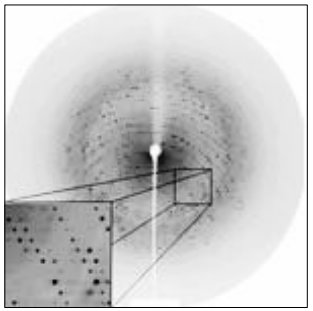
Automated Model Building

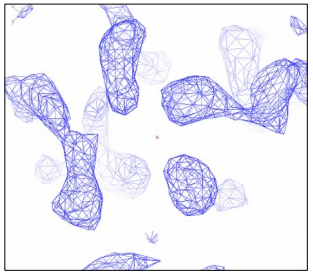
Buccaneer and Nautilus

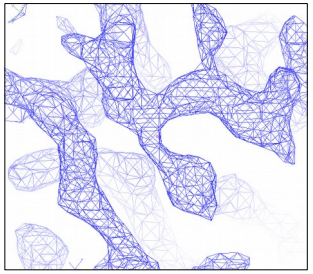
Paul Bond, Kevin Cowtan

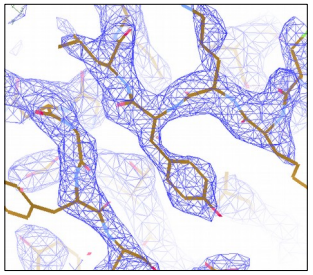
kevin.cowtan@york.ac.uk

X-ray structure solution pipeline...









Data collection

Data processing

Experimental phasing

Molecular Replacement

Density Modification

Model building

Refinement

Rebuilding Validation

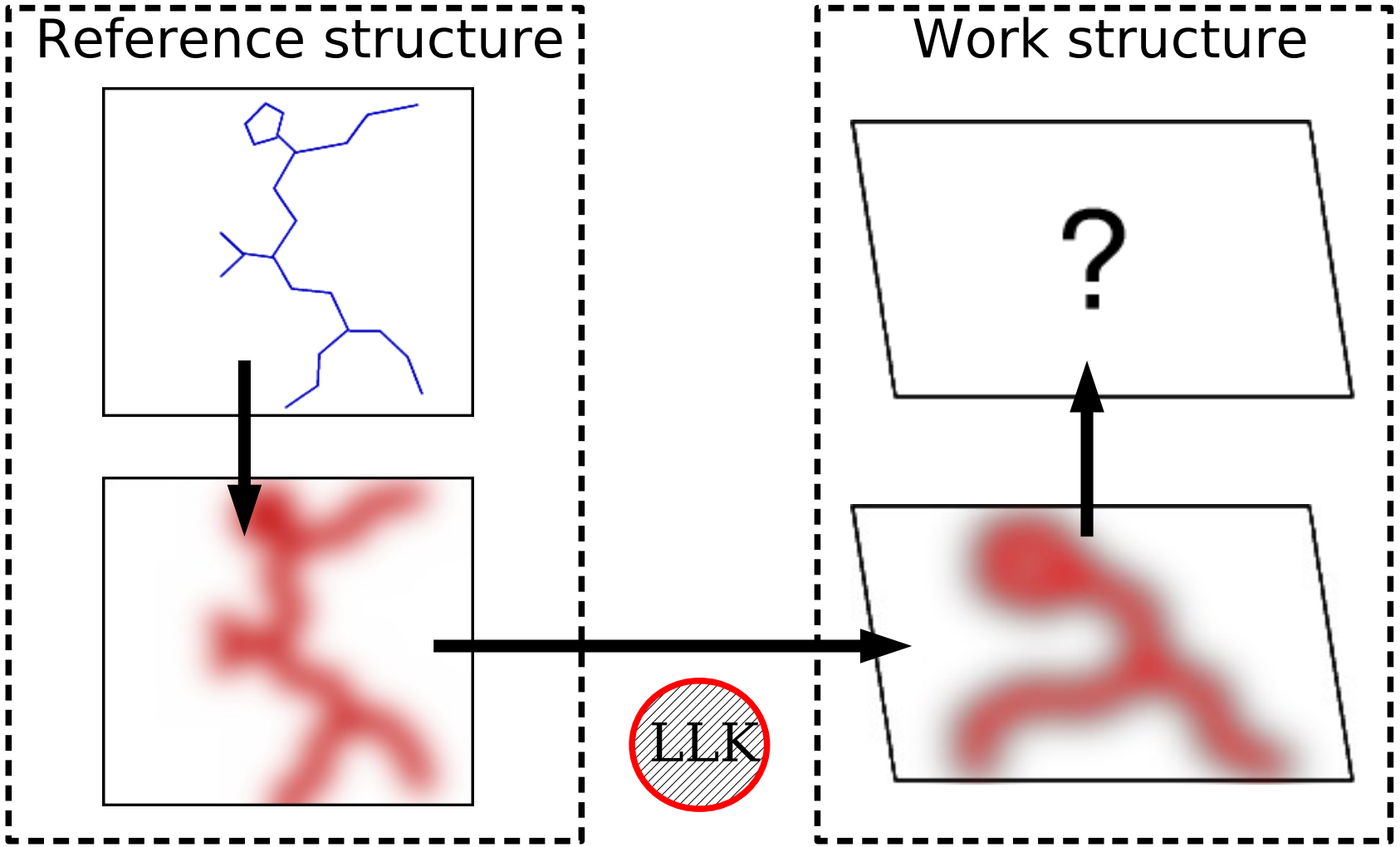
Buccaneer

Statistical model building software based on the use of a reference structure to construct likelihood targets for protein features.

- 2006 – Initial release, main chain tracing
  K. Cowtan, Acta Cryst. (2006). D**62**, 1002-1011 DOI
- 2008 – Sequencing, NCS
  K. Cowtan, Acta Cryst. (2008). D**64**, 83-89 DOI
- 2012 – Loop building, sloop
  K. Cowtan, Acta Cryst. (2012). D**68**, 328-335 DOI

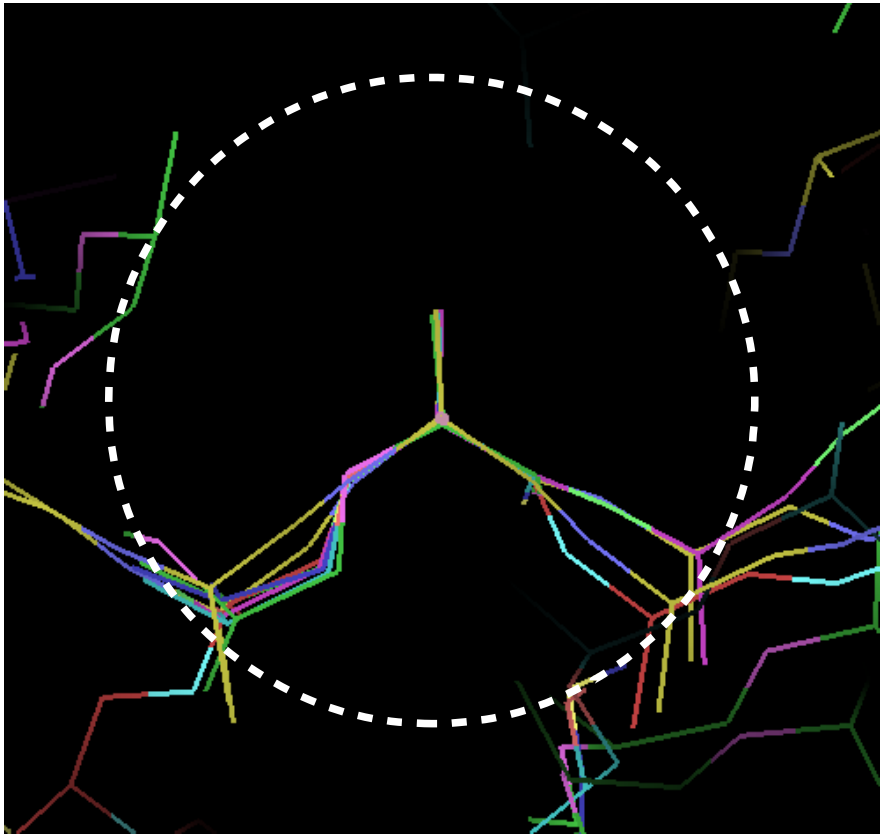
Buccaneer: Method

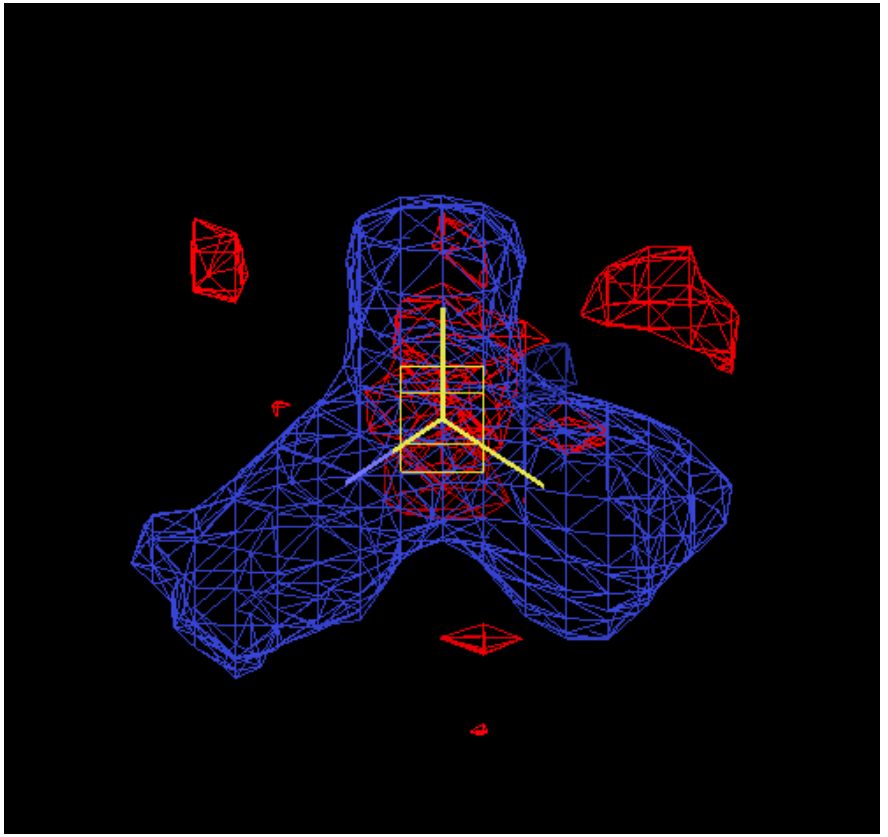
Compare simulated map and known model to obtain likelihood target, then search for this target in the unknown map.



Buccaneer: Method

- Compile statistics for reference map in 4A sphere about Cα => LLK target.





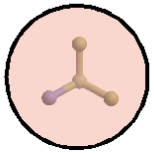
- Use mean/variance.

Buccaneer

Use a likelihood function based on conserved density features.

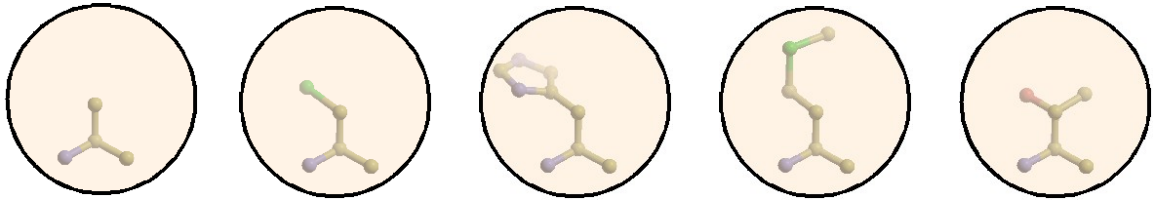
The same likelihood function is used several times. This makes the program very simple, and the whole calculation works over a range of resolutions.

Finding, growing: Look for C-alpha environment



(4.0A sphere about Cα)

Sequencing: Look for C-beta environment



(5.5A sphere about Cβ)

ALA CYS HIS MET THR ... x20

Buccaneer

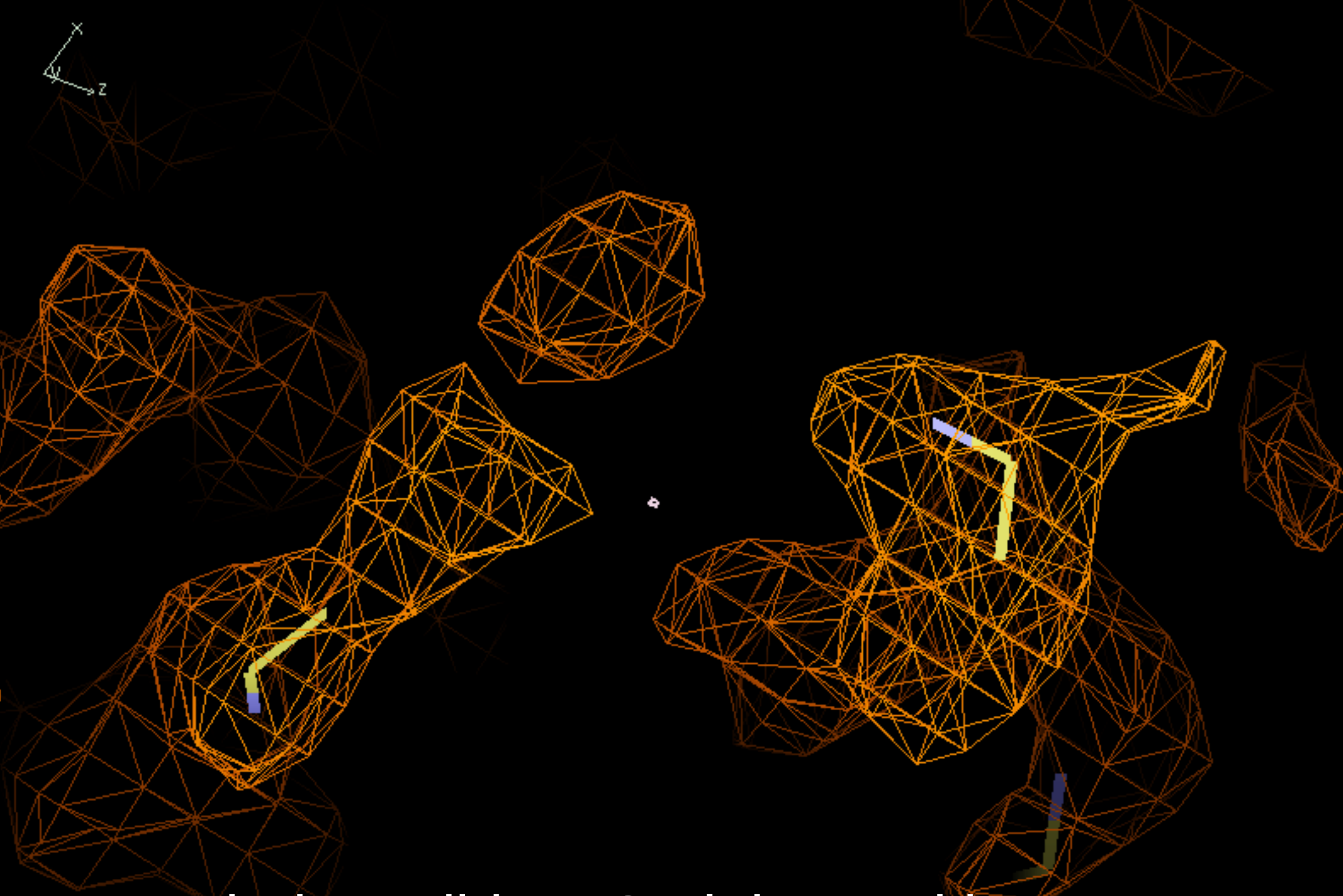
10 Steps per cycle:

- **Find** candidate C-alpha positions
- **Grow** them into chain fragments
- **Join** and merge the fragments, resolving branches
- **Link** nearby N and C terminii
- **Sequence** the chains (i.e. dock sequence)
- **Correct** insertions/deletions
- **Filter** based on poor density
- **NCS Rebuild** to complete NCS copies of chains
- **Prune** any remaining clashing chains
- **Rebuild** side chains

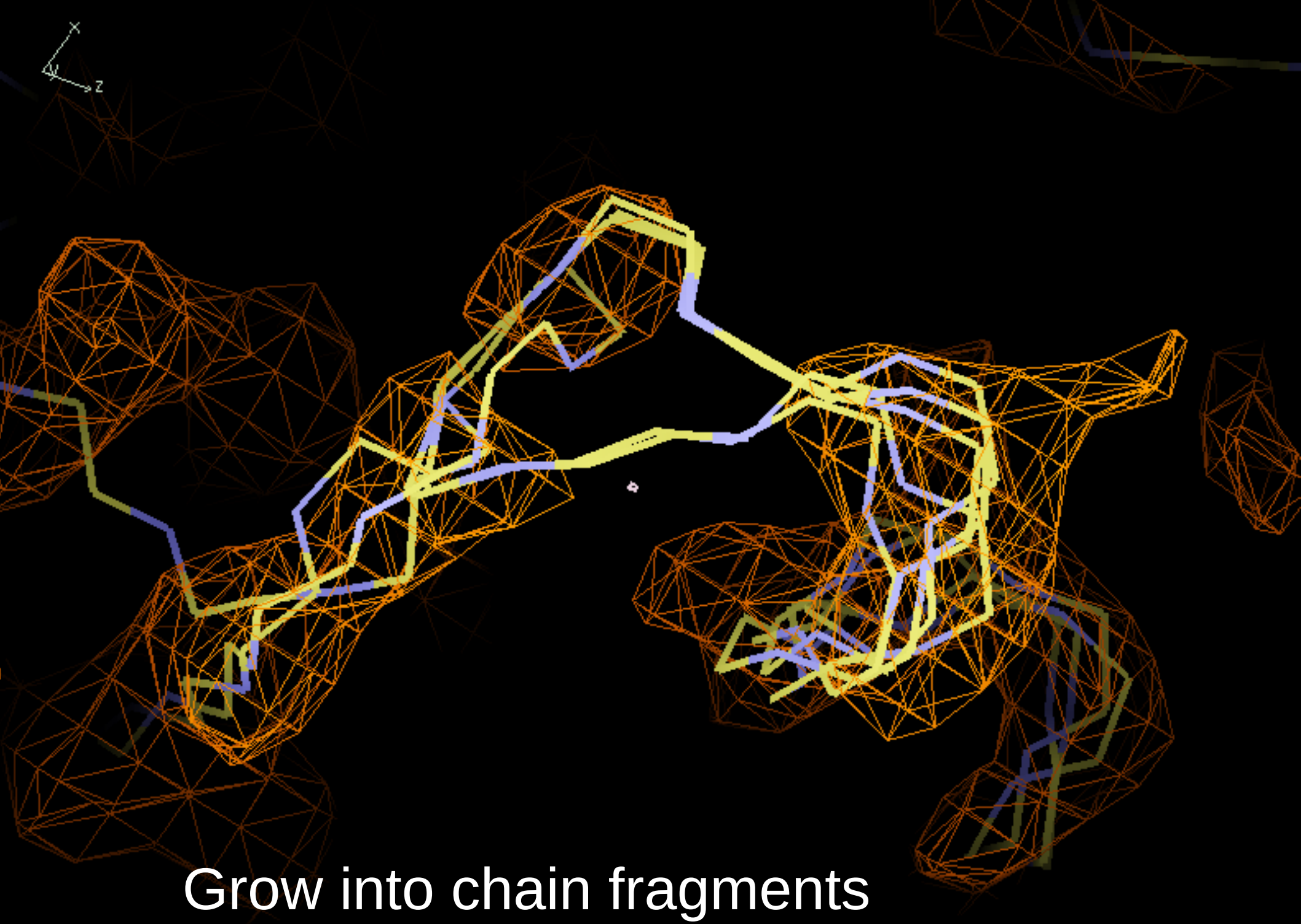
Buccaneer

Case Study:

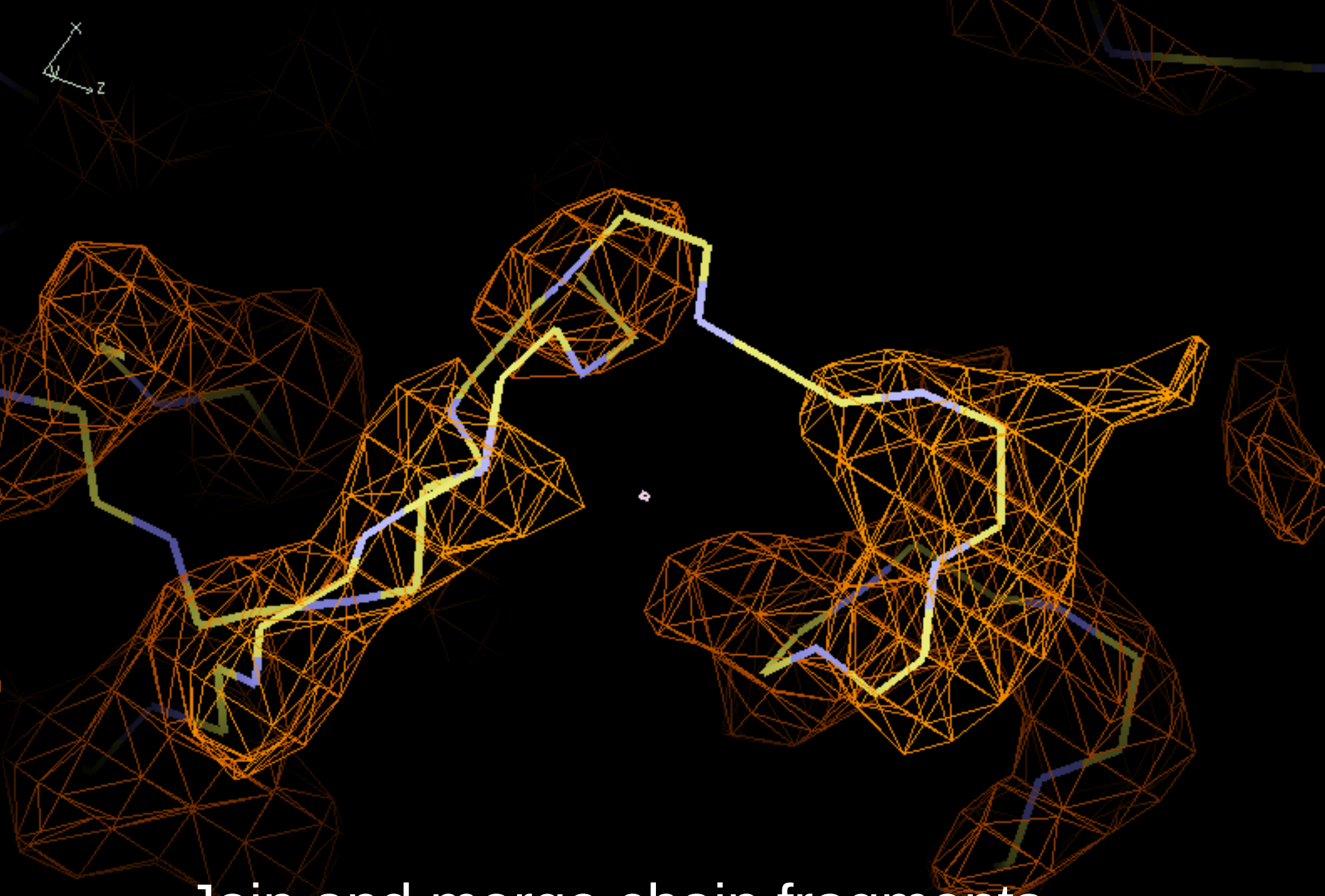
A difficult loop in a 2.9A map, calculated using real data from the JCSG.



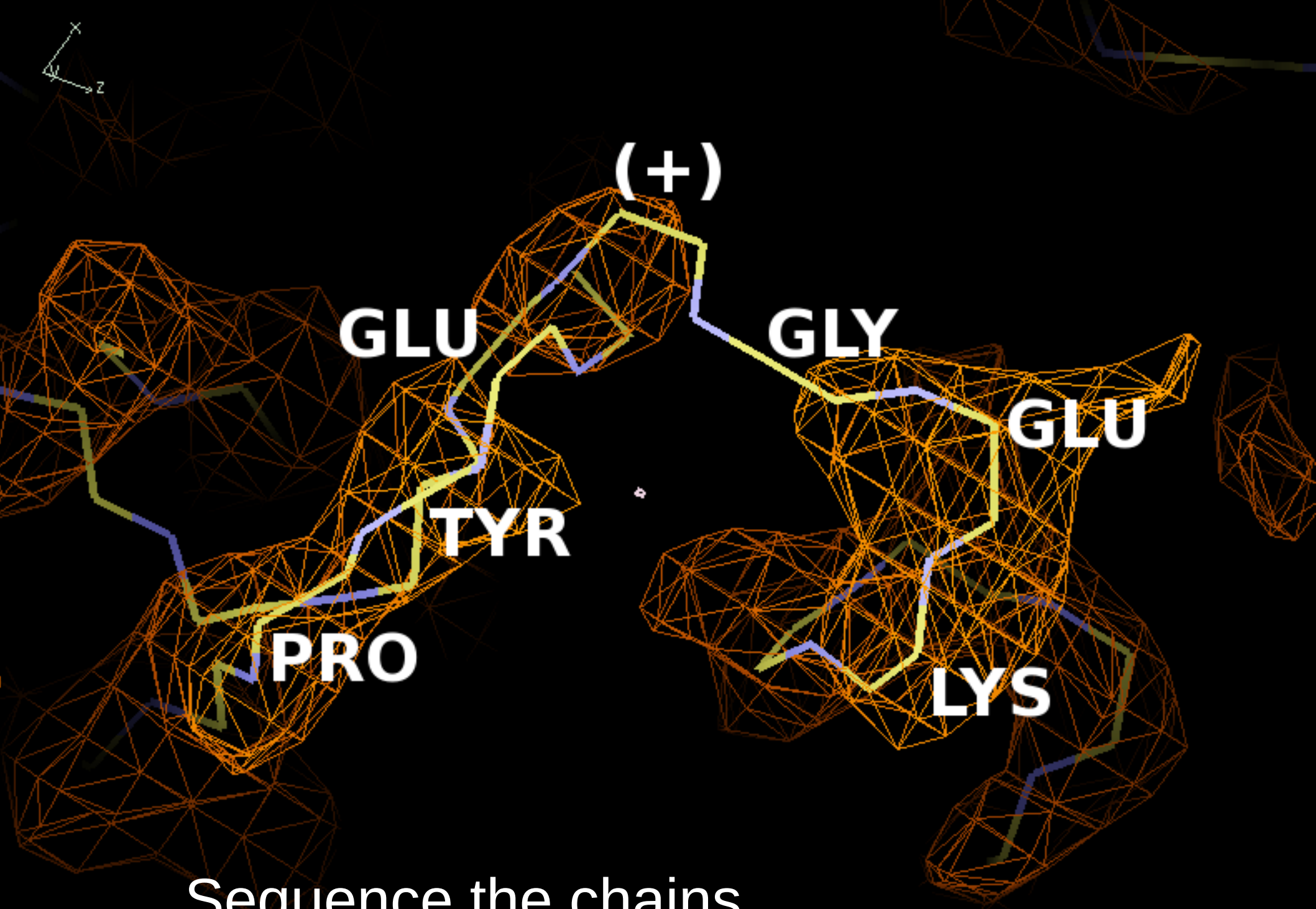
Find candidate C-alpha positions



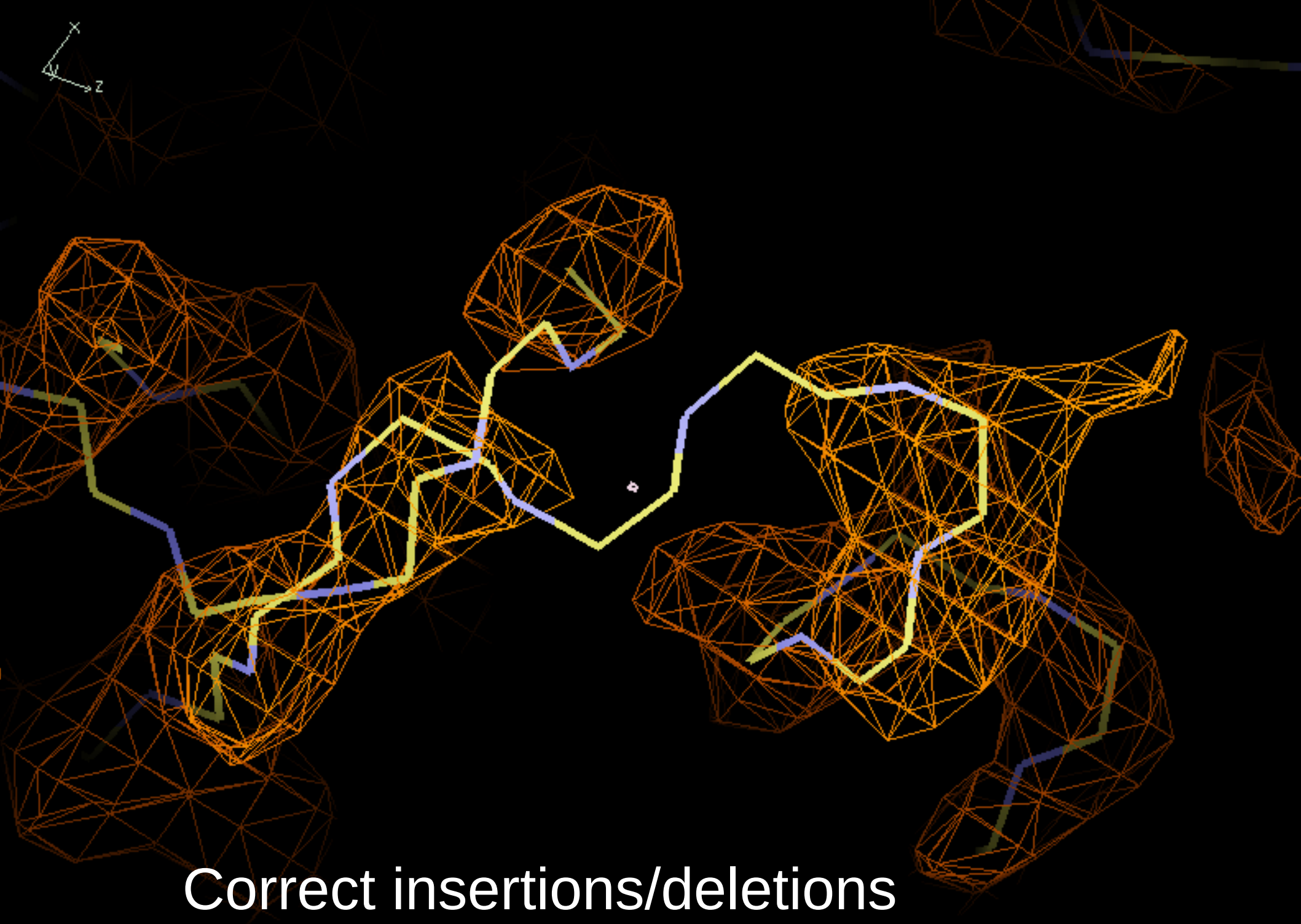
Grow into chain fragments



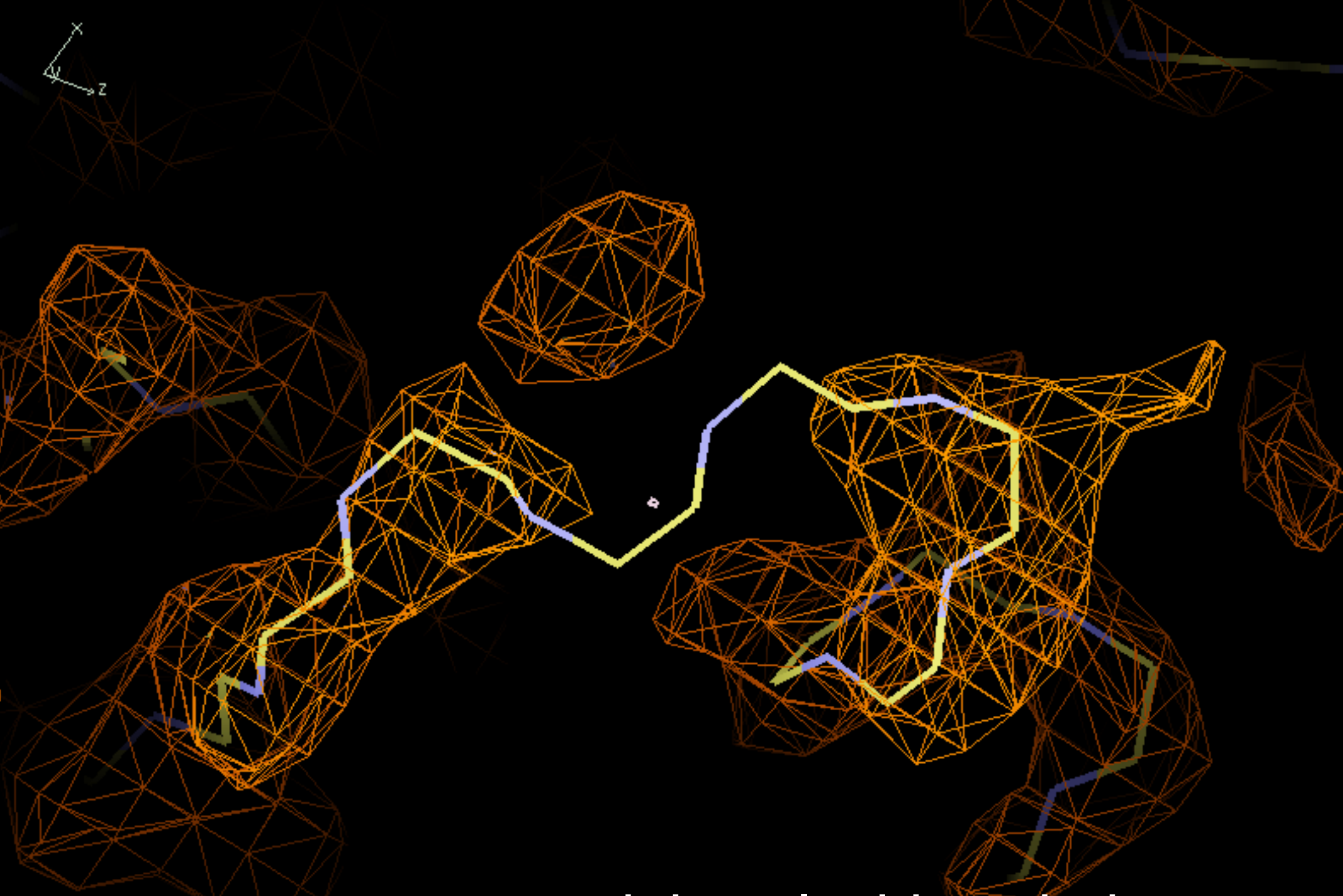
Join and merge chain fragments



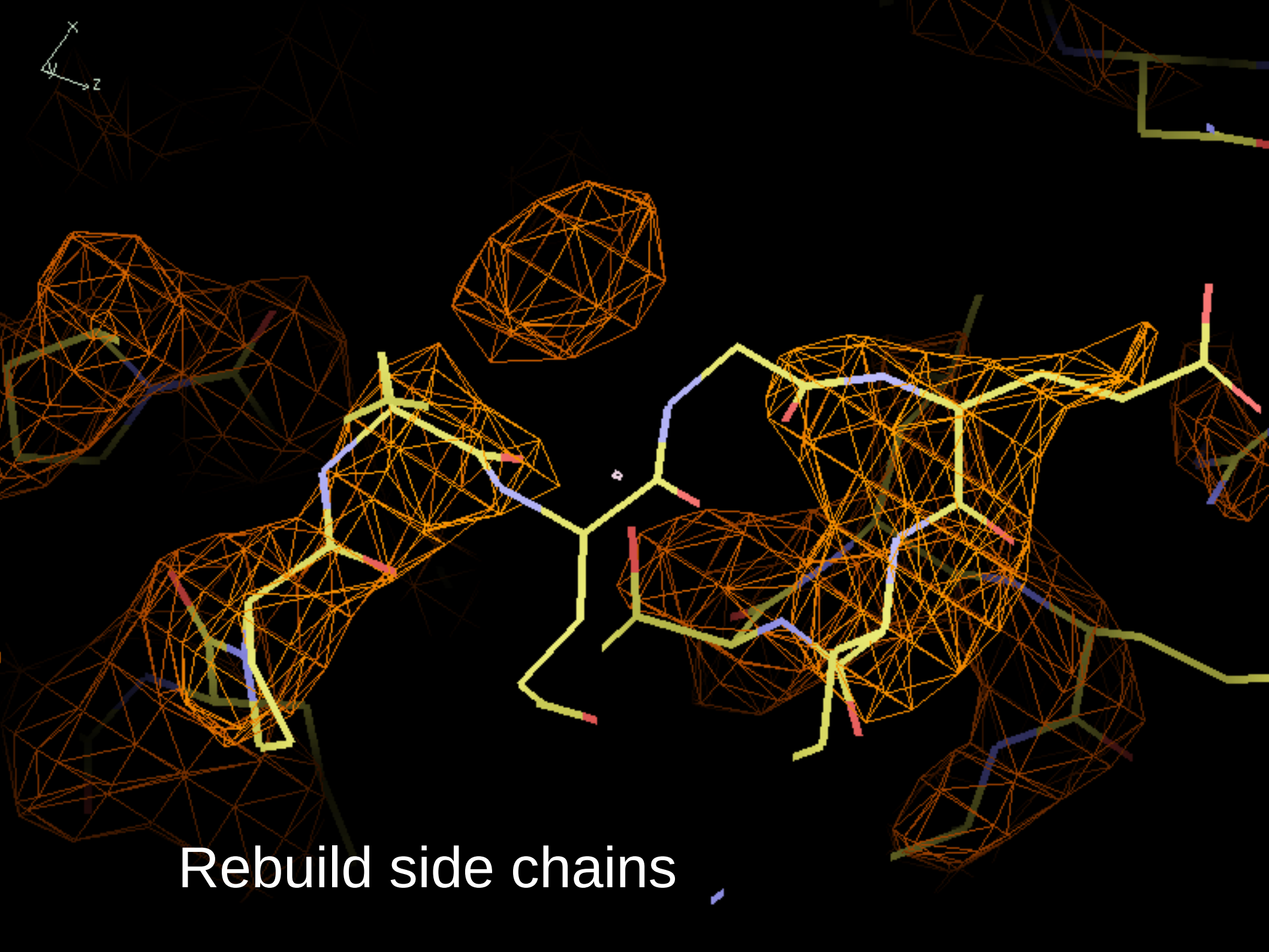
Sequence the chains

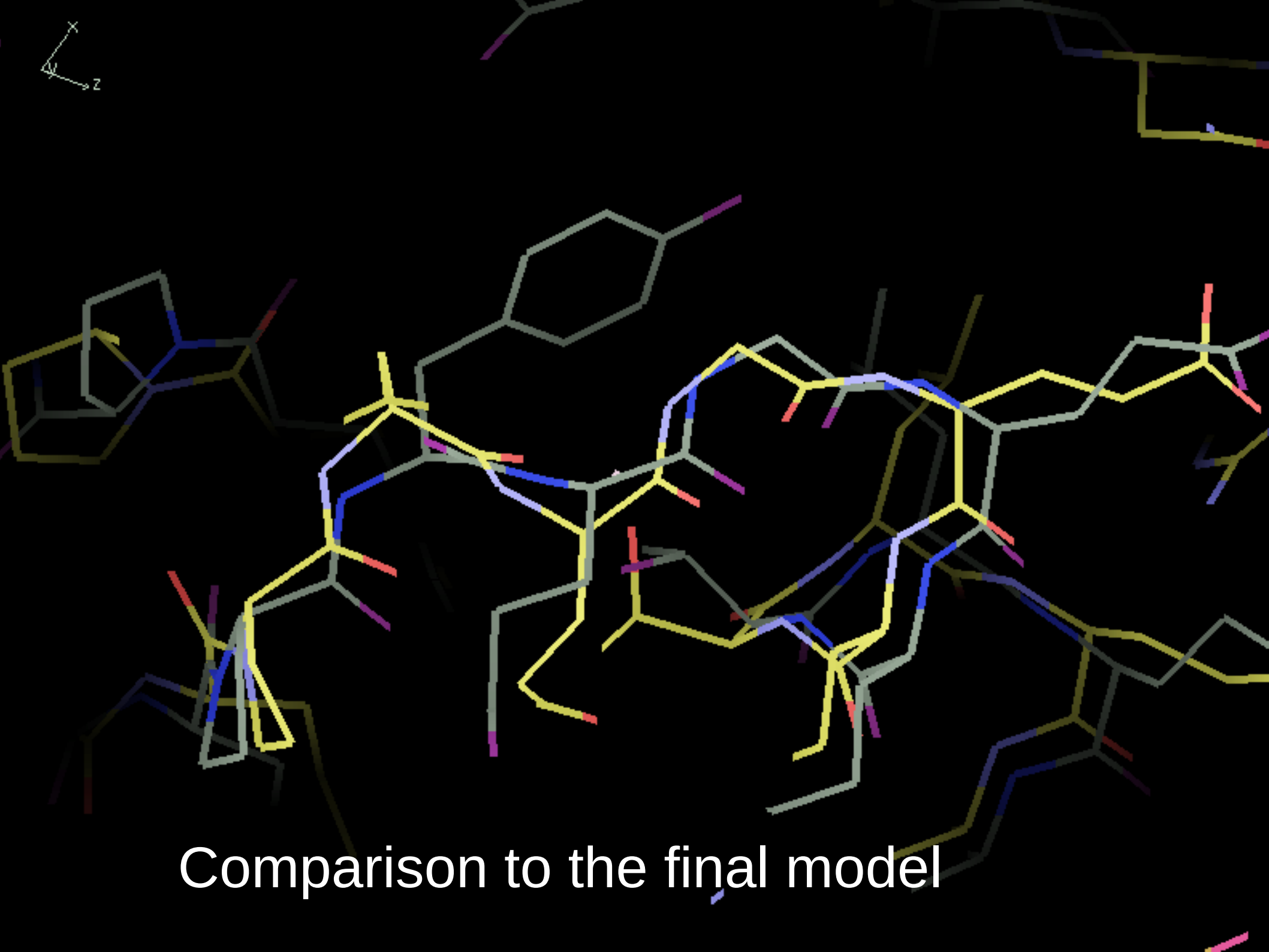


Correct insertions/deletions

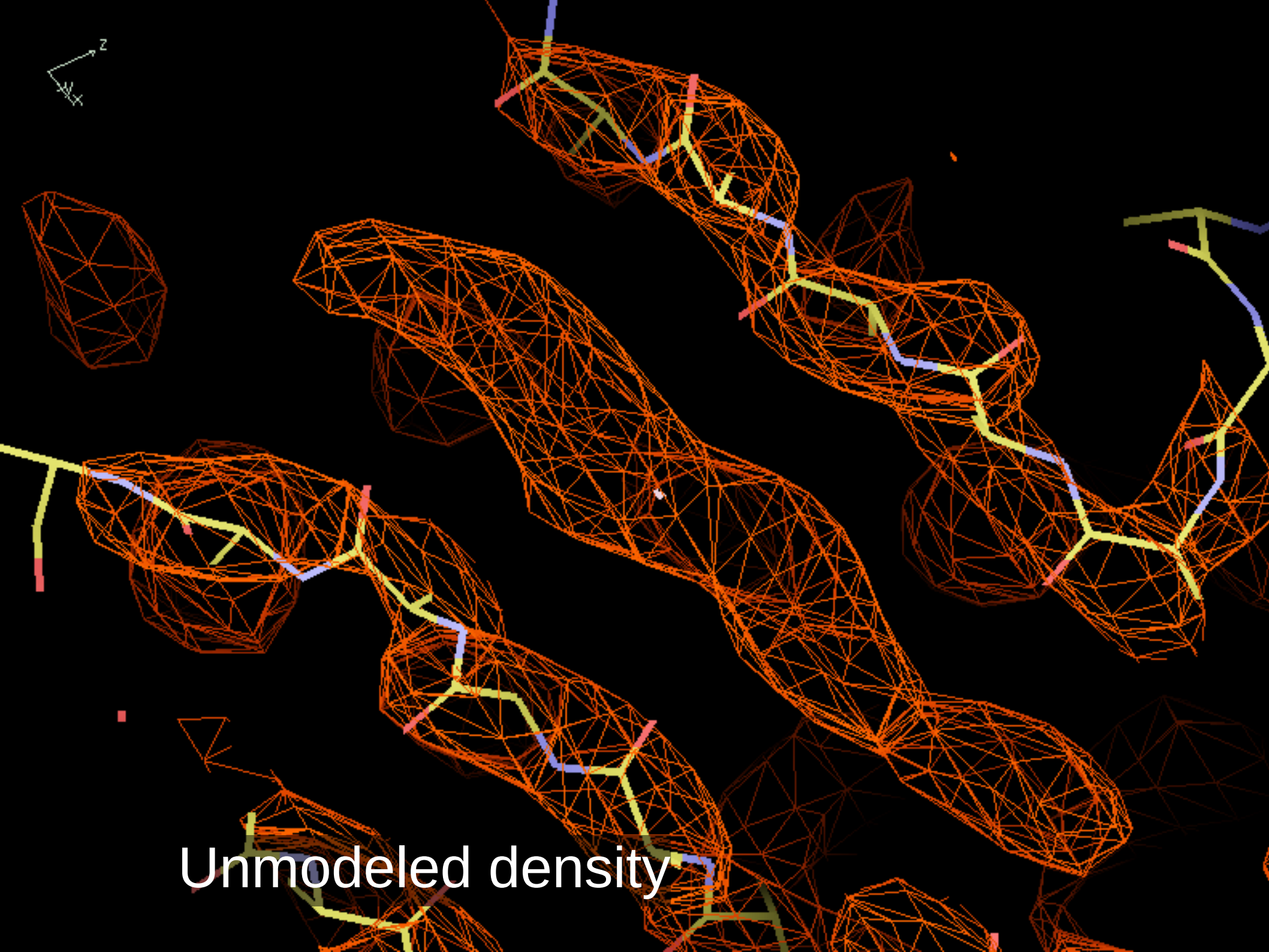


Prune any remaining clashing chains


Rebuild side chains



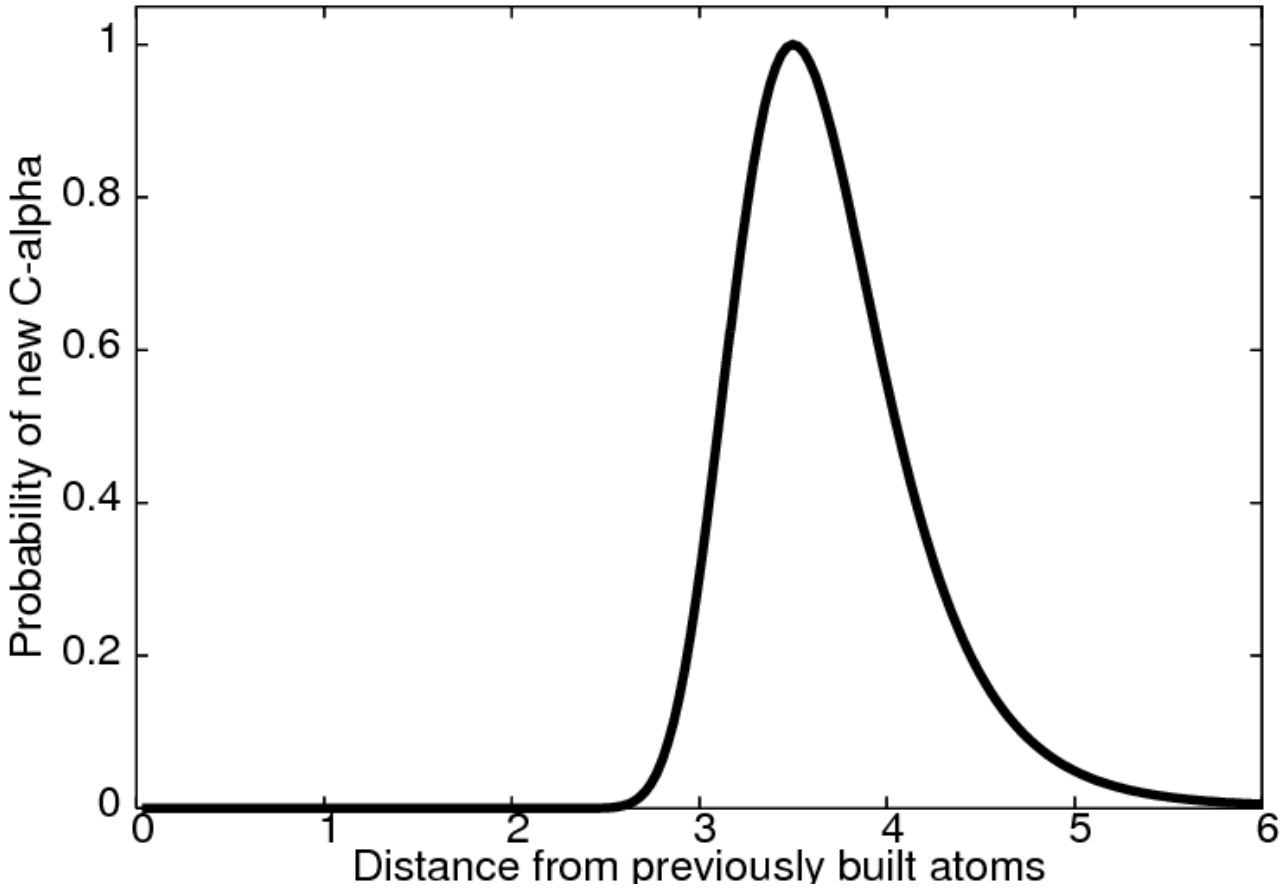
Comparison to the final model

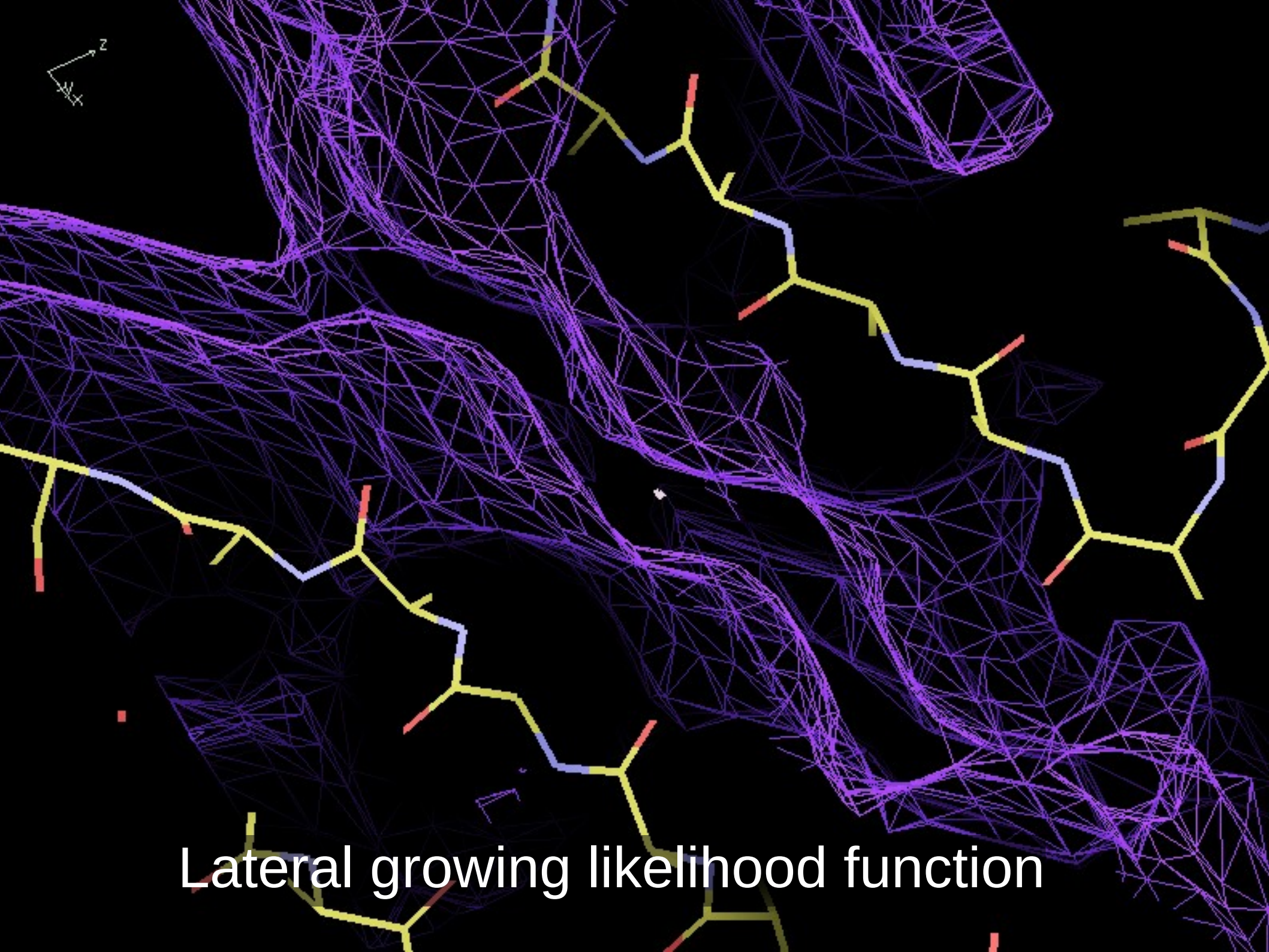

Unmodeled density

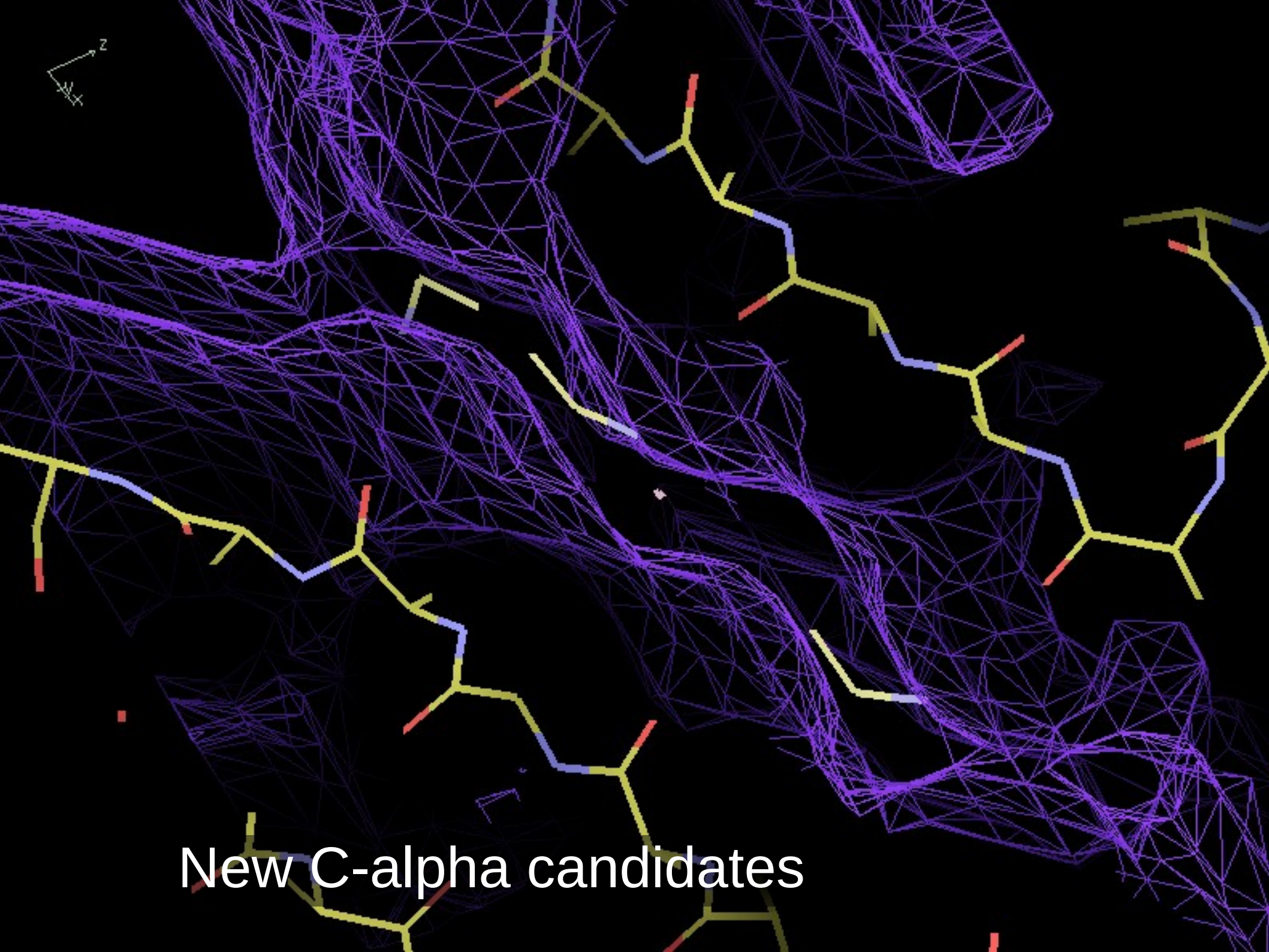
Buccaneer

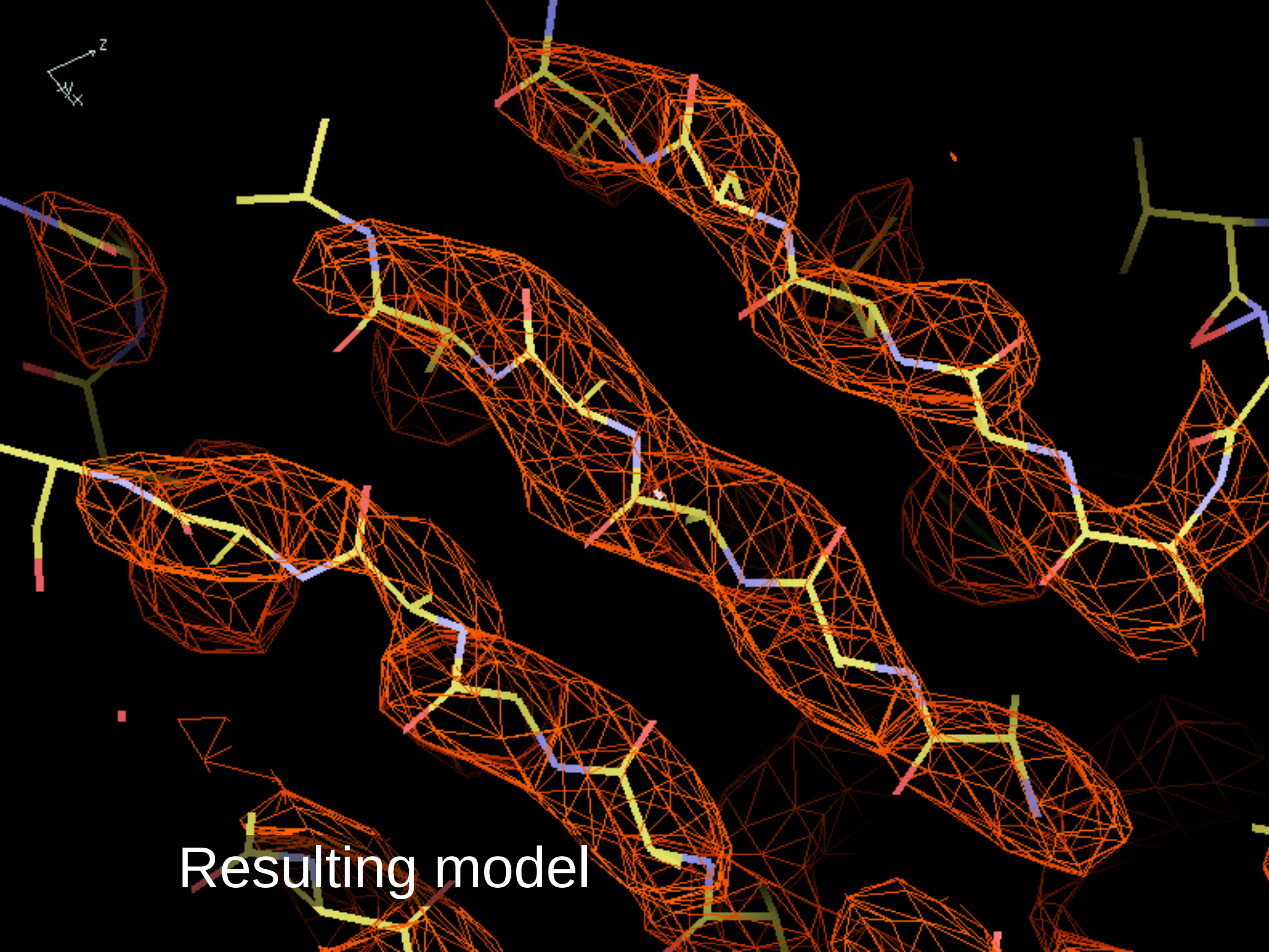
Model completion uses “**Lateral growing**”:

Grow sideways from existing chain fragments by looking for new C-alphas at an appropriate distance “sideways” from the existing chain:



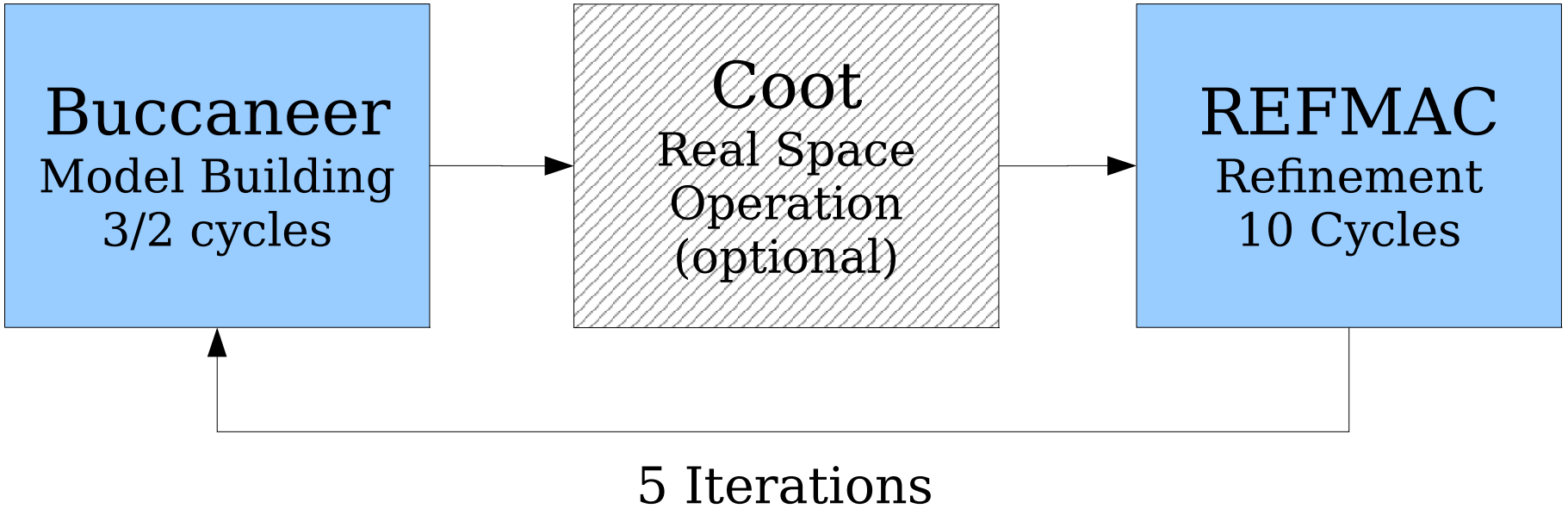

Lateral growing likelihood function


New C-alpha candidates


Resulting model

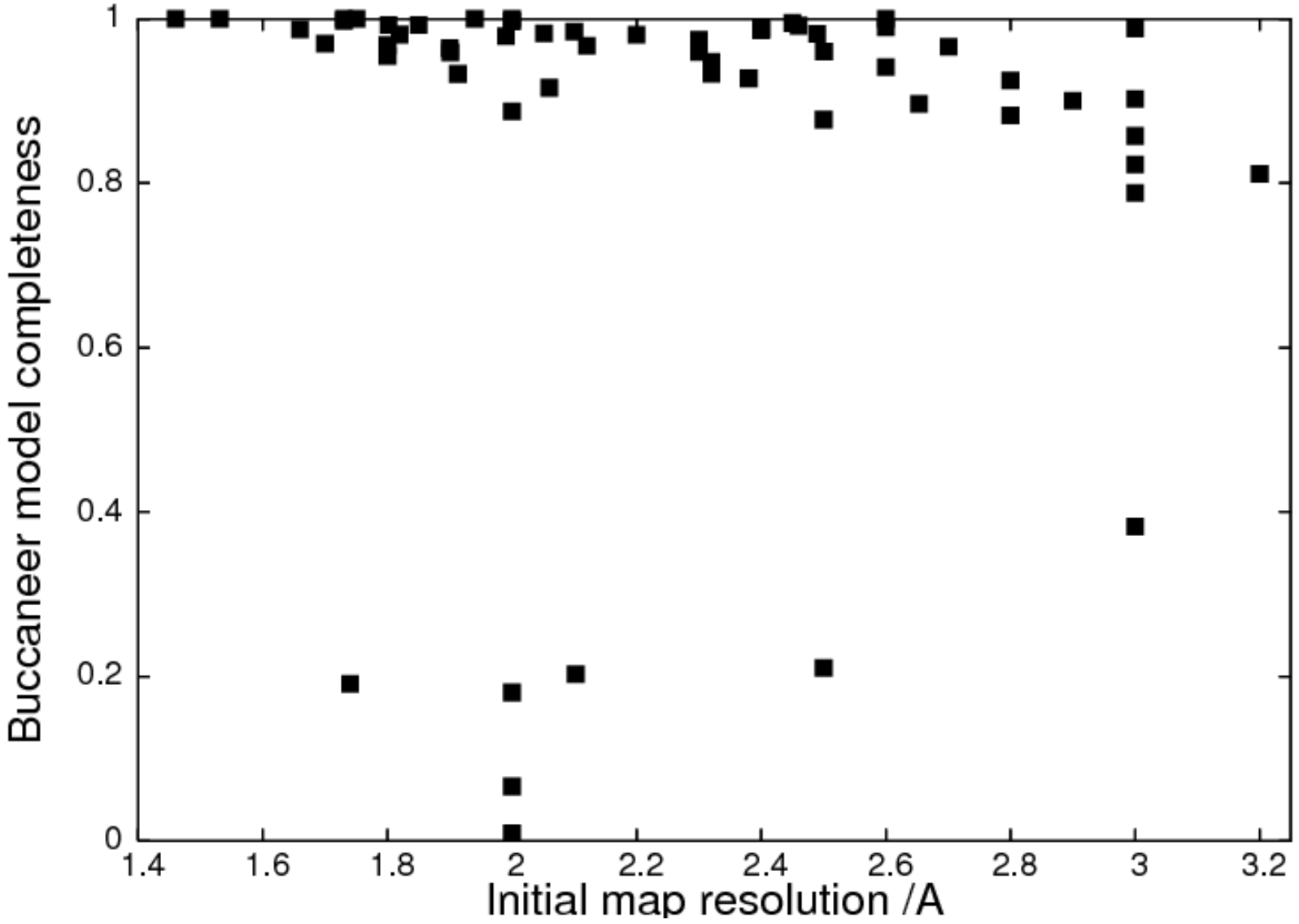
Buccaneer: Pipeline

CCP4i2 pipeline that iterates model building and refinement:



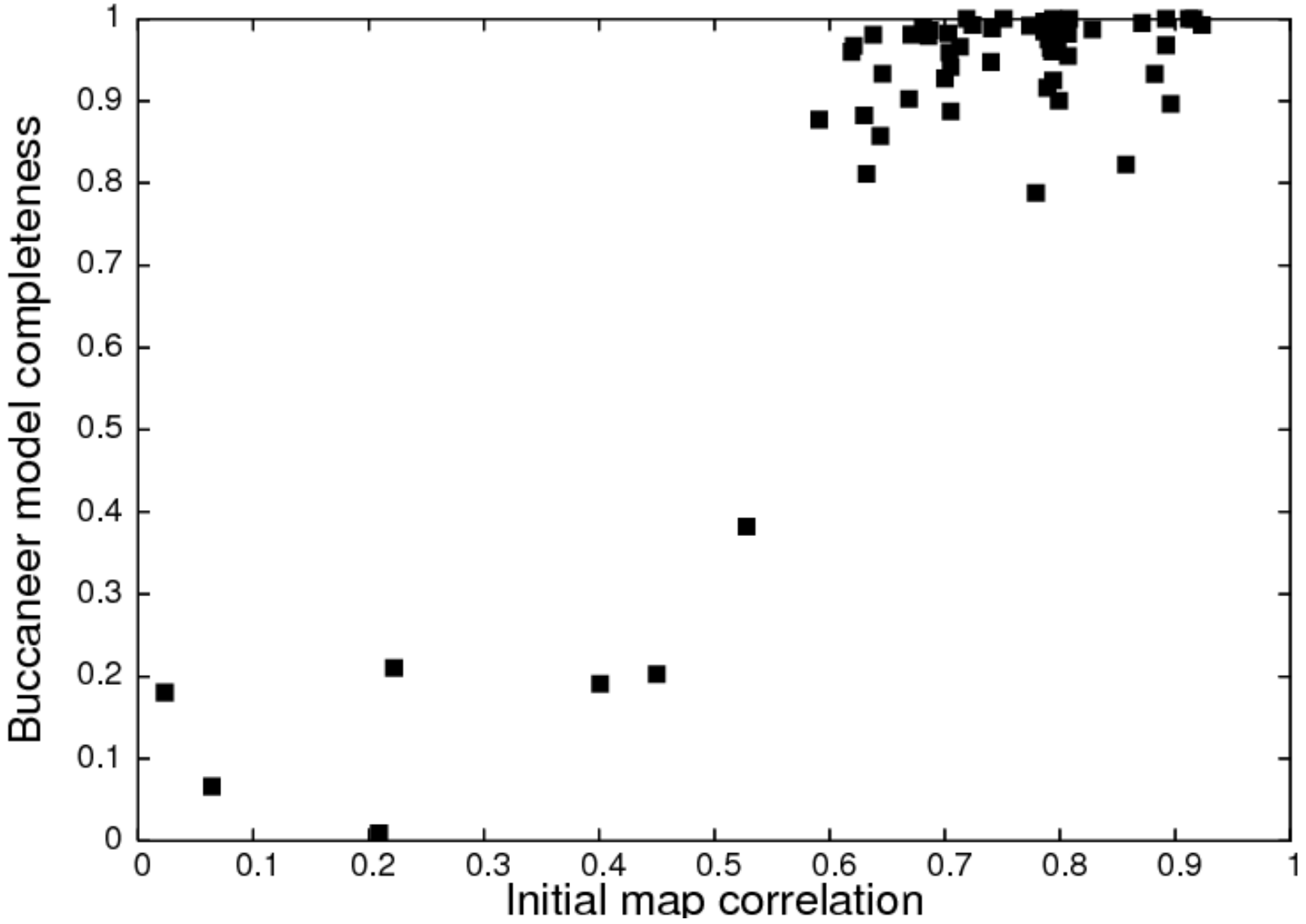
Buccaneer: Results

Model completeness not very dependent on resolution:

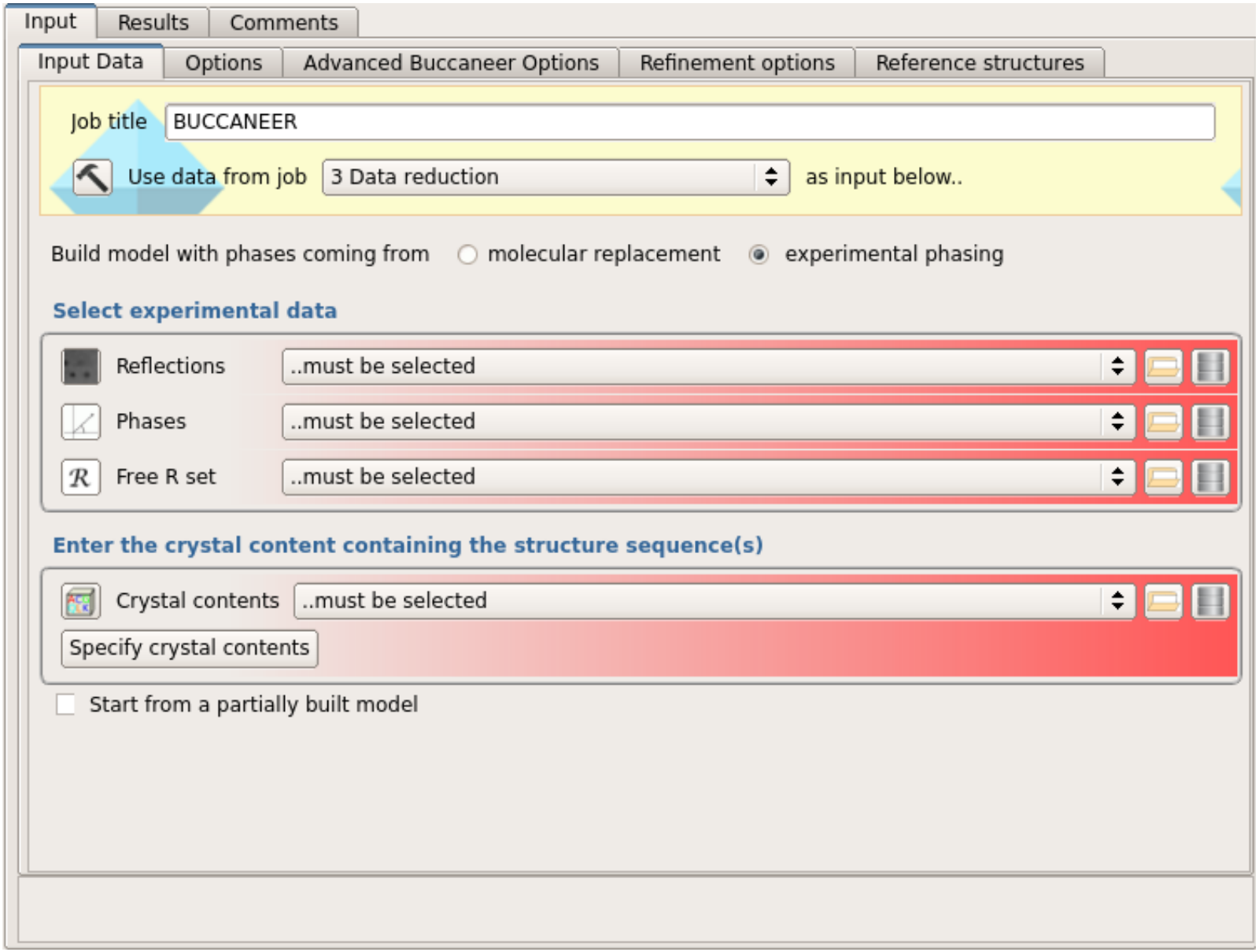


Buccaneer: Results

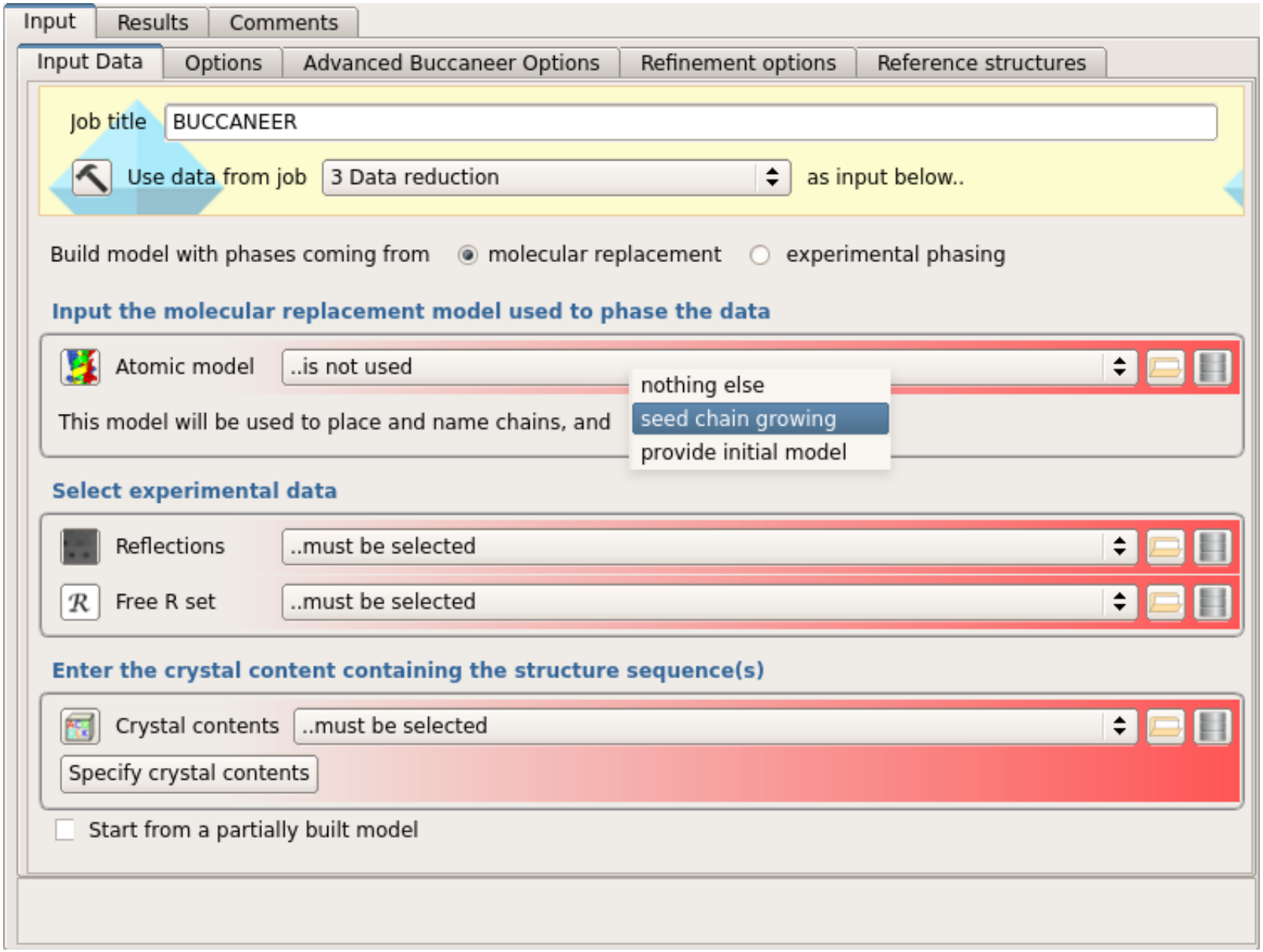
Model completeness dependent on initial phases:



Buccaneer: CCP4i2



Buccaneer: CCP4i2



Buccaneer: CCP4i2

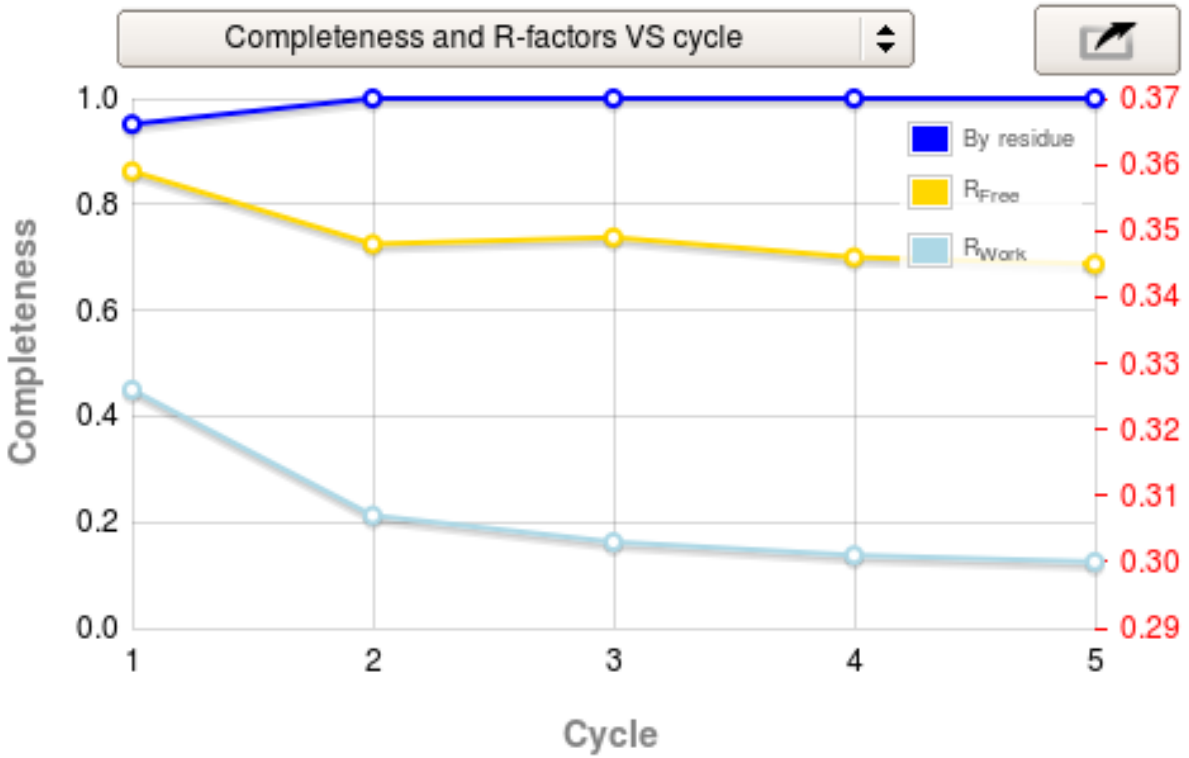
Results

118 residues were built in 2 fragments. Of these, 114 residues were assigned to the sequence.

The number of chains is estimated to be 1. Of these chains, 88.1% of the residues have been built.
Of the residues that were built, 100.0% were assigned to a chain.

The refinement R-factor is 0.30, and the free-R factor is 0.34. The RMS bond deviation is 0.017 A.
On the basis of the refinement statistics, the model is approaching completion.

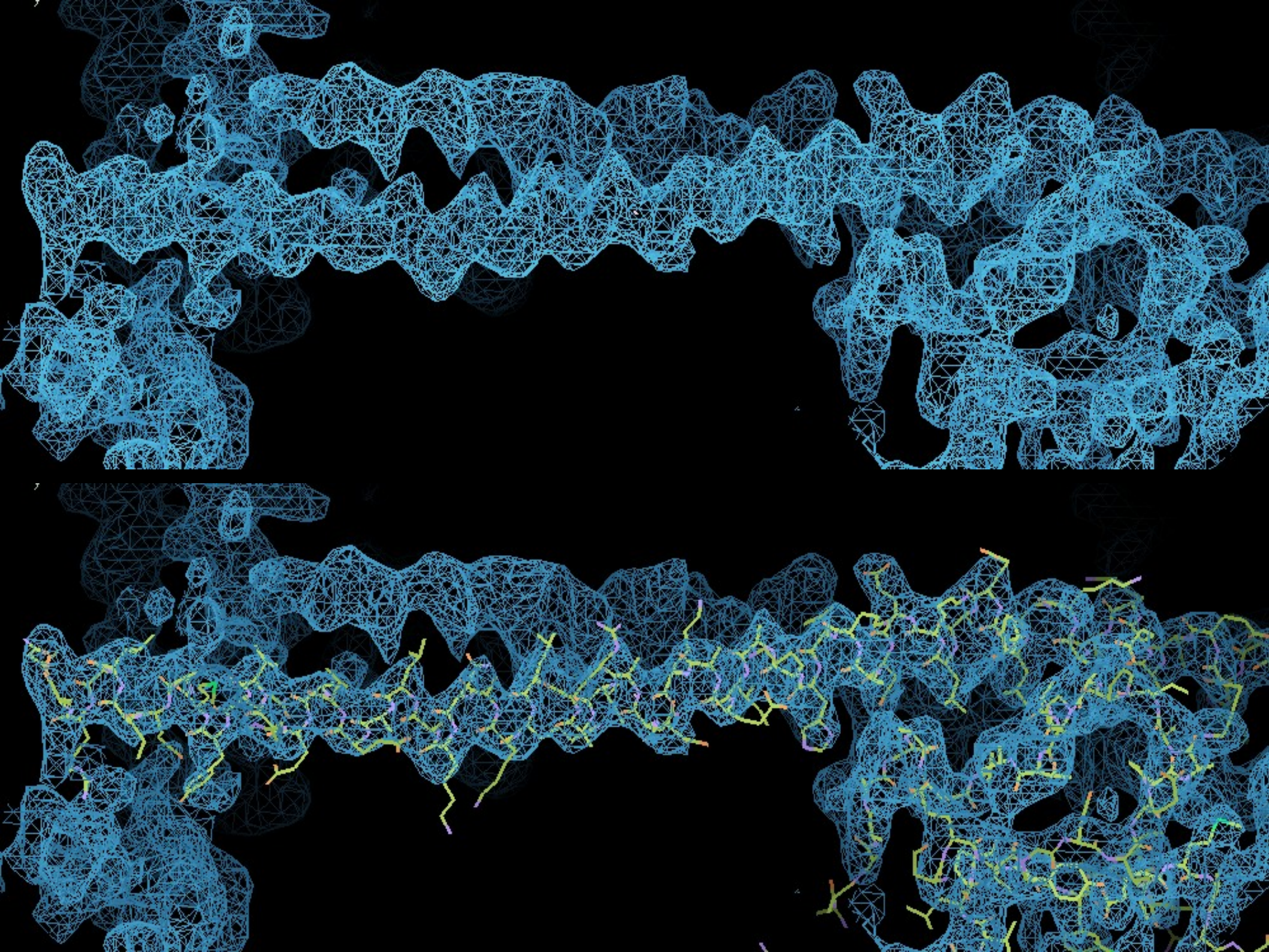
Completeness by residue	1.0
Completeness by chains	0.88
Number of chains	1
Residues built	118
Residues sequenced	114
Longest fragment	108
Number of fragments	2
R_{Work}	0.3
R_{Free}	0.345
RMS_{Bonds}	0.017
RMS_{Angles}	2.33



Buccaneer

What you need to do afterwards:

- Tidy up with Coot:
  - Connect up any broken chains.
  - Use density fit and rotamer analysis.
  - Check Ramachandran, molprobity, etc.
  - Add waters, ligands, check un-modeled blobs.
  - Re-refine, examine difference maps.
- If completion is very low:
  - Increase number of pipeline iterations.
  - Try using different options.
  - Pass partially built buccaneer model to ARP/wARP.



Buccaneer: Summary

- A simple, (i.e. MTZ and sequence), very fast method of model building which is robust against resolution.
- User reports for structures down to 3.7A when phasing is good.
- Results can be further improved by iterating with refinement in refmac (and in future, density modification).
- Proven on real world problems.
- Use it when resolution is poor or you are in a hurry. If resolution is good and phases are poor, then ARP/wARP may do better. Best approach: Run both!

Nautilus

- Automatic model building of nucleotide structures in electron density maps.

- Automated (CCP4i2) or interactive (Coot)

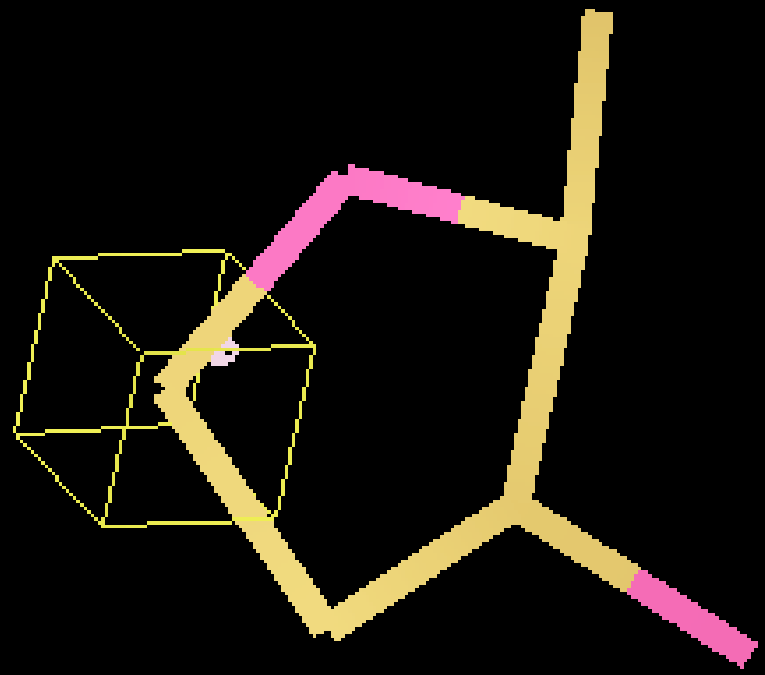
- Able to:
  - Start from an empty map
  - Extend an existing nucleotide model
  - Add nucleotide to a protein complex

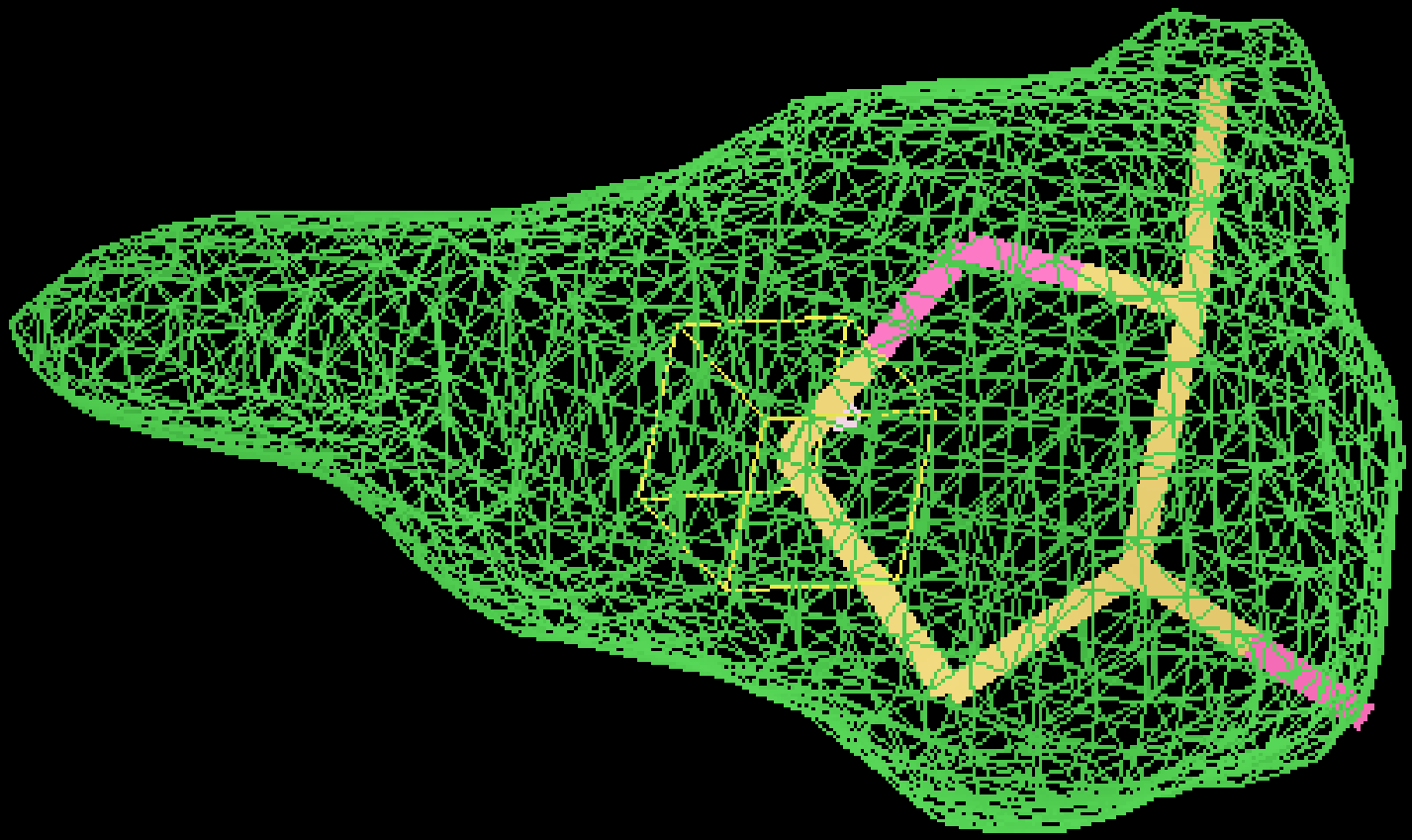
- K. Cowtan, IUCrJ (2014). **1**, 387-392 DOI

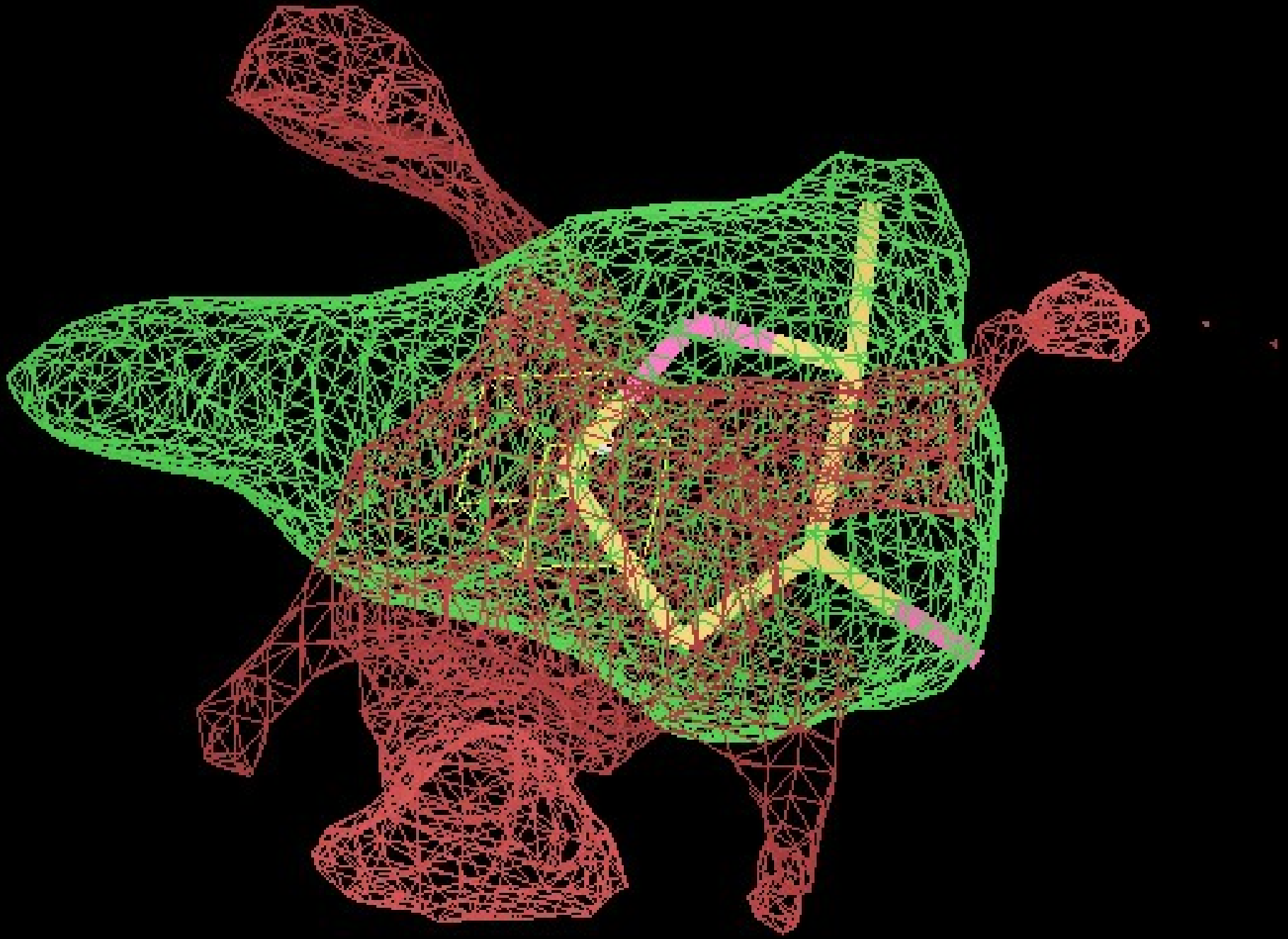
Nautilus

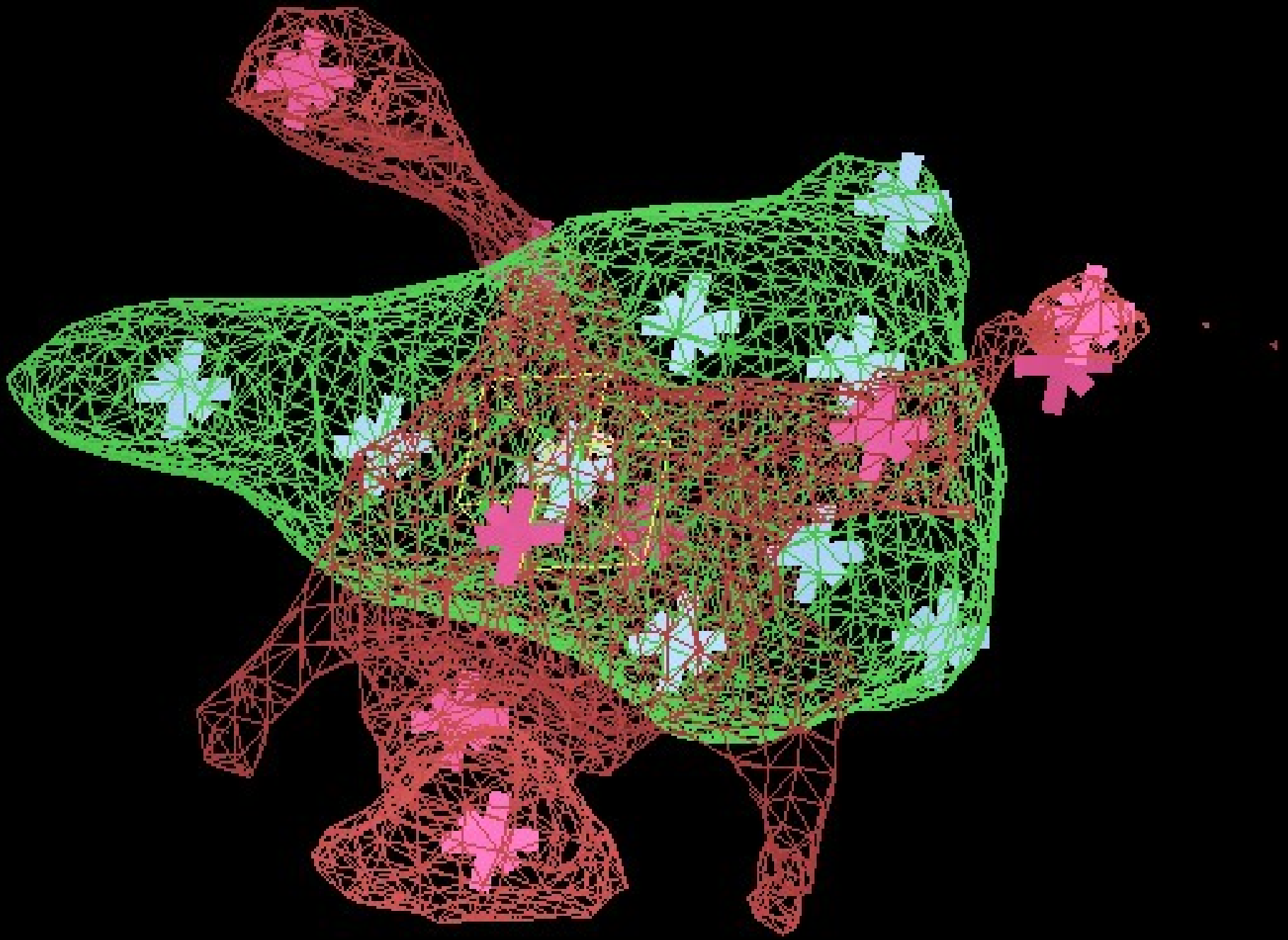
'Fingerprint' detection:

- Identify high and low density features consistent with the presence of nucleic acid features.
- Sugar / phosphate / base
- Very fast.
- Related to 'Essens' (Kleywegt and Jones), but with looks at both ridges and troughs.

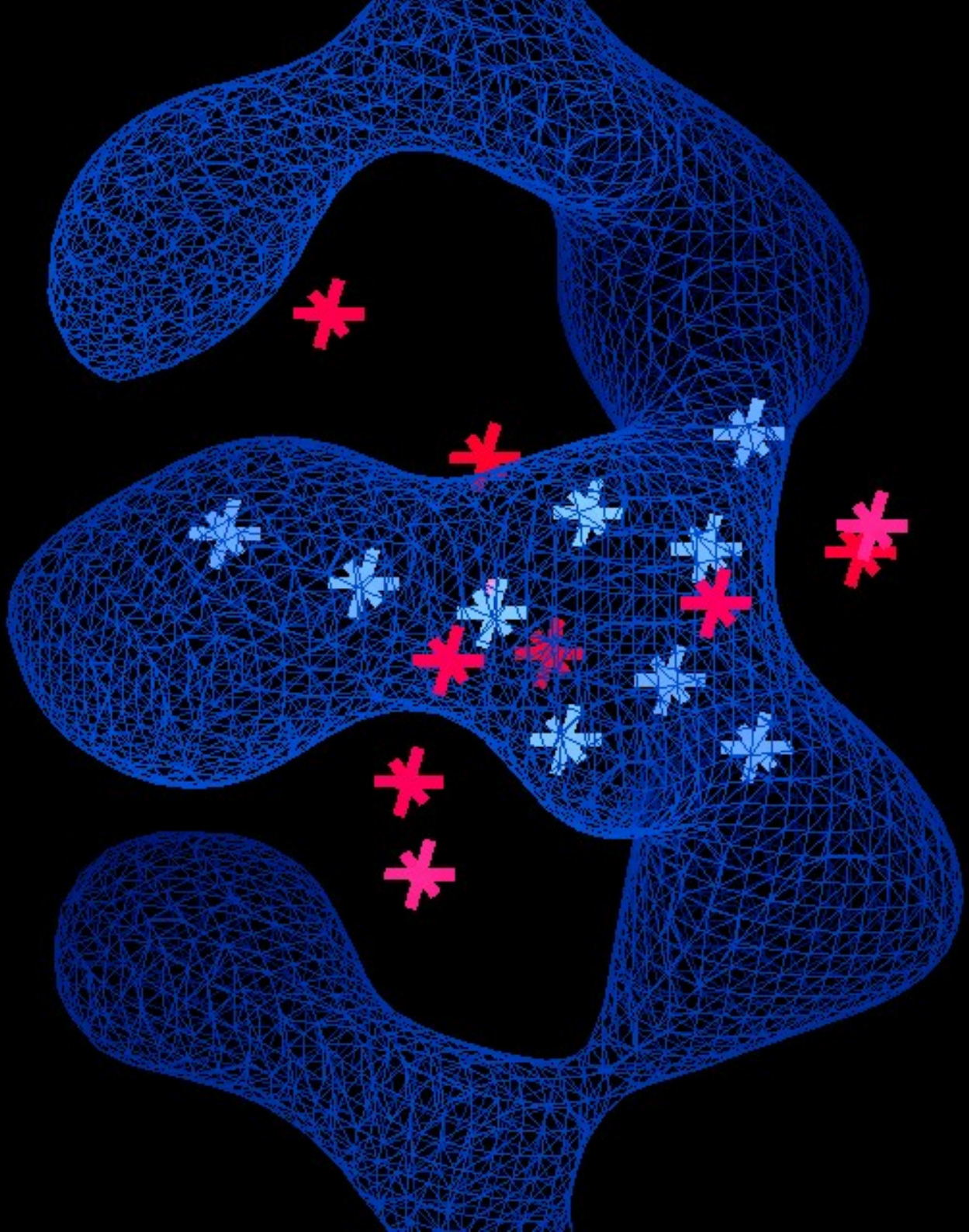




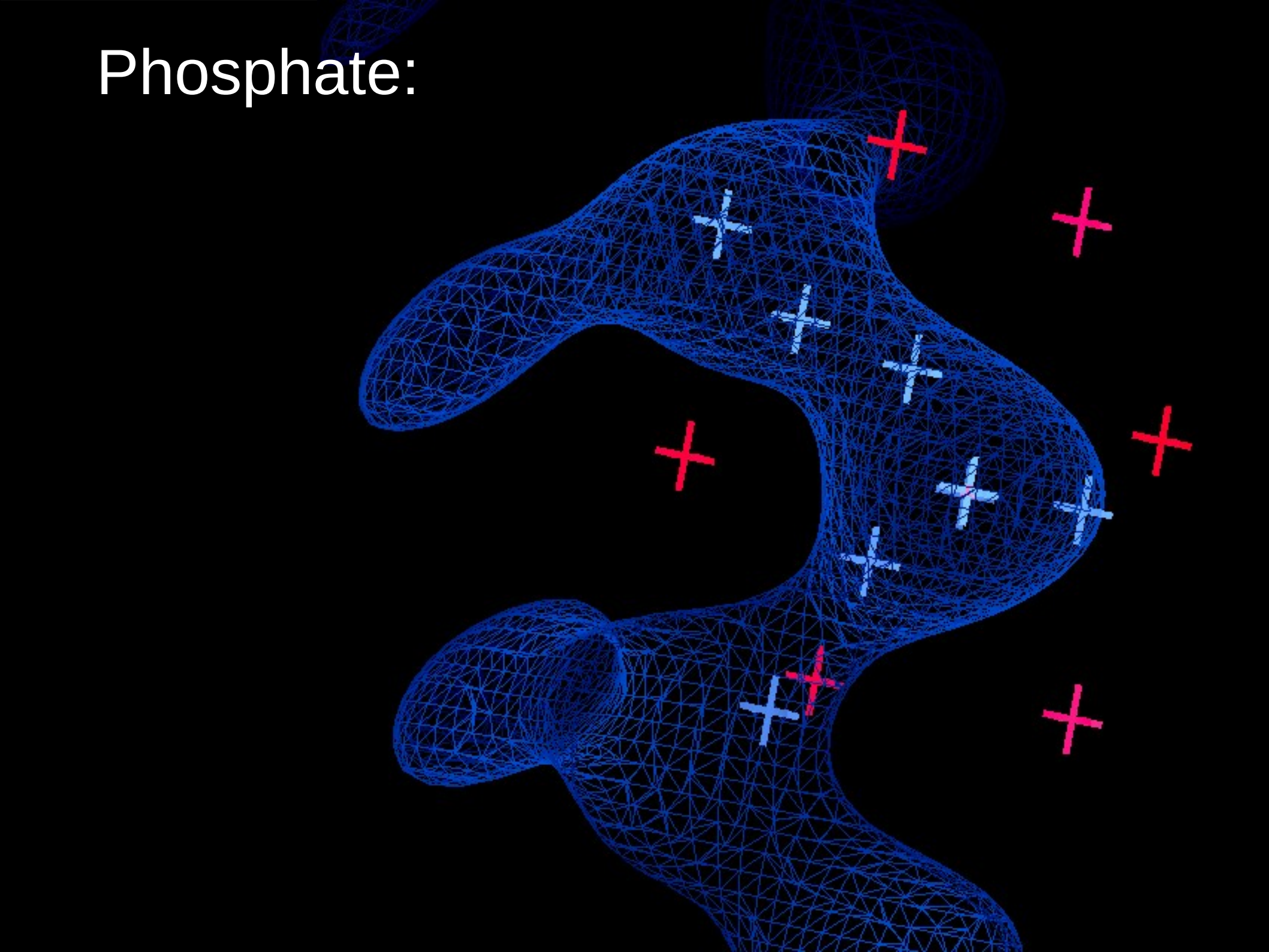




Sugar:



Phosphate:



Nautilus: Target Scoring

S-mean

Use the difference between the mean of the 'high' points and the mean of the 'low' points as a score indicating how likely it is the given group is present at a given position and orientation.

S-minmax

Need to search positions and orientations – a more optimized version of the same target uses the minimum of the highs minus the maximum of the lows – can often stop the calculation before testing all the sample points.

Nautilus

Steps:

- Find chain seeds
- Grow into chains
- Join overlapping chains
- Link nearby chains
- Prune clashing chains
- Rebuild chains to ensure connectivity
- Assign sequence
- Build bases

Nautilus

Find:

- Optimised 6-d rotation-translation using the sugar or phosphate fingerprint.
  - ~5 seconds for whole ASU
- Sugar:
  - Build a single nucleic acid using the best matching equivalent from the database, scored by 1 x sugar + 2 x phosphate fingerprints
- Phosphate:
  - Build a pair of nucleic acids using the best matching equivalent from the database, scored by 1 x phosphate + 2 x sugar fingerprints

Nautilus

Grow:

- Try adding additional nucleic acids to either end of each fragment, scored by the sugar fingerprint and the intermediate phosphate fingerprint.
  - ~1-2 second

Join:

- Merge overlapping fragments into longer fragments
  - <0.1 second

Link:

- Join fragments with nearby 3' and 5' terminii
  - ~0.5 second

Nautilus

Prune:

- Eliminate clashing regions
  - <0.1 second

Rebuild:

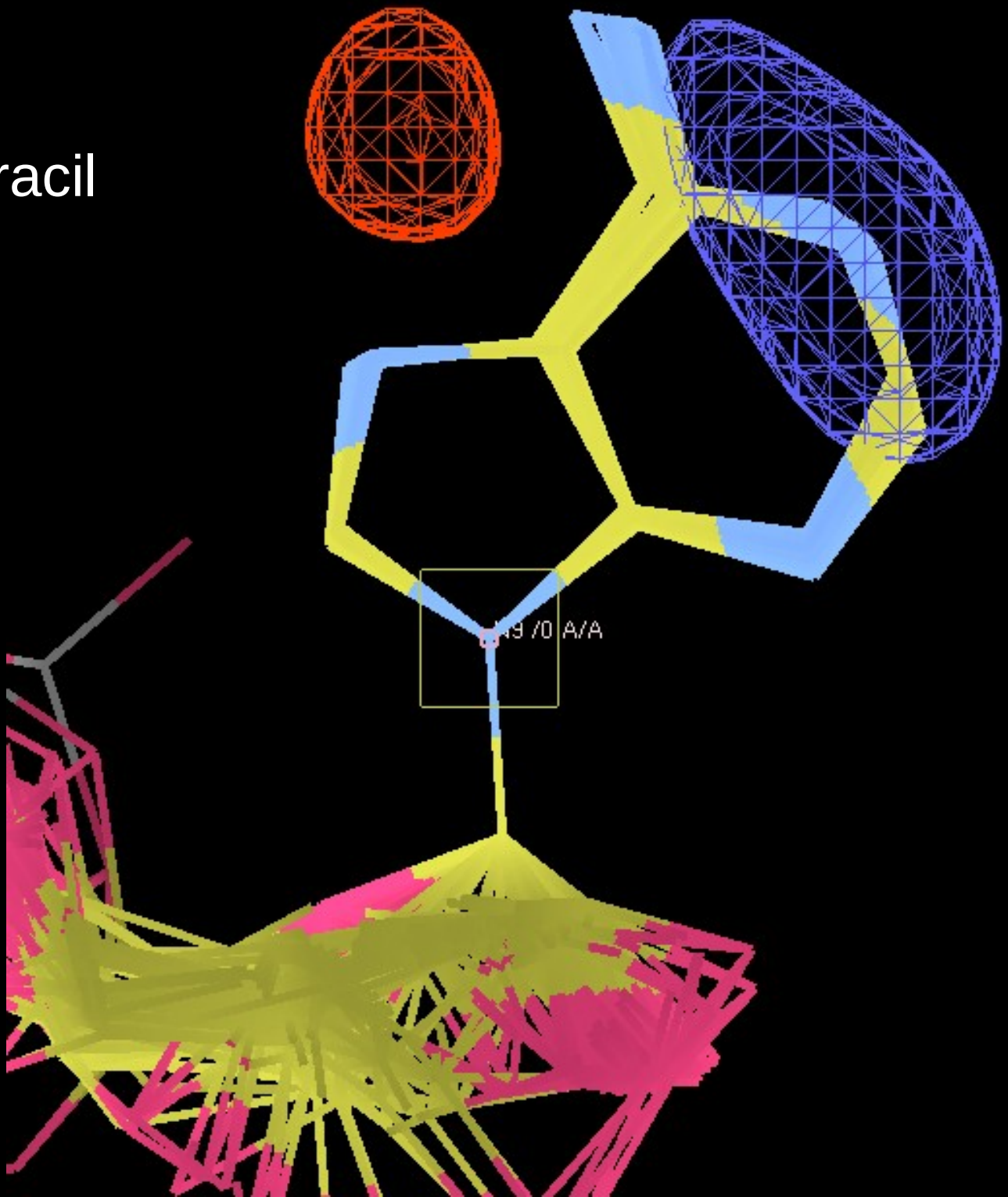
- Rebuild each sugar-sugar link using a fragment from the database
  - ~0.3 seconds

Sequence:

- Score base-type fingerprints at each position and assign sequence
  - <0.1 second

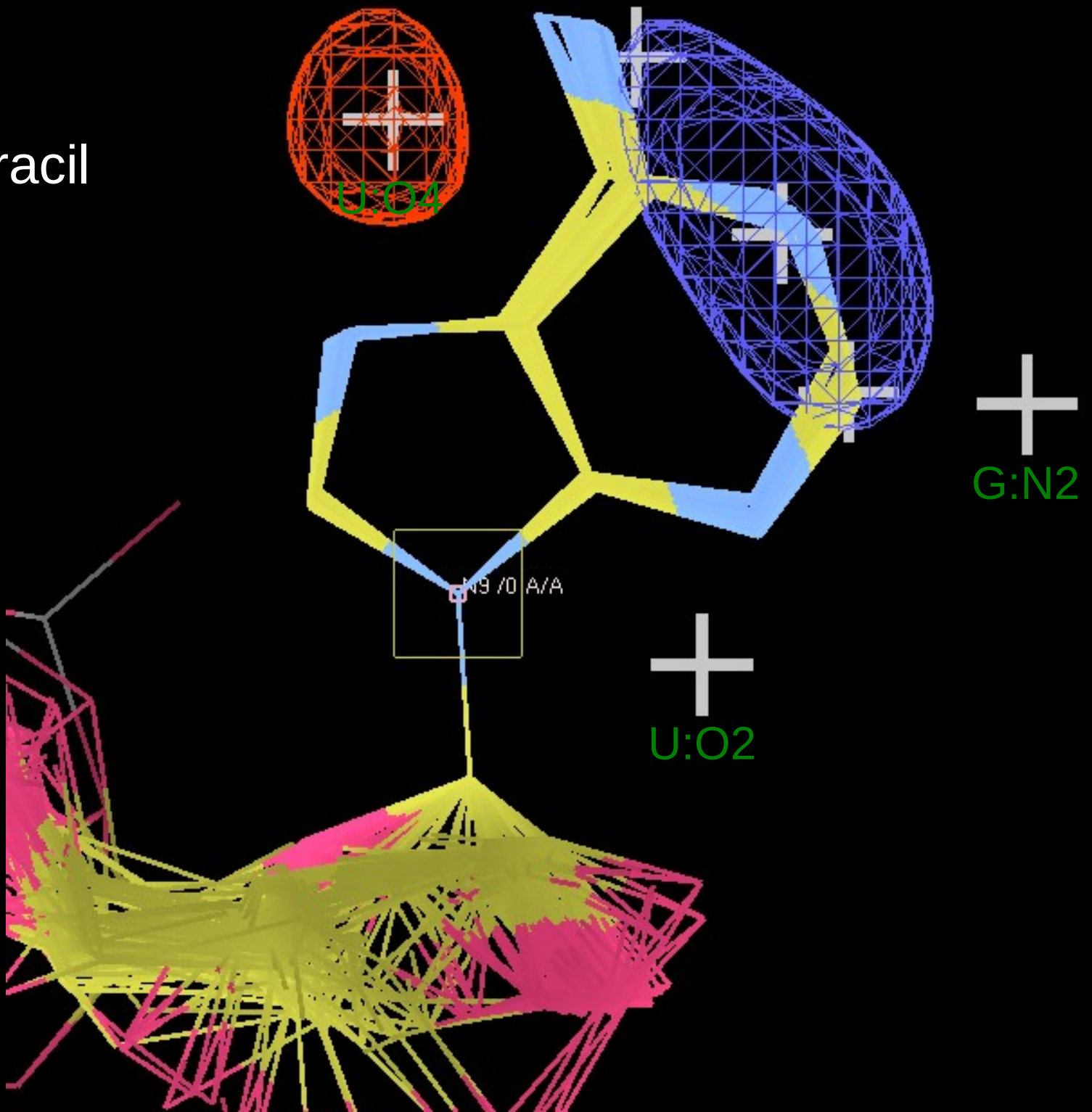
Base:

Adenine-Uracil



Base:

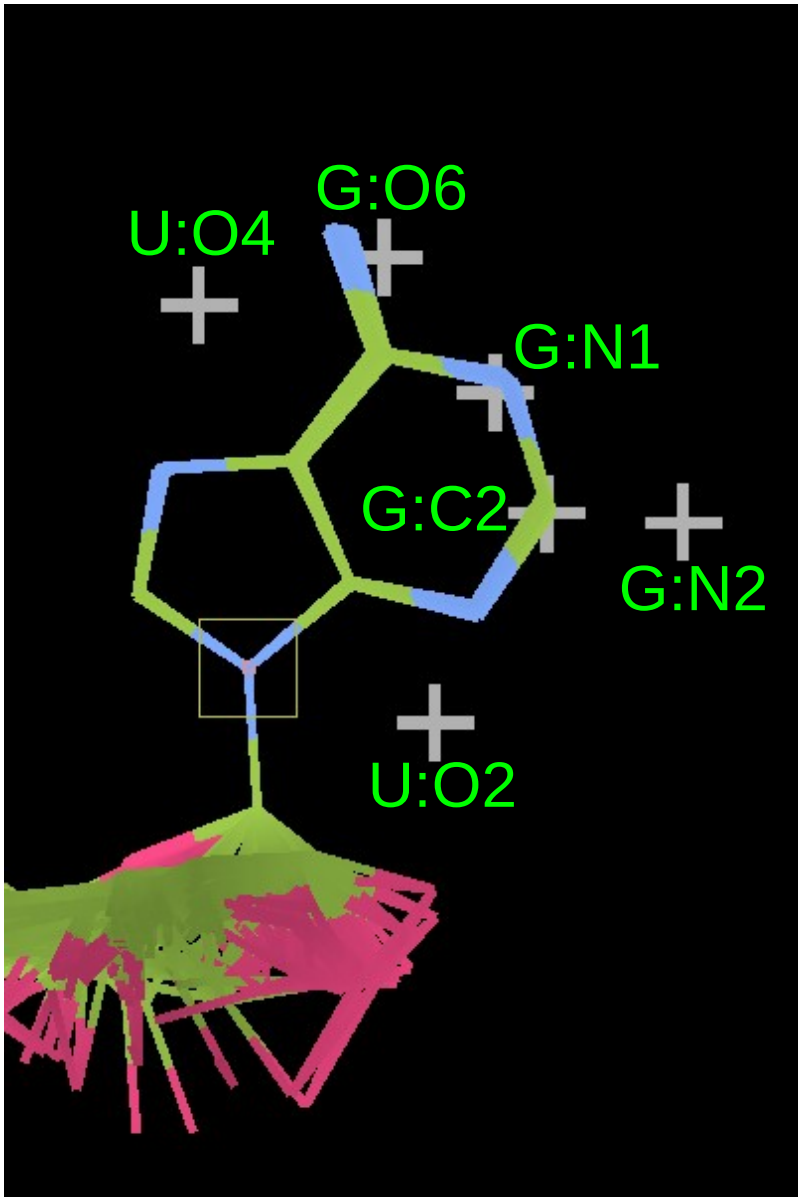
Adenine-Uracil



Nautilus

Adenine:

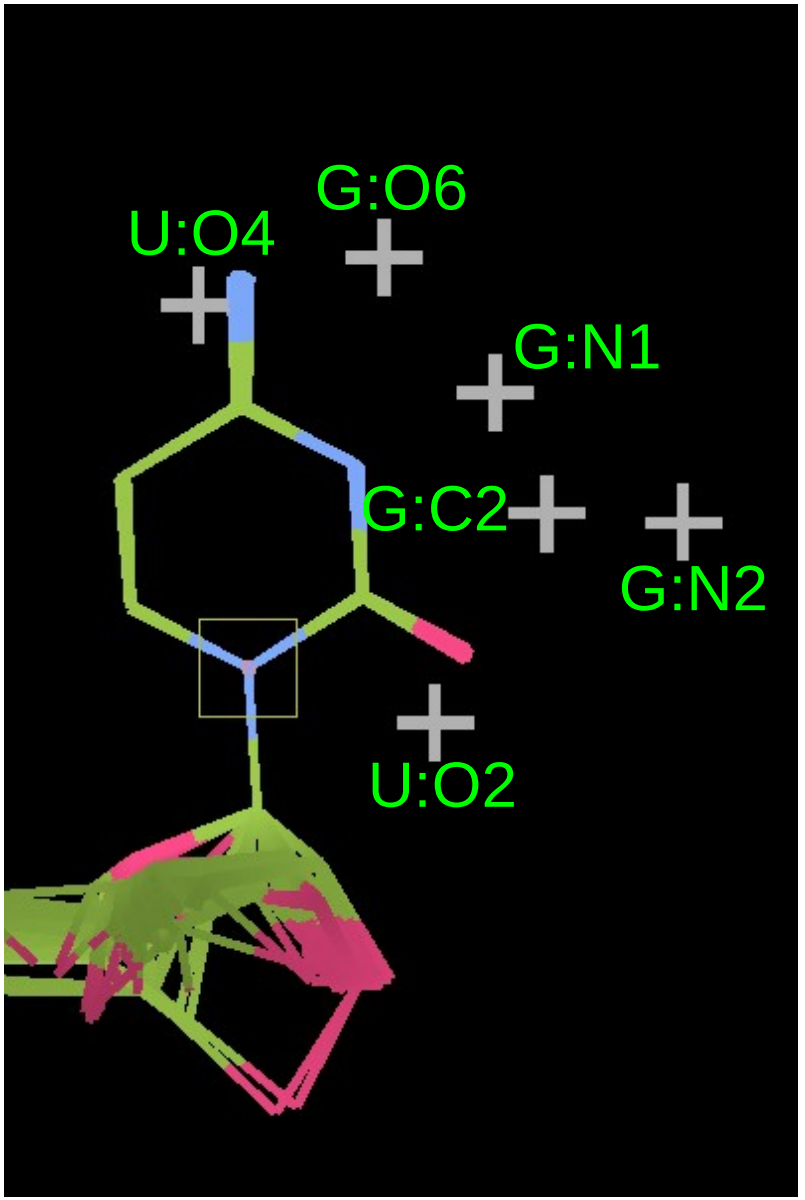
	U: O4	U: O2	G: O6	G: N1	G: C2	G: N2
A	-	-	+	+	+	-
C						
G						
U						



Nautilus

Cytosine:

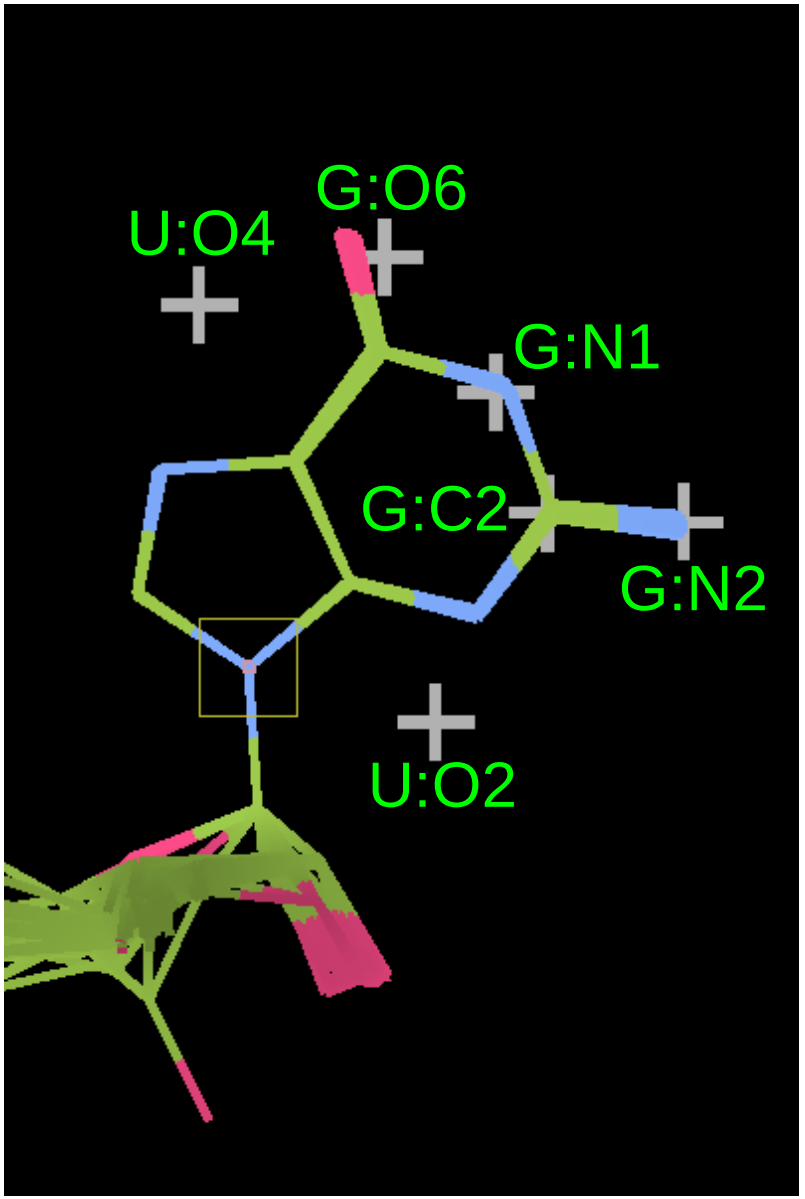
	U: O4	U: O2	G: O6	G: N1	G: C2	G: N2
A	-	-	+	+	+	-
C	+	+	-	-	-	-
G						
U						



Nautilus

Guanine:

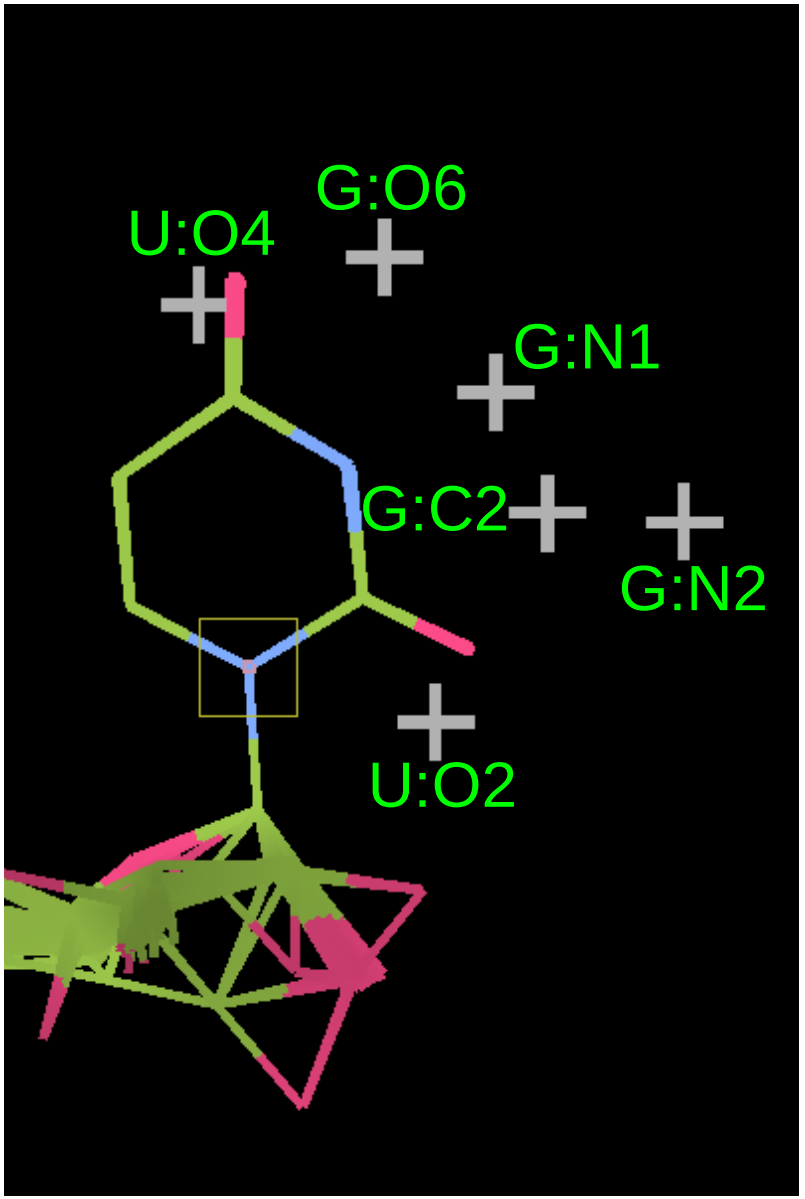
	U: O4	U: O2	G: O6	G: N1	G: C2	G: N2
A	-	-	+	+	+	-
C	+	+	-	-	-	-
G	-	-	+	+	+	+
U						



Nautilus

Uracil:

	U: O4	U: O2	G: O6	G: N1	G: C2	G: N2
A	-	-	+	+	+	-
C	+	+	-	-	-	-
G	-	-	+	+	+	+
U	+	+	-	-	-	-



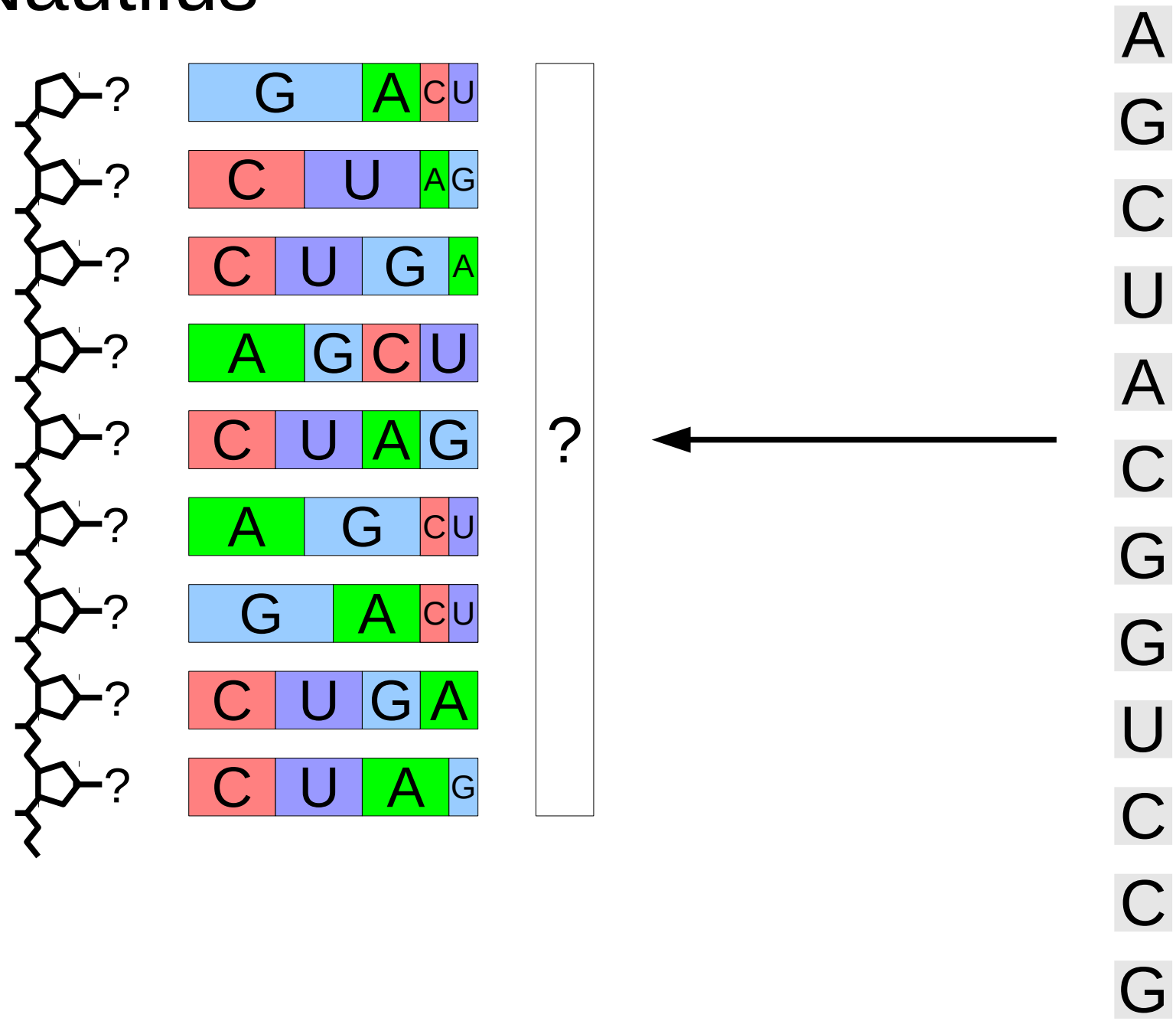
Nautilus

But the real world isn't black and white. Ideally we want a probability of a base being of a particular type.

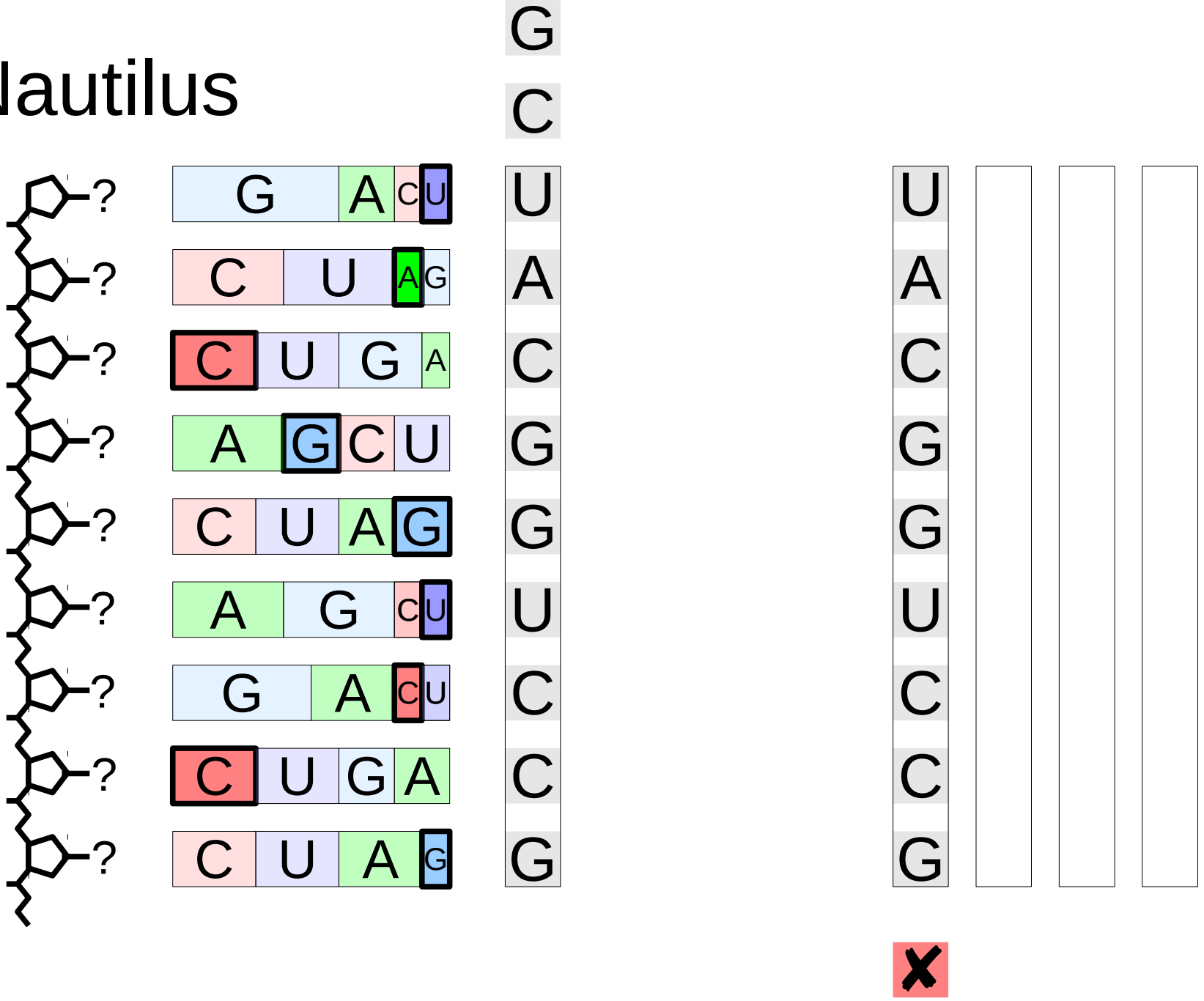
- Calculate z-scored densities for the density at each of the 6 sample positions for 200 bases (50 of each type), to form a sample database.
- Calculate z-scored densities for the 6 sample positions of the unknown base.
- Find the 50 closest matches to the unknown base from the database.
- Assign probability of being A/C/G/U on the basis of the proportion of the 50 closest matches being of each type (+ an error term).

Google: k-NN (k-Nearest Neighbour)

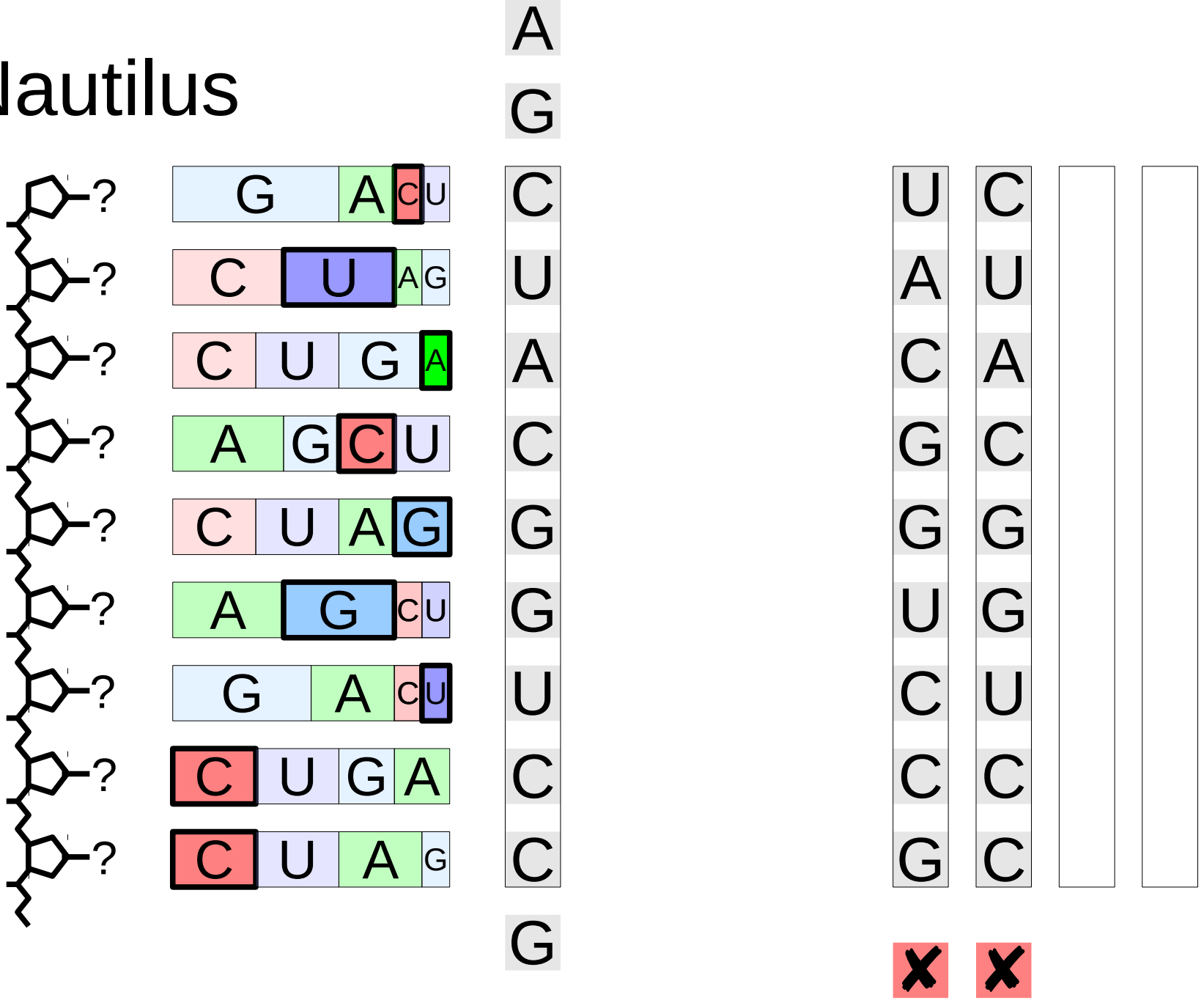
Nautilus



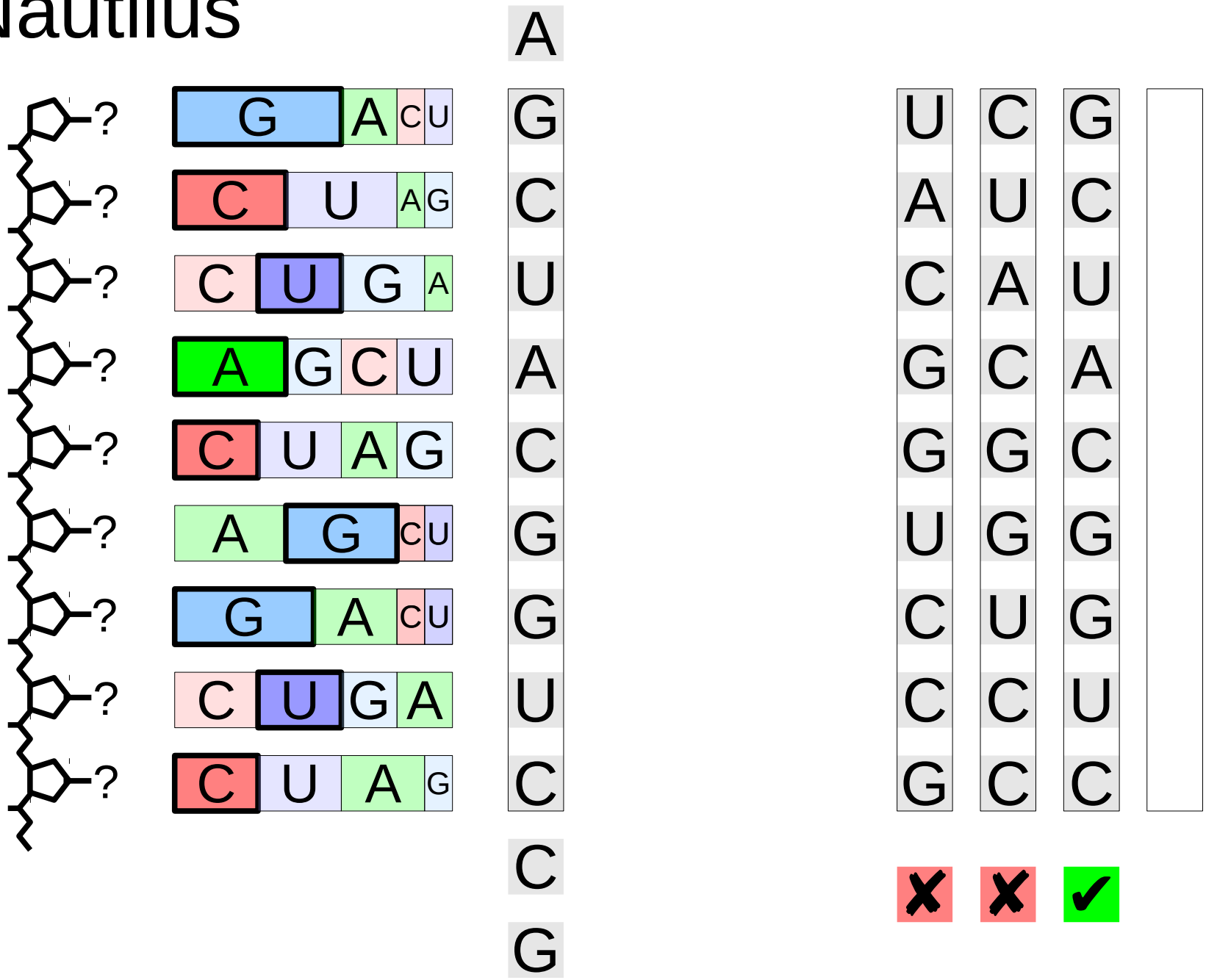
Nautilus


G
C
U
A
C
G
G
U
C
C
G
U
A
C
G
G
U
C
C
G
?
?
?
?
?
?
?
?
?
?
G A C U
C U A G
C U G A
A G C U
C U A G
A G C U
G A C U
C U G A
C U A G

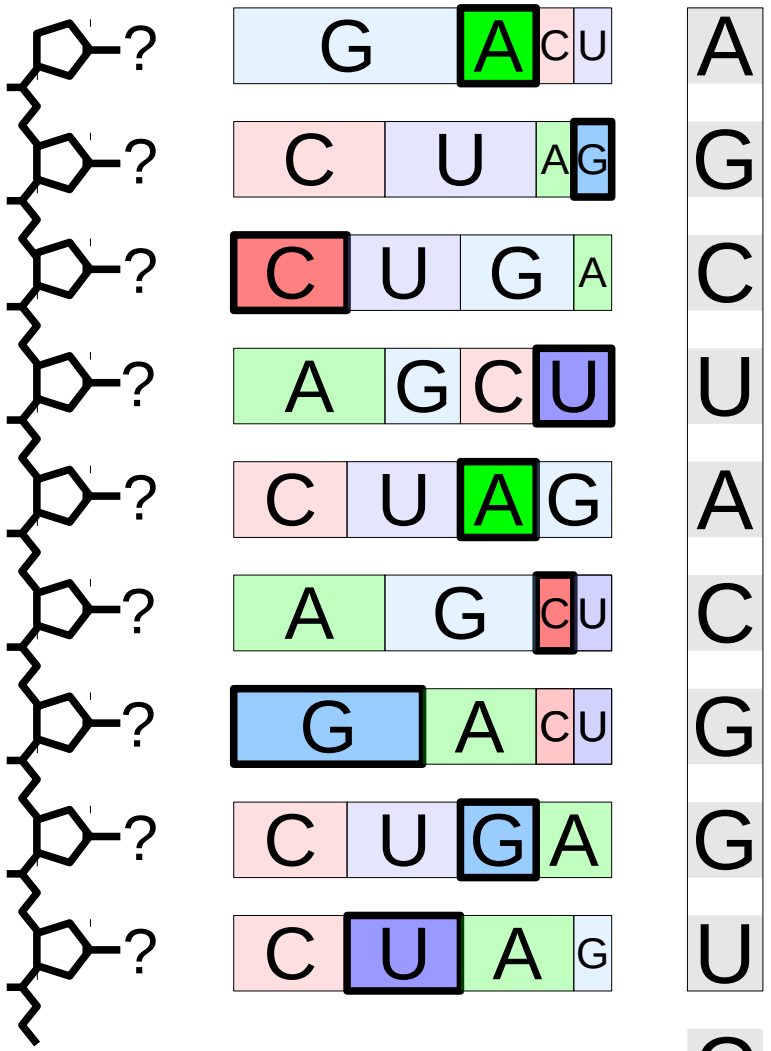
Nautilus


?
?
?
?
?
?
?
?
?
G A C U
C U A G
C U G A
A G C U
C U A G
A G C U
G A C U
C U G A
C U A G
A
G
C
U
A
C
G
G
U
C
C
G
U C
A U
C A
G C
G G
U G
C U
C C
G C

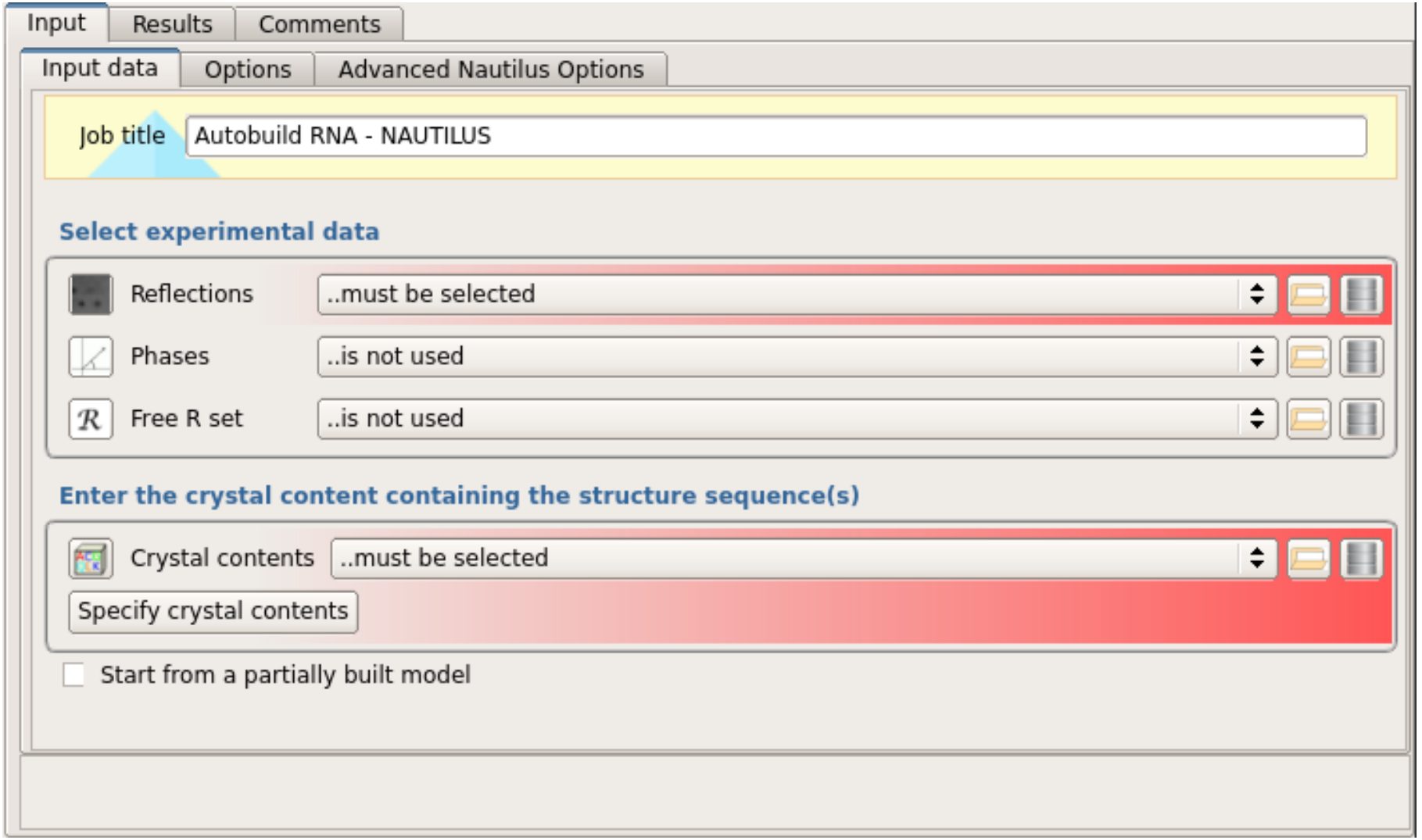
Nautilus



Nautilus


?
?
?
?
?
?
?
?
?
G A C U
C U A G
C U G A
A G C U
C U A G
A G C U
G A C U
C U G A
C U A G
A
G
C
U
A
C
G
G
C
C
G
U C G A
A U C G
C A U C
G C A U
G G C A
U G G C
C U G G
C C U G
G C C U
✘ ✘ ✔ ✘

Nautilus: CCP4i2



Nautilus

Results:

- Good results on synthetic noisy data at 3.5A and user reports on real data at 3.8A.
  - Need more data
- Like '*buccaneer*', phases are more important than resolution.
- Failed on a quadruplex structure with good phases.
  - Try a different database?

Acknowledgments

Help:

- JCSG data archive: www.jcsg.org
- Garib Murshudov, Raj Pannu, Pavol Skubak
- Eleanor Dodson, Paul Emsley, Randy Read, Clemens Vonrhein

Funding:

- The Royal Society, BBSRC