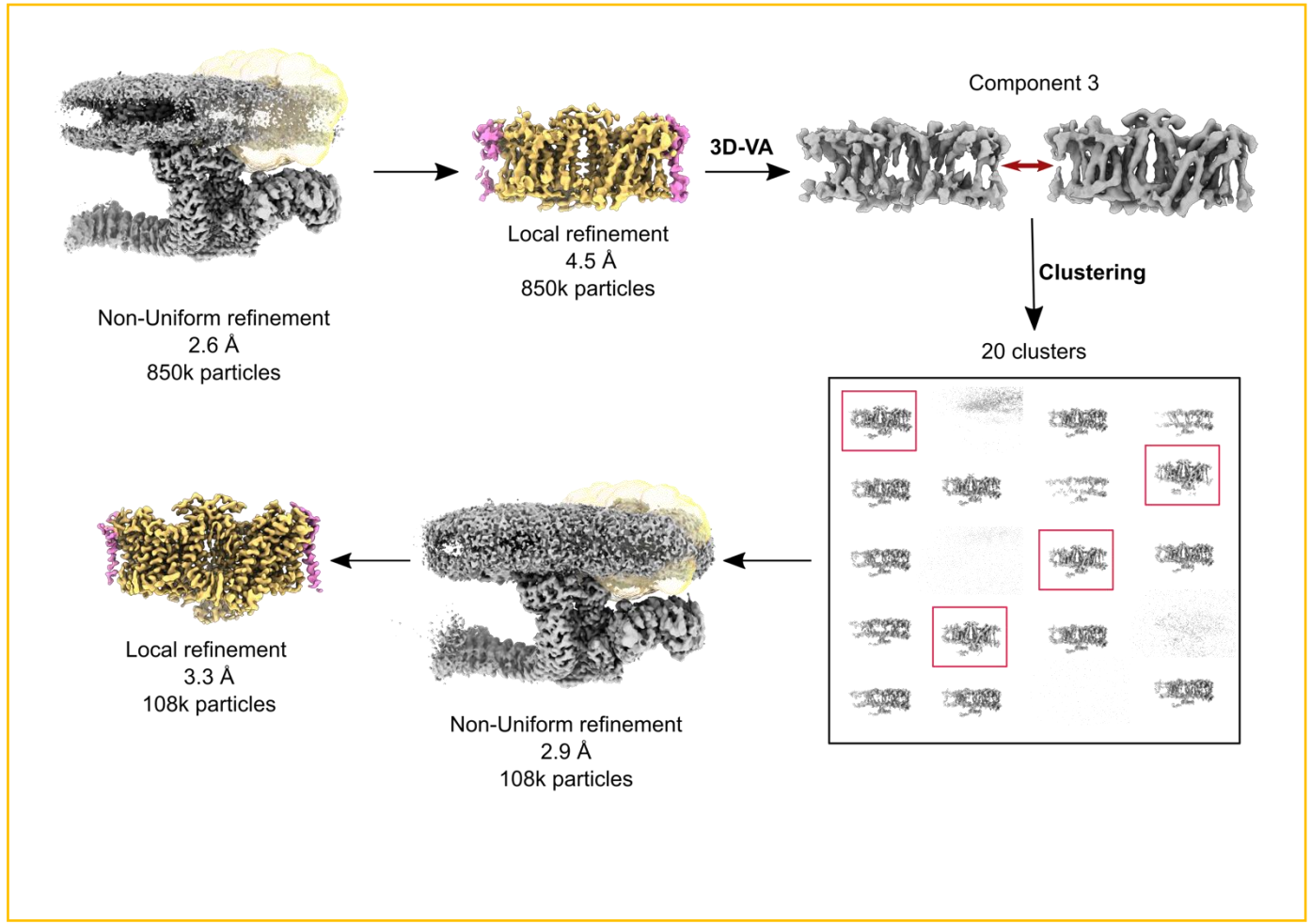
- Map anisotropy hinders interpretation, even when resolution in “good” direction is high
- Can derive from either preferred orientation, or interdomain mobility (or combination).
- In latter case, masked refinement can improve local map quality to aid model building and map interpretation. **Always better to improve the map than build in marginal density**
- If anisotropy derives from preferred orientation, it is best to address this by improving the sample (additives, constructs, substrates) or altering data collection strategy (tilt). If all else fails, ML-based map improvement using deepEMhancer can improve map interpretability.



Map anisotropy can be mitigated by classification

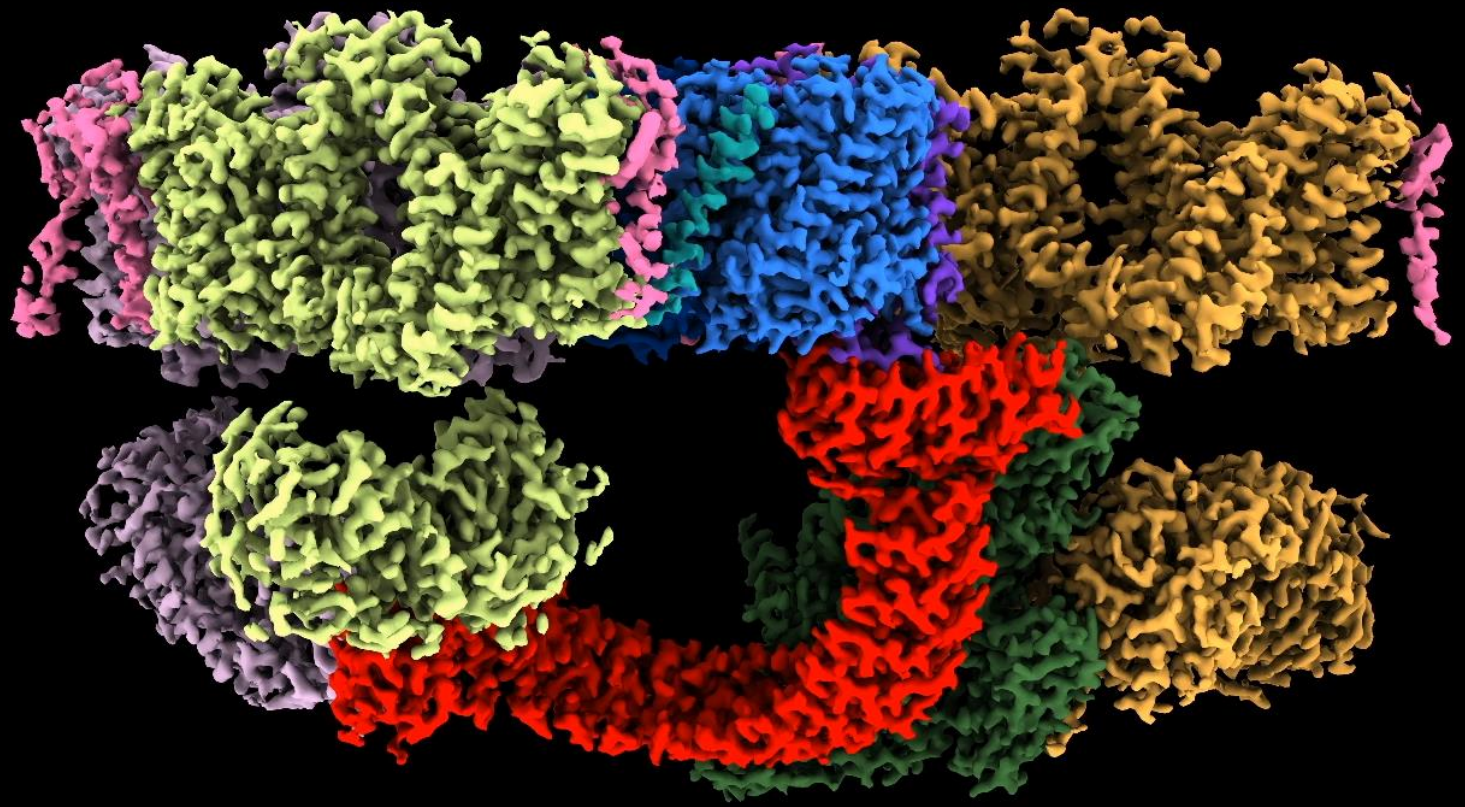


3D-VA clustering for resolution improvement



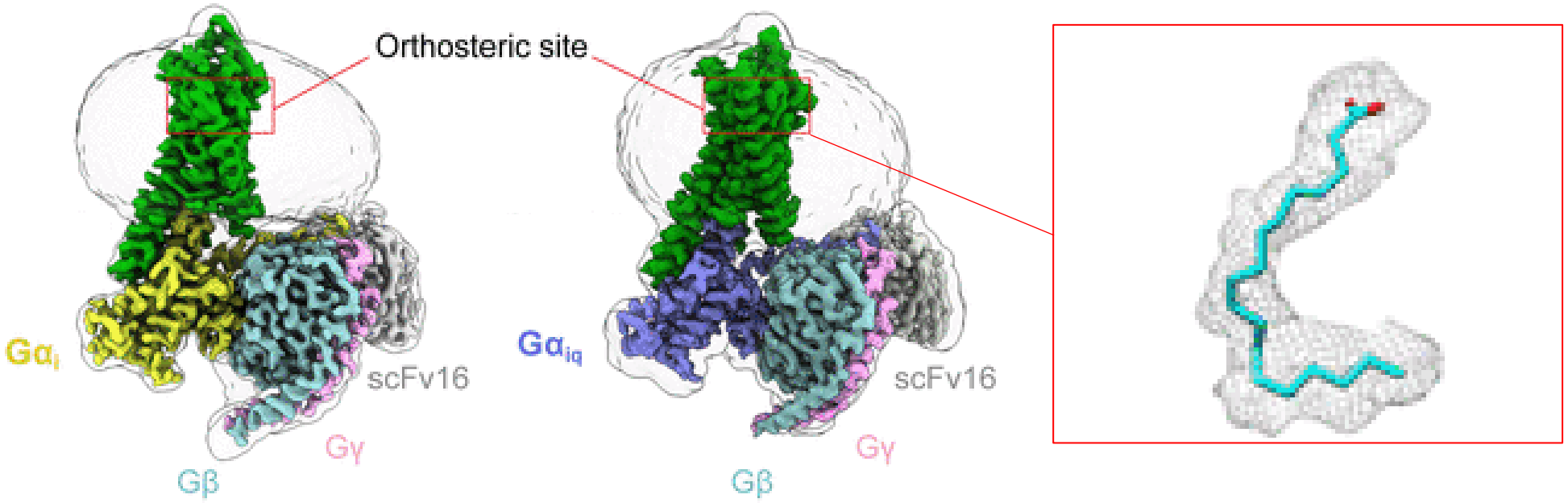
Vallese et al., NSMB 2022

Composite map example

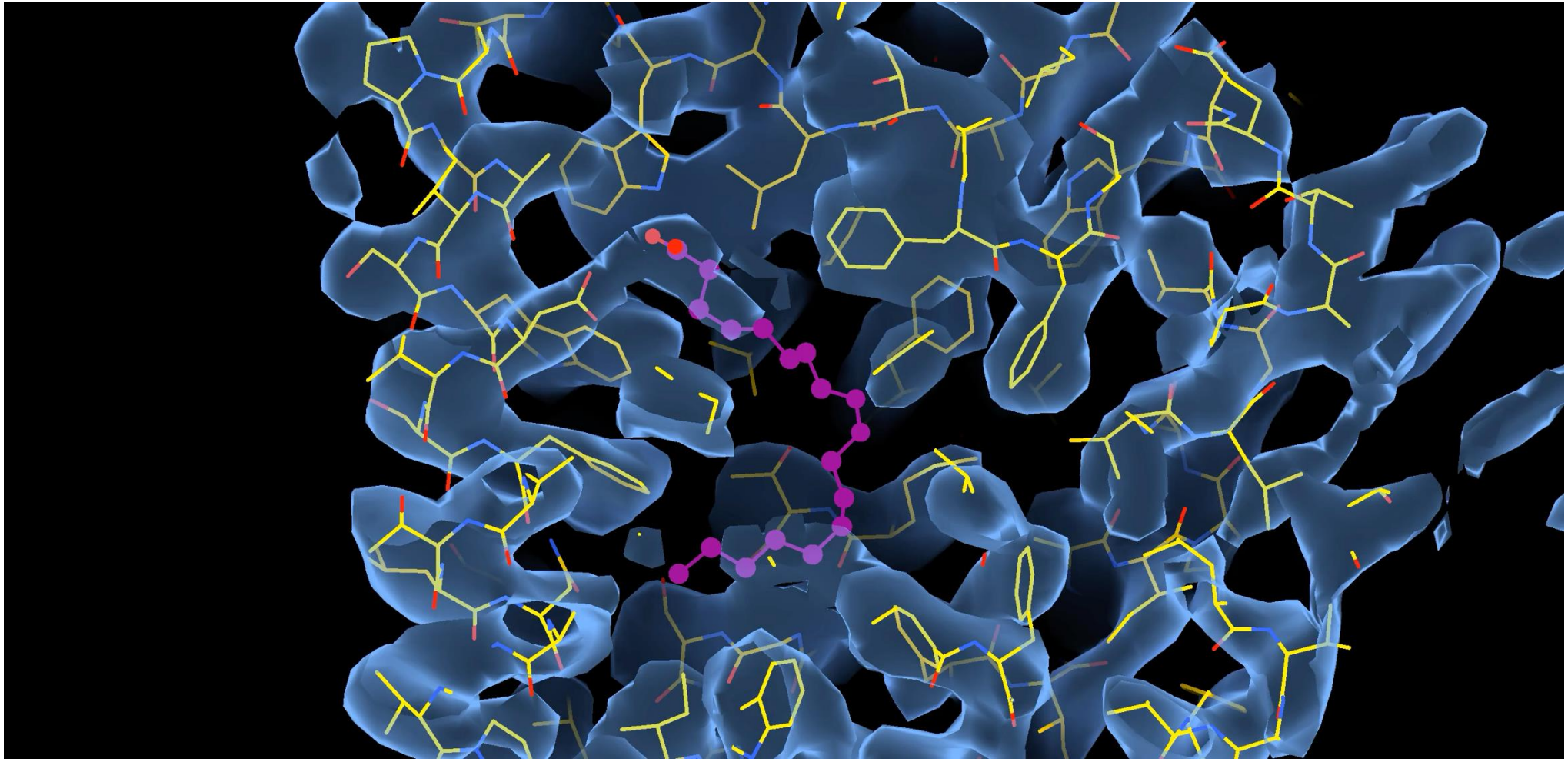


Vallese et al., NSMB 2022

Cautionary note – don't believe everything you read in the (Nature/Science/Cell) paper…



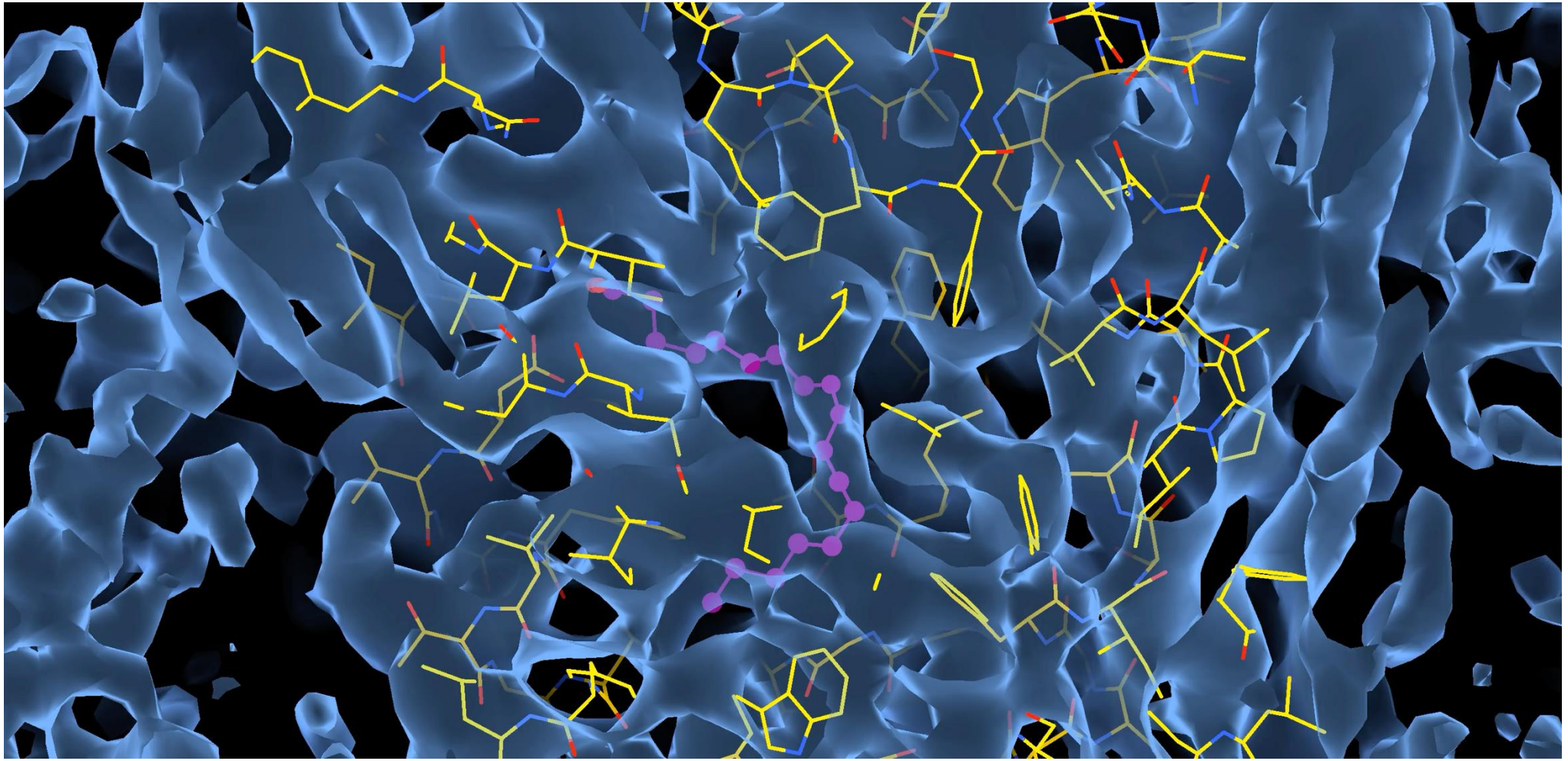
Cautionary note – don't believe everything you read in the (Nature/Science/Cell) paper…



Ligand invisible at any reasonable contour; protein density anisotropic and broken.

So how was nice density figure for ligand made? By carving the map around the ligand. Please don't do this!

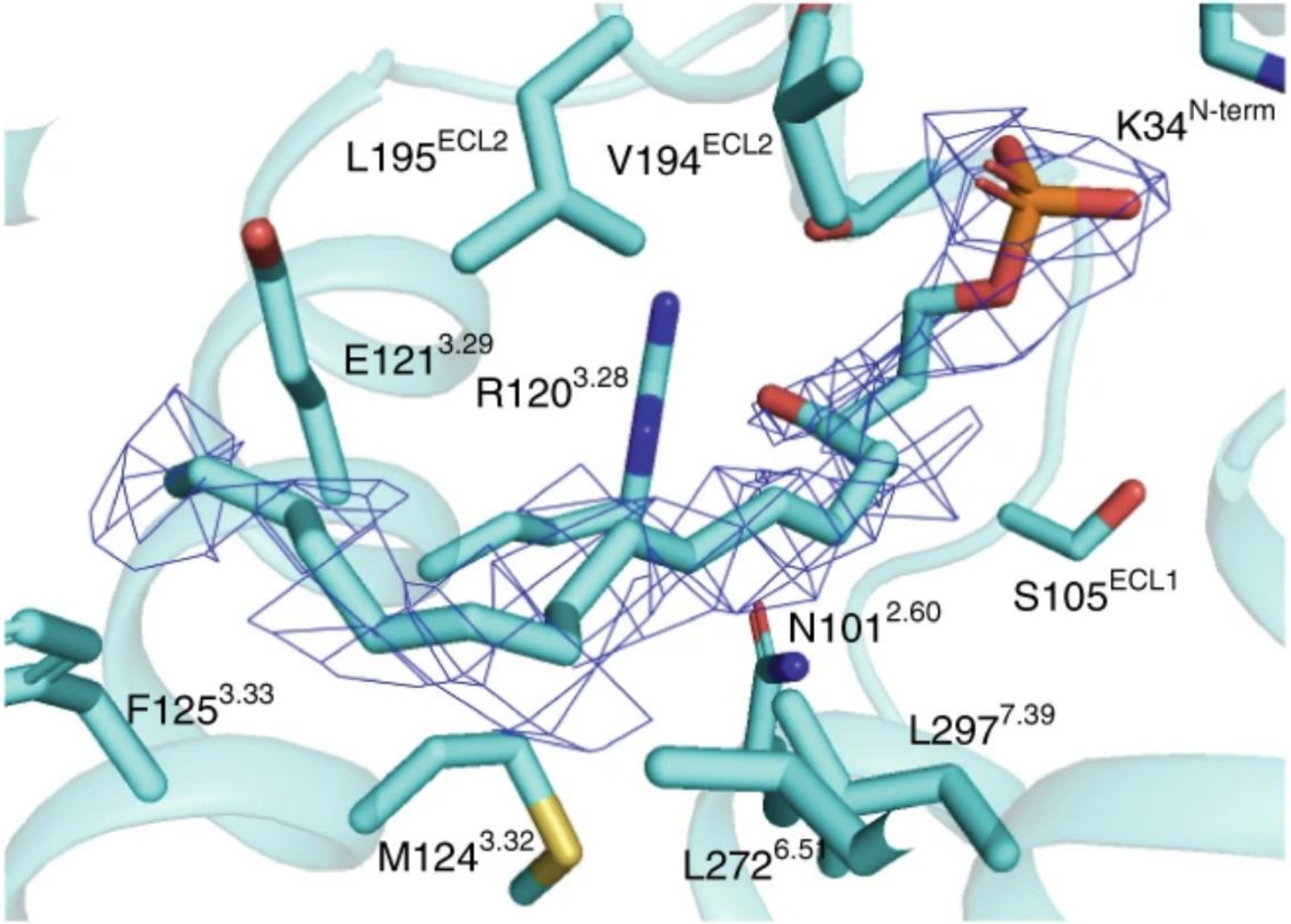
Cautionary note – don't believe everything you read in the (Nature/Science/Cell) paper…



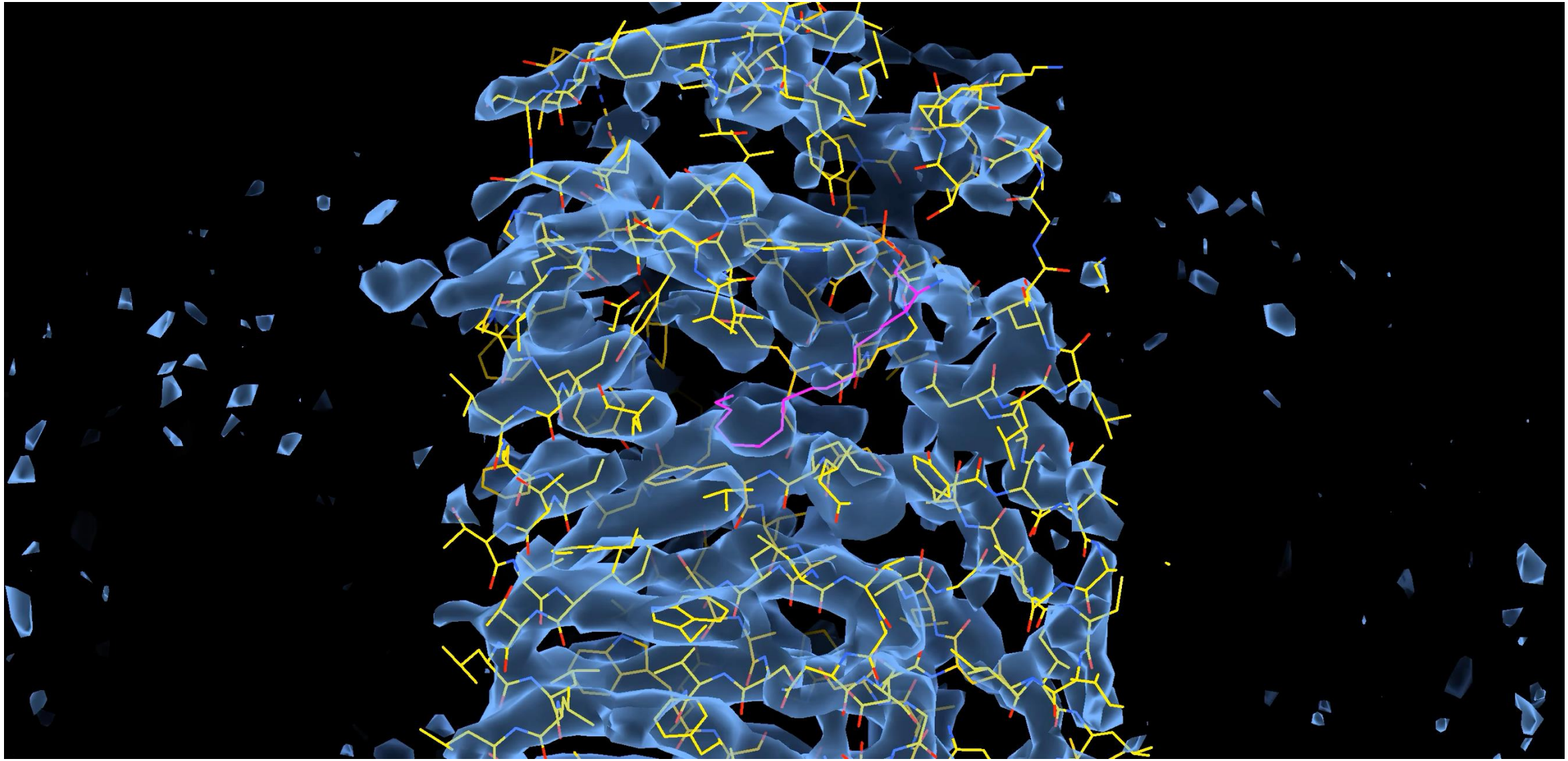
Ligand invisible at any reasonable contour; protein density anisotropic and broken.

So how was nice density figure for ligand made? By carving the map around the ligand. Please don't do this!

Cautionary note – don’t believe everything you read in the (Nature/Science/Cell) paper…



Cautionary note – don’t believe everything you read in the (Nature/Science/Cell) paper…



Prep for model building - what can we learn from the sequence alone?

Your protein sequence contains a lot of useful information which you can use to aid model building:

- Start by identifying boundaries of conserved domains (NCBI CDD: https://www.ncbi.nlm.nih.gov/Structure/cdd/; DELTA-BLAST also performs CD-search by default)

- Then identify and/or generate suitable structural templates for building known domains: FUGUE, PHYRE 2, MUSTER. (Alphafold/ROSETTAfold dominate now!). Modelangelo useful for generating initial model, especially if sequence unknown.

- Secondary structure, TM & disorder prediction (XtalPRED for overall summary; specific tools such as SPOT-DISORDER, SPIDER3 for best accuracy).

- Contact prediction from evolutionary couplings: EVFOLD & GREMLIN.

- Conservation analysis: Use favorite MSA algorithm (MUSCLE & CLUSTAL-OMEGA work well; TM-COFFEE, PRALINE-TM useful for membrane proteins) to create a sequence alignment of your protein with a few orthologs; gaps & insertions most commonly occur in loops/disordered regions. Useful as a guide during building.

Prep for model building - what can we learn from the sequence alone?

Your protein sequence contains a lot of useful information which you can use to aid model building:

- **Start by identifying boundaries of conserved domains (NCBI CDD: https://www.ncbi.nlm.nih.gov/Structure/cdd/; DELTA-BLAST also performs CD-search by default)**

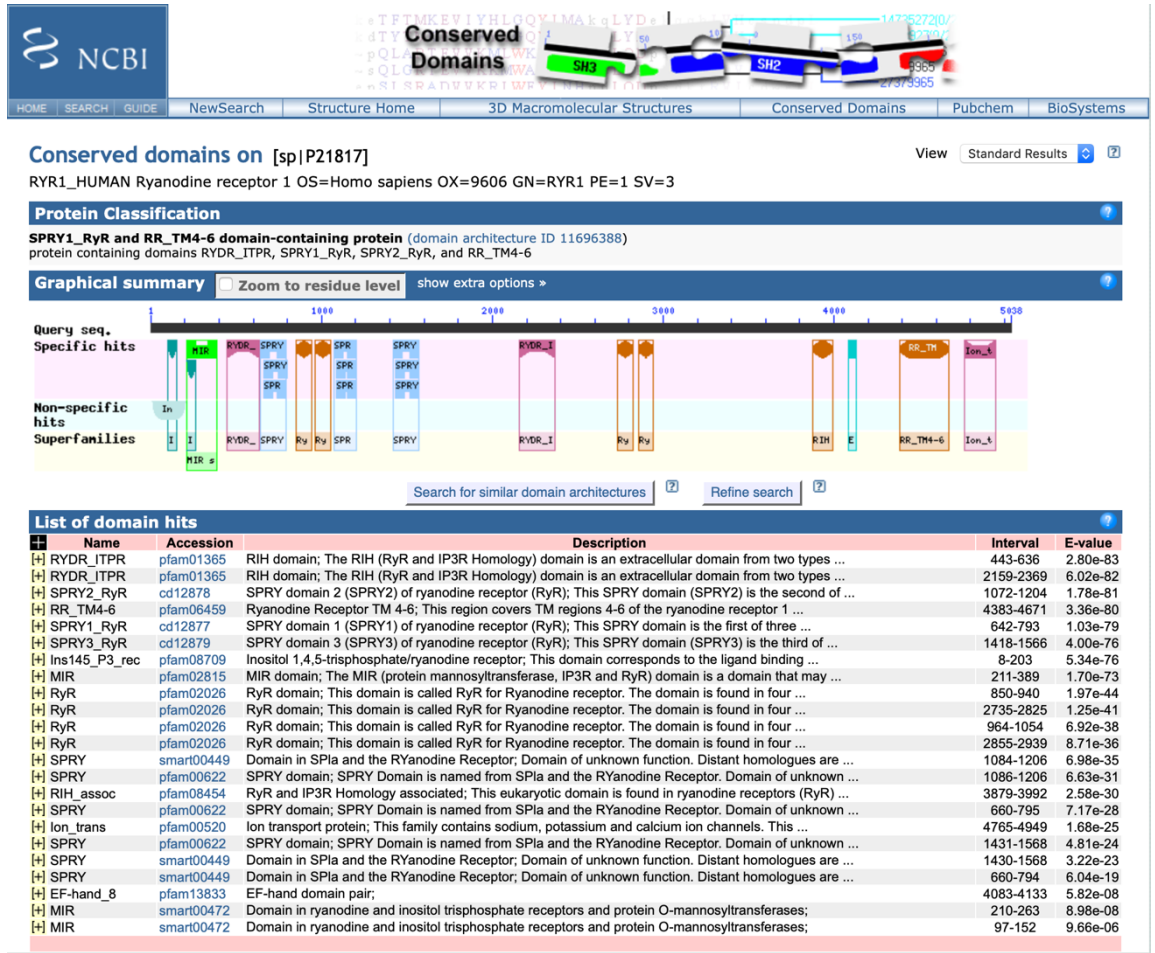
- Then identify and/or generate suitable structural templates for building known domains: FUGUE, PHYRE2, MUSTER. (Alphafold/ROSETTAfold dominate now!). Modelangelo useful for generating initial model, especially if sequence unknown.

- Secondary structure, TM & disorder prediction (XtalPRED for overall summary; specific tools such as SPOT-DISORDER, SPIDER3 for best accuracy).

- Contact prediction from evolutionary couplings: EVFOLD & GREMLIN.

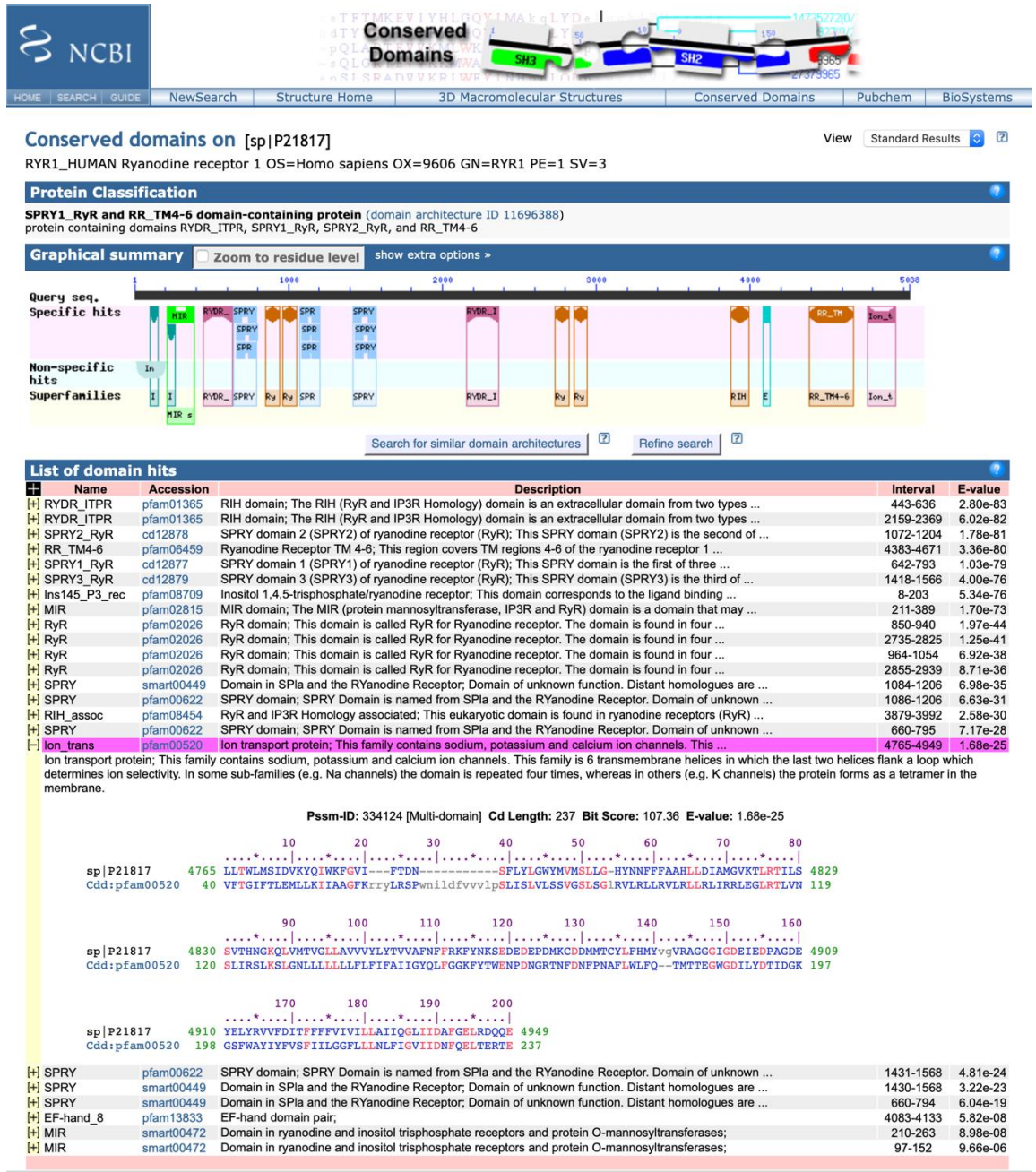
- Conservation analysis: Use favorite MSA algorithm (MUSCLE & CLUSTAL-OMEGA work well; TM-COFFEE, PRALINE-TM useful for membrane proteins) to create a sequence alignment of your protein with a few orthologs; gaps & insertions most commonly occur in loops/disordered regions. Useful as a guide during building.

CDD provides a guide to domain level architecture, including sequence alignments & representative structures.

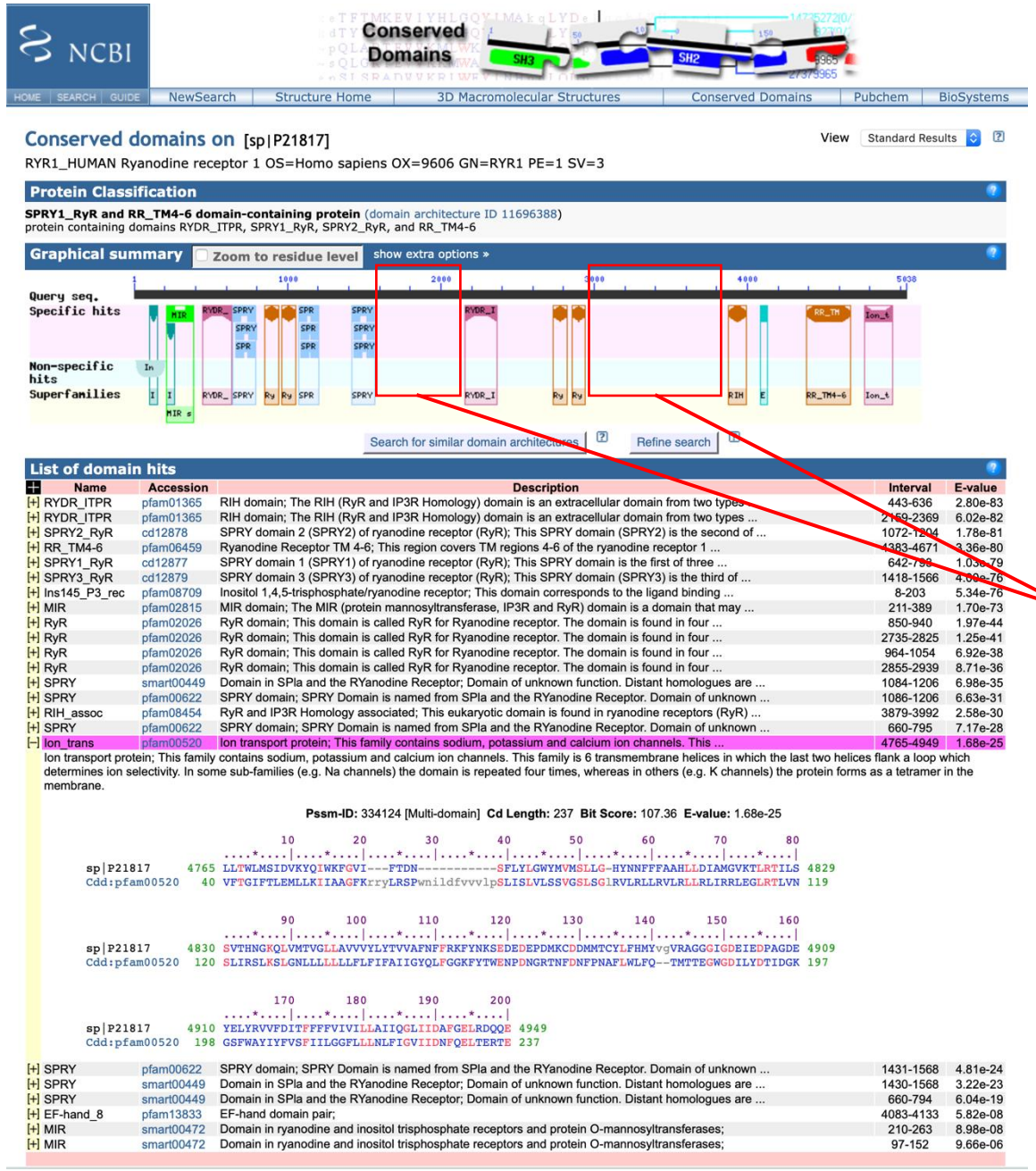


Name	Accession	Description	Interval	E-value
[+] RYDR_ITPR	pfam01365	RIH domain; The RIH (RyR and IP3R Homology) domain is an extracellular domain from two types ...	443-636	2.80e-83
[+] RYDR_ITPR	pfam01365	RIH domain; The RIH (RyR and IP3R Homology) domain is an extracellular domain from two types ...	2159-2369	6.02e-82
[+] SPRY2_RyR	cd12878	SPRY domain 2 (SPRY2) of ryanodine receptor (RyR); This SPRY domain (SPRY2) is the second of ...	1072-1204	1.78e-81
[+] RR_TM4-6	pfam06459	Ryanodine Receptor TM 4-6; This region covers TM regions 4-6 of the ryanodine receptor 1 ...	4383-4671	3.36e-80
[+] SPRY1_RyR	cd12877	SPRY domain 1 (SPRY1) of ryanodine receptor (RyR); This SPRY domain is the first of three ...	642-793	1.03e-79
[+] SPRY3_RyR	cd12879	SPRY domain 3 (SPRY3) of ryanodine receptor (RyR); This SPRY domain (SPRY3) is the third of ...	1418-1566	4.00e-76
[+] Ins145_P3_rec	pfam08709	Inositol 1,4,5-trisphosphate/ryanodine receptor; This domain corresponds to the ligand binding ...	8-203	5.34e-76
[+] MIR	pfam02815	MIR domain; The MIR (protein mannosyltransferase, IP3R and RyR) domain is a domain that may ...	211-389	1.70e-73
[+] RyR	pfam02026	RyR domain; This domain is called RyR for Ryanodine receptor. The domain is found in four ...	850-940	1.97e-44
[+] RyR	pfam02026	RyR domain; This domain is called RyR for Ryanodine receptor. The domain is found in four ...	2735-2825	1.25e-41
[+] RyR	pfam02026	RyR domain; This domain is called RyR for Ryanodine receptor. The domain is found in four ...	964-1054	6.92e-38
[+] RyR	pfam02026	RyR domain; This domain is called RyR for Ryanodine receptor. The domain is found in four ...	2855-2939	8.71e-36
[+] SPRY	smart00449	Domain in SPla and the RYanodine Receptor; Domain of unknown function. Distant homologues are ...	1084-1206	6.98e-35
[+] SPRY	pfam00622	SPRY domain; SPRY Domain is named from SPla and the RYanodine Receptor. Domain of unknown ...	1086-1206	6.63e-31
[+] RIH_assoc	pfam08454	RyR and IP3R Homology associated; This eukaryotic domain is found in ryanodine receptors (RyR) ...	3879-3992	2.58e-30
[+] SPRY	pfam00622	SPRY domain; SPRY Domain is named from SPla and the RYanodine Receptor. Domain of unknown ...	660-795	7.17e-28
[+] Ion_trans	pfam00520	Ion transport protein; This family contains sodium, potassium and calcium ion channels. This ...	4765-4949	1.68e-25
[+] SPRY	pfam00622	SPRY domain; SPRY Domain is named from SPla and the RYanodine Receptor. Domain of unknown ...	1431-1568	4.81e-24
[+] SPRY	smart00449	Domain in SPla and the RYanodine Receptor; Domain of unknown function. Distant homologues are ...	1430-1568	3.22e-23
[+] SPRY	smart00449	Domain in SPla and the RYanodine Receptor; Domain of unknown function. Distant homologues are ...	660-794	6.04e-19
[+] EF-hand_8	pfam13833	EF-hand domain pair;	4083-4133	5.82e-08
[+] MIR	smart00472	Domain in ryanodine and inositol trisphosphate receptors and protein O-mannosyltransferases;	210-263	8.98e-08
[+] MIR	smart00472	Domain in ryanodine and inositol trisphosphate receptors and protein O-mannosyltransferases;	97-152	9.66e-06

CDD provides a guide to domain level architecture, including sequence alignments & representative structures.



CDD provides a guide to domain level architecture, including sequence alignments & representative structures.



Once an initial trace is obtained for these regions, use Foldseek to identify structural homologs that could not be identified by sequence alone.

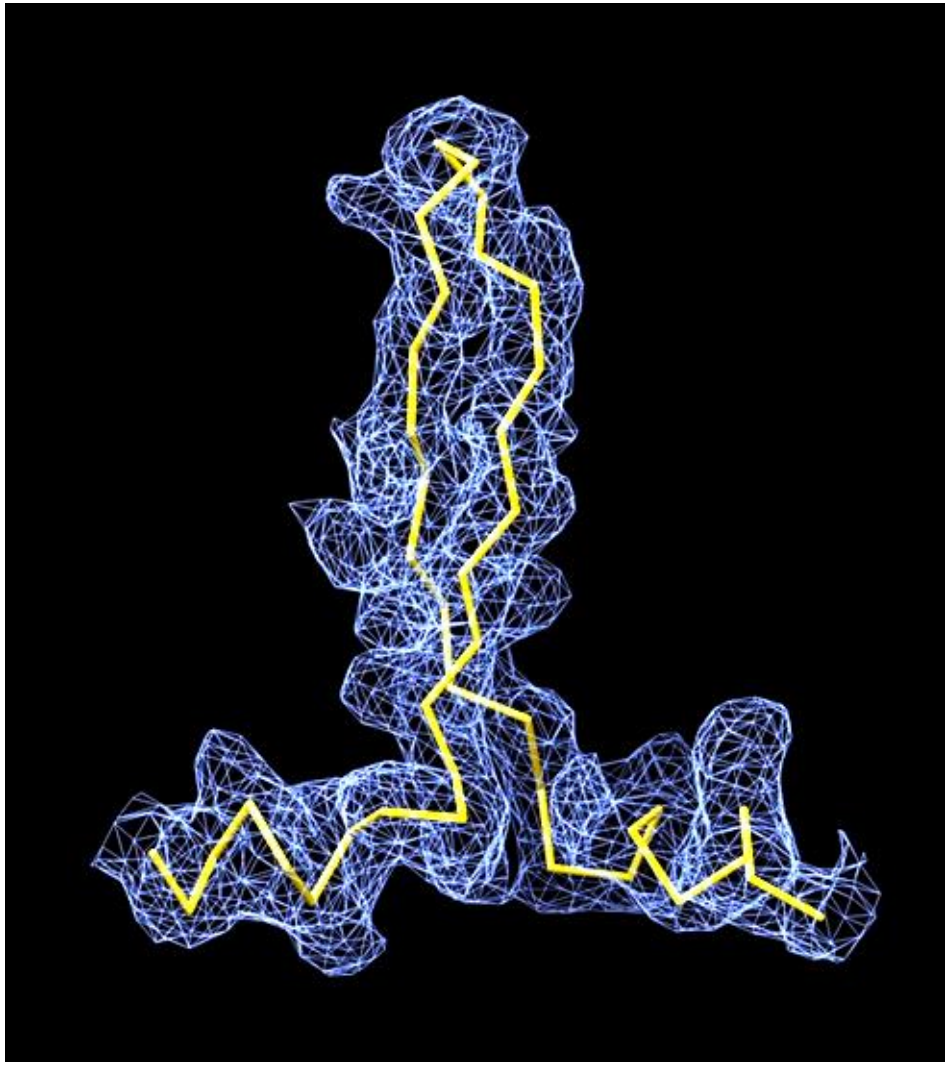
Prep for model building - what can we learn from the sequence alone?

Your protein sequence contains a lot of useful information which you can use to aid model building:

- Start by identifying boundaries of conserved domains (NCBI CDD: https://www.ncbi.nlm.nih.gov/Structure/cdd/; DELTA-BLAST also performs CD-search by default)
- Then identify and/or generate suitable structural templates for building known domains: FUGUE, PHYRE 2, MUSTER. trROSETTA useful in cases where sequence homology is limited (and Alphafold/ROSETTAfold!).
- **Secondary structure, TM & disorder prediction.**
- Contact prediction from evolutionary couplings: EVFOLD & GREMLIN.
- Conservation analysis: Use favorite MSA algorithm (MUSCLE & CLUSTAL-OMEGA work well; TM-COFFEE, PRALINE-TM useful for membrane proteins) to create a sequence alignment of your protein with a few orthologs; gaps & insertions most commonly occur in loops/disordered regions. Useful as a guide during building.

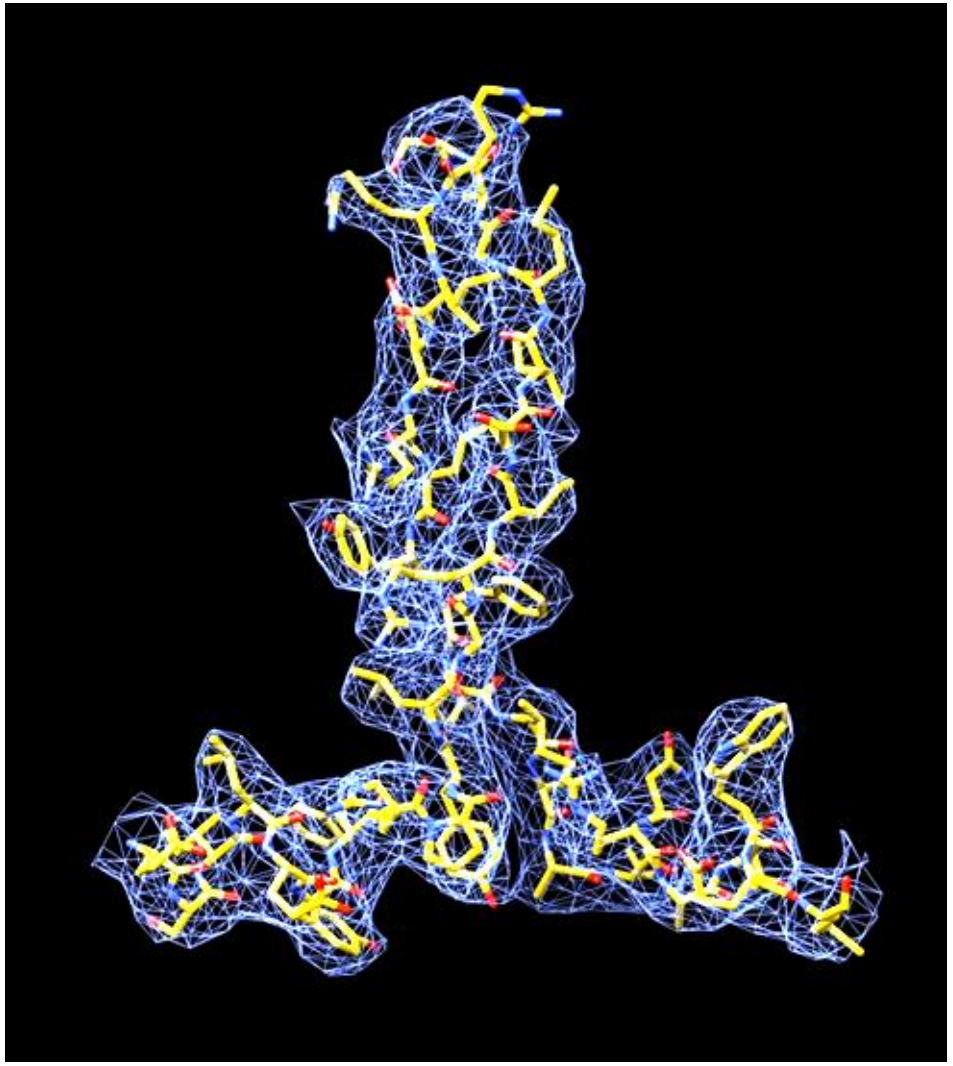
Secondary structure prediction is a very useful guide when building.

```
....*.3210....*.3220....*.3230....*.3240....*.3250....*.3260....*.3270....*.3280....*.3290....*.3300
MPVAFLEPQLNEYNACSVYTTKSPRERAILGLPNSVEEMCPDIPVLDRLMADIGGLAESGARYTEMPHVIEITLPMLCSYLPRWWERGPEAPPPALPAGA

....*.3310....*.3320....*.3330....*.3340....*.3350....*.3360....*.3370....*.3380....*.3390....*.3400
PPPCTAVTSDHLNSLLGNILRIIVNNLGIDEATWMKRLAVFAQPIVSRARPELLHSHFIPTIGRLRKRAGKVVAEEEQLRLEAKAEAEEGELLVRDEFSV

....*.3410....*.3420....*.3430....*.3440....*.3450....*.3460....*.3470....*.3480....*.3490....*.3500
LCRDLYALYPLLIRYVDNNRAHWLTEPNANAEELFRMVGEIFIYWSKSHNFKREEQNFVVQNEINNMSFLTADSKSKMAKAGDAQSGGSDQERTKKKRRG

....*.3510....*.3520....*.3530....*.3540....*.3550....*.3560....*.3570....*.3580....*.3590....*.3600
DRYSVQTSLIVATLKKMLPIGLNMCAPTDQDLIMLAKTRYALKDTDEEVREFLQNNLHLQGKVEGSPSLRWQMALYRGLPGREEDADDPEKIVRRVQEVS

....*.3610....*.3620....*.3630....*.3640....*.3650....*.3660....*.3670....*.3680....*.3690....*.3700
AVLYHLEQTEHPYKSKKAVWHKLLSKQRRRAVVACFRMTPLYNLPTHRACNMFLESYKAAWILTEDHSFEDRMIDDLSKAGEQEEEEEEVEEKKPDPLHQ

....*.3710....*.3720....*.3730....*.3740....*.3750....*.3760....*.3770....*.3780....*.3790....*.3800
LVLHFSRTALTEKSKLDEDYLYMAYADIMAKSCHLEEGGENGEAEEEEVEVSFEEKEMEKQRLLYQQSRLHTRGAAEMVLQMISACKGETGAMVSSTLKL

....*.3810....*.3820....*.3830....*.3840....*.3850....*.3860....*.3870....*.3880....*.3890....*.3900
GISILNGGNAEVQQKMLDYLKDKKEVGFFQSIQALMQTCSVLDLNAFERQNKAEGLGMVNEDGTVINRQNGEKVMADDEFTQDLFRFLQLLCEGHNNDFQ

....*.3910....*.3920....*.3930....*.3940....*.3950....*.3960....*.3970....*.3980....*.3990....*.4000
NYLRTQTGNTTTINIIICTVDYLLRLQESISDFYWYYSGKDVIEEQGKRNFSKAMSVAKQVFNSLTEYIQGPCTGNQQSLAHSRLWDAVVGFLHVFAHMM

....*.4010....*.4020....*.4030....*.4040....*.4050....*.4060....*.4070....*.4080....*.4090....*.4100
MKLAQDSSQIELLKELLDLQKDMVVMLLSLLEGNVVNGMIARQMVDMLVESSSNVEMILKFFDMFLKLKDIVGSEAFQDYVTDPRGLISKKDFQKAMDSQ

....*.4110....*.4120....*.4130....*.4140....*.4150....*.4160....*.4170....*.4180....*.4190....*.4200
KQFTGPEIQFLLSCSEADENEMINFEEFANRFQEPARDIGFNVAVLLTNLSEHVPHDPRLRNFLELAESILEYFRPYLGRIEIMGASRRIERIYFEISET

....*.4210....*.4220....*.4230....*.4240....*.4250....*.4260....*.4270....*.4280....*.4290....*.4300
NRAQWEMPQVKESKRQFIFDVVNEGGEAEKMELFVSFCEDTIFEMQIAAQISEPEGEPEADEDEGMGEAAAEGAEEGAAGAEGAAGTVAAGATARLAAAA

....*.4310....*.4320....*.4330....*.4340....*.4350....*.4360....*.4370....*.4380....*.4390....*.4400
ARALRGLSYRSLRRRVRRLRRLTAREAATALAALLWAVVARAGAAGAGAAAGALRLLWGSLFGGGLVEGAKKVTVTELLAGMPDPTSDEVHGEQPAGPGG

....*.4410....*.4420....*.4430....*.4440....*.4450....*.4460....*.4470....*.4480....*.4490....*.4500
DADGAGEGEGEGDAAEGDGDEEVAGHEAGPGGAEGVVAVADGGPFRPEGAGGLGDMGDTTPAEPPTPEGSPILKRKLGVDGEEEELVPEPEPEPEPEPEK

....*.4510....*.4520....*.4530....*.4540....*.4550....*.4560....*.4570....*.4580....*.4590....*.4600
ADEENGEKEEVPEAPPEPPKKAPPSPPAKKEEAGGAGMEFWGELEVQRVKFLNYLSRNFYTLRFLALFLAFAINFILLFYKVSDSPPGEDDMEGSAAGDL
```



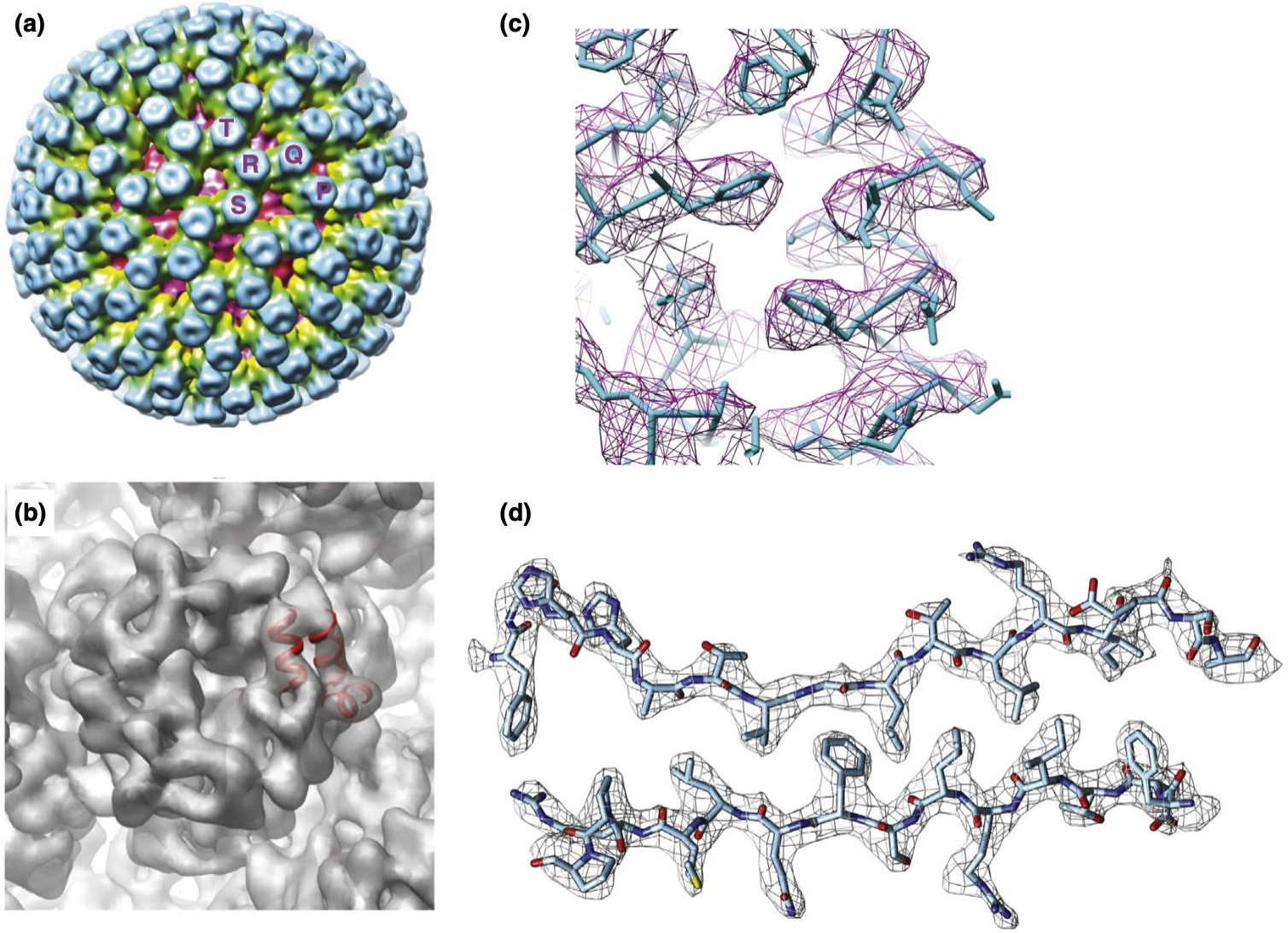
Where is this motif in the sequence?

Secondary structure prediction is a very useful guide when building.



Secondary structure prediction is ~80% accurate. So if your model consistently disagrees with predicted secondary structure, look at it very closely!

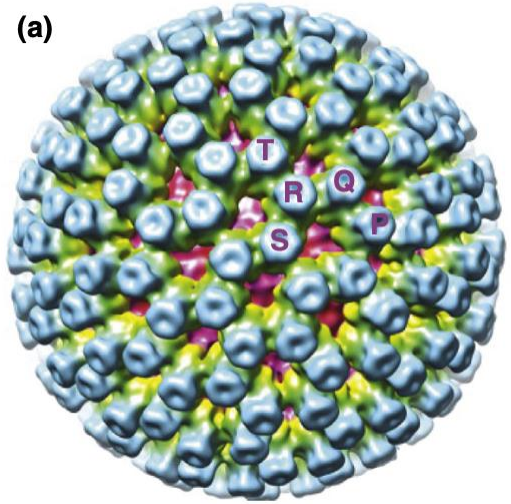
What do maps look like at different resolutions?

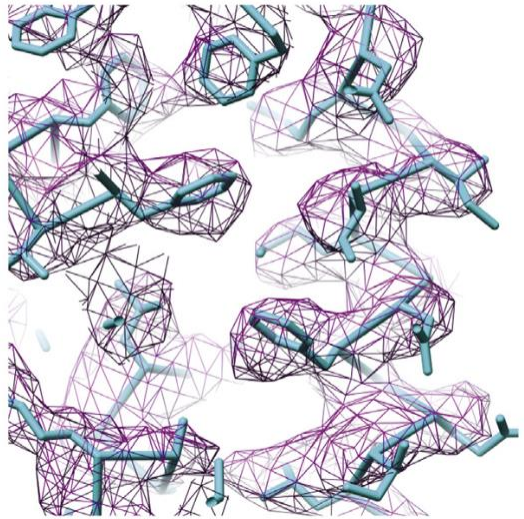


Rosenthal & Rubinstein, Curr. Opinion Struct. Bio. 2015

Map resolution should be consistent with observed features!

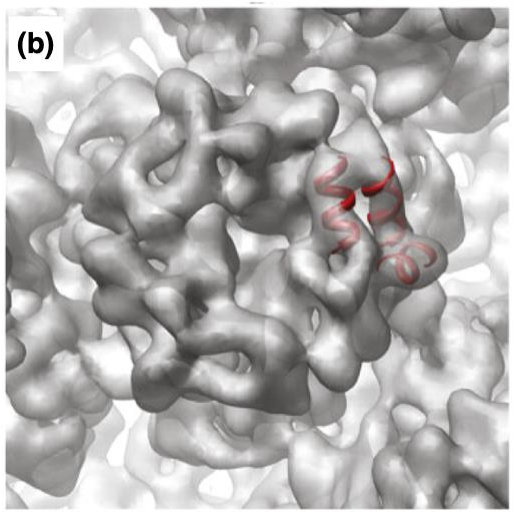
20Å

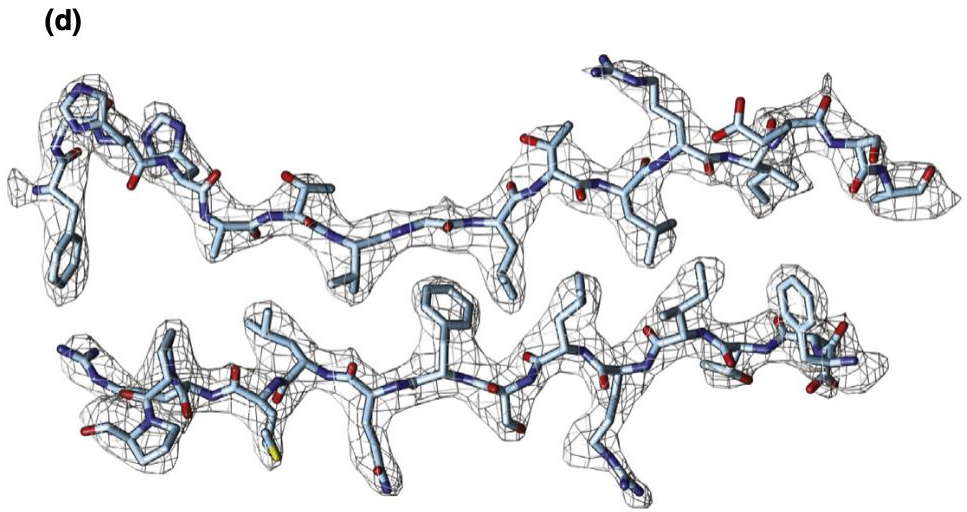




3.8Å

7Å

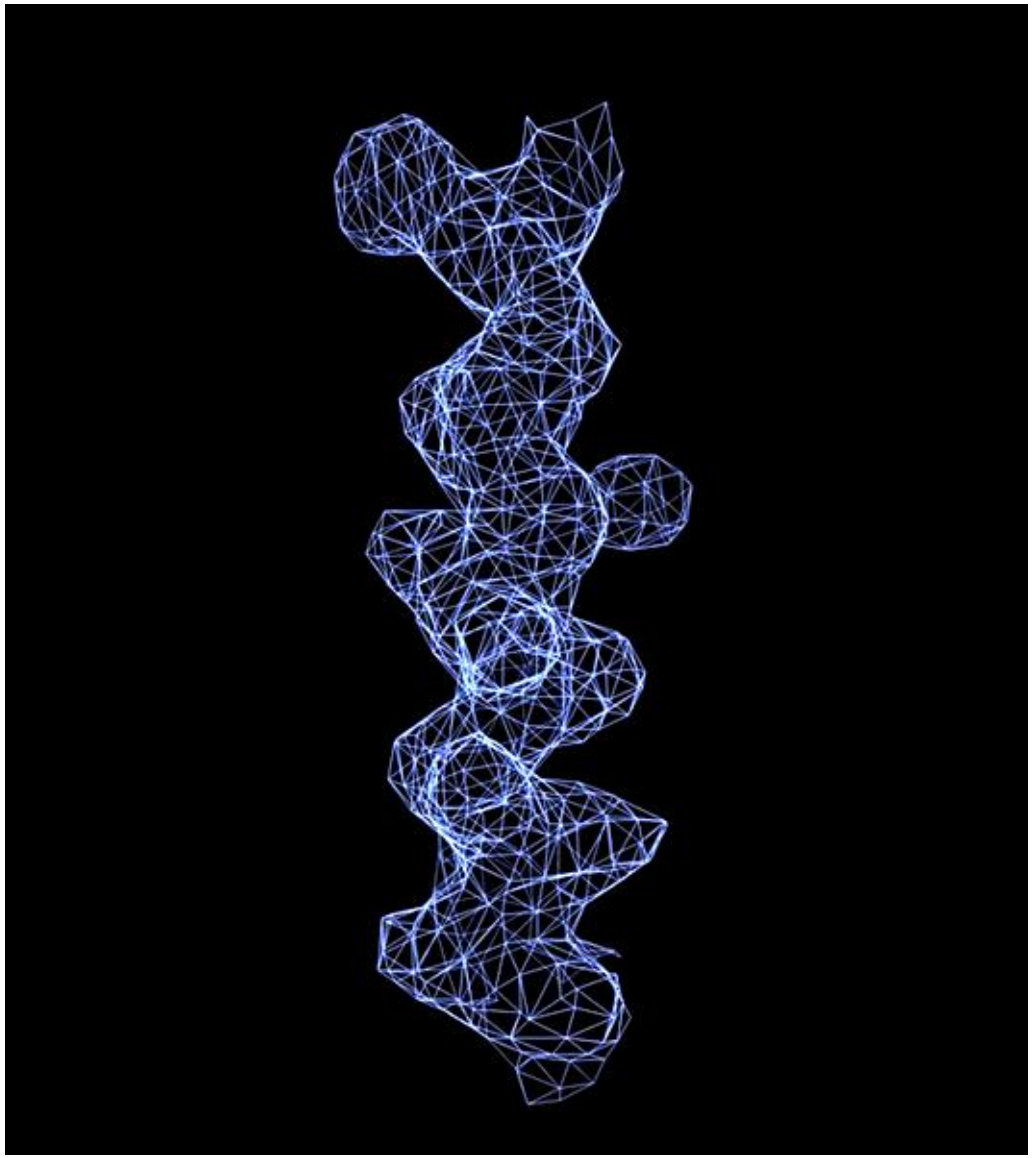




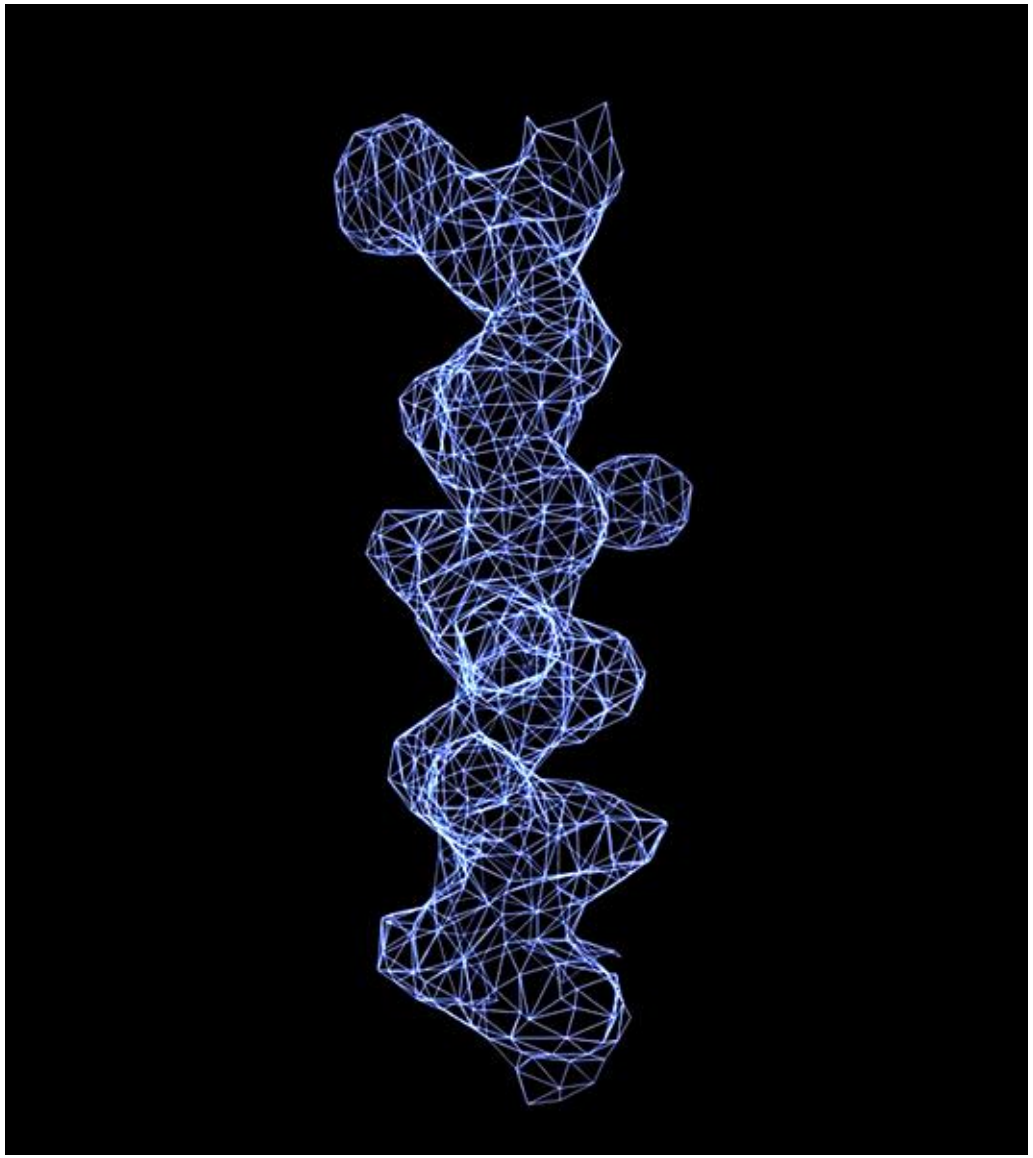
2.6Å

Rosenthal & Rubinstein, Curr. Opinion Struct. Bio. 2015

What can we learn from the map alone?

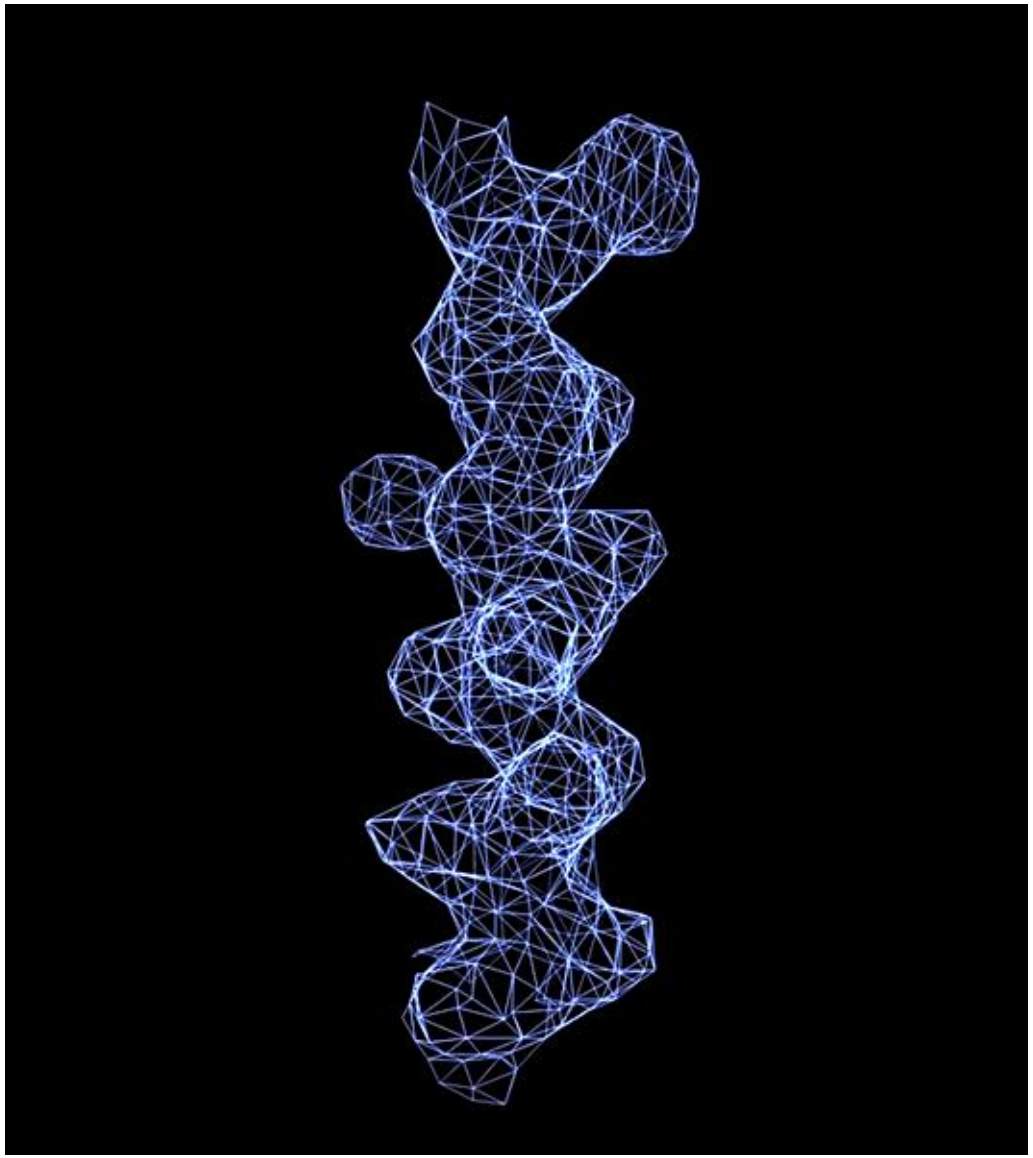


What can we learn from the map alone?

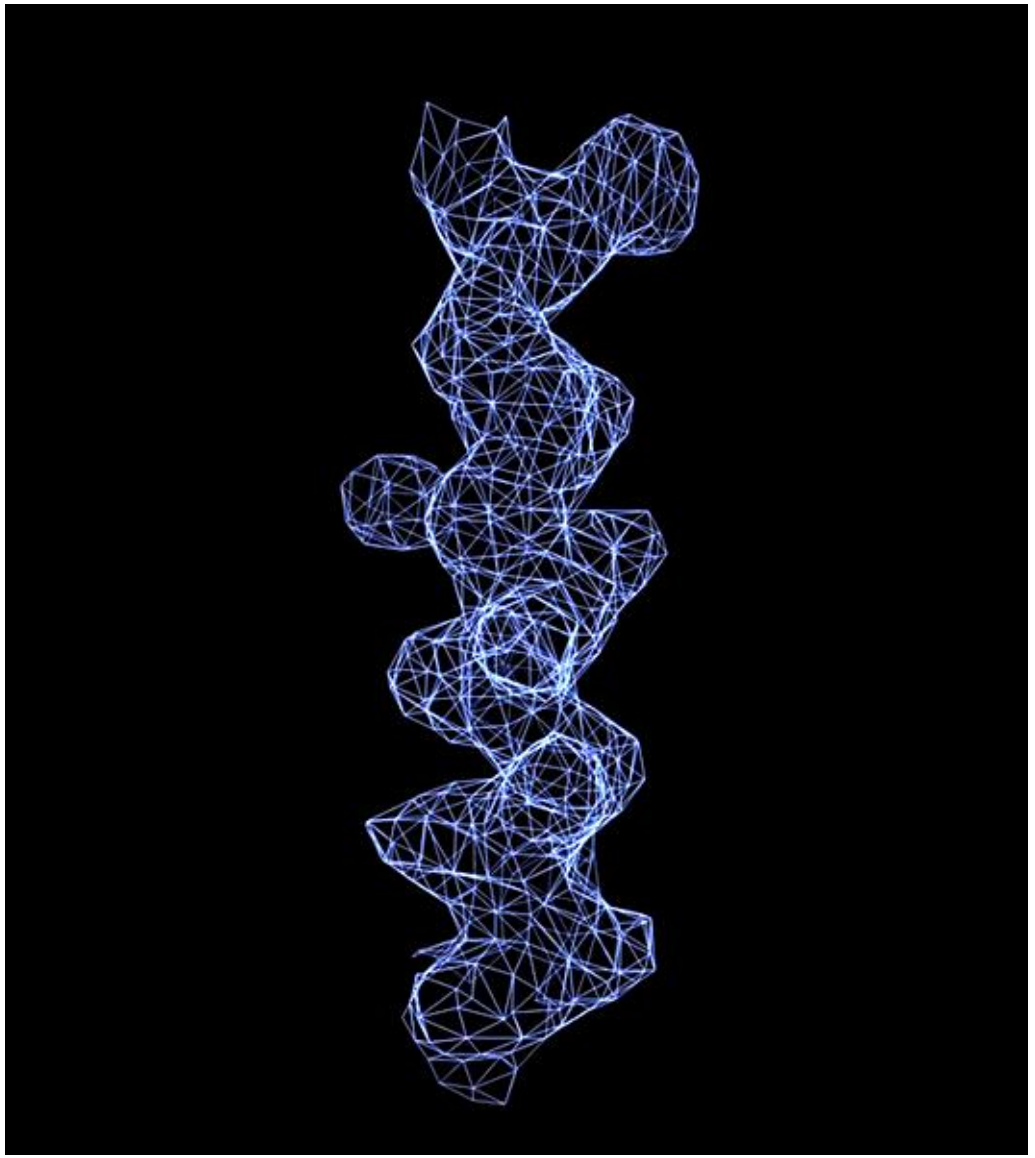


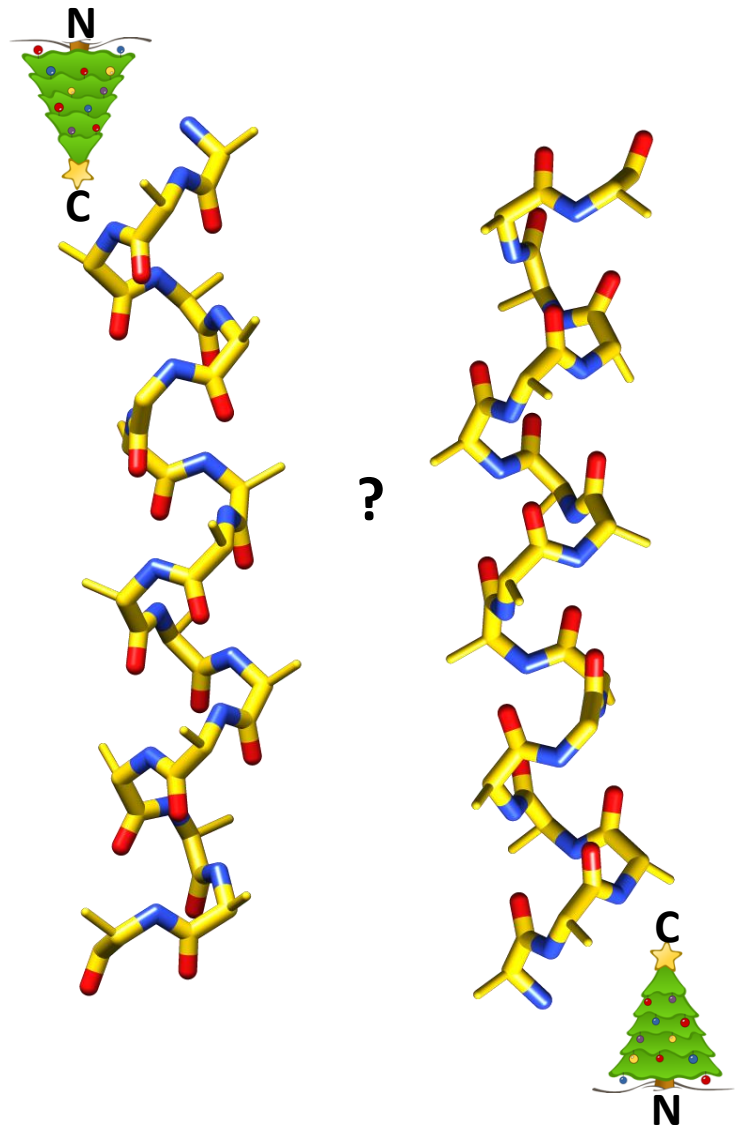
Left handed! Obvious here – can be less clear at lower res, so be careful.

OK, that’s better! What can we learn from the map alone?

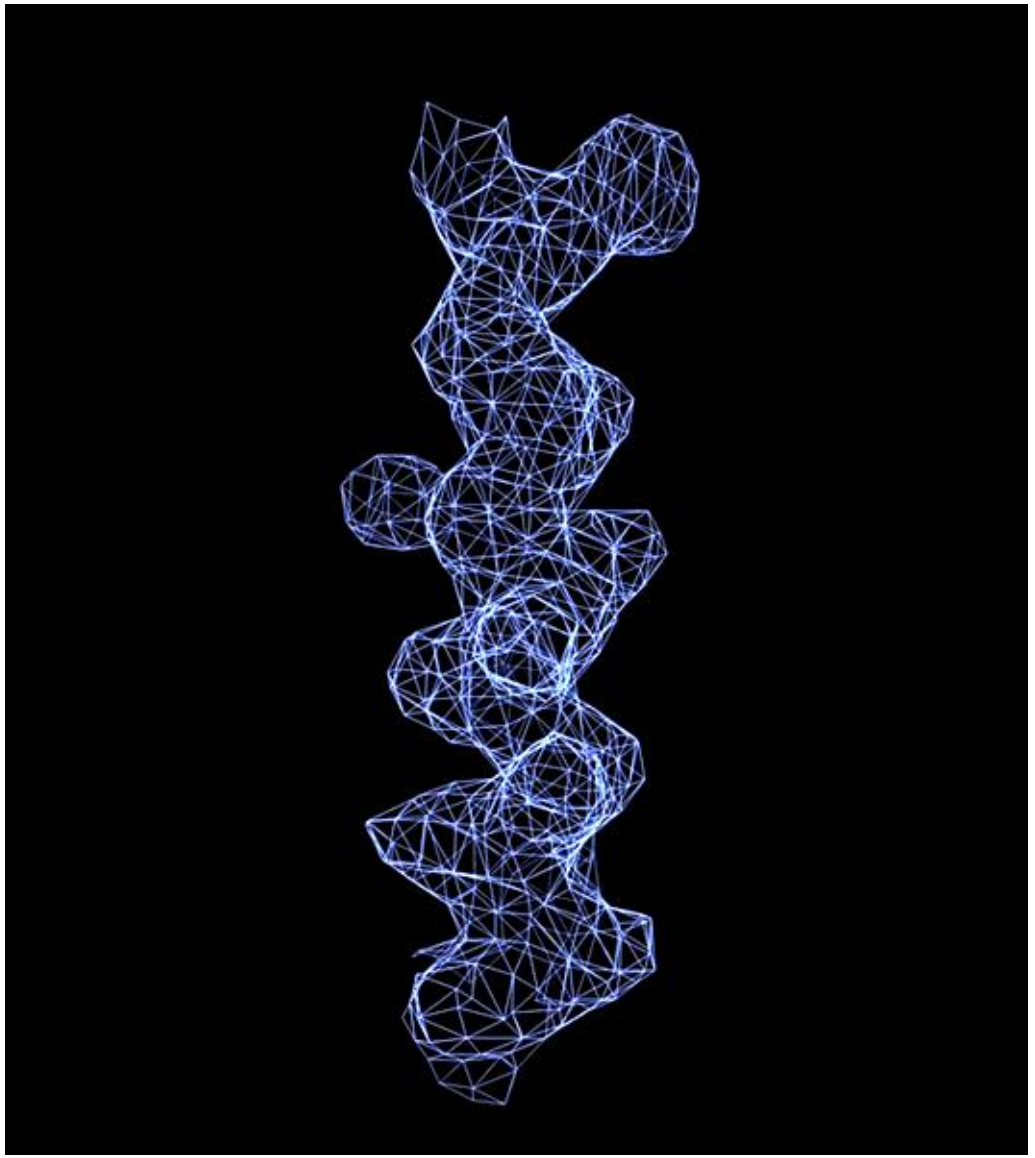


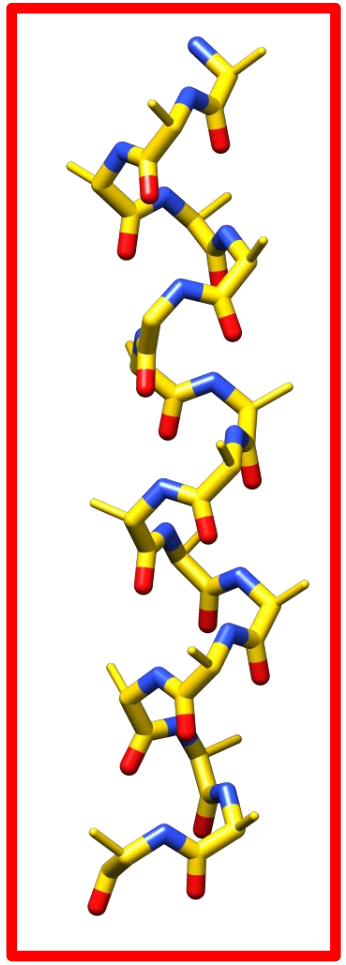
Which direction does the helix point?



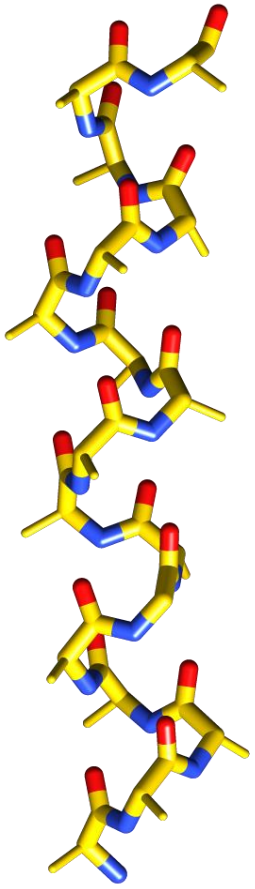


Which direction does the helix point?

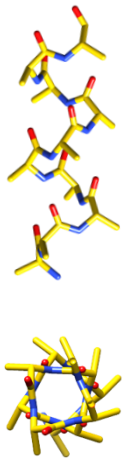




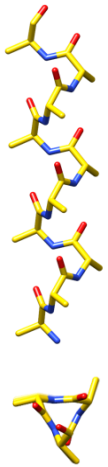
?



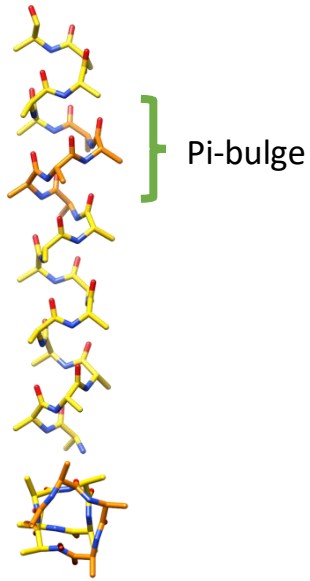
Helices – alpha and 3_{10}



Alpha
- ~90%
- 3.6 residues per turn
- Fat

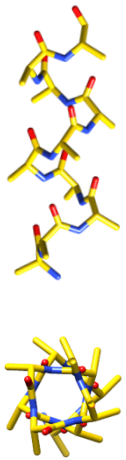


3_{10}
- ~10%. More common in TM? (e.g. S4 of VSD)
- 3 residues per turn. Triangular cross section.
- Skinny
- Can be tricky to identify at low resolution, can lead to register errors.

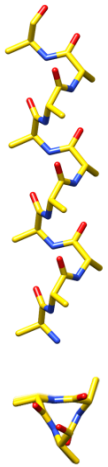


Pi bulge
- Rare as a proportion of helical residues, but occur in ~15% of proteins
- 4.1 residues per turn; i+5 H-bond pattern, pentagonal cross-section
- Can be tricky to identify at low resolution, can lead to register errors.
- Never present as extended helix, but often insertion in alpha helix (e.g. S6 of TRPV1, RyR).
- Often biased sequence composition (rich in aromatics, e.g. 4xPhe in RyR1)

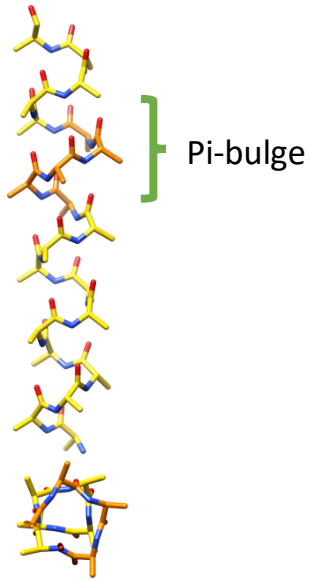
Helices – alpha and 3_{10}



Alpha
- ~90%
- 3.6 residues per turn
- Fat



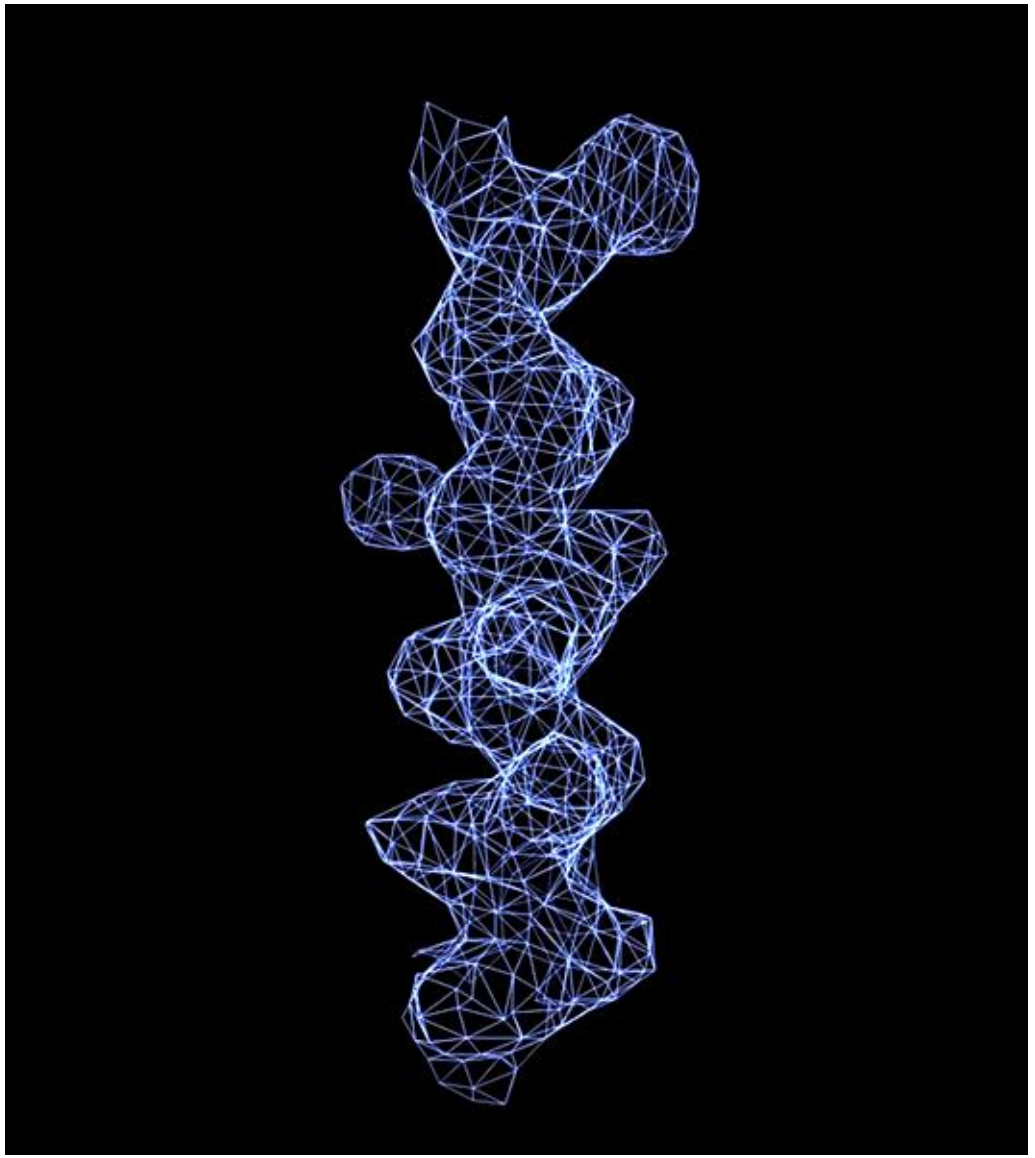
3_{10}
- ~10%. More common in TM? (e.g. S4 of VSD)
- 3 residues per turn. Triangular cross section.
- Skinny
- Can be tricky to identify at low resolution, can lead to register errors.

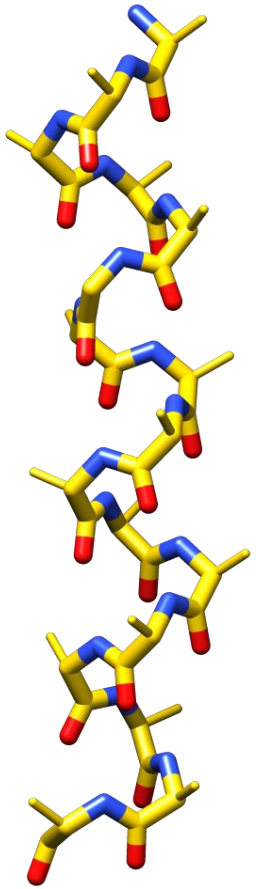


Pi bulge
- Rare as a proportion of helical residues, but occur in ~15% of proteins
- 4.1 residues per turn; i+5 H-bond pattern, pentagonal cross-section
- Can be tricky to identify at low resolution, can lead to register errors.
- Never present as extended helix, but often insertion in alpha helix (e.g. S6 of TRPV1, RyR).
- Often biased sequence composition (rich in aromatics, e.g. 4xPhe in RyR1)

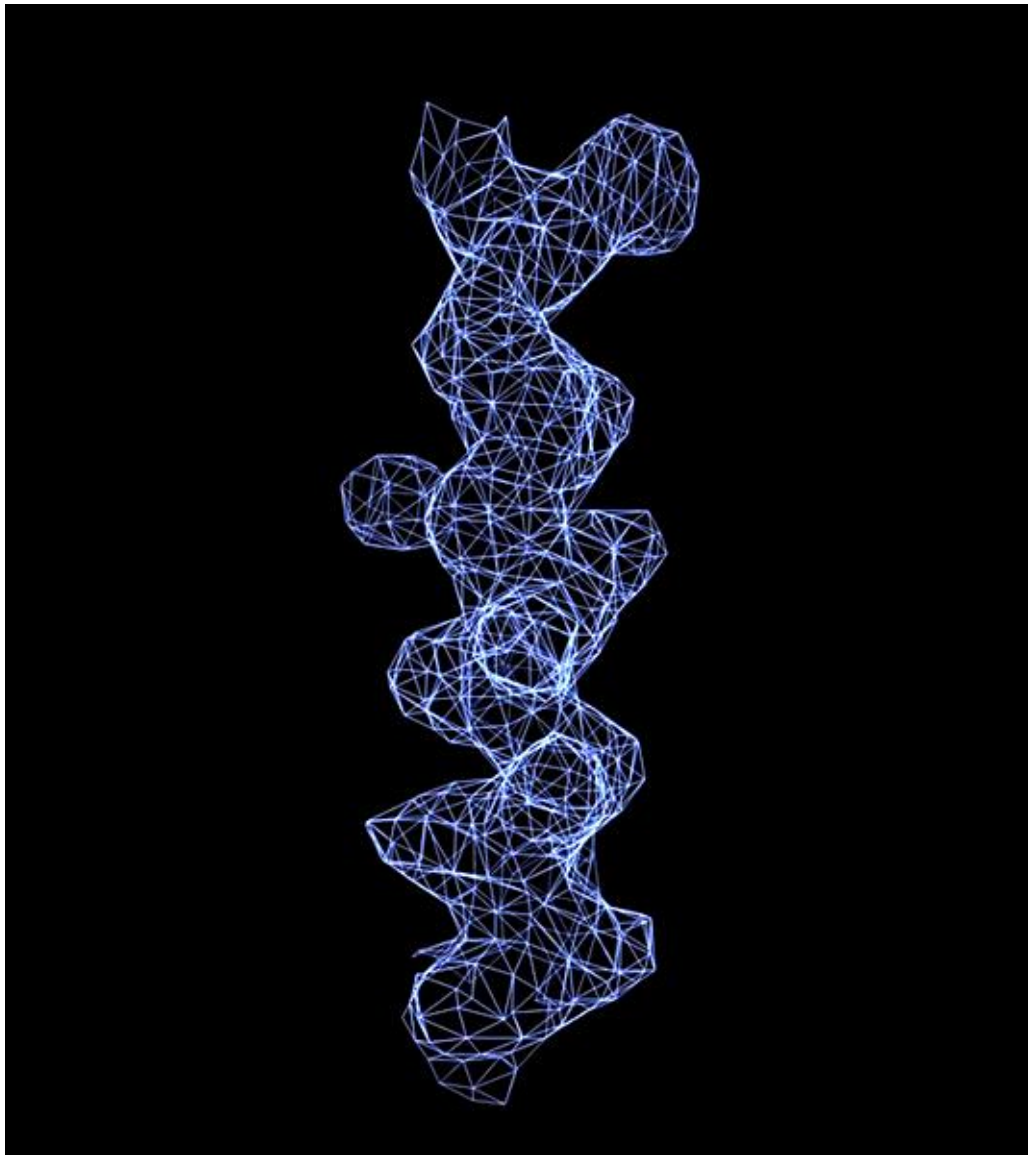
Require manual definition of restraints at low res – do not use alpha helical restraints!

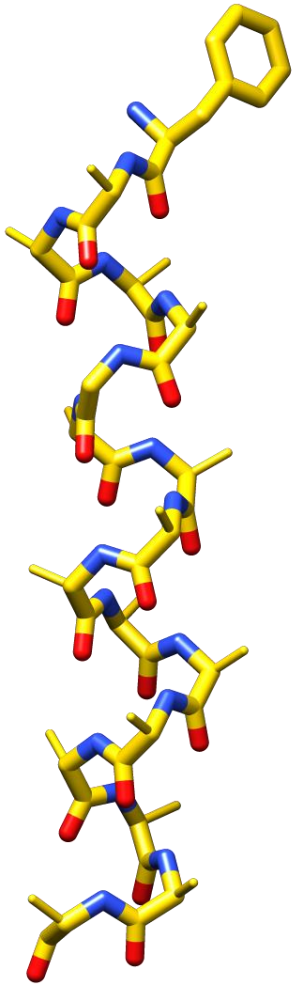
Can we identify any probable sidechains from the density?



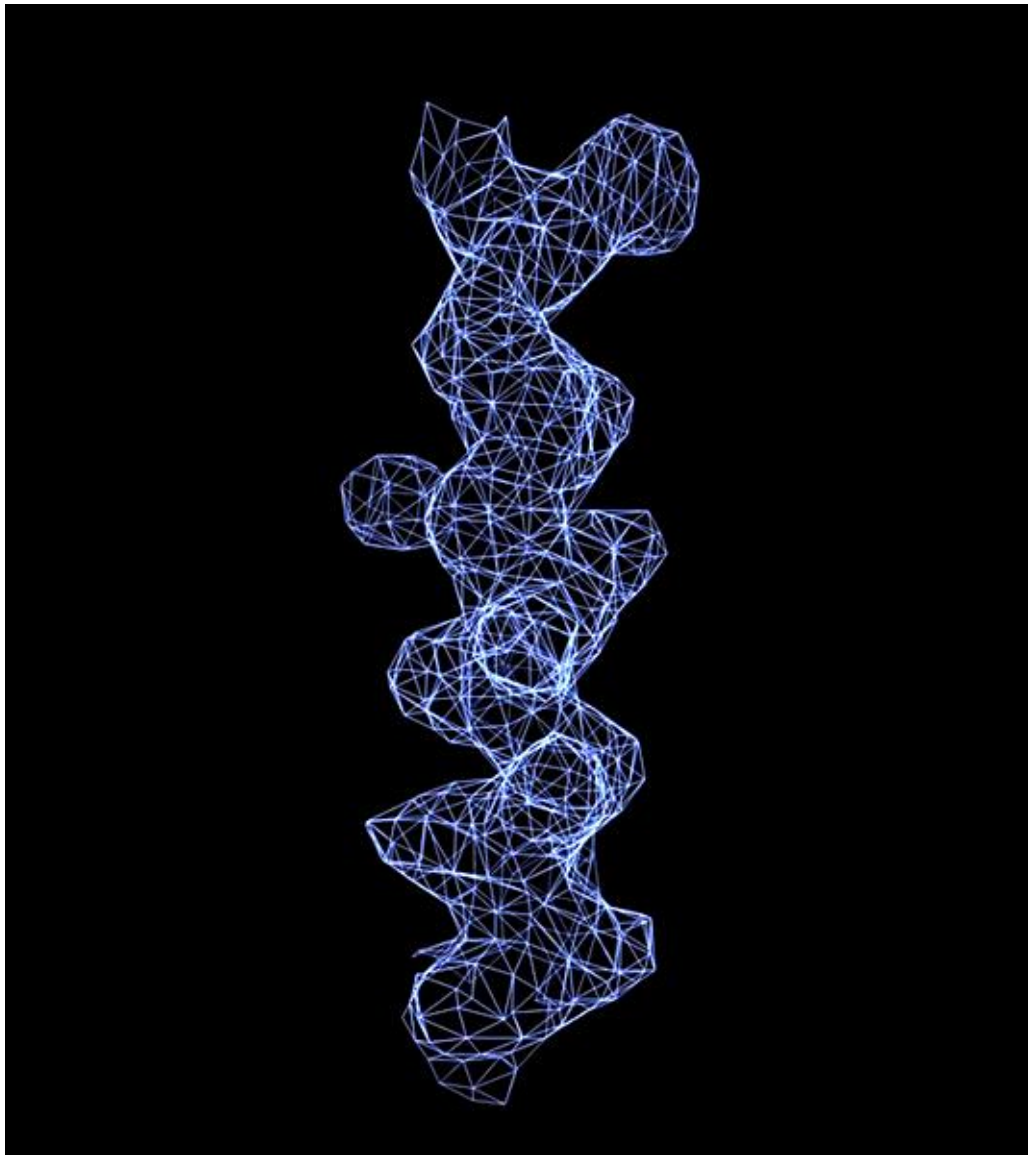


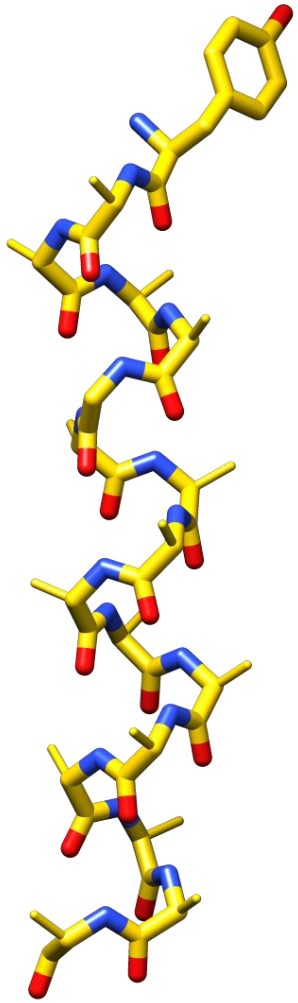
Can we identify any probable sidechains from the density?



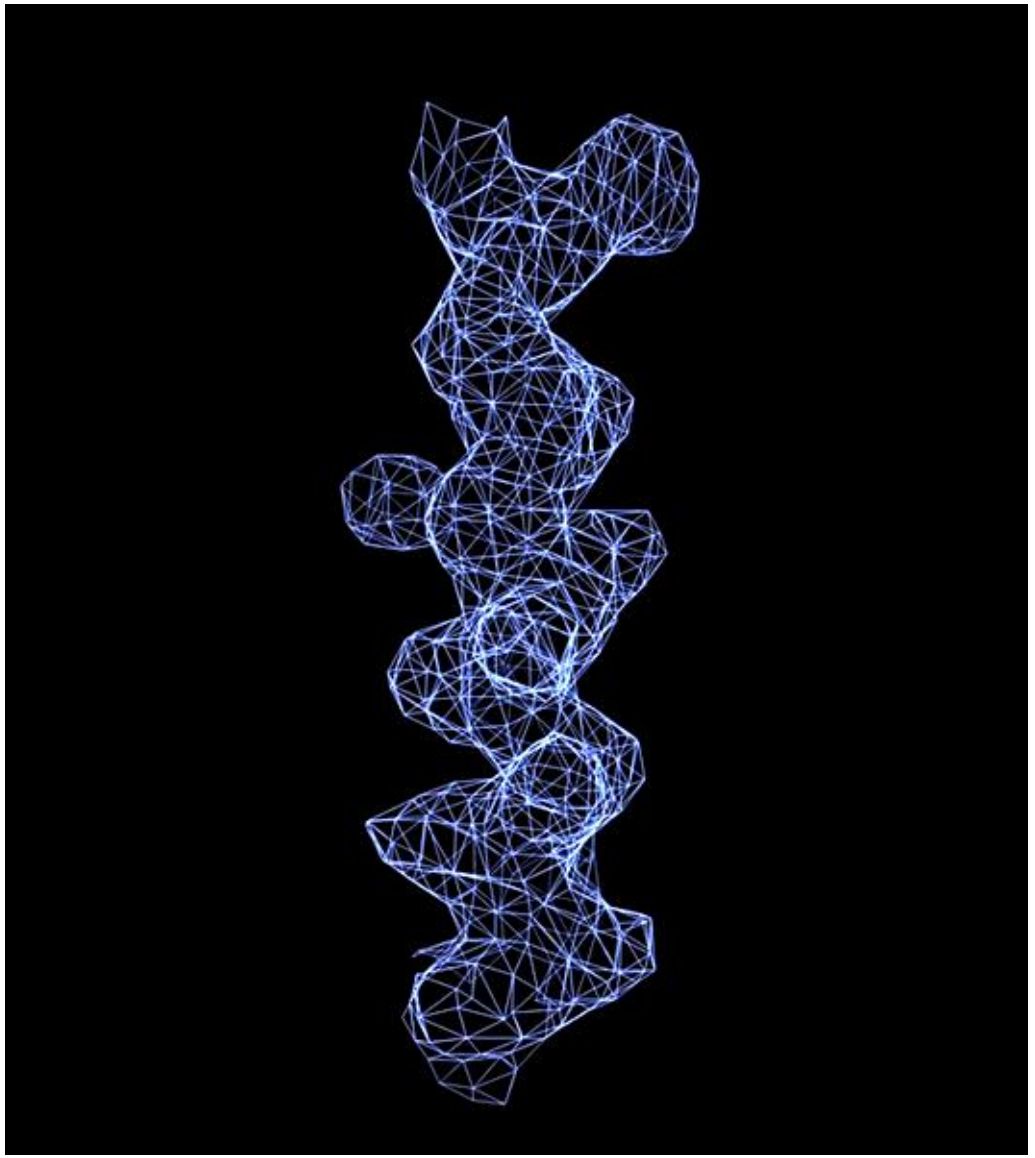


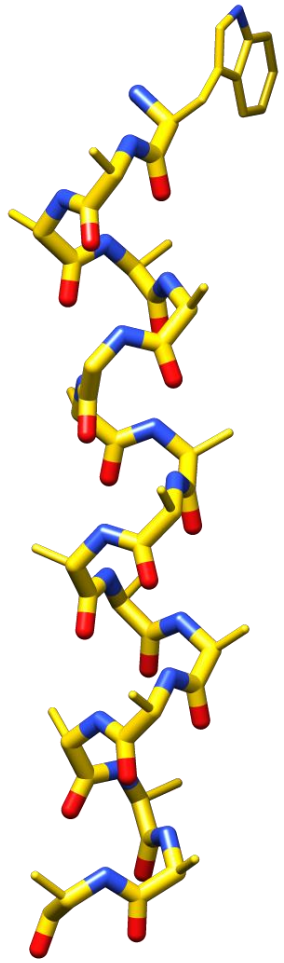
Can we identify any probable sidechains from the density?



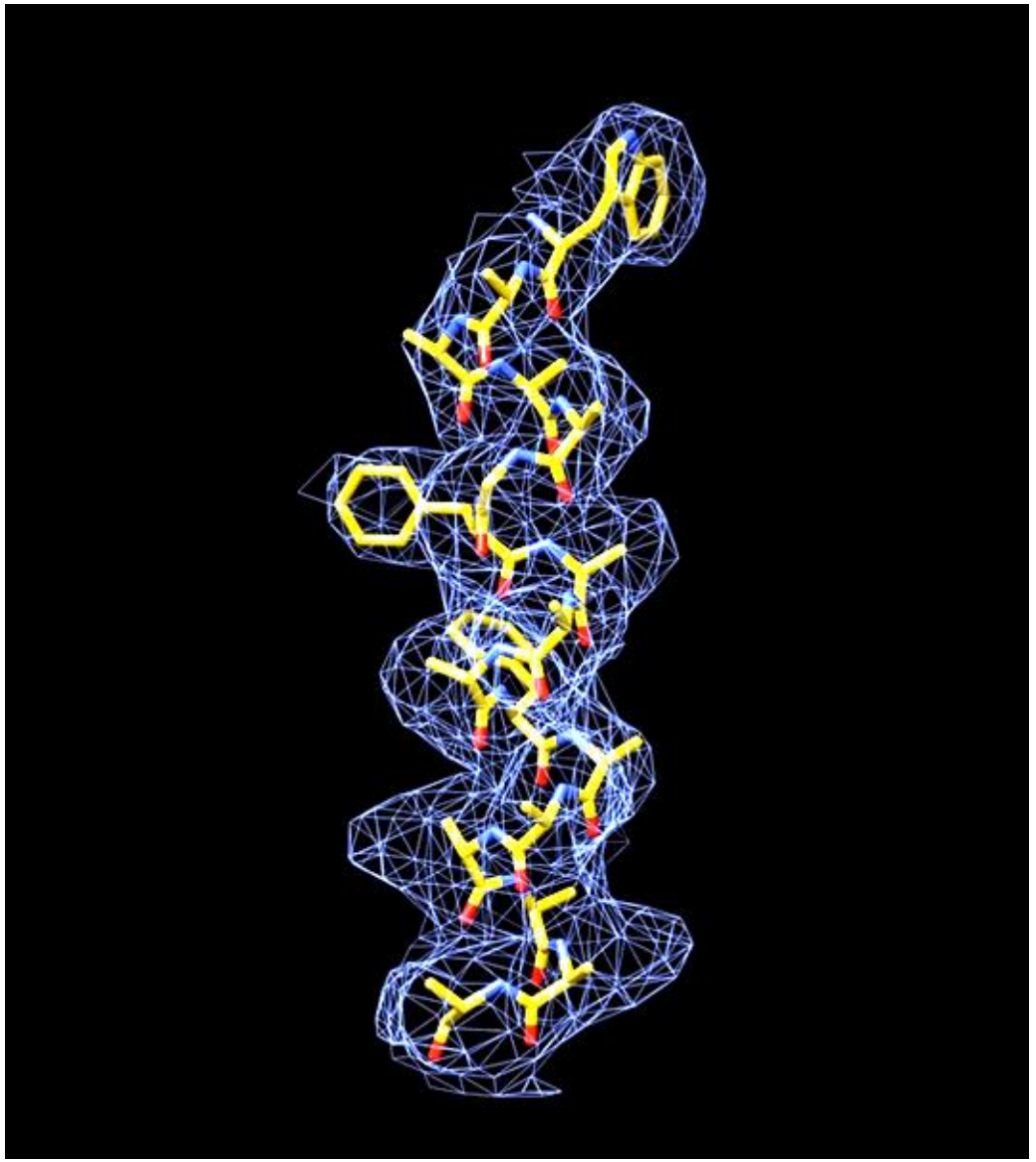


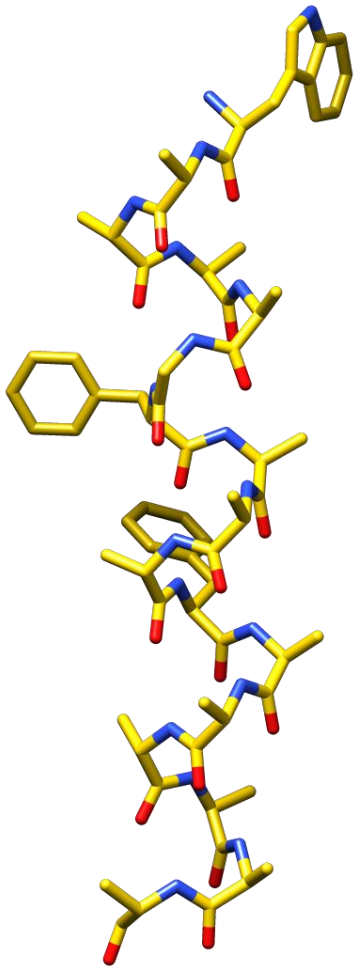
Can we identify any probable sidechains from the density?





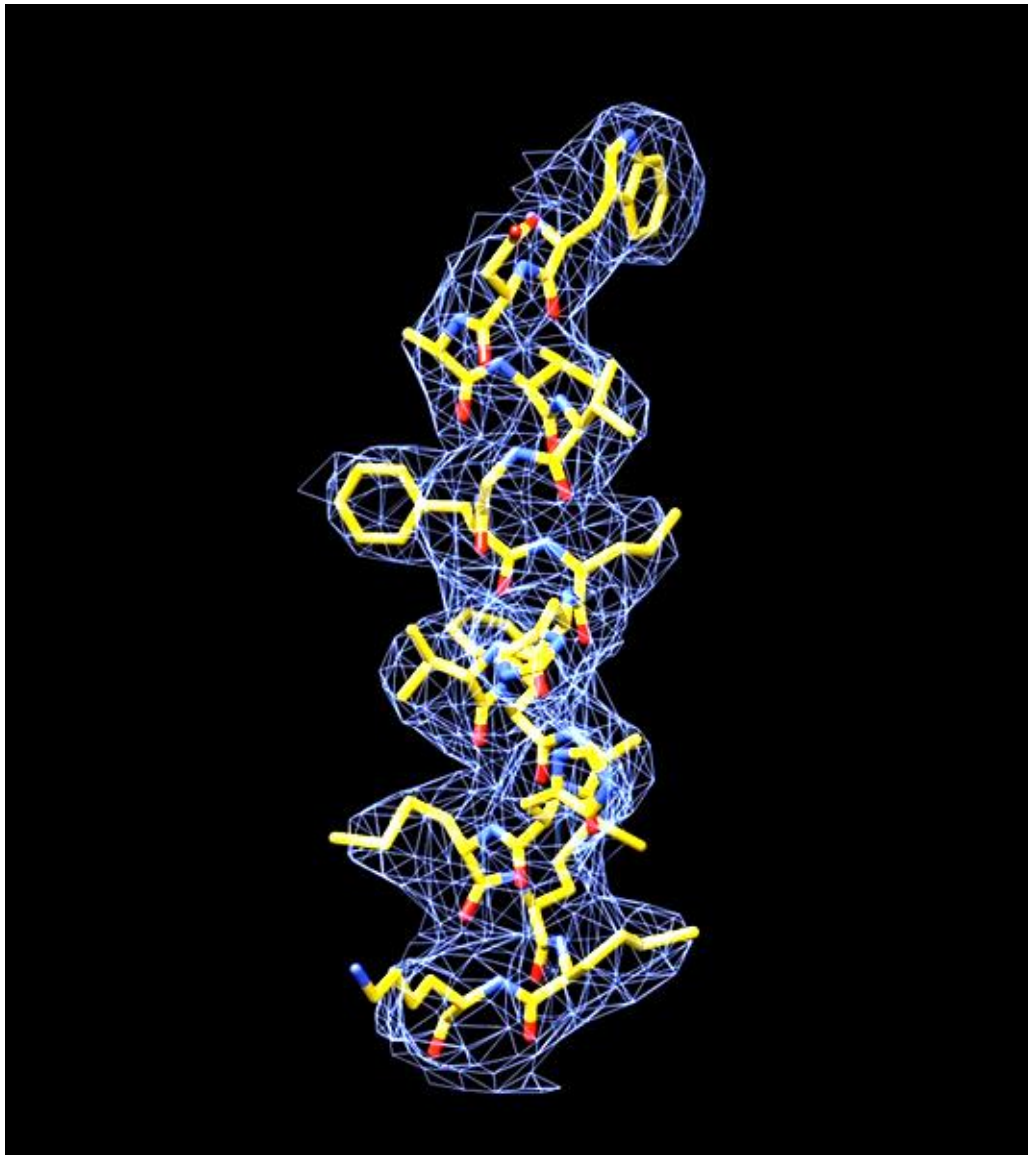
Test the initial hypothesis by extending sequence assignment along the chain.

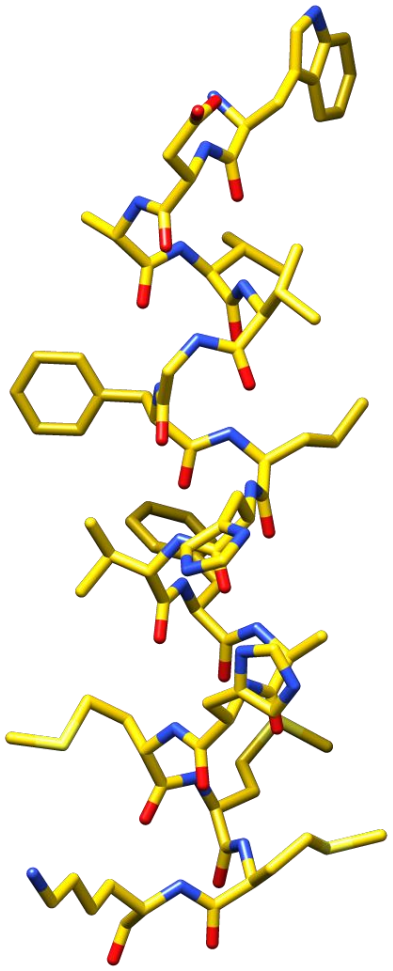




…VFNSLTEYIQGPCTGNQQSLAHSRL**W**DAVVG**F**LHV**F**AHMMMMKLAQDSSQIELLKELLDLQ…

Test the initial hypothesis by extending sequence assignment along the chain.

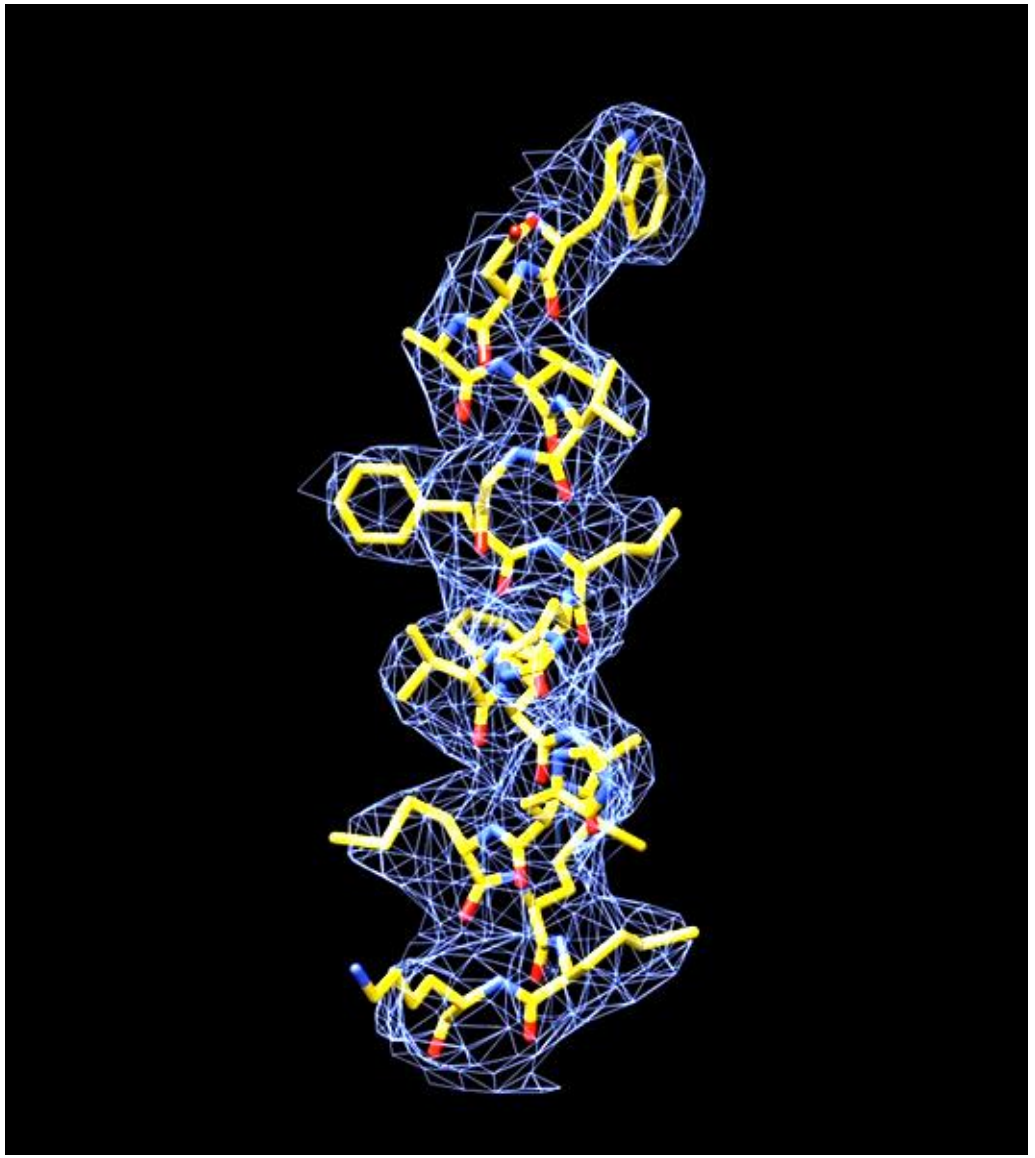


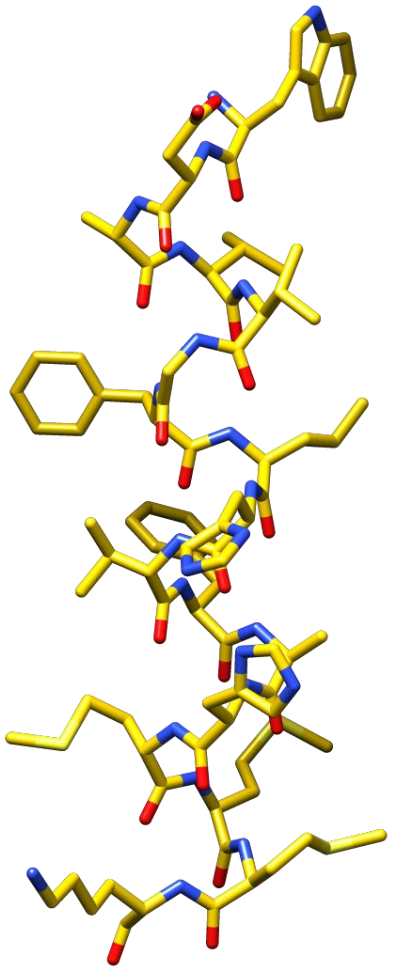


…VFNSLTEYIQGPCTGNQQSLAHSRL**WDAVVGFLHVFAHMMMK**LAQDSSQIELLKELLDLQ…

Test the initial hypothesis by extending sequence assignment along the chain.

Notice that the **absence** of large sidechain densities at small residue positions is just as valuable in validating the fit as the fit of large sidechains to the density.

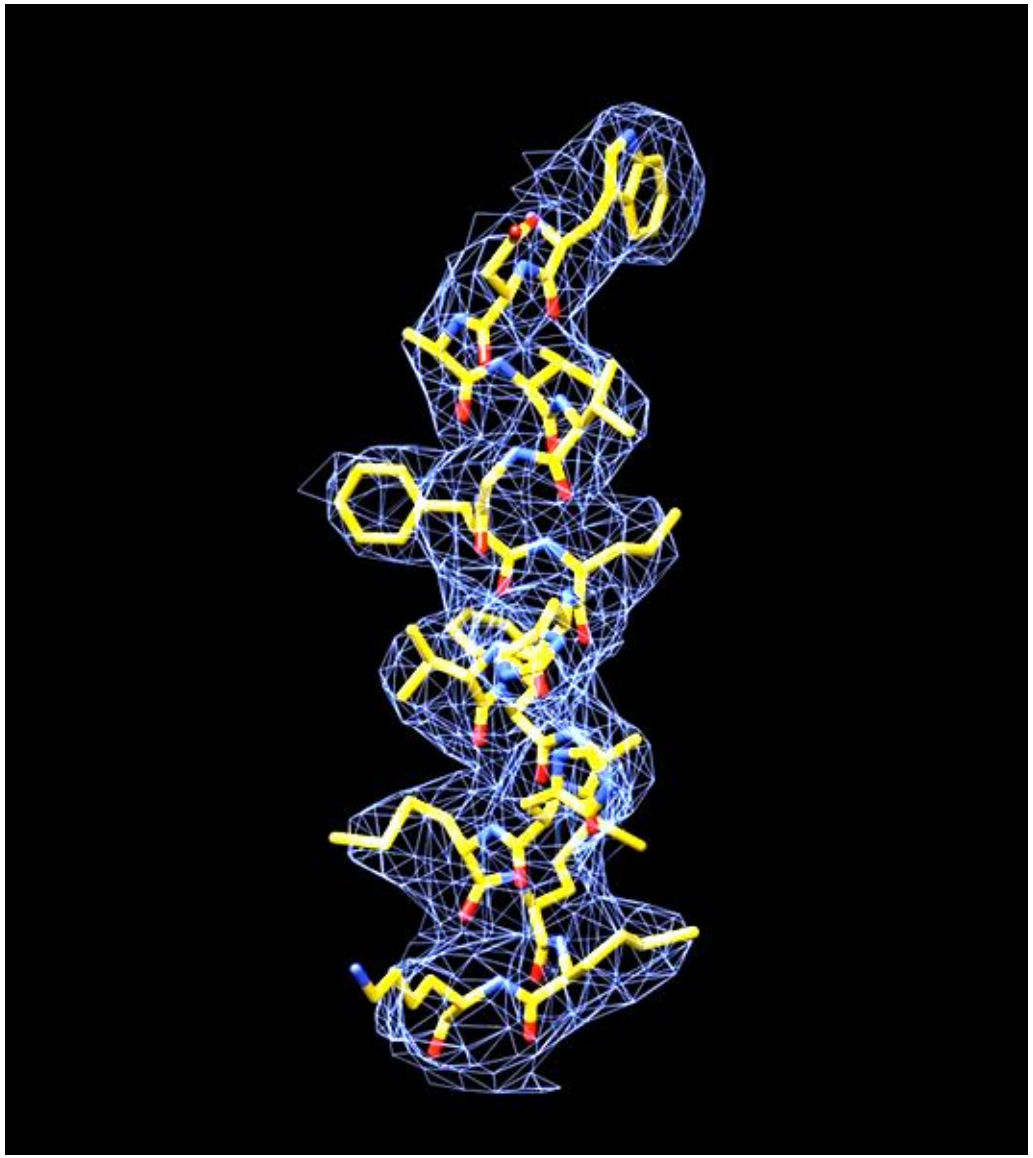


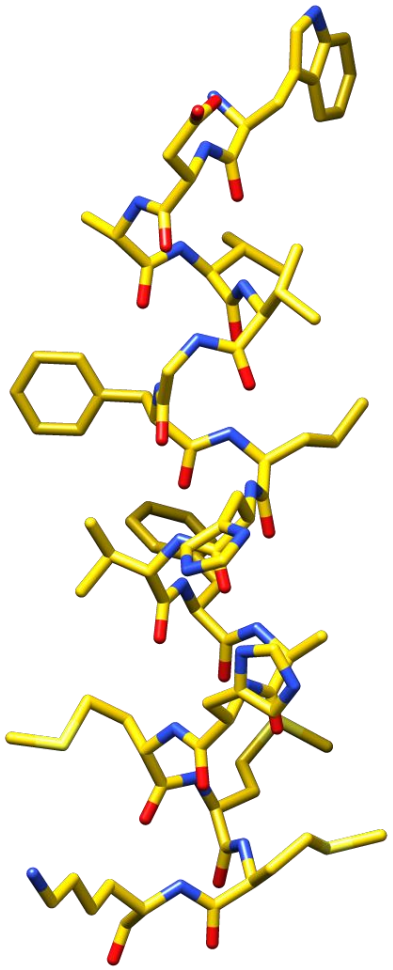


…VFNSLTEYIQGPCTGNQQSLAHSRL**WDAVVGFLHVFAHMMMMK**LAQDSSQIELLKELLDLQ…

Test the initial hypothesis by extending sequence assignment along the chain.

Also, note that the information content of local regions varies. Consider "VTVVAASSTVV" vs "FGAAYWVTRA" – which is more likely to be uniquely identifiable from the map?





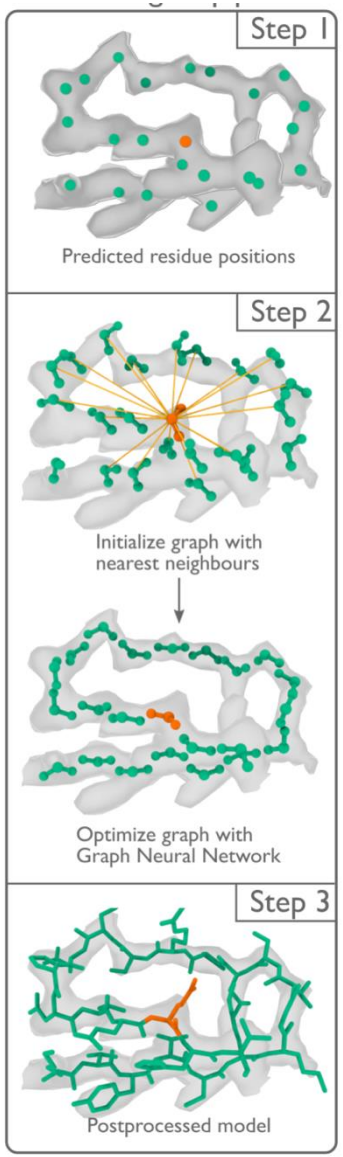
...VFNSLTEYIQGPCTGNQQSLAHSRL**WDAVVGFLHVFAHMMMK**LAQDSSQIELLKELLDLQ...

Modelangelo – applying neural networks to map interpretation

Here, we introduce a machine-learning approach, called ModelAngelo, for the automated building of atomic models and the identification of proteins in cryo-EM maps. Machine learning approaches often require large amounts of training data. For example, recent protein language models were trained on tens of millions of sequences (*14*) and AlphaFold2 was trained on more than 200,000 structures (*15*). In contrast, fewer than 13,000 cryo-EM structures with resolutions better than 4 Å have been determined to date and many of these are redundant. The limited amount of available training data prompted us to design a multi-modal machine-learning approach that combines local information from the cryo-EM map surrounding each protein or nucleic acid residue with additional information from the protein sequences in the sample and the local geometry of the structure. Similar sources of information are exploited by human experts when manually building atomic models in cryo-EM maps.

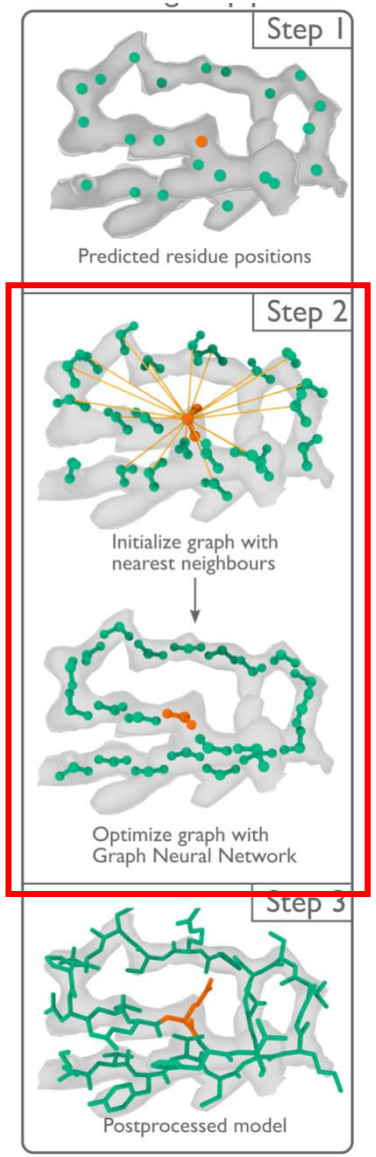
(Jamali et al., BiorXiv, 2023)

Modelangelo – applying neural networks to map interpretation



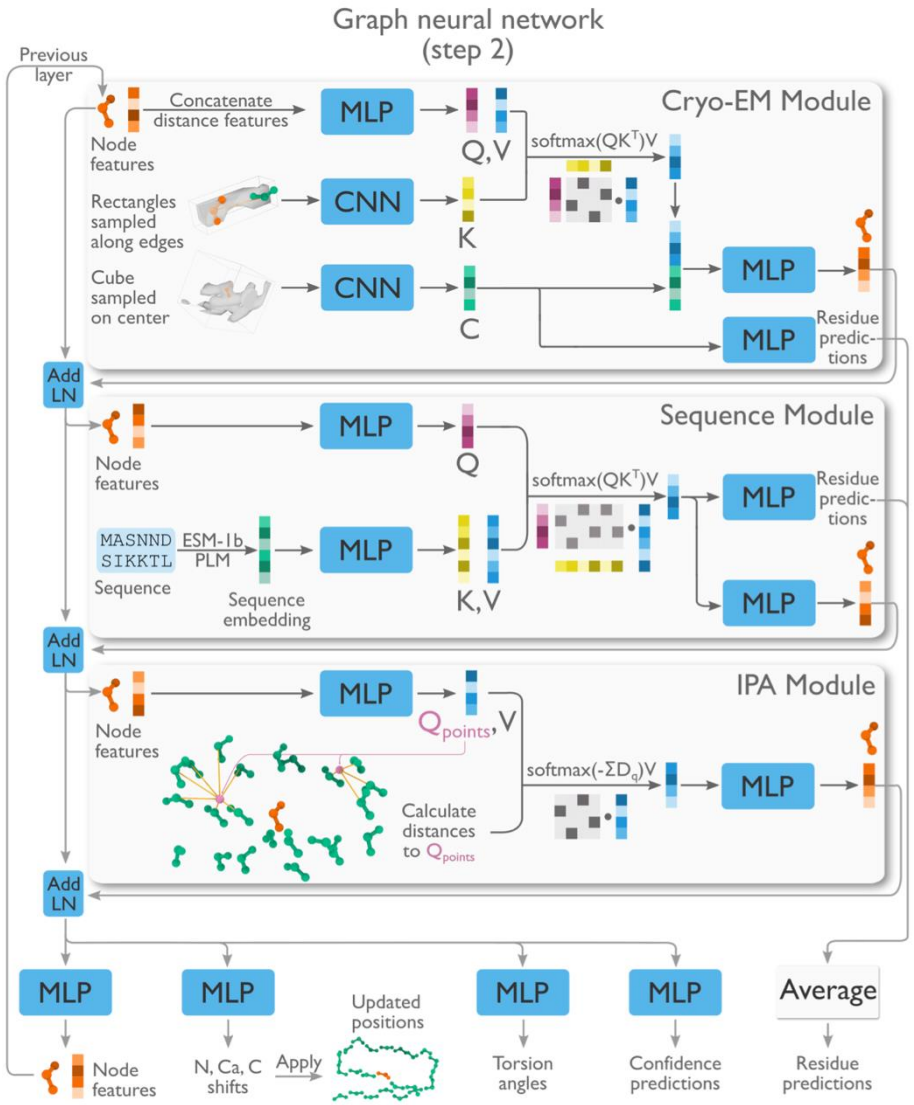
(Jamali et al., BiorXiv, 2023)

Modelangelo – applying neural networks to map interpretation



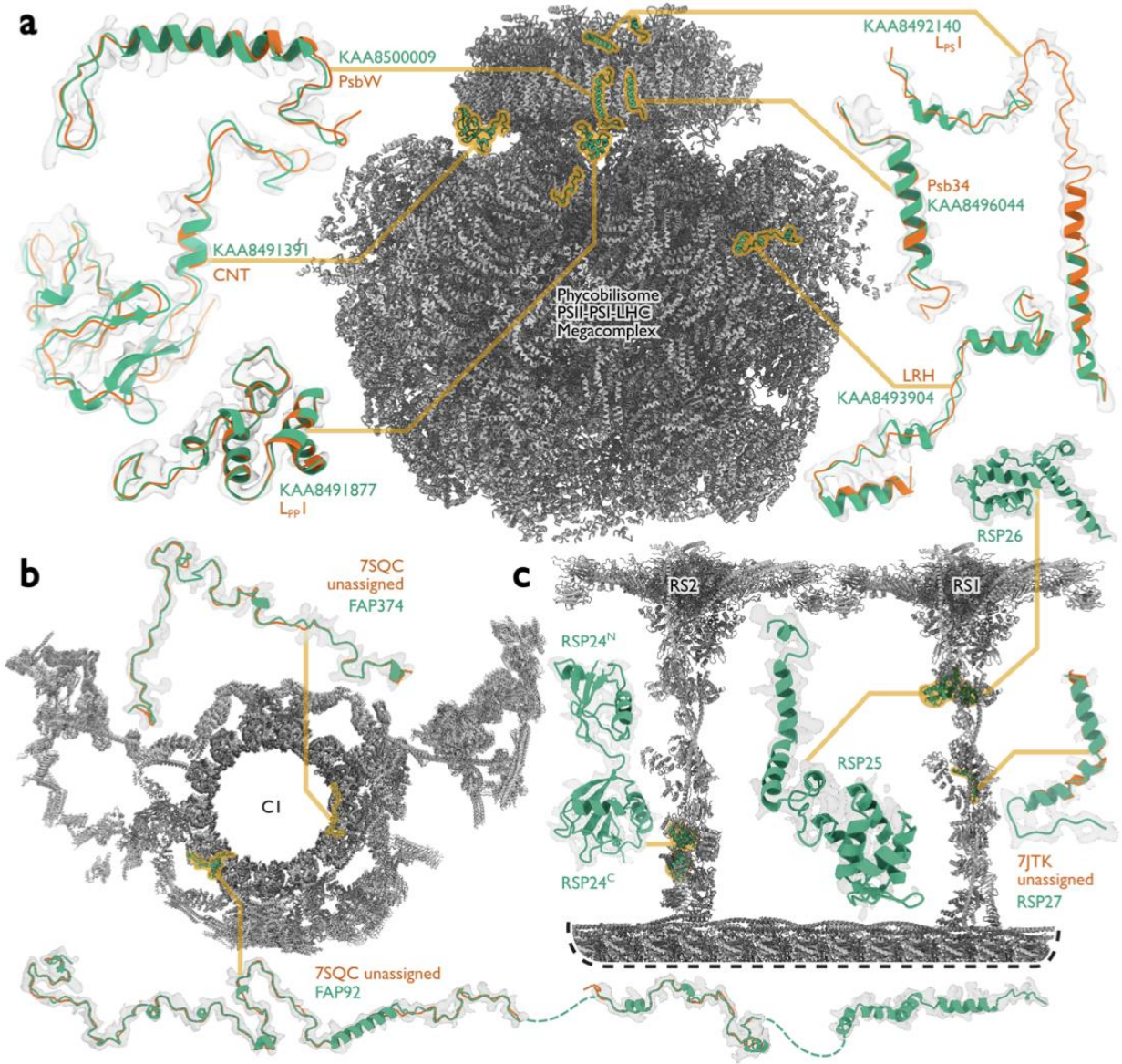
(Jamali et al., BiorXiv, 2023)

Modelangelo – applying neural networks to map interpretation



(Jamali et al., BiorXiv, 2023)

Modelangelo – applying neural networks to map interpretation



(Jamali et al., BiorXiv, 2023)

Modelangelo – applying neural networks to map interpretation

- ***Fantastic*** starting point for model building – generates near-complete models with good geometry in favorable areas.

- Difficult regions (e.g. flexible, anisotropic) still require manual building (for now!)

- Model still requires manual inspection and analysis (both for completion/correction/validation, and understanding!)

- Can't build waters/ions/ligands (yet!)

- ***Allows us to build better, faster, and focus on difficult/important regions.***

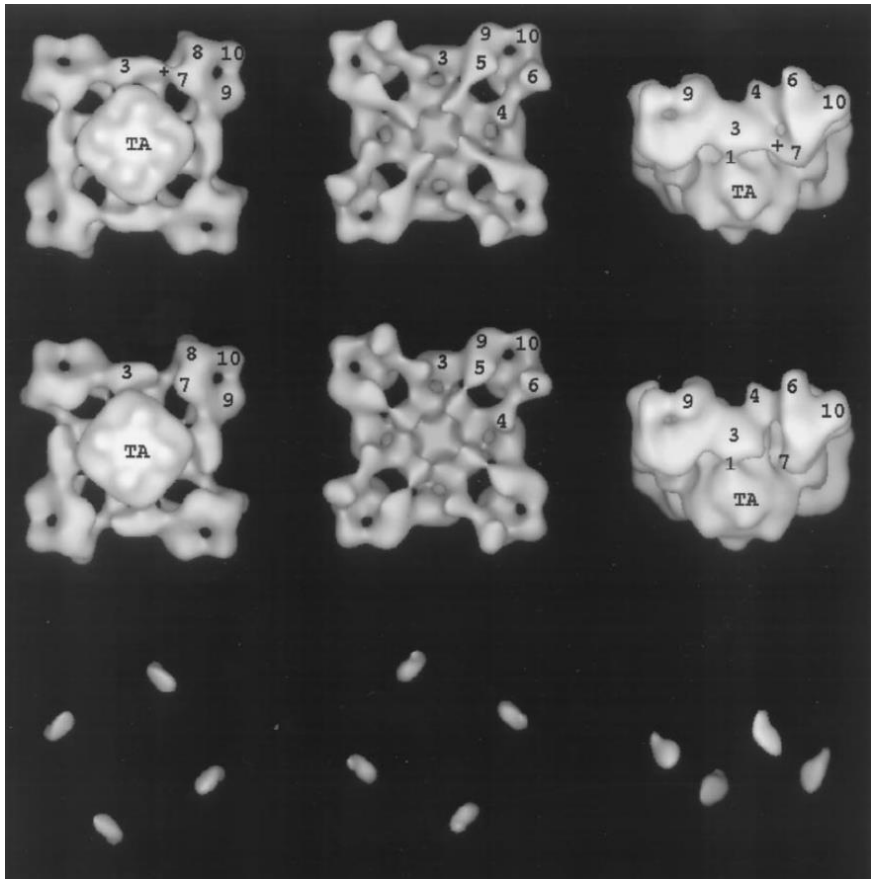
(Jamali et al., BiorXiv, 2023)

How to deal with uncertainty in sequence assignment and sidechain placement

- You will likely encounter situations where you cannot be certain of the local sequence register – what to do?
- No clear consensus, but I suggest assigning residue code as “UNK” and numbering to “best guess” value. A more granular way to quantify/convey uncertainty would be helpful!
- Sidechain placement – two main camps – trim sidechains to density vs place them all (+/- zero occ.). The former may sound more conservative, but it can hide errors during validation (during analysis of clashes).
- Either is acceptable, just be consistent, and preferably document the approach taken when writing up the structure. You can read long discussions of the merits of both approaches on the CCP4bb should you so desire 🙃

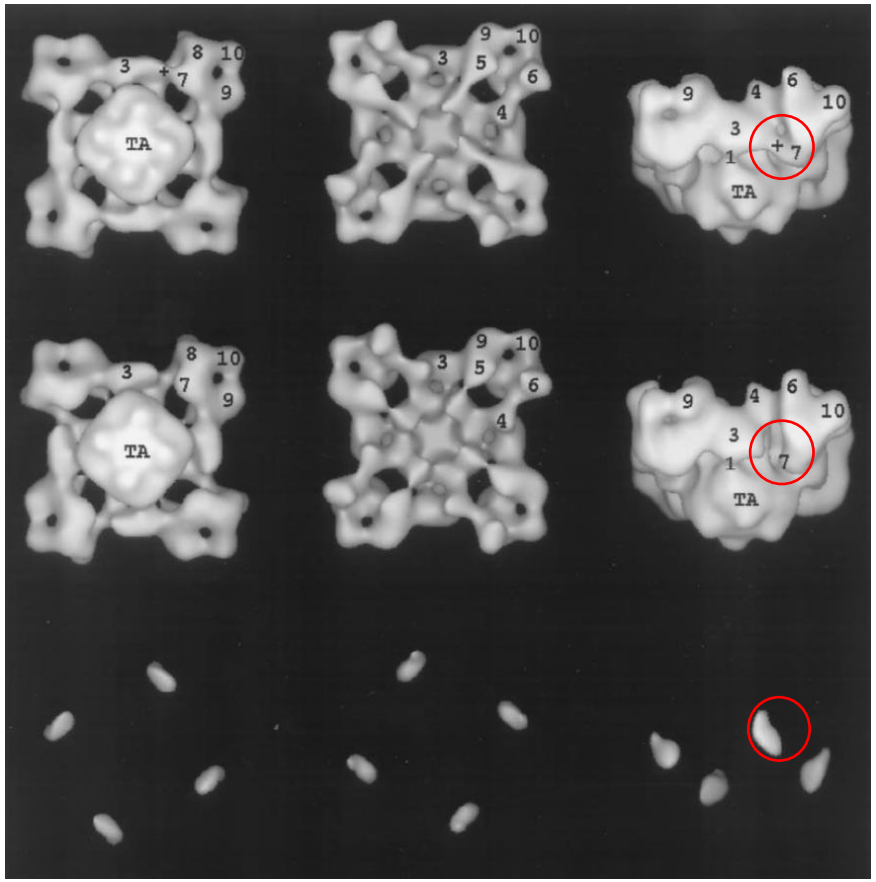
Identifying density features (labels & diff maps)

- Difference maps useful even at very low resolution (37Å!)
- Can verify binding of protein (apo vs holo), position of epitope or motif (using Fab, tag insertion)
- At higher resolution, can use to verify binding sites of ligands & ions



Identifying density features (labels & diff maps)

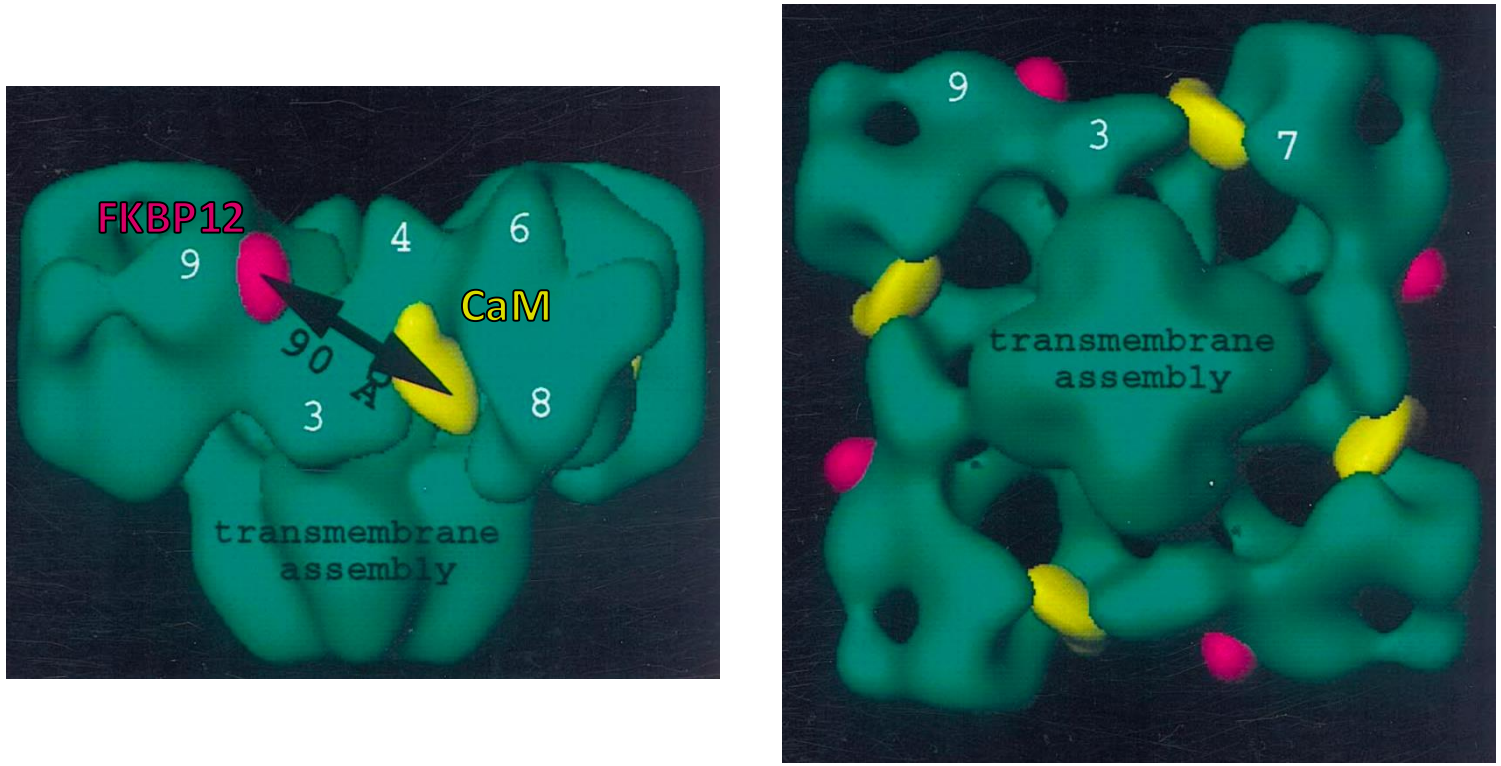
- Difference maps useful even at very low resolution (37Å!)
- Can verify binding of protein (apo vs holo), position of epitope or motif (using Fab, tag insertion)
- At higher resolution, can use to verify binding sites of ligands & ions



RyR1-CaM difference map

(Wagenknecht et al, JBC, 1997)

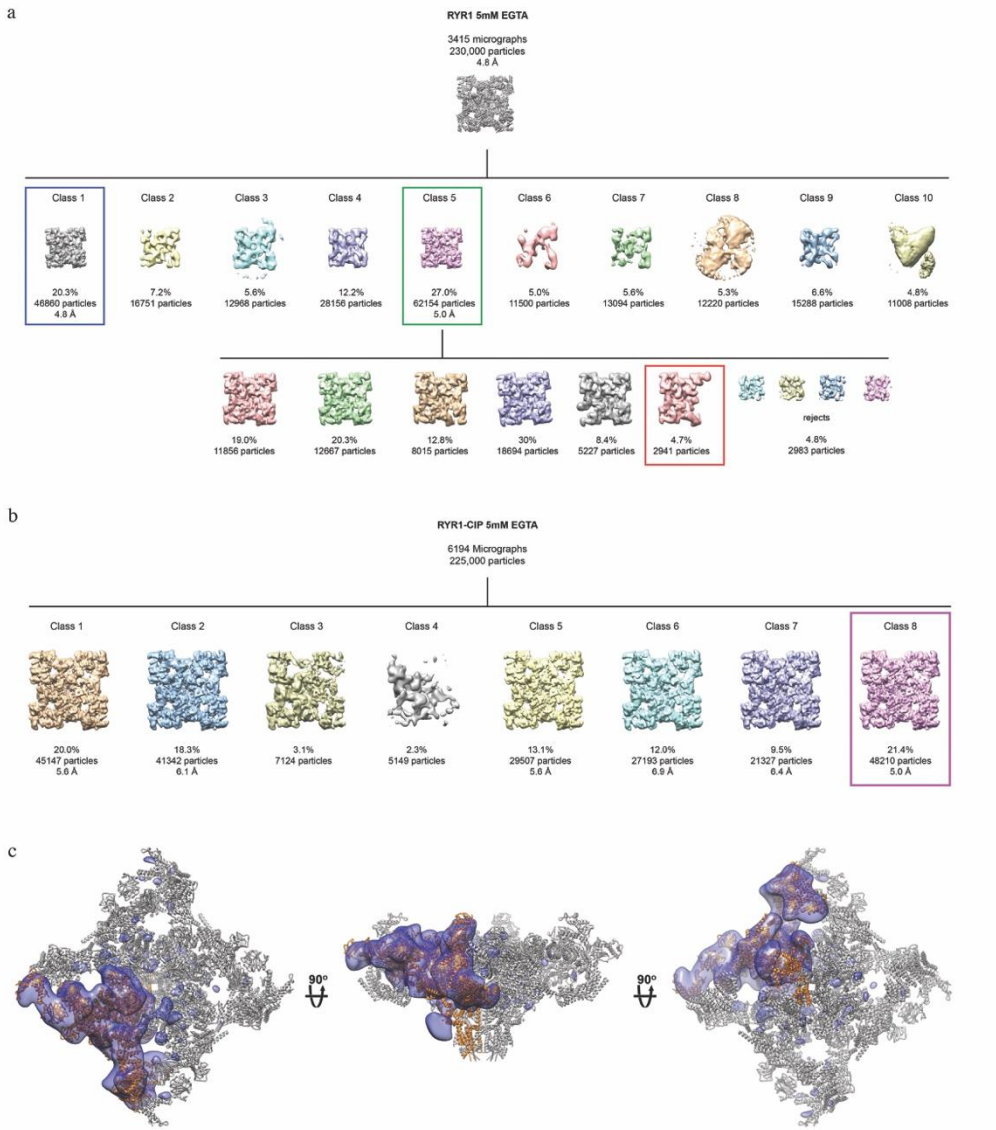
Identifying density features (labels & diff maps)



RyR1-CaM & RyR1-FKBP12 difference maps

(Wagenknecht et al, JBC, 1997)

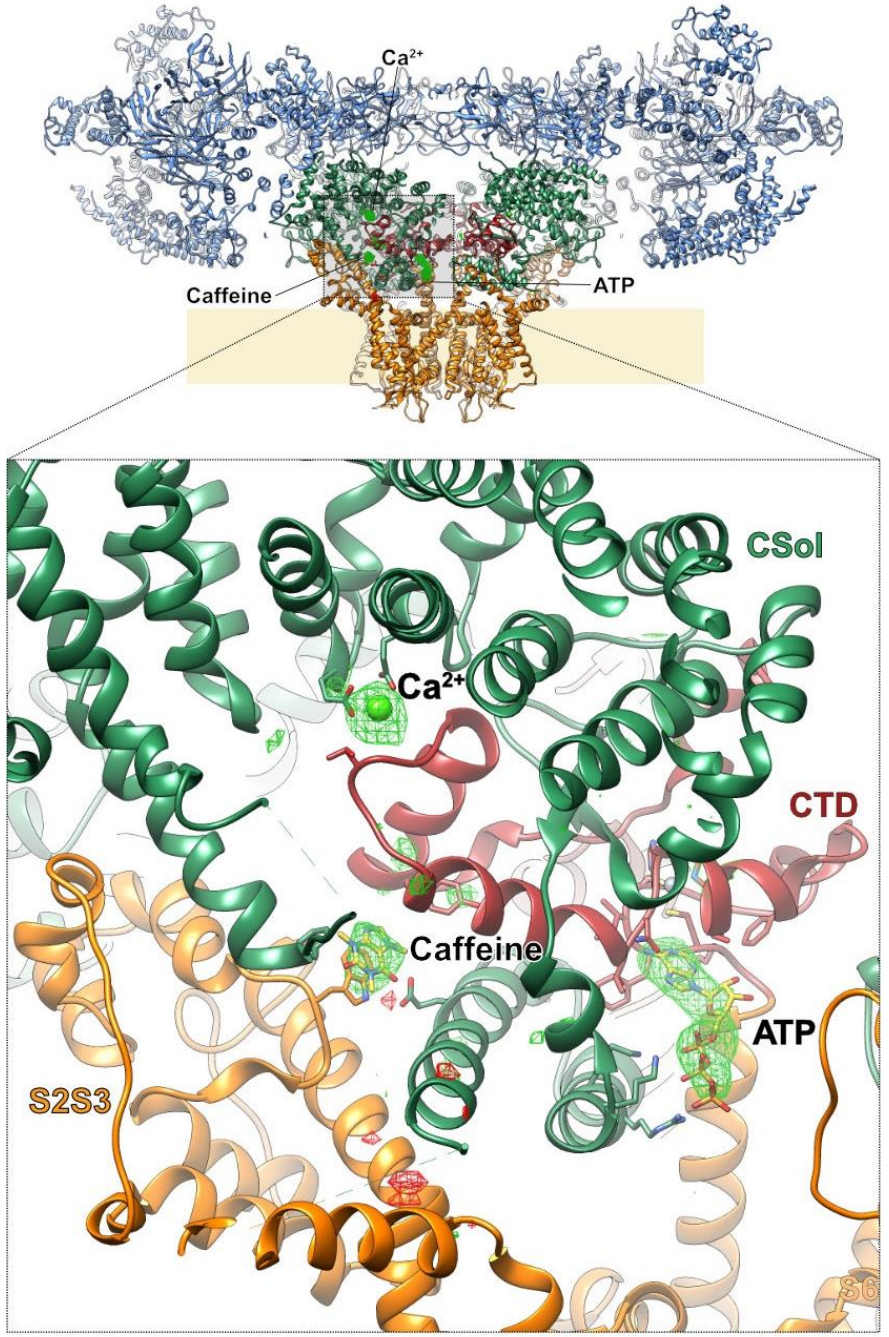
Prior knowledge can come in many forms – use any and all available info to guide model building.



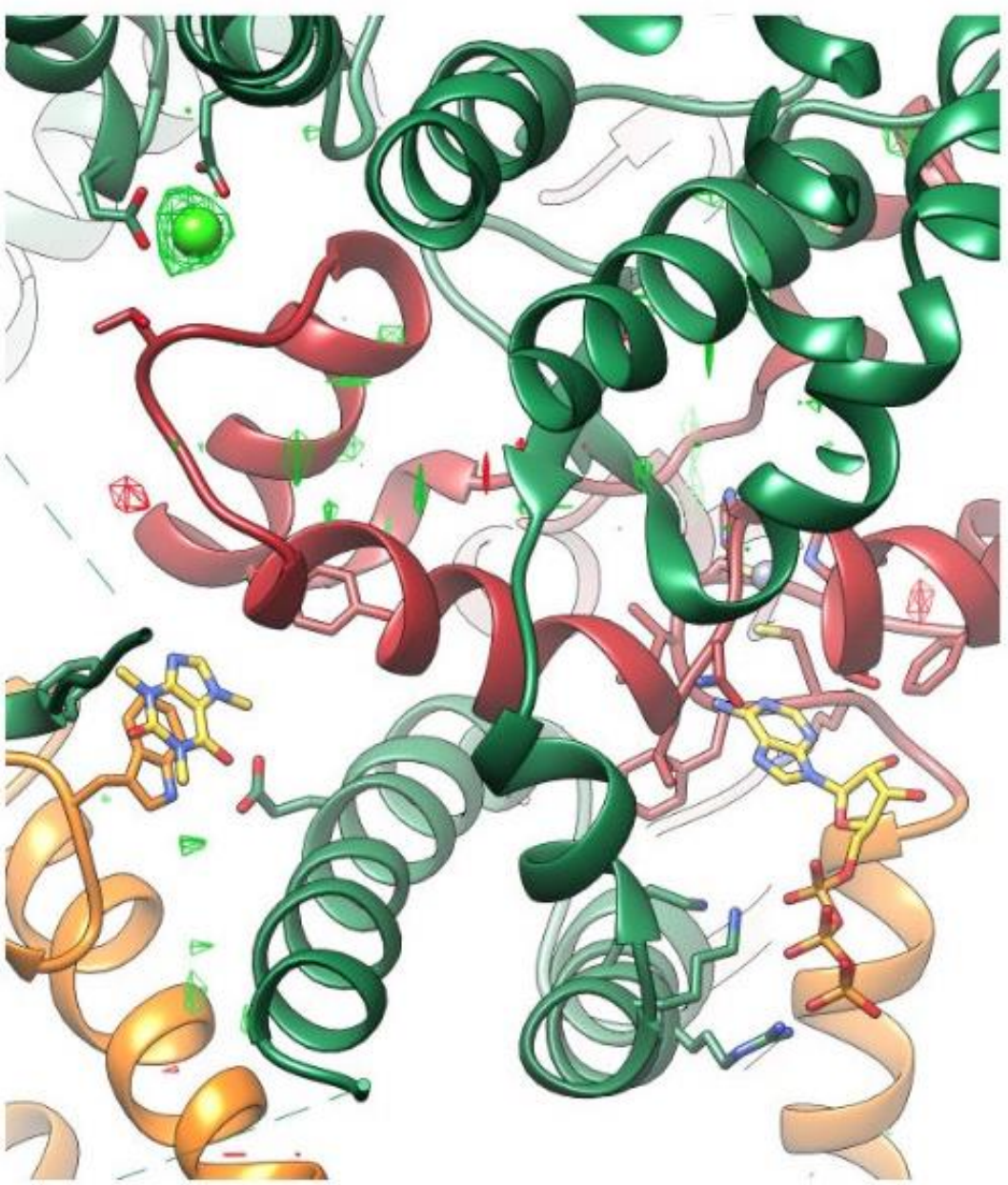
Here, serendipitous identification of a conformational class of RyR1 lacking density for one subunit aided identification of protomer boundaries. In other cases, cross-linking data or NS data on subcomplexes or Fab-complexes may be helpful.

(Zalk et al, Nature 2015)

In a similar manner, we can use locally aligned difference maps between holo and apo structures to locate ligands.

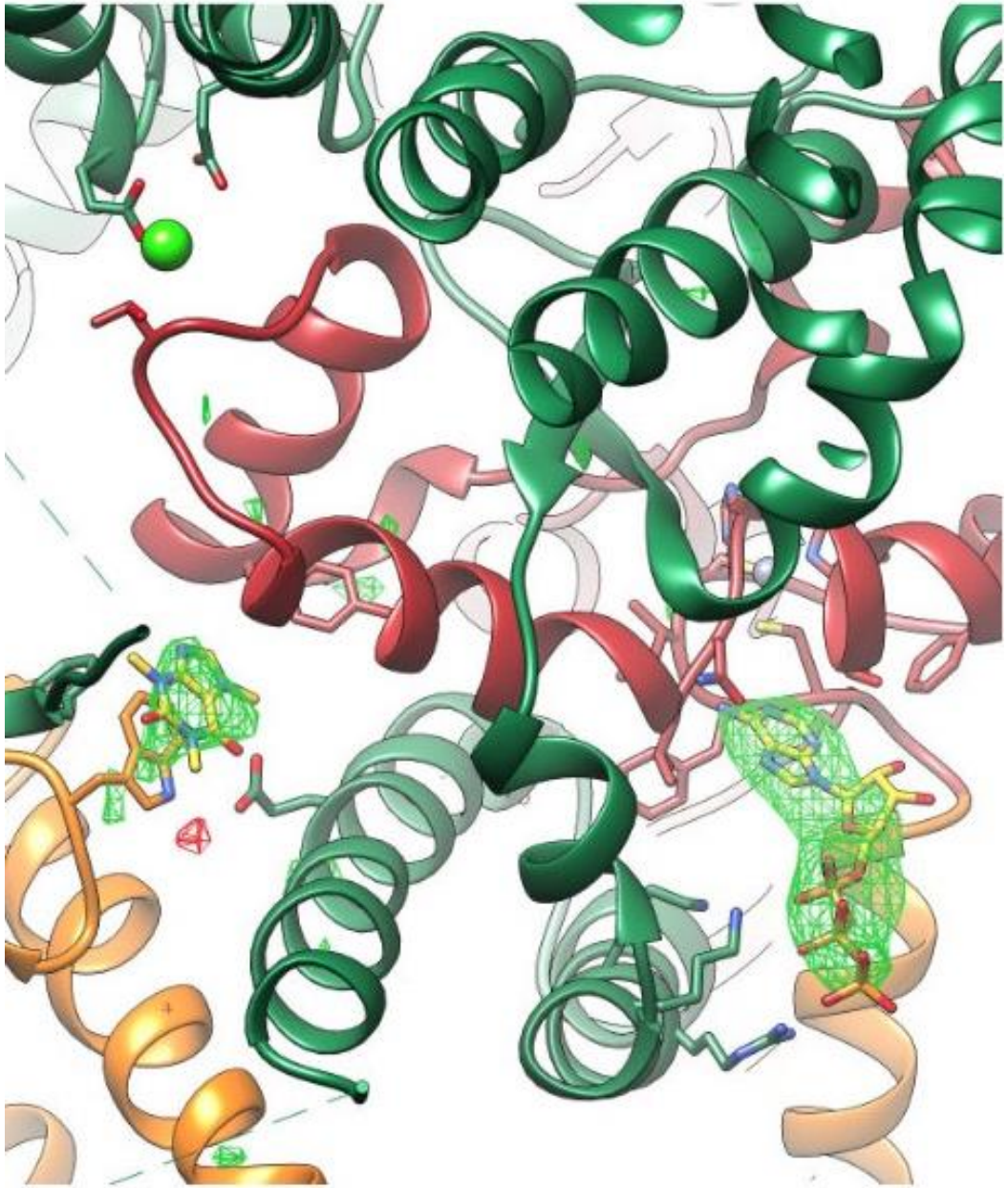


The three ligands are clustered around the C-terminal domain.



(Ca^{2+} only) minus (EGTA only)

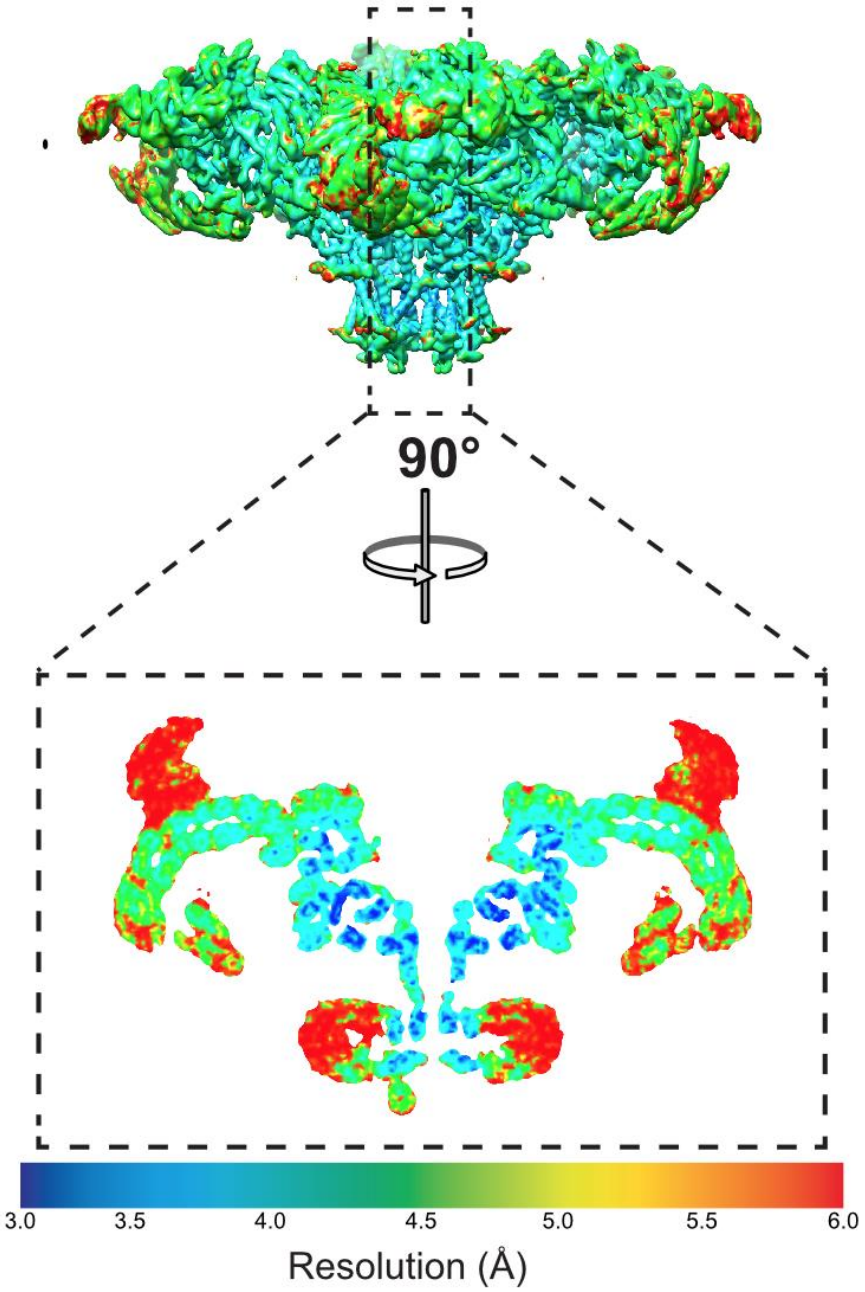
The three ligands are clustered around the C-terminal domain.



(ATP/Caffeine) minus (EGTA only)

EM-specific considerations

- No unambiguous sequence/elemental markers at low resolution (no equivalent of SeMet yet).
- No feedback from phase improvement, but also no model bias – WYSIWIG.
- Often substantial variation in local resolution – different strategies and levels of detail required for different regions. Map sharpening essential.
- ”Medium” resolution (4-6Å) much more common than for crystallography.
- Often have more than one map, with different composition or conformation (may be convenient to combine focused refinements in Chimera by taking max value at each voxel after alignment, e.g.: *`vop maximum #1,2 ongrid #1`*)

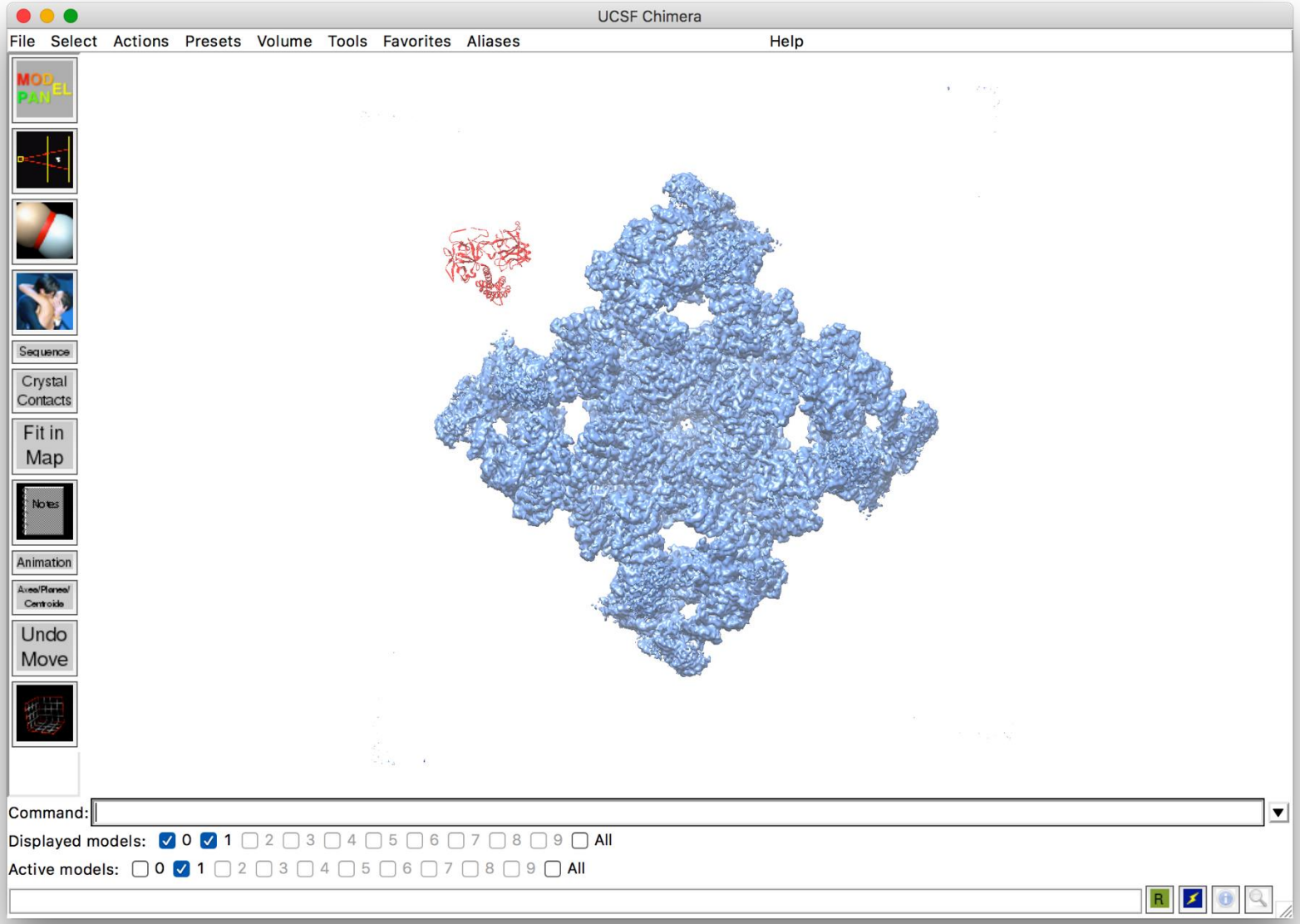


Building an initial model - where to start?

- If you have a crystal structure, of a fragment or a homology model of a domain, place it, and extend into density. (***Now, Alphafold & Rosettafold mean this is almost always the case****)*

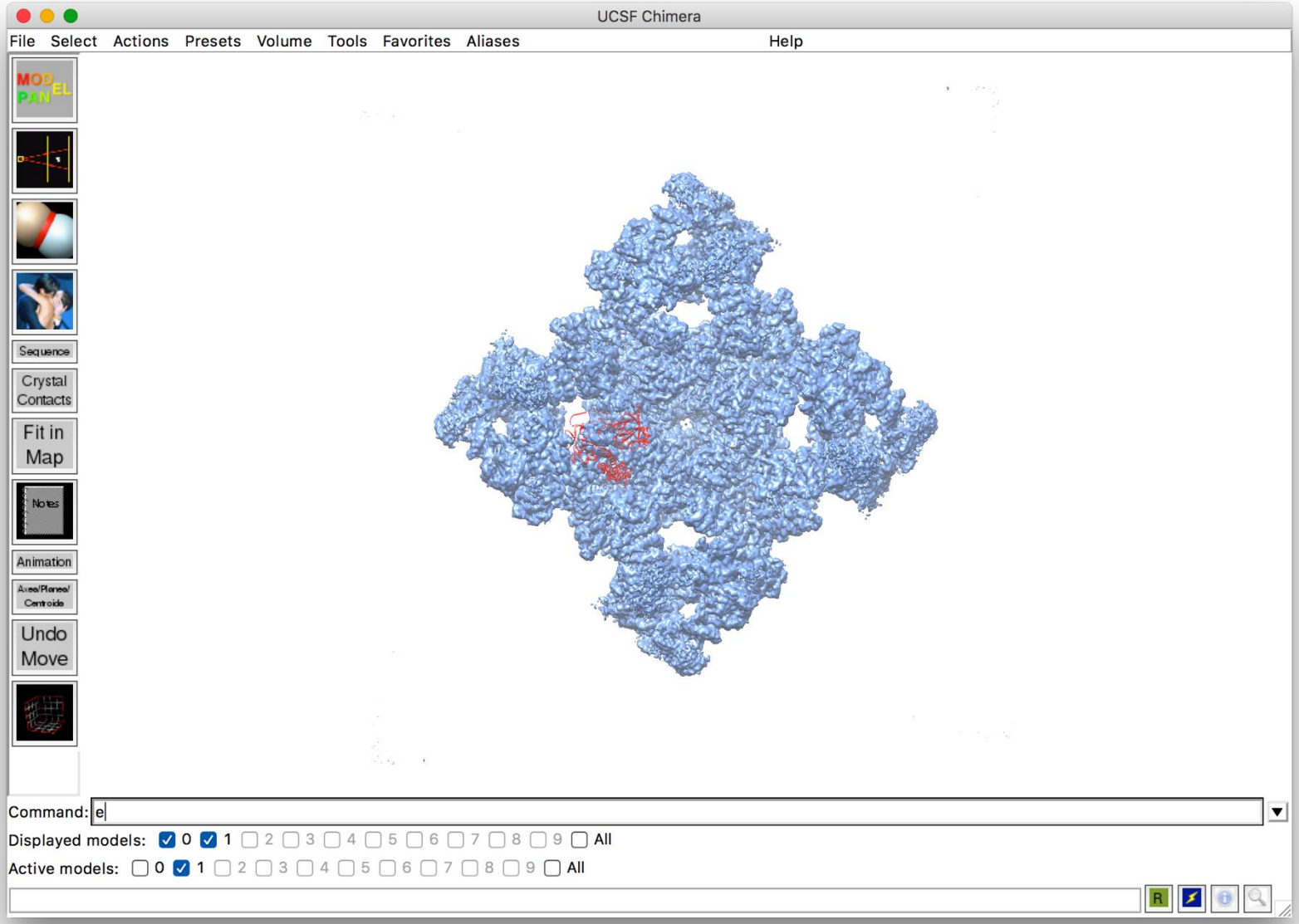
- Otherwise, identify structurally distinctive motifs in the sequence – for example, a strongly predicted helix with three aromatic residues near the N-term end – and identify candidate locations in the density map. Extend and see if hypothesis still holds.

Using UCSF Chimera to fit solved domains



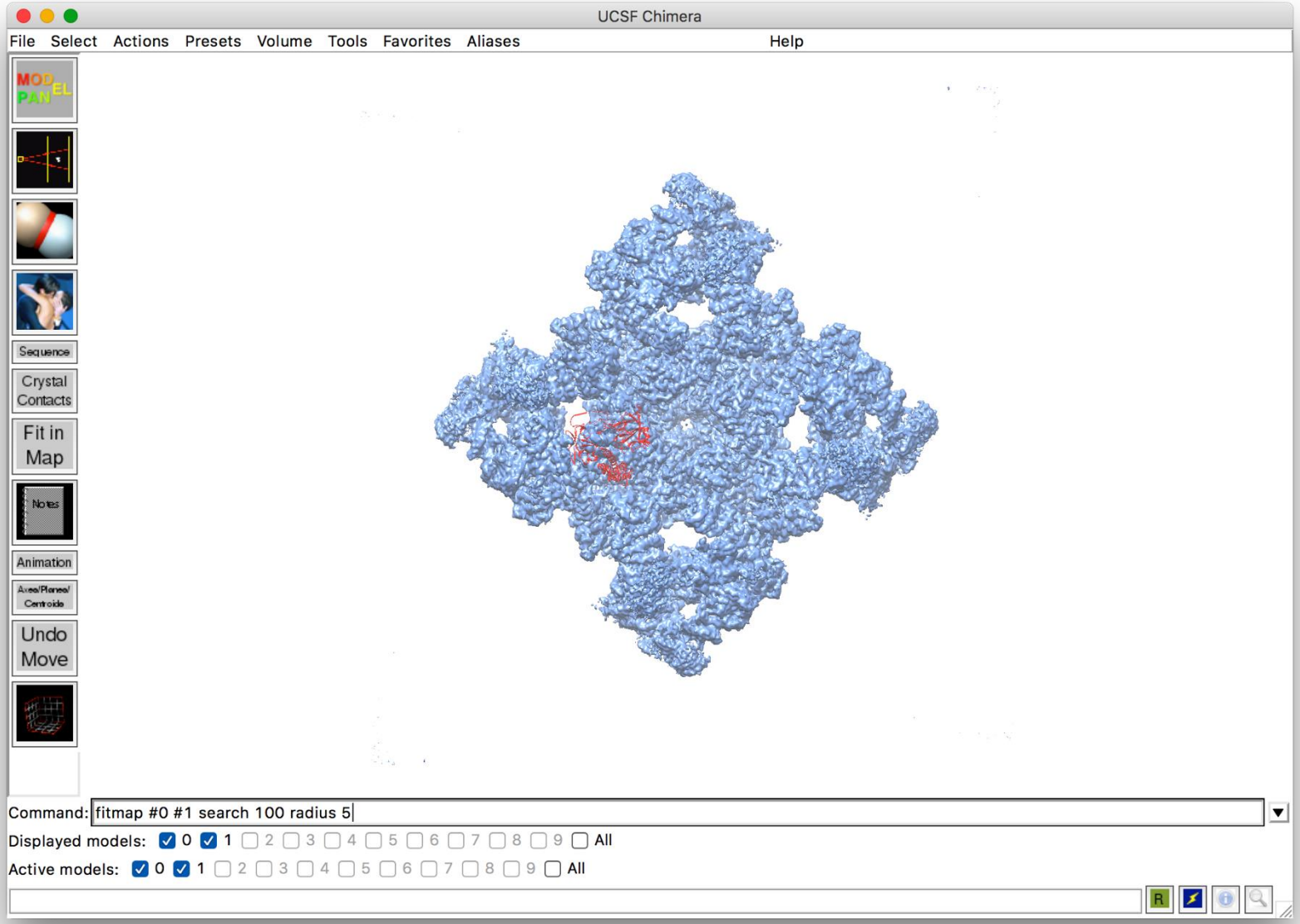
Start with map and model.

Using UCSF Chimera to fit solved domains



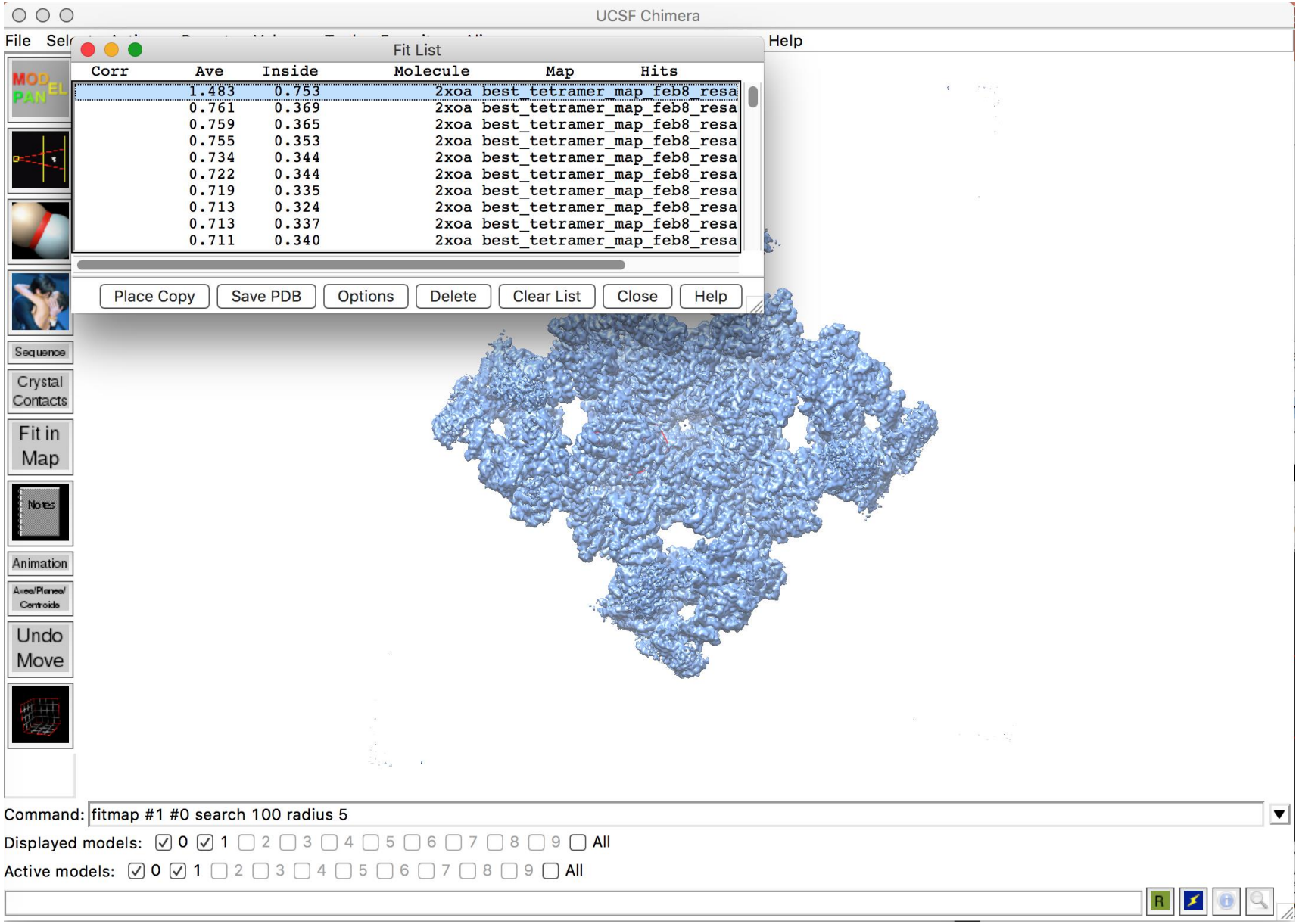
Move model to approximate position (if known, to save computation)

Using UCSF Chimera to fit solved domains



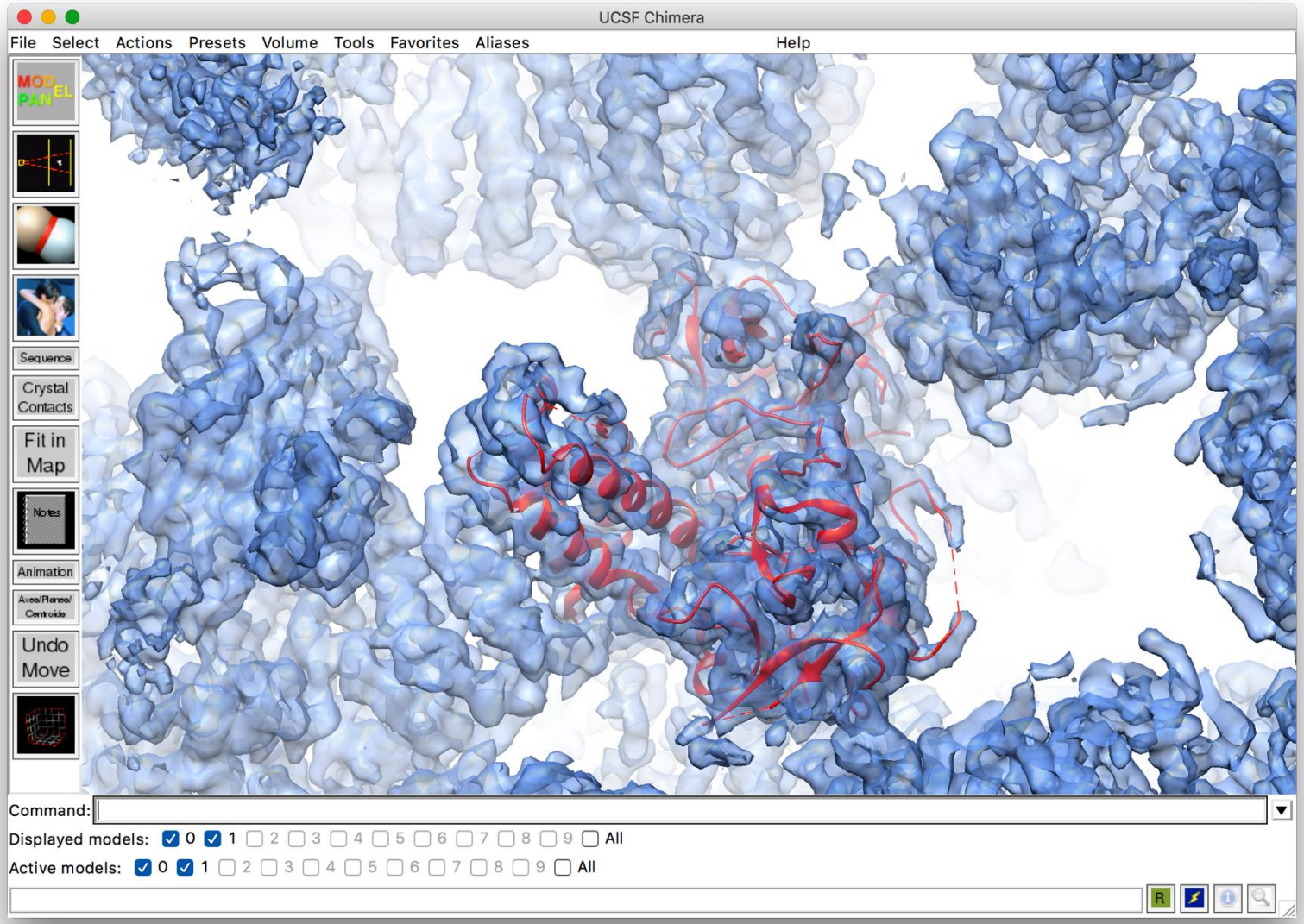
Run fitmap with ‘search’ (here 100 orientations) and ’radius’ (here 5 Å)

Using UCSF Chimera to fit solved domains



Chimera will return a list of candidate orientations, ranked by agreement with the map. Hopefully there will be a clear separation between the correct and incorrect solutions.

Using UCSF Chimera to fit solved domains



Chimera will return a list of candidate orientations, ranked by agreement with the map. Hopefully there will be a clear separation between the correct and incorrect

Voxel size calibration

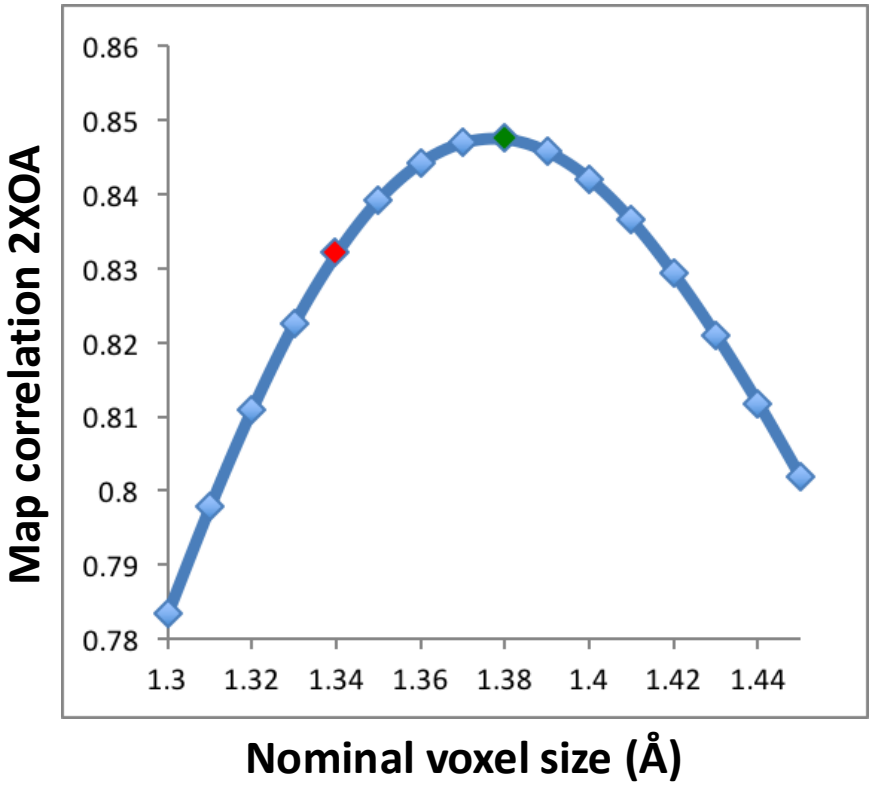
- Voxel size generally requires calibration against a crystal structure or some other source of ground truth.

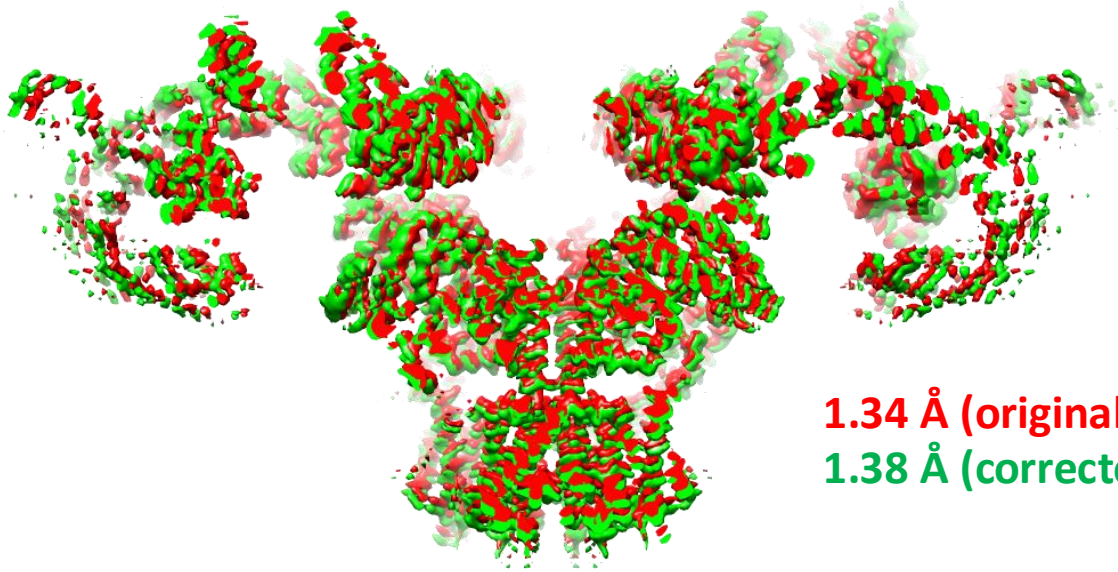
- Once calibrated, generally stable between samples/datasets at same magnification.

- Can calibrate by fitting in Chimera at range of nominal voxel sizes and measuring correlation. **Can also do this automatically with *phenix.magref!***

- Incorrect voxel sizes are common in deposited maps - **be aware of this when comparing structures**. E.g. here there is a 3% difference – affects structural alignment, reported resolution (3.8 vs 3.9Å).

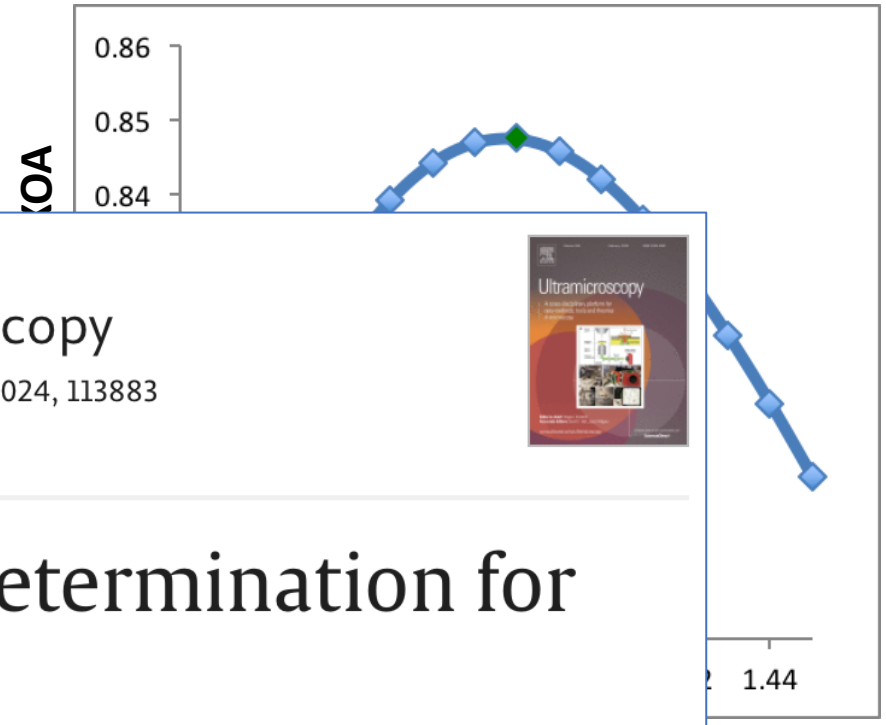
- Alternative recent approach if using gold grids – use MagCalEM, which fits pixel size based on position of gold diffraction peaks.





Voxel size calibration

- Voxel size generally requires calibration against a crystal structure or some other source of ground truth.
- Once calibrated, samples/dataset
- Can calibrate by nominal voxel si
- Incorrect voxel s maps - **be aware structures**. E.g. affects structura resolution (3.8 v
- Alternative recent approach if using gold grids – use MagCalEM, which fits pixel size based on position of gold diffraction peaks.



ELSEVIER

Ultramicroscopy

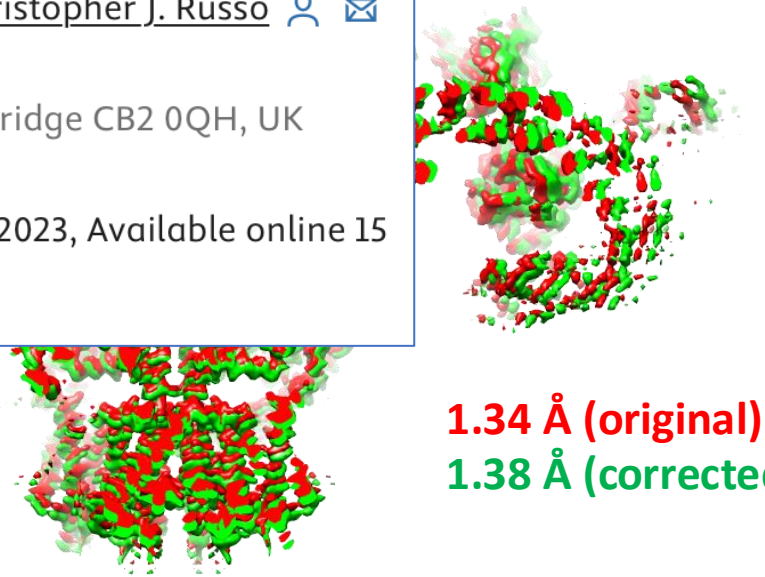
Volume 256, February 2024, 113883

Accurate magnification determination for cryoEM using gold

Joshua L. Dickerson, Erin Leahy, Mathew J. Peet, Katerina Naydenova, Christopher J. Russo

MRC Laboratory of Molecular Biology, Francis Crick Avenue, Cambridge CB2 0QH, UK

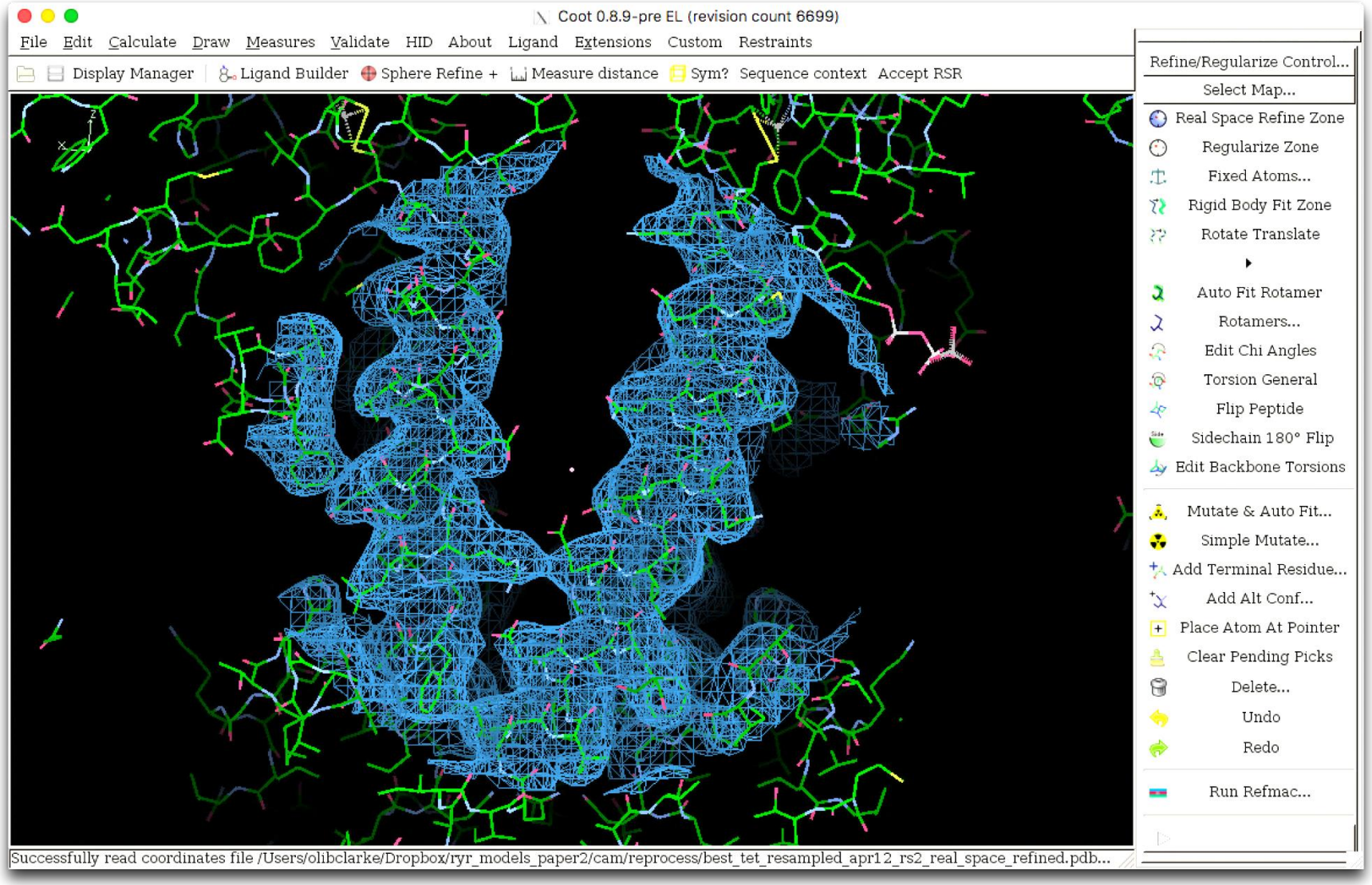
Received 17 August 2023, Revised 27 October 2023, Accepted 7 November 2023, Available online 15 November 2023, Version of Record 25 November 2023.



1.34 Å (original)
1.38 Å (corrected)

COOT – Crystallographic Object Oriented Toolkit

- Simple, intuitive interface for building and manipulating atomic models in density maps.
- Low computational requirements
- Extensive API – easy to script or modify (using simple Python code)
- On-the-fly sharpening, resampling and low pass filtering

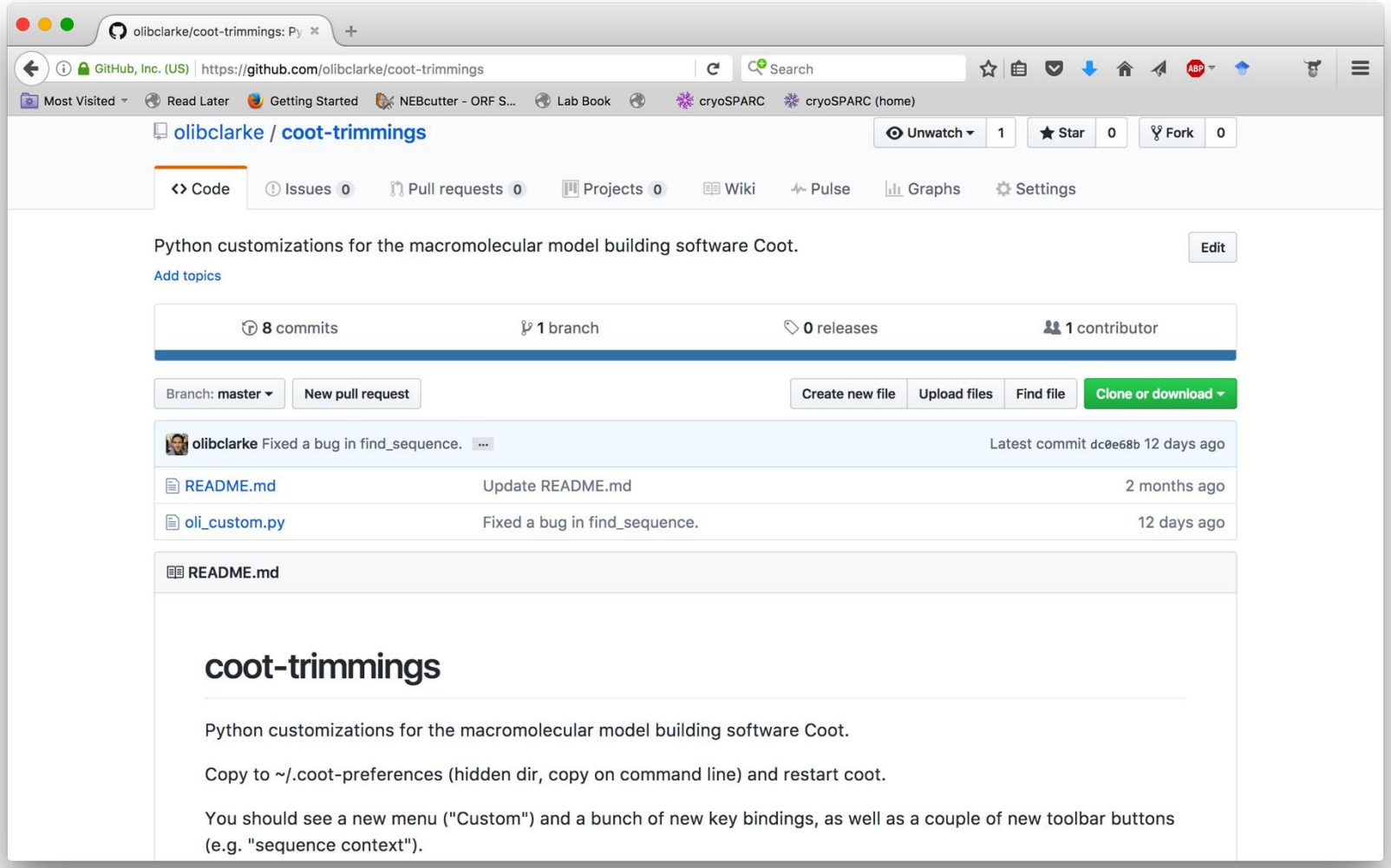


(Try the latest nightly with new features for EM, improved RSR: http://www.ccpem.ac.uk/download.php)

(*Emsley P. 2004, Acta Cryst. D; Casañal A. et al. 2020, Protein Science*)

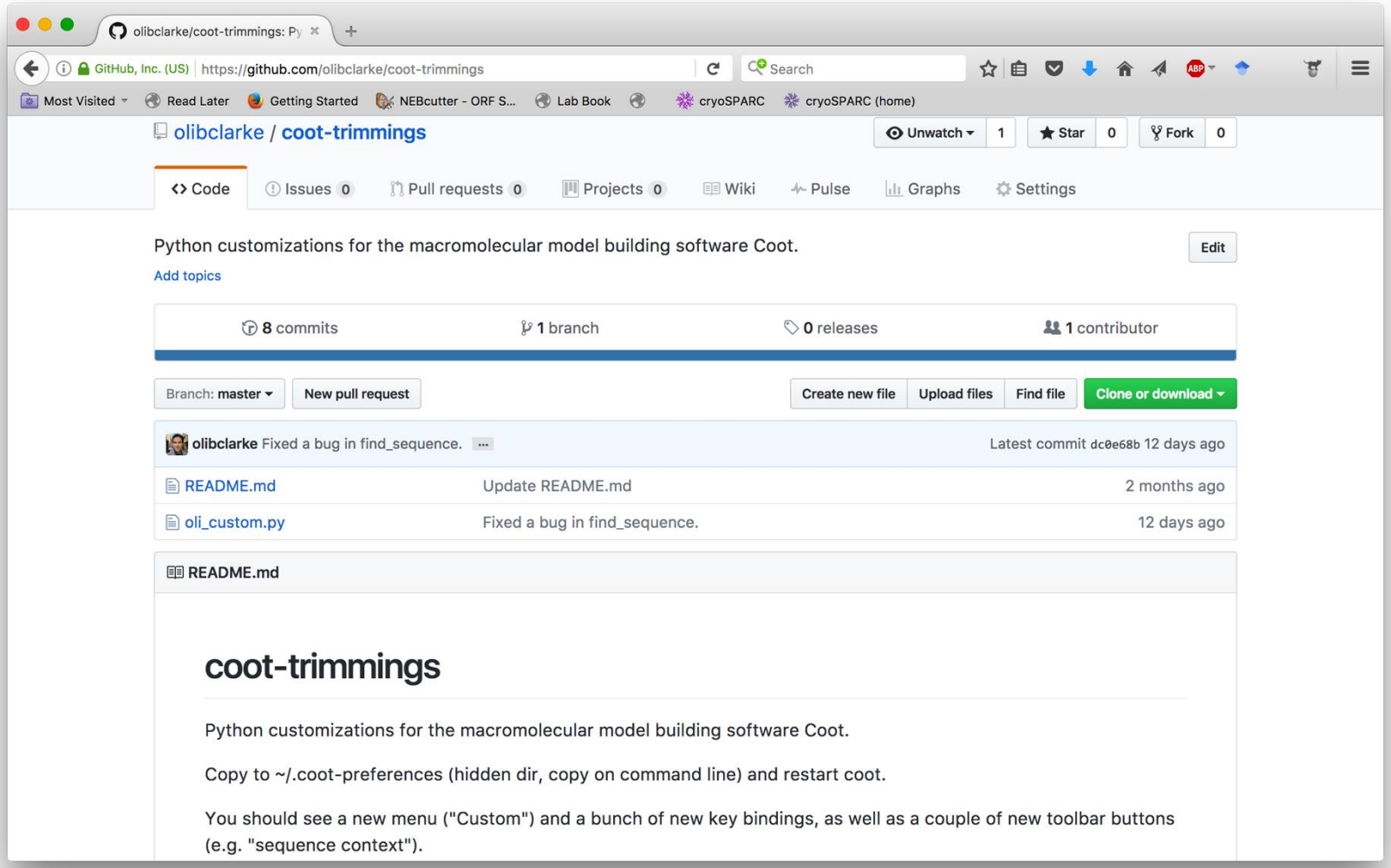
COOT – Crystallographic Object Oriented Toolkit

- Simple, intuitive interface for building and manipulating atomic models in density maps.
- Low computational requirements
- **Extensive API – easy to script or modify (using simple Python code)**
- On-the-fly sharpening, resampling and low pass filtering



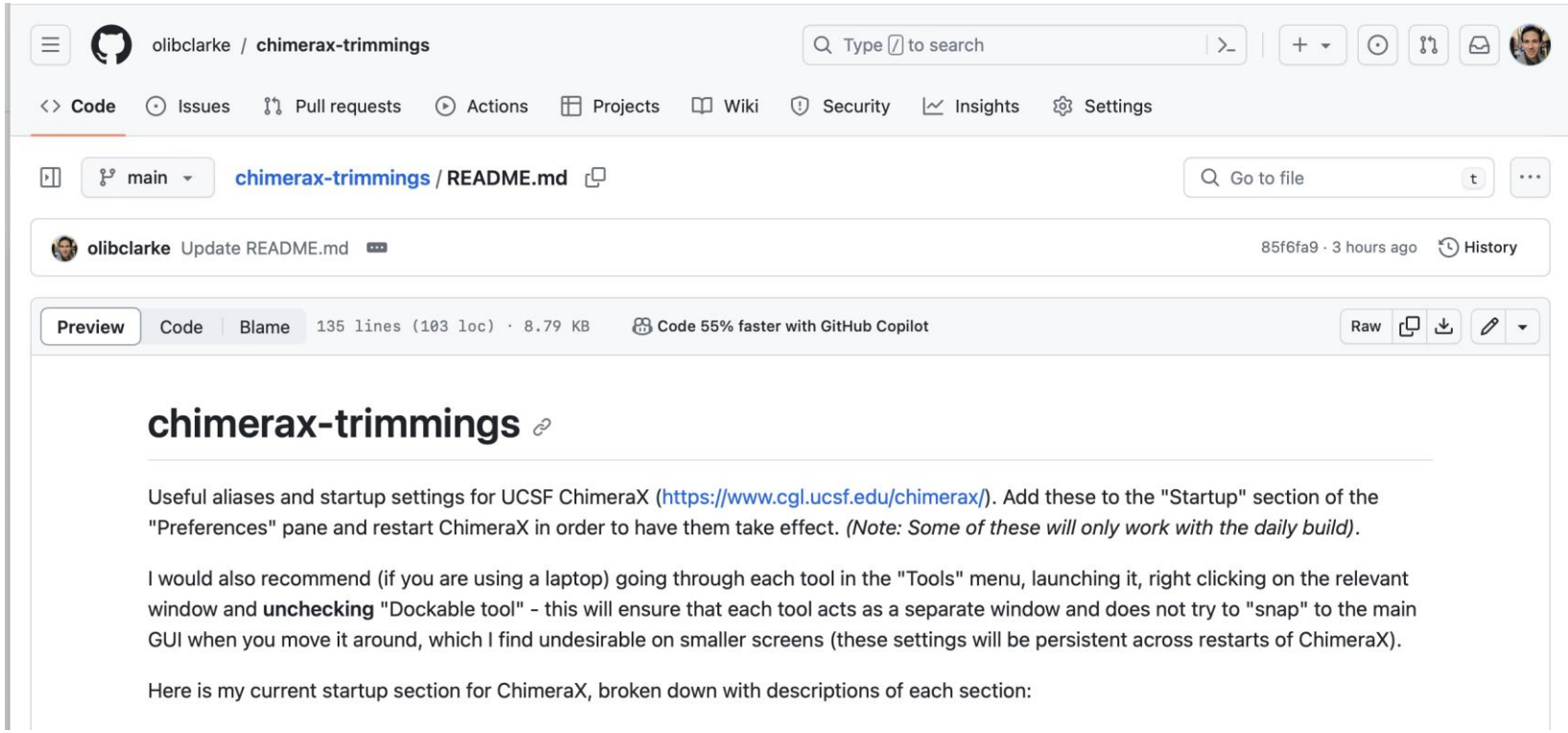
COOT – Crystallographic Object Oriented Toolkit

- Simple, intuitive interface for building and manipulating atomic models in density maps.
- Low computational requirements
- **Extensive API – easy to script or modify (using simple Python code)**
- On-the-fly sharpening, resampling and low pass filtering

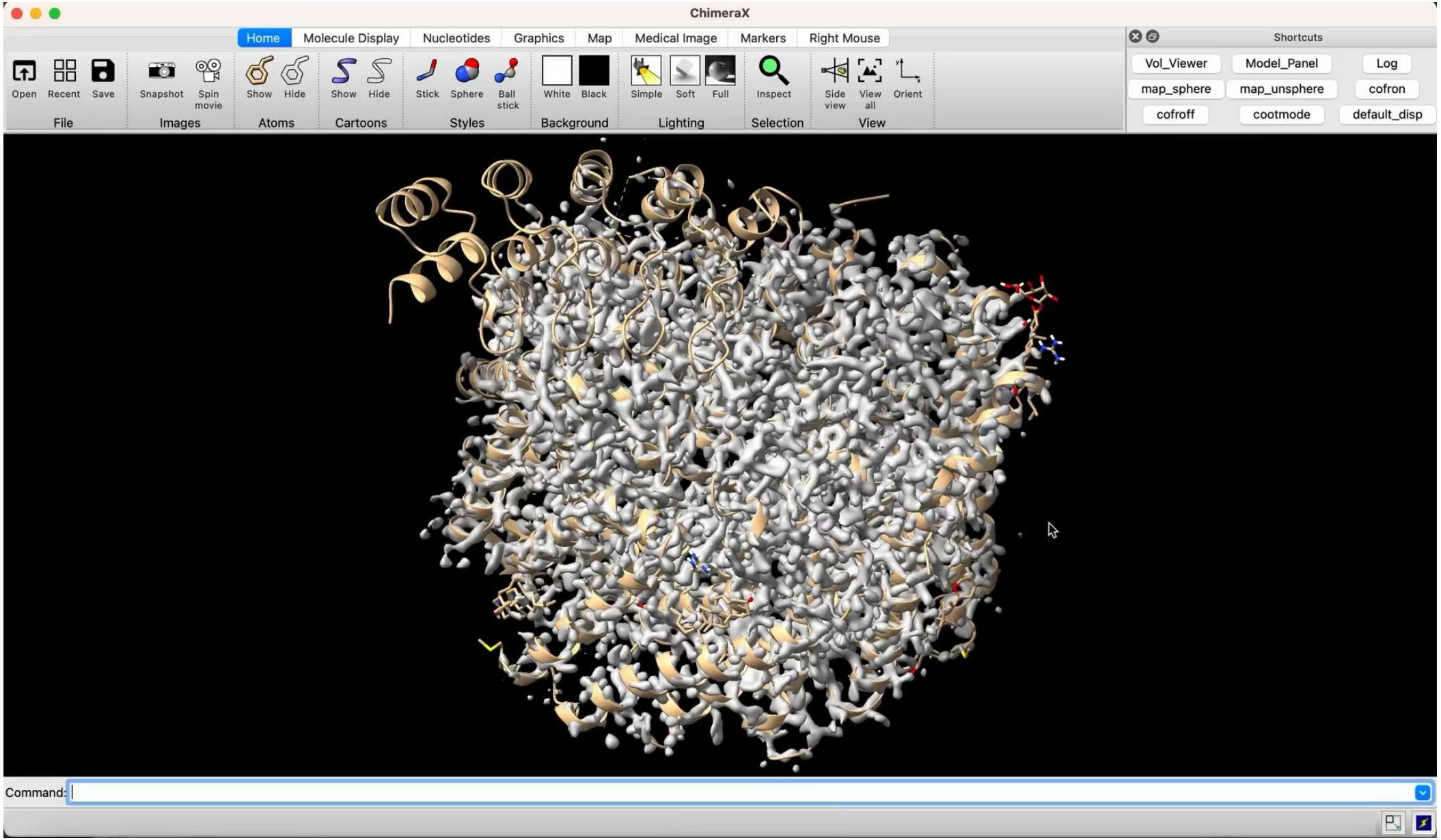


Sidenote – trimmings for ChimeraX:

- Added some shortcuts and a bunch of aliases that I find helpful for using ChimeraX with maps/models.
- Use and modify as you like!



Sidenote – trimmings for ChimeraX:



COOT – Crystallographic Object Oriented Toolkit

- Simple, intuitive interface for building and manipulating atomic models in density maps.
- Low computational requirements
- **Extensive API – easy to script or modify (using simple Python code)**
- On-the-fly sharpening, resampling and low pass filtering

```
def mutate_by_entered_code():
  def mutate_single_letter(X):
    entry=str(X).upper()
    mol_id=active_residue()[0]
    ch_id=active_residue()[1]
    resno=active_residue()[2]
    ins_code=active_residue()[3]
    resname=residue_name(mol_id,ch_id,resno,ins_code)
    map_id=imol_refinement_map()
    aa_dic={'A':'ALA','R':'ARG','N':'ASN','D':'ASP','C':'CYS','E':'GLU','Q':'GLN','G':'GLY','H':'HIS','I':'ILE','L':'LEU','K':'LY
    nt_list=['A','C','T','G','U']
    if (resname in aa_dic.values()) and (aa_dic.get(entry,0)!=0):
      mutate(mol_id,ch_id,resno,ins_code,aa_dic.get(entry,0))
    elif (resname in nt_list) and (entry in nt_list):
      mutate_base(mol_id,ch_id,resno,ins_code,entry)
    else:
      info_dialog("Invalid target residue! Must be protein or nucleic acid, and entered code must be single letter.")
  generic_single_entry("New residue? (single letter code)","A","Mutate by single-letter code",mutate_single_letter)
```

```
#mutate active residue to entered residue code (upper or lower case single-letter)
add_key_binding("Mutate by single letter code","M",
lambda: mutate_by_entered_code())
```

COOT – Crystallographic Object Oriented Toolkit

- Simple, intuitive interface for building and manipulating atomic models in density maps.
- Low computational requirements
- **Extensive API – easy to script or modify (using simple Python code)**
- On-the-fly sharpening, resampling and low pass filtering

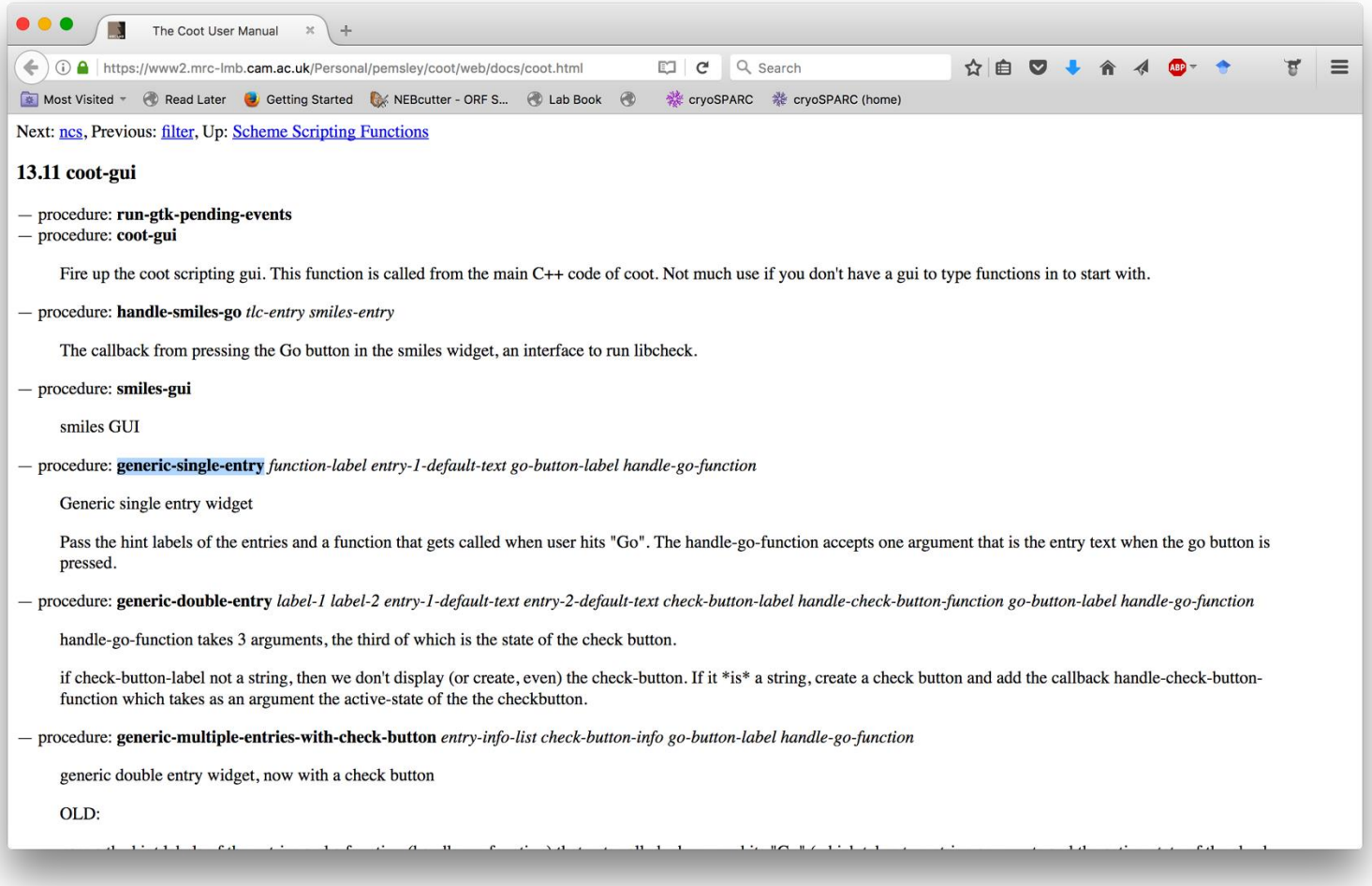
```
def mutate_by_entered_code():
  def mutate_single_letter(X):
    entry=str(X).upper()
    mol_id=active_residue()[0]
    ch_id=active_residue()[1]
    resno=active_residue()[2]
    ins_code=active_residue()[3]
    resname=residue_name(mol_id,ch_id,resno,ins_code)
    map_id=imol_refinement_map()
    aa_dic={'A':'ALA','R':'ARG','N':'ASN','D':'ASP','C':'CYS','E':'GLU','Q':'GLN','G':'GLY','H':'HIS','I':'ILE','L':'LEU','K':'LY
    nt_list=['A','C','T','G','U']
    if (resname in aa_dic.values()) and (aa_dic.get(entry,0)!=0):
      mutate(mol_id,ch_id,resno,ins_code,aa_dic.get(entry,0))
    elif (resname in nt_list) and (entry in nt_list):
      mutate_base(mol_id,ch_id,resno,ins_code,entry)
    else:
      info_dialog("Invalid target residue! Must be protein or nucleic acid, and entered code must be single letter.")
  generic_single_entry("New residue? (single letter code)","A","Mutate by single-letter code",mutate_single_letter)
```

```
#mutate active residue to entered residue code (upper or lower case single-letter)
add_key_binding("Mutate by single letter code","M",
lambda: mutate_by_entered_code())
```

Many pre-packaged functions available in COOT API. Mostly documented in online manual. Very easy to write your own! Useful e.g. for scripting domain-wise rigid body refinement.

COOT – Crystallographic Object Oriented Toolkit

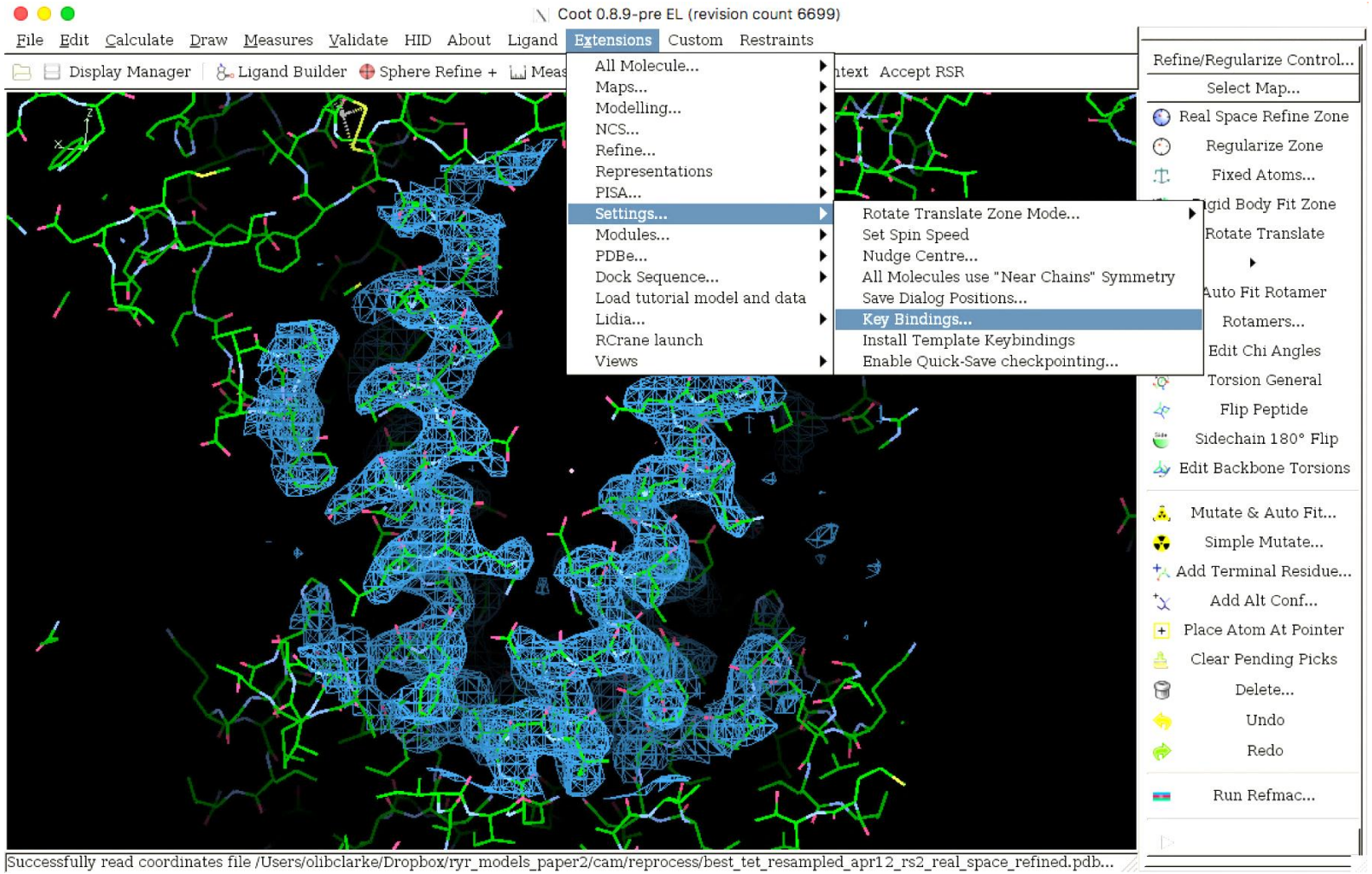
- Simple, intuitive interface for building and manipulating atomic models in density maps.
- Low computational requirements
- **Extensive API – easy to script or modify (using simple Python code)**
- On-the-fly sharpening, resampling and low pass filtering



Many pre-packaged functions available in COOT API. Mostly documented in online manual.

COOT – Crystallographic Object Oriented Toolkit

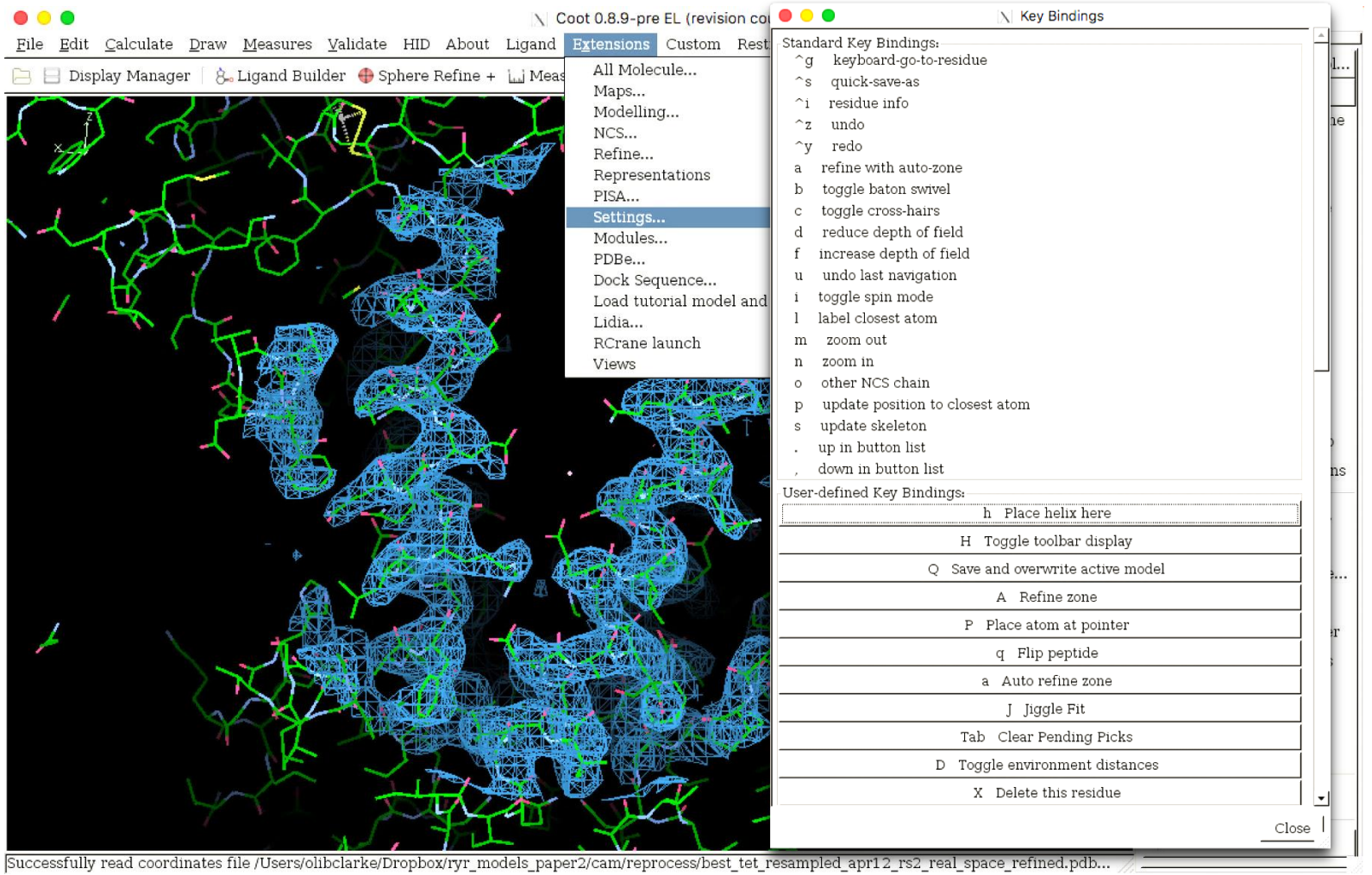
- Simple, intuitive interface for building and manipulating atomic models in density maps.
- Low computational requirements
- **Extensive API – easy to script or modify (using simple Python code)**
- On-the-fly sharpening, resampling and low pass filtering



Lots of key bindings, and easy to define custom keys. Learn them. They make everything much faster.

COOT – Crystallographic Object Oriented Toolkit

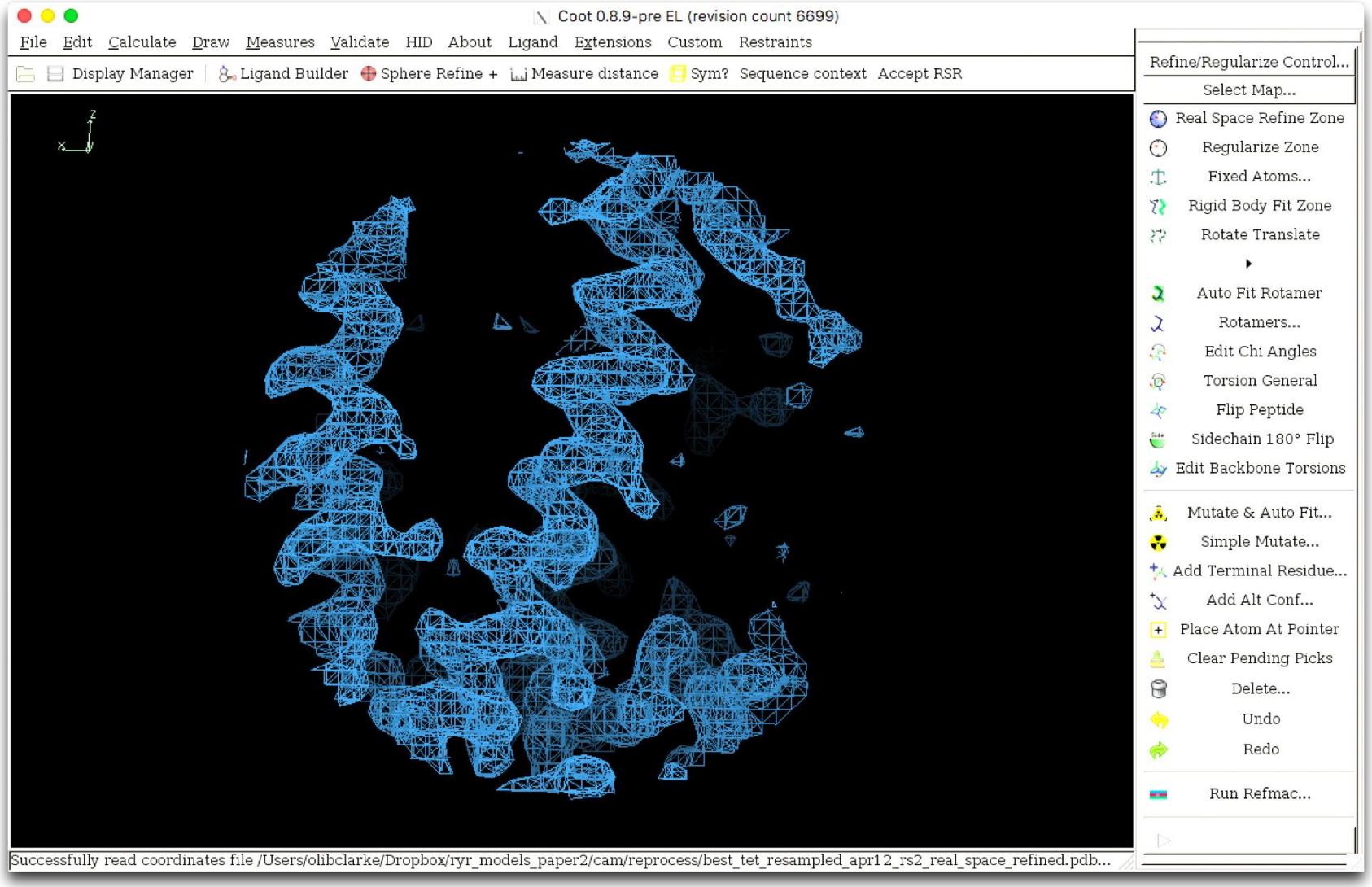
- Simple, intuitive interface for building and manipulating atomic models in density maps.
- Low computational requirements
- **Extensive API – easy to script or modify (using simple Python code)**
- On-the-fly sharpening, resampling and low pass filtering



Lots of key bindings, and easy to define custom keys. Learn them. They make everything much faster.

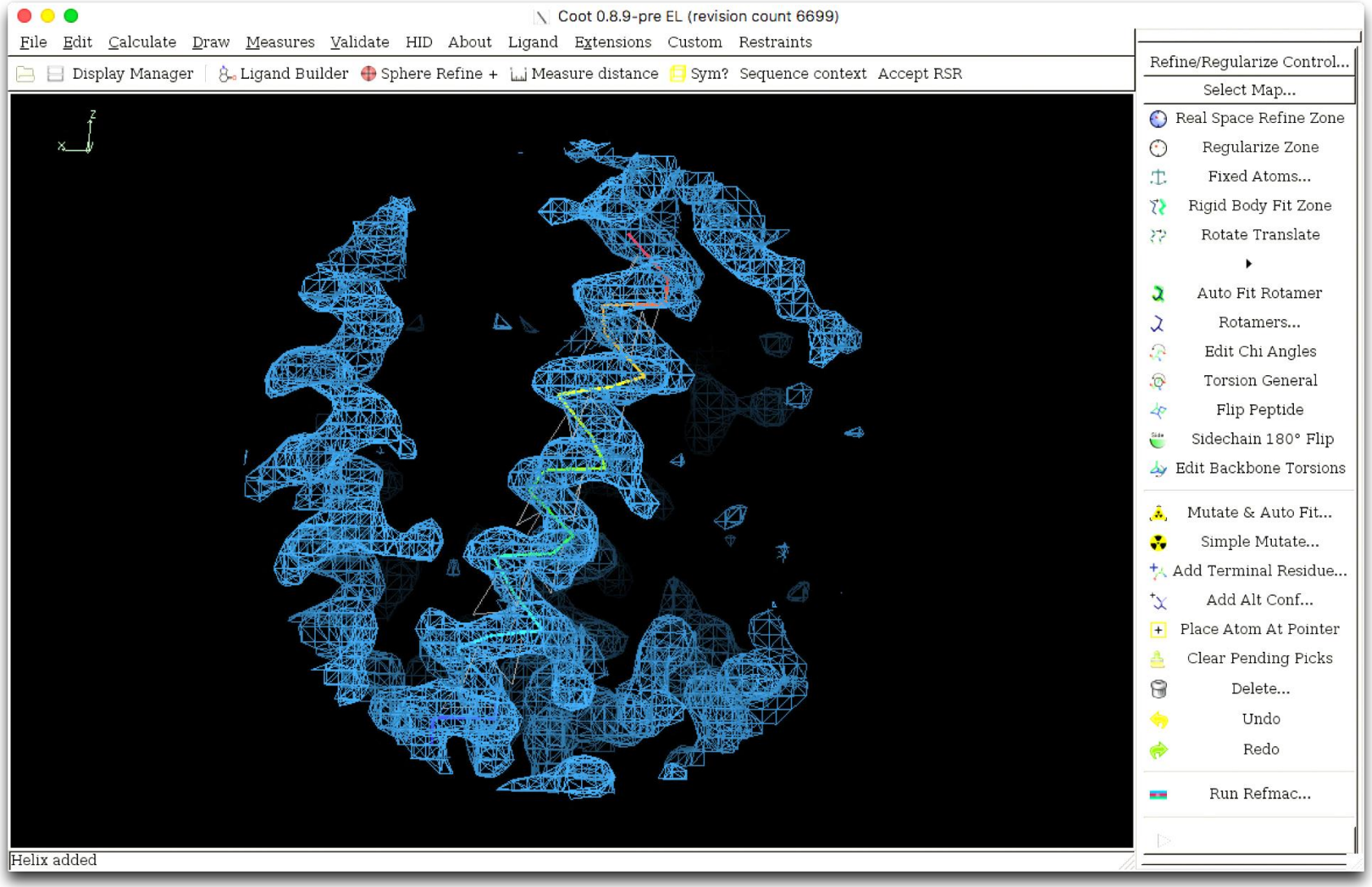
COOT – Crystallographic Object Oriented Toolkit

- Semi-automated helix placement
- Place cursor at the center of the helix and trigger ”Place helix here” (I suggest via a key binding - “h” with `coot-trimmings`)
- Coot will attempt to automatically determine the length and direction of the helix.
- Trim/extend, adjust weights, then refine using real-space refine zone. Drag into density to adjust fit.



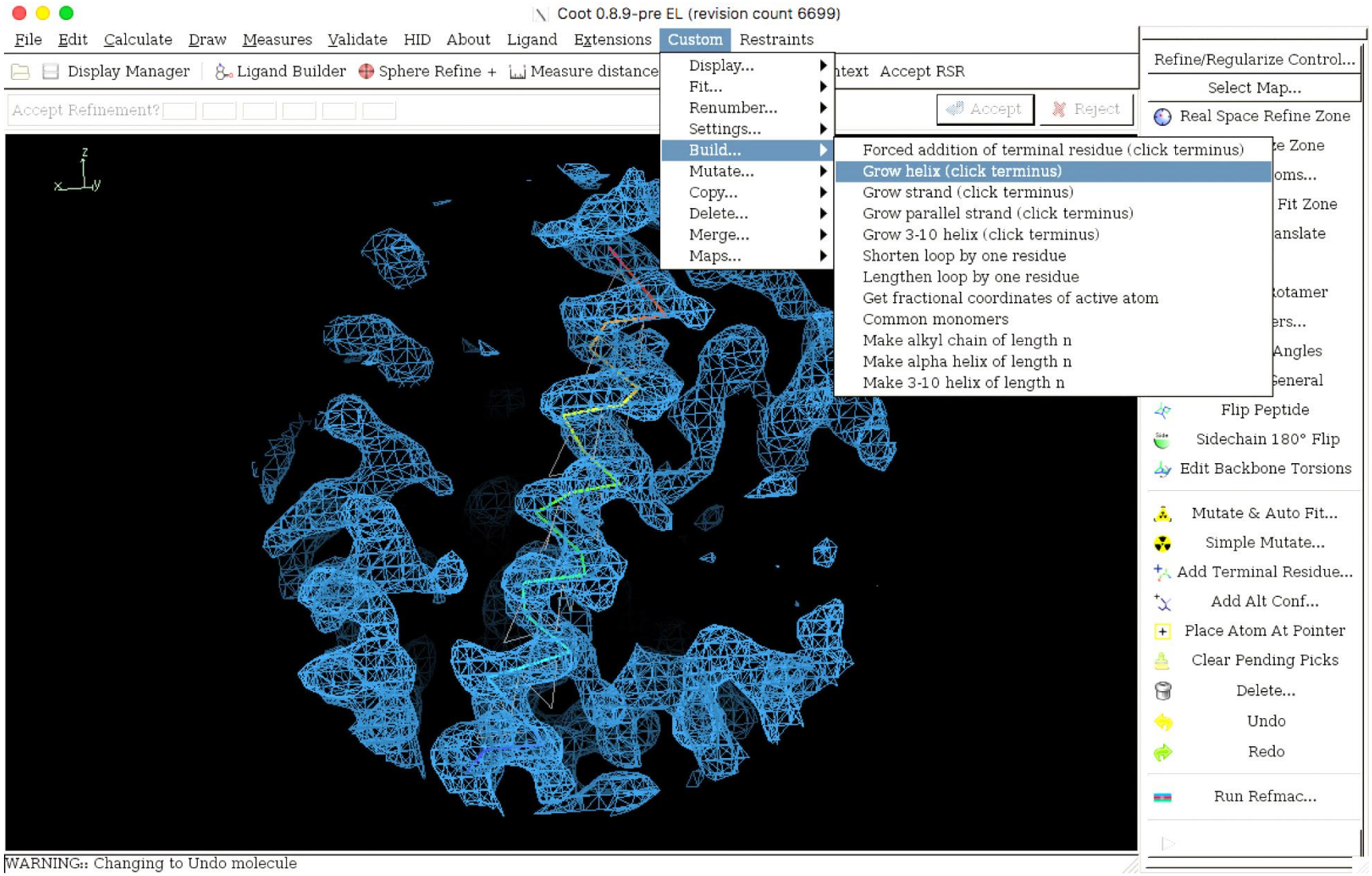
COOT – Crystallographic Object Oriented Toolkit

- Semi-automated helix placement
- **Place cursor at the center of the helix and trigger ”Place helix here” (I suggest via a key binding - “h” with *`coot-trimmings`*)**
- Coot will attempt to automatically determine the length and direction of the helix.
- Trim/extend, adjust weights, then refine using real-space refine zone. Drag into density to adjust fit.



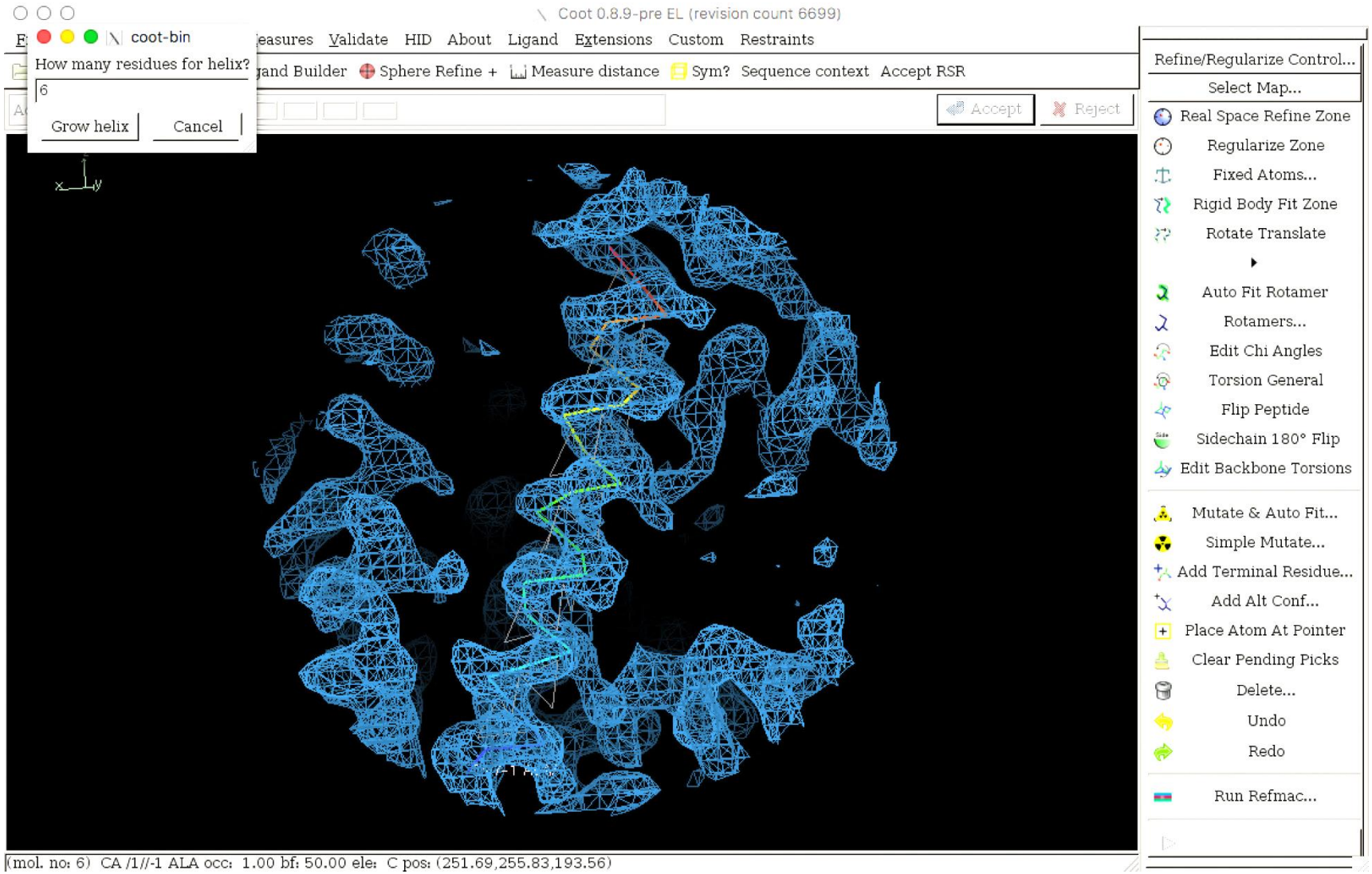
COOT – Crystallographic Object Oriented Toolkit

- Semi-automated helix placement
- Place cursor at the center of the helix and trigger ”Place helix here” (I suggest via a key binding - “h” with *coot-trimmings*)
- Coot will attempt to automatically determine the length and direction of the helix.
- **Trim/extend, adjust weights, then refine using real-space refine zone. Drag into density to adjust fit.**



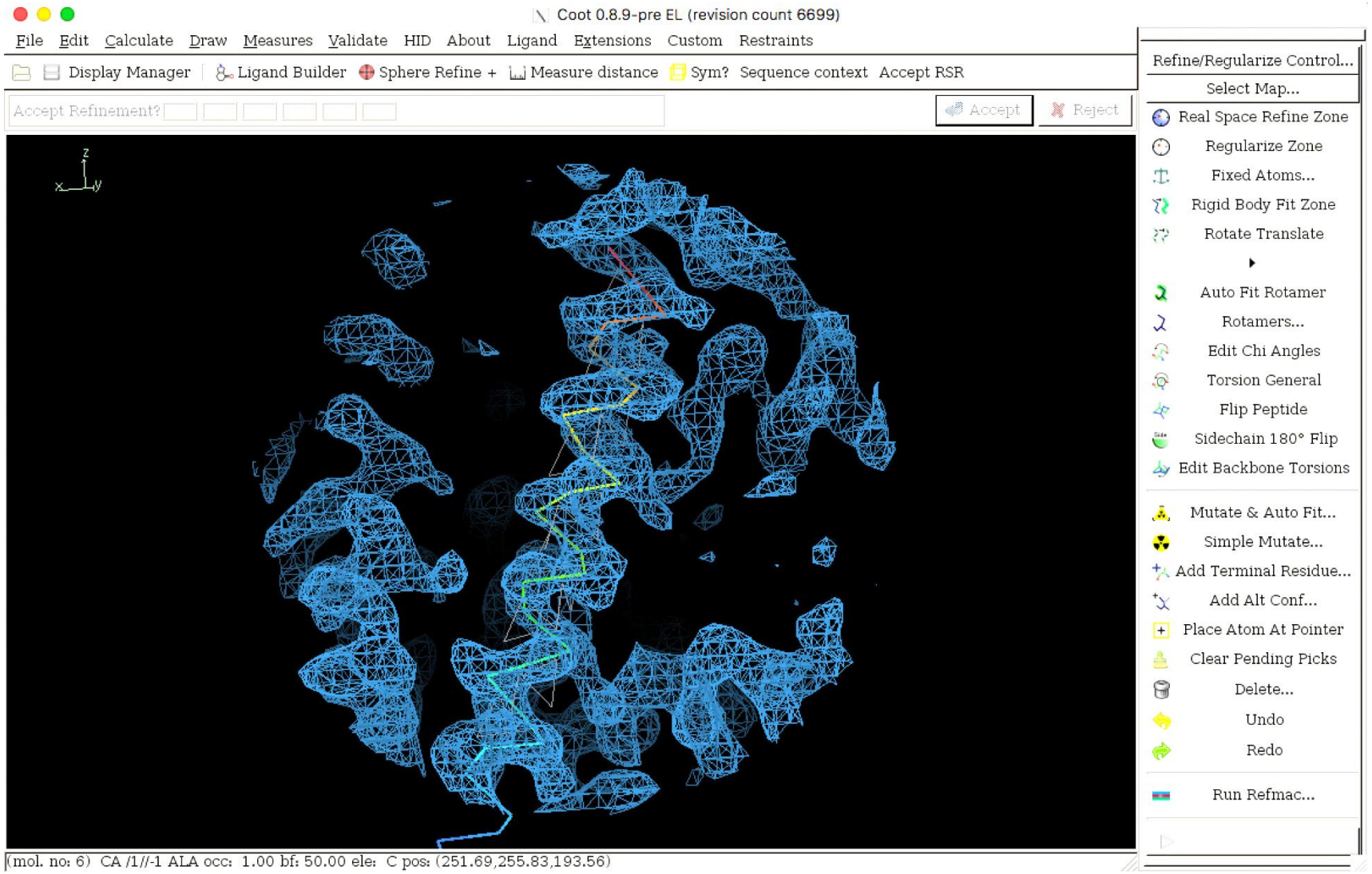
COOT – Crystallographic Object Oriented Toolkit

- Semi-automated helix placement
- Place cursor at the center of the helix and trigger ”Place helix here” (I suggest via a key binding - “h” with `coot-trimmings`)
- Coot will attempt to automatically determine the length and direction of the helix.
- **Trim/extend, adjust weights, then refine using real-space refine zone. Drag into density to adjust fit.**



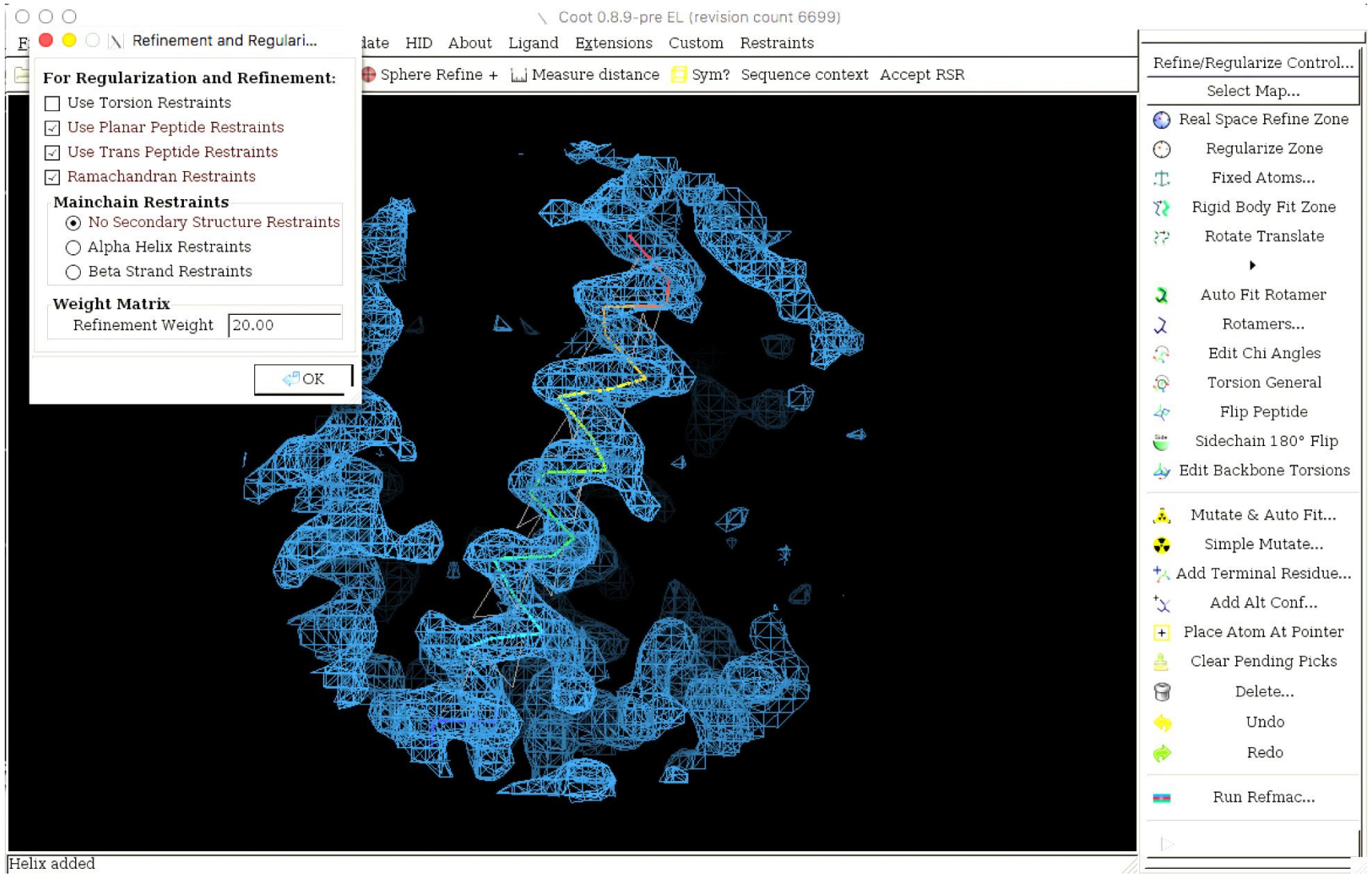
COOT – Crystallographic Object Oriented Toolkit

- Semi-automated helix placement
- Place cursor at the center of the helix and trigger ”Place helix here” (I suggest via a key binding - “h” with `coot-trimmings`)
- Coot will attempt to automatically determine the length and direction of the helix.
- **Trim/extend, adjust weights, then refine using real-space refine zone. Drag into density to adjust fit.**



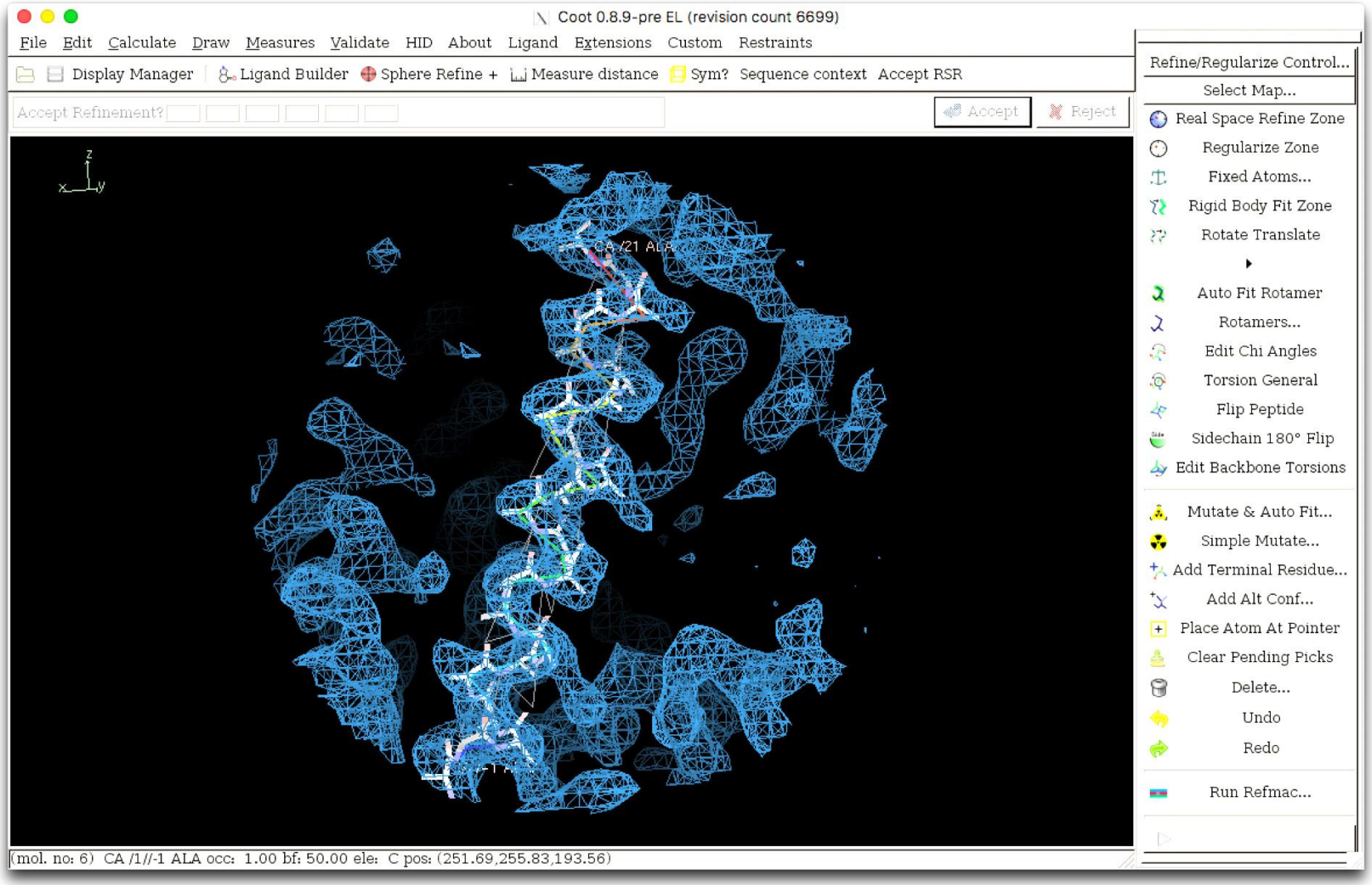
COOT – Crystallographic Object Oriented Toolkit

- Semi-automated helix placement
- Place cursor at the center of the helix and trigger ”Place helix here” (I suggest via a key binding - “h” with `coot-trimmings`)
- Coot will attempt to automatically determine the length and direction of the helix.
- **Trim/extend, adjust weights, then refine using real-space refine zone. Drag into density to adjust fit.**



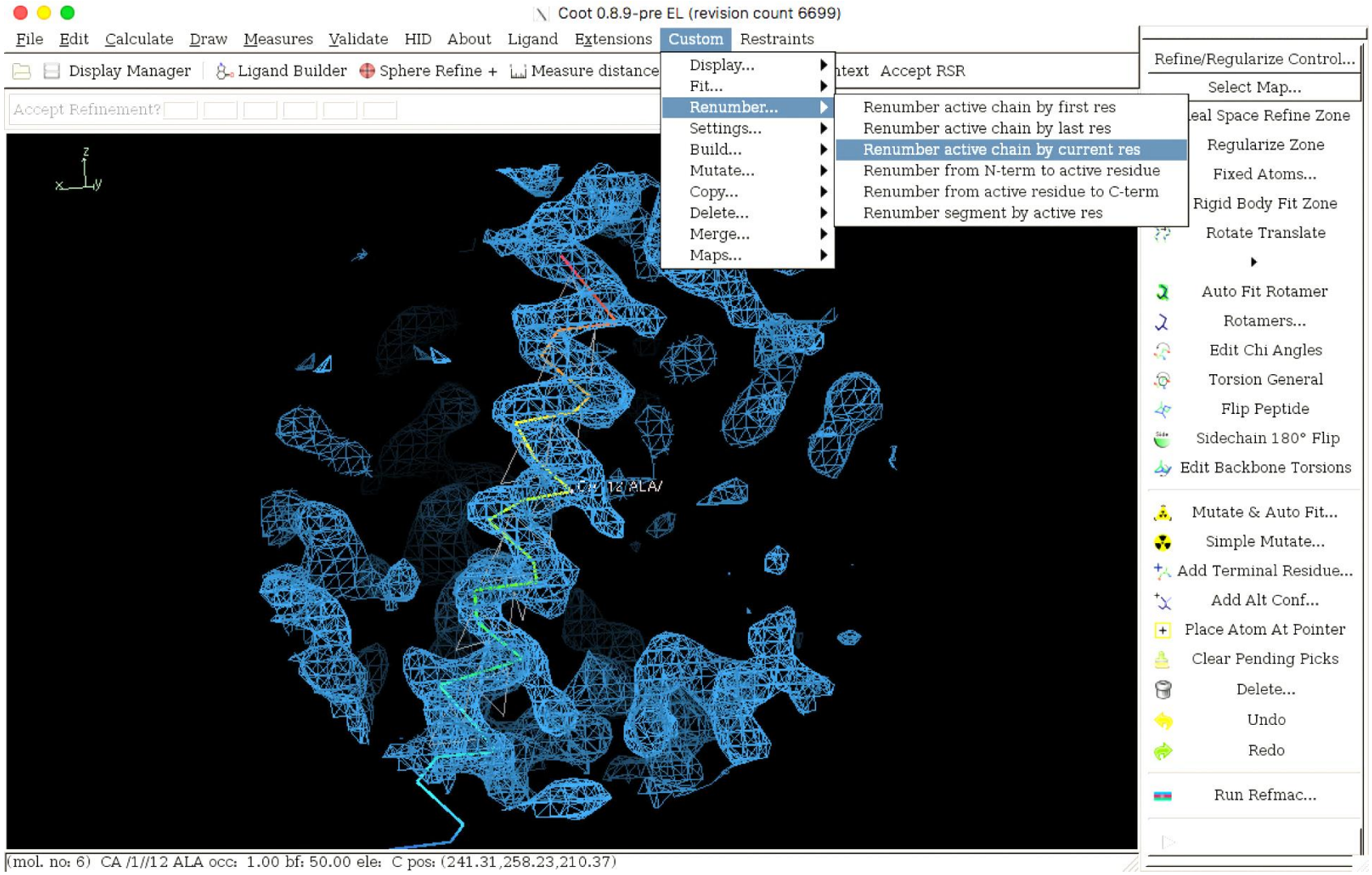
COOT – Crystallographic Object Oriented Toolkit

- Semi-automated helix placement
- Place cursor at the center of the helix and trigger ”Place helix here” (I suggest via a key binding - “h” with `coot-trimmings`)
- Coot will attempt to automatically determine the length and direction of the helix.
- **Trim/extend, adjust weights, then refine using real-space refine zone. Drag into density to adjust fit.**



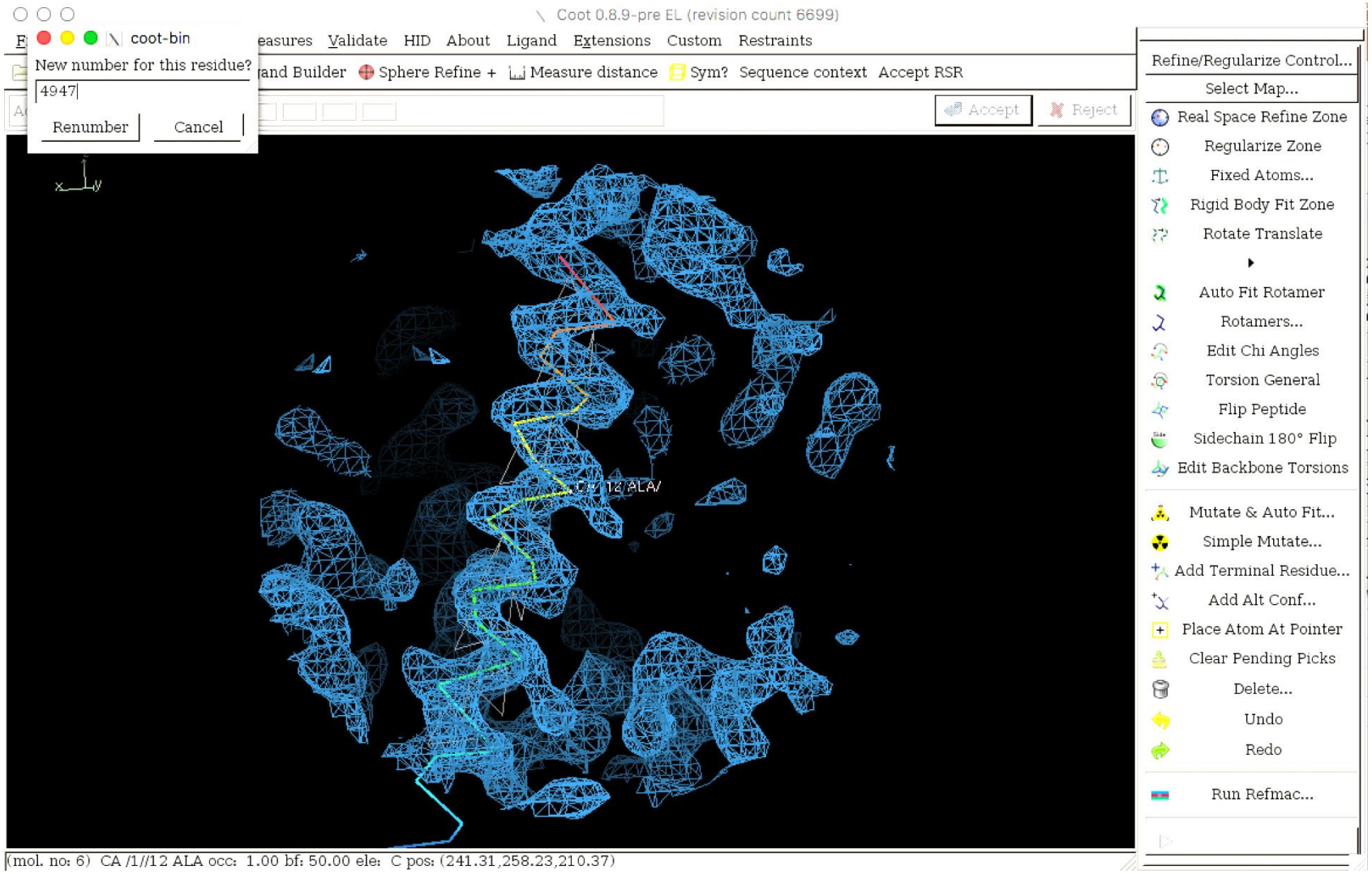
COOT – Crystallographic Object Oriented Toolkit

- Sequence assignment.
- **Adjust numbering to match expected position in sequence.**
- Mutate to match sequence
- Fill sidechains manually.
- Adjust sequence register to optimize local fit to sidechain densities.



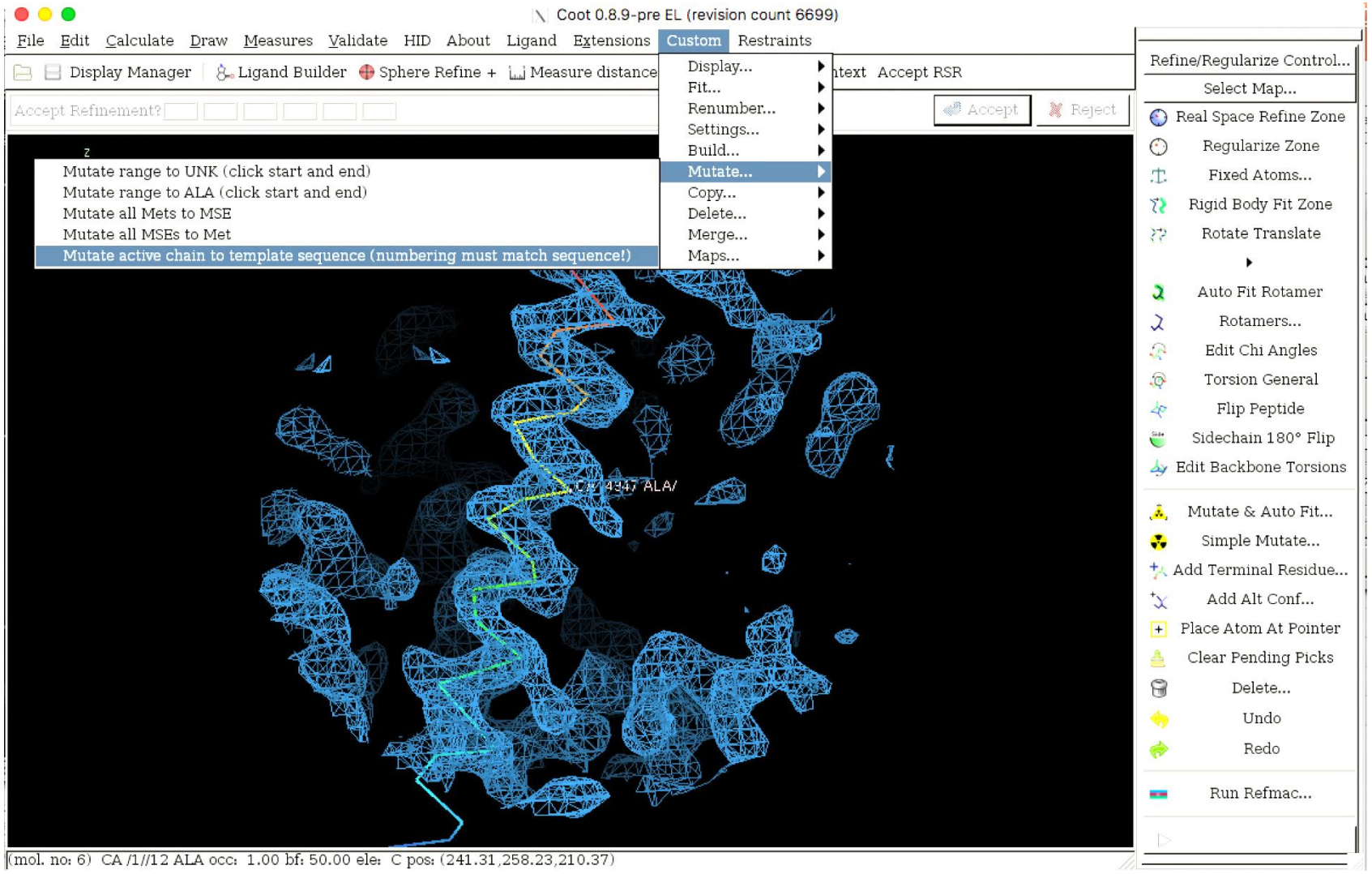
COOT – Crystallographic Object Oriented Toolkit

- Sequence assignment.
- **Adjust numbering to match expected position in sequence.**
- Mutate to match sequence
- Fill sidechains manually.
- Adjust sequence register to optimize local fit to sidechain densities.



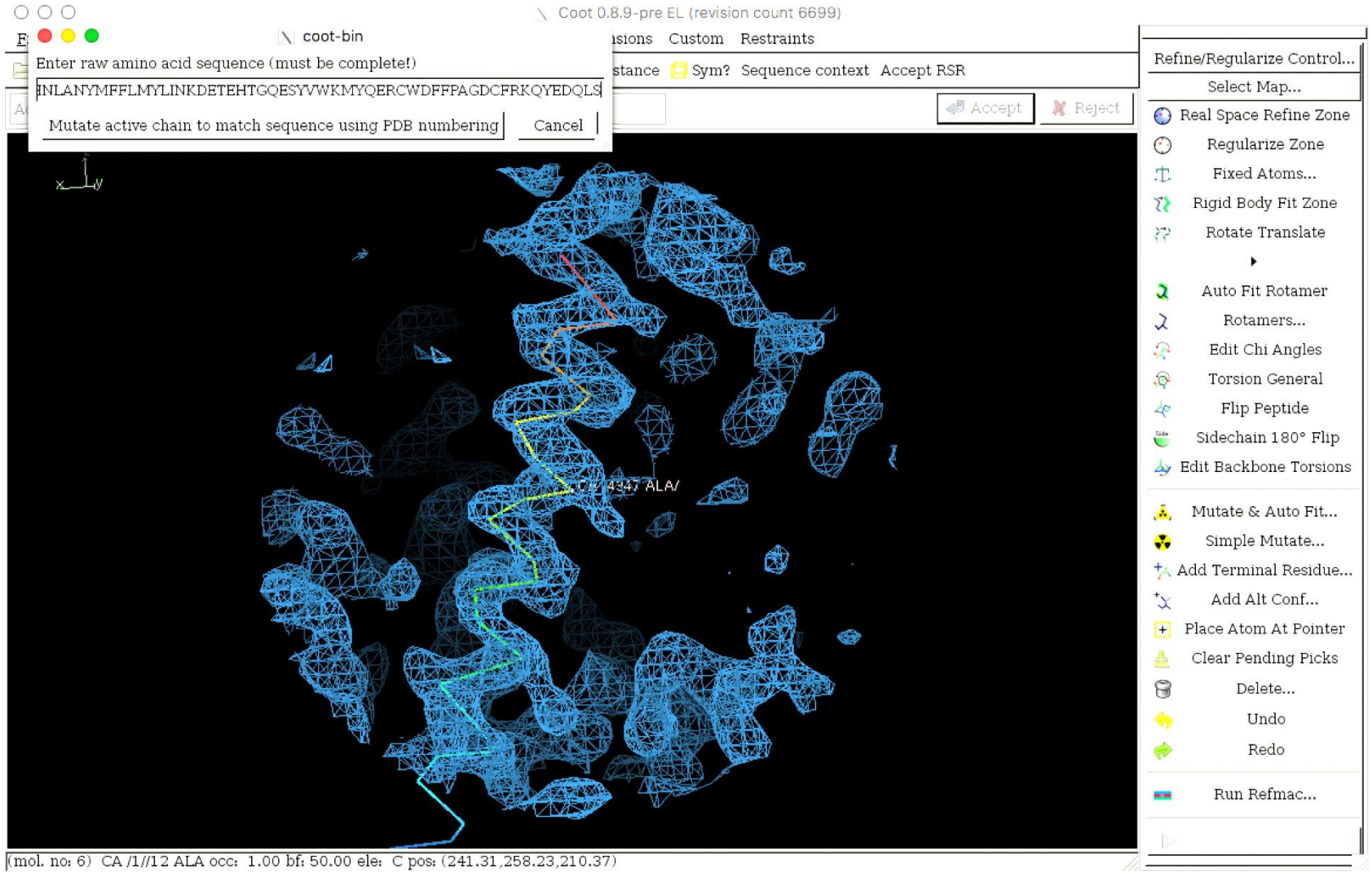
COOT – Crystallographic Object Oriented Toolkit

- Sequence assignment.
- Adjust numbering to match expected position in sequence.
- **Mutate to match sequence**
- Fill sidechains manually.
- Adjust sequence register to optimize local fit to sidechain densities.



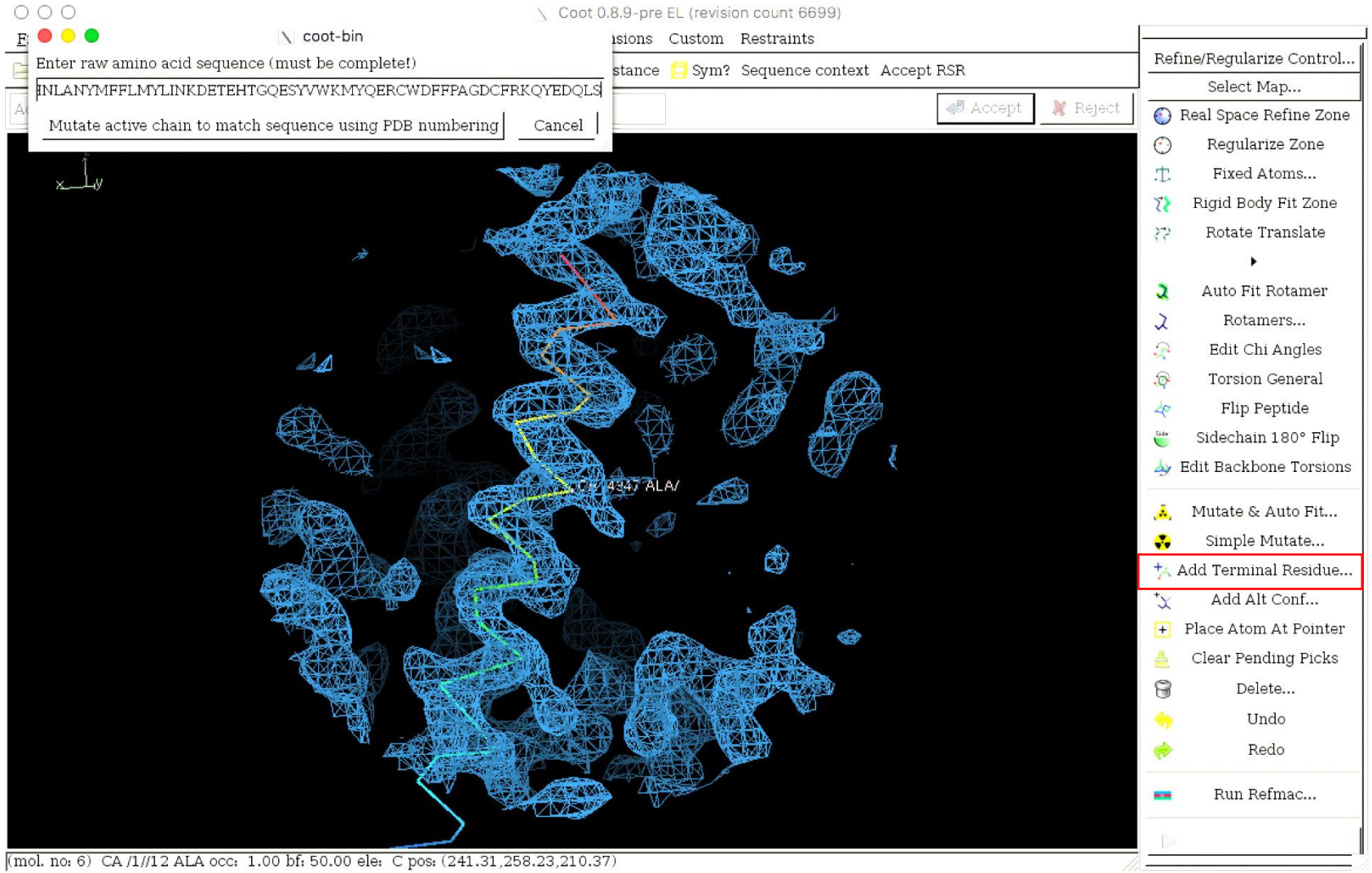
COOT – Crystallographic Object Oriented Toolkit

- Sequence assignment.
- Adjust numbering to match expected position in sequence.
- **Mutate to match sequence**
- Fill sidechains manually.
- Adjust sequence register to optimize local fit to sidechain densities.



COOT – Crystallographic Object Oriented Toolkit

- Sequence assignment.
- Adjust numbering to match expected position in sequence.
- **Mutate to match sequence**
- Fill sidechains manually.
- Adjust sequence register to optimize local fit to sidechain densities.



Use 'Add Terminal residue' to extend chain.

ISOLDE

- Interactive molecular dynamics flexible fitting, implemented as plugin for ChimeraX

- Useful during “polishing” stage of generating a final model, identifying and fixing otherwise difficult to correct errors in geometry, non-bonded contacts. Physically realistic simulation guided by map, user input.

- Complementary to COOT – COOT better for de novo building and assembly, ligand placement, ISOLDE very useful for final round of real space fitting.

(Croll, 2018, Acta. Cryst. D)

Types of errors in macromolecular models

- Identity (e.g. wrong domain)
- Directionality
- Topology/connectivity
- Register
- Rotamer
- Backbone torsion
- Ligand identification and placement

Types of errors in macromolecular models

- Identity (e.g. wrong domain)
- Directionality

Low resolution (<4.5 Å)

- **Topology/connectivity**
- **Register**

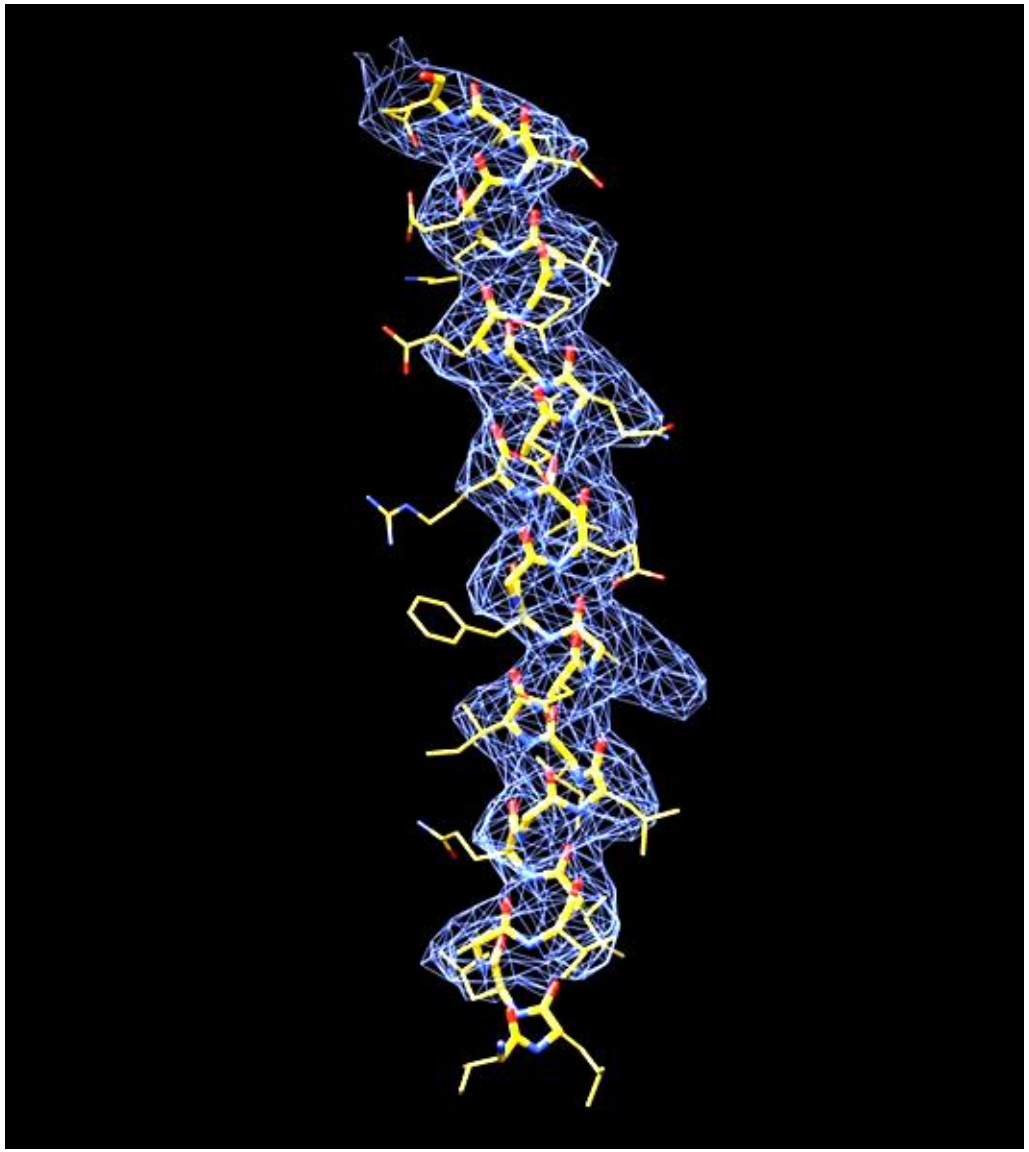
Medium resolution (3.5-4.5 Å)

- Rotamer
- Backbone torsion
- Ligand identification and placement

Medium/high resolution (2.5-4 Å)

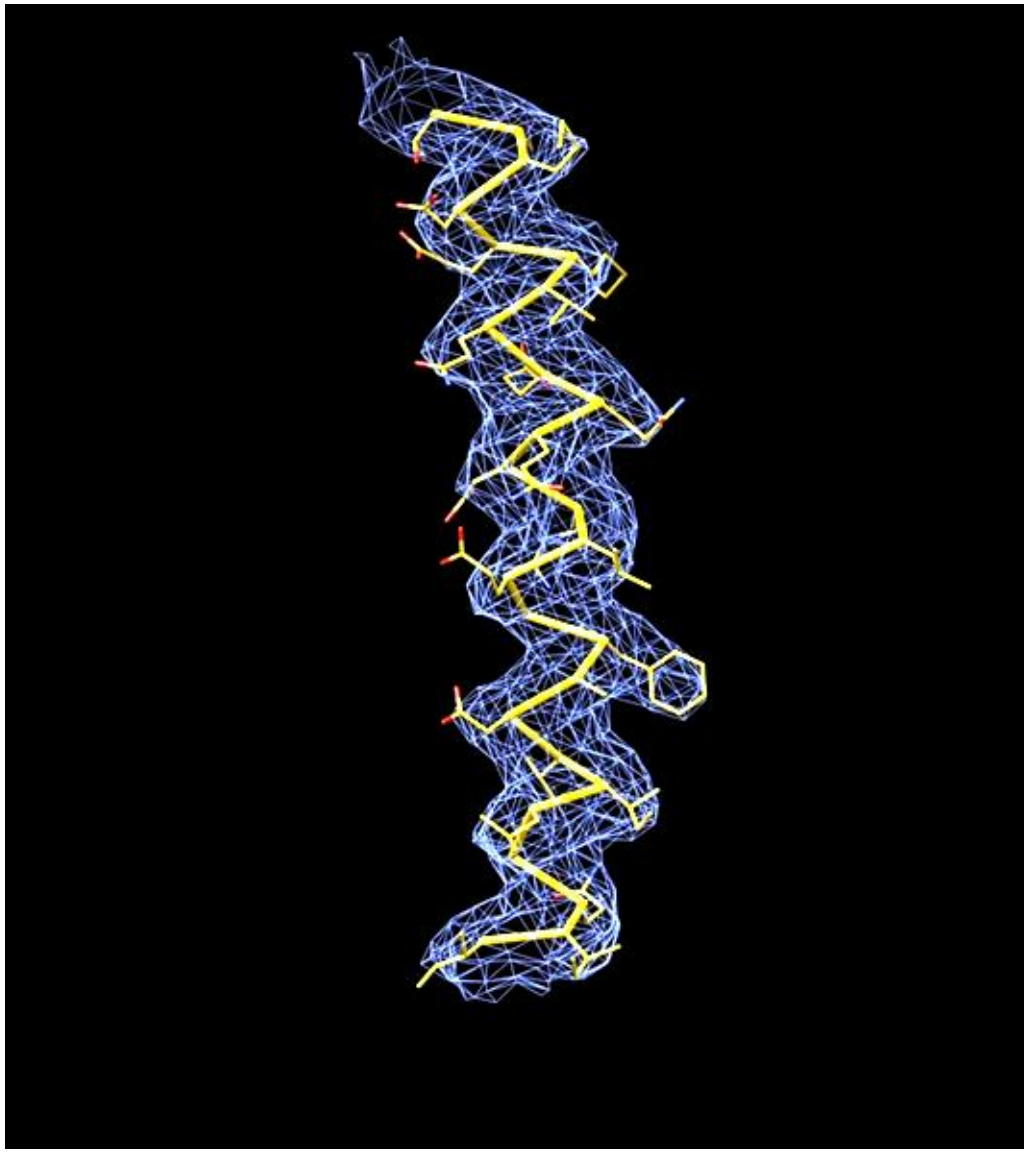
Types of errors in macromolecular models

- Identity (e.g. wrong domain)
- Directionality
- Topology/connectivity
- **Register**
- Rotamer
- Backbone torsion
- Ligand identification and placement



Types of errors in macromolecular models

- Identity (e.g. wrong domain)
- Directionality
- Topology/connectivity
- **Register**
- Rotamer
- Backbone torsion
- Ligand identification and placement



Strategy for identifying and correcting errors.

- Analyse as you go – “sanity checks” on chemistry, nonbonded interactions, surface composition. Use Molprobity for clashes, Chimera(X) or pymol to check e.g. for buried polars, exposed hydrophobics. Monitor agreement with secondary structure, disorder predictions.

- Use EM-ringer (or other local metrics – Q-score, Strudel score, MEDIC all useful) to identify errors in backbone and rotamer geometry.

- **Look at everything! Manually check and recheck the fit of every residue. Tedious but necessary.**

- Sometimes, you just can’t tell the right answer. Don’t be afraid to specify sequence ambiguity (use UNKs).

- Half-map FSCs are useful to analyze/identify overfitting during refinement, but they tell you little about the local quality or correctness of the model.

Finally…

“ALL MODELS ARE WRONG, BUT SOME ARE USEFUL” – *George P. Box*

* It should be remembered that just as the Declaration of Independence promises the pursuit of happiness rather than happiness itself, so the iterative scientific model building process offers only the pursuit of the perfect model. For even when we feel we have carried the model building process to a conclusion some new initiative may make further improvement possible. Fortunately to be useful a model does not have to be perfect.

George P. Box, “Robustness in the Strategy of Scientific Model Building”, 1979

Thank you for listening!

COLUMBIA UNIVERSITY
MEDICAL CENTER