Automated model building with **Buccaneer**

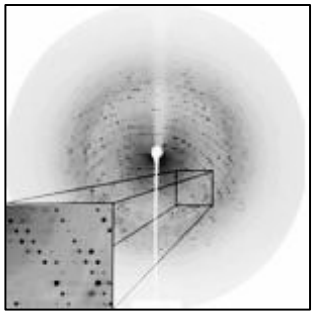
The buccaneer software for automated model building of protein structures across a broad range of resolutions.

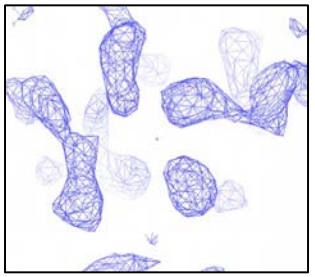
Kevin Cowtan

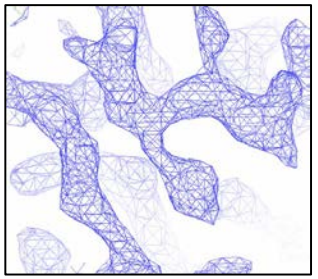
YSBL,University of York

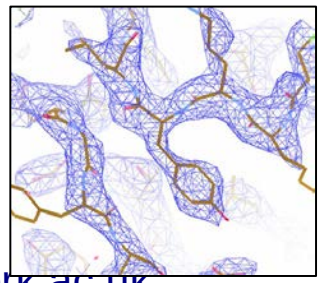
kevin.cowtan@york.ac.uk

X-ray structure solution pipeline...









Data collection

Data processing

Experimental phasing

Molecular Replacement

Density Modification

Model building

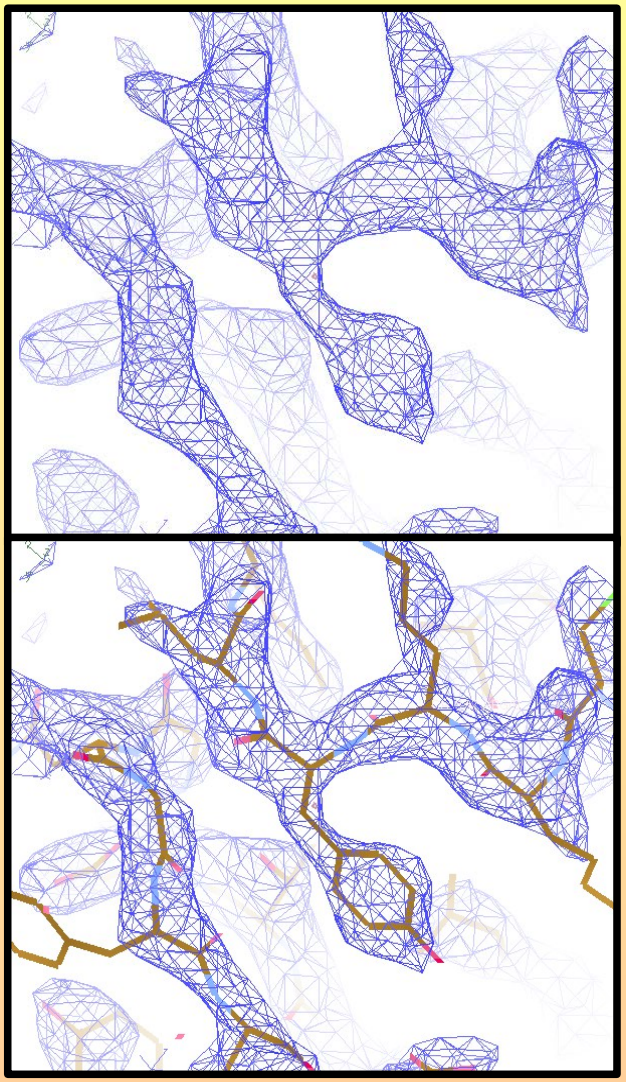
Refinement

Rebuilding Validation

Model Building

Model building software:

- Buccaneer



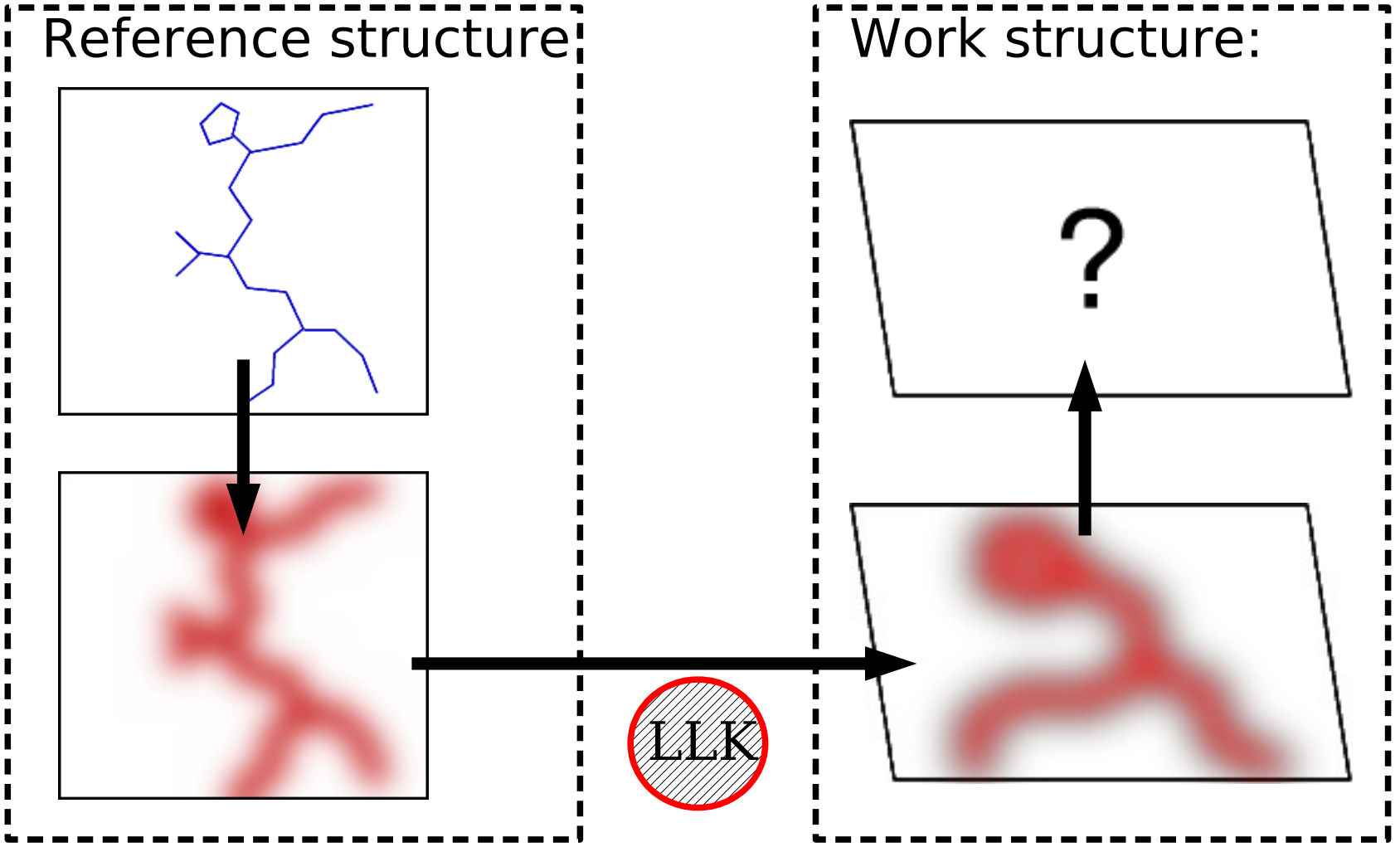
Buccaneer

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

- 1.0 release in 2008
  - Tracing, sequencing, ncs and refinement
- 1.2 release (1.1.9) in CCP4 6.1.2
  - Faster, better sequencing
- 1.4 release in CCP4 6.1.13
  - MR support, known structure, command line
- 1.5 release in CCP4 6.2
  - Model tidying
  - Minor sequencing improvements
- 1.6 release in CCP4 6.3(4)
  - Different MR modes

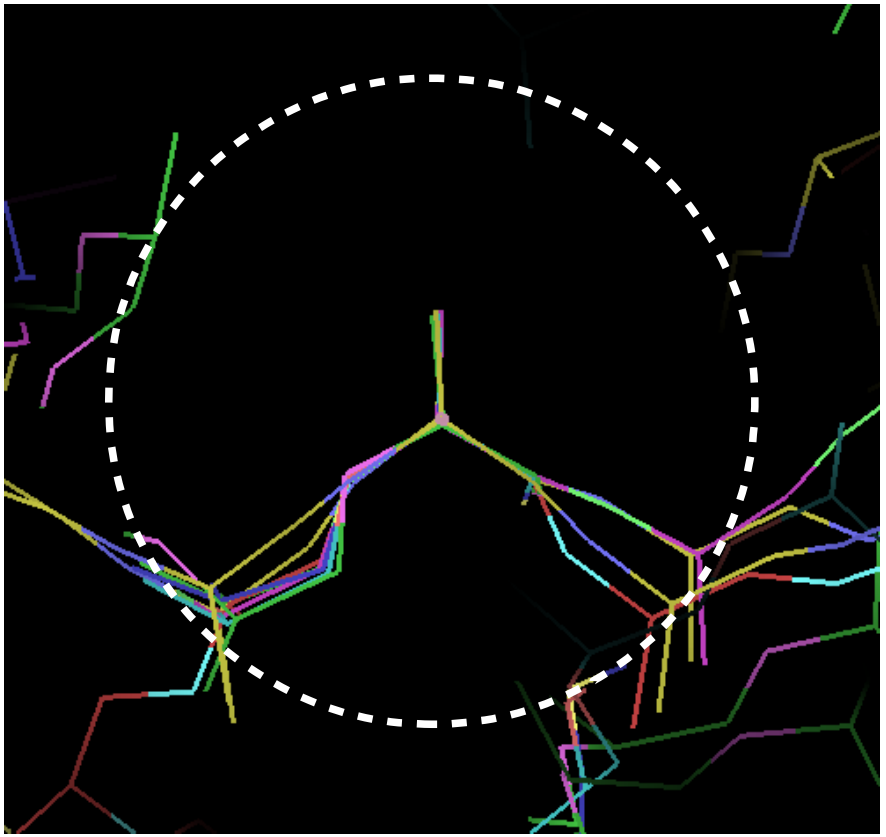
Buccaneer: Method

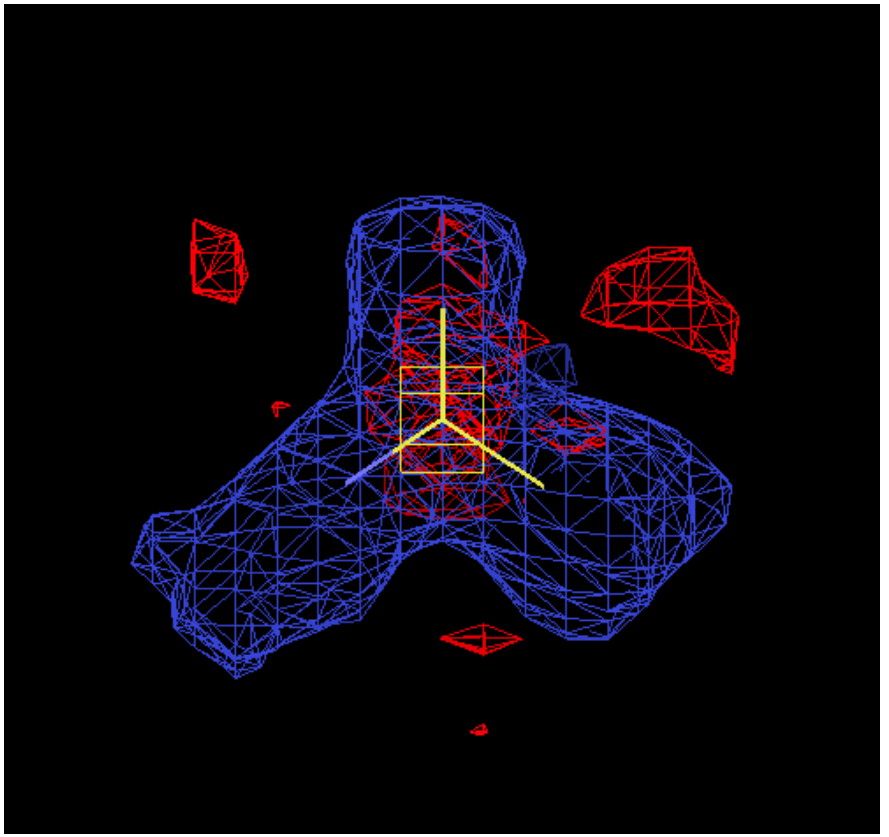
- Compare simulated map and known model to obtain likelihood target, then search for this target in the unknown map.



Buccaneer: Method

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





- Use mean/variance.

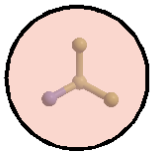
4Å sphere about Cα also used by 'CAPRA' loeger et al. (but different target function).

Buccaneer

Use a likelihood function based on conserved density features.

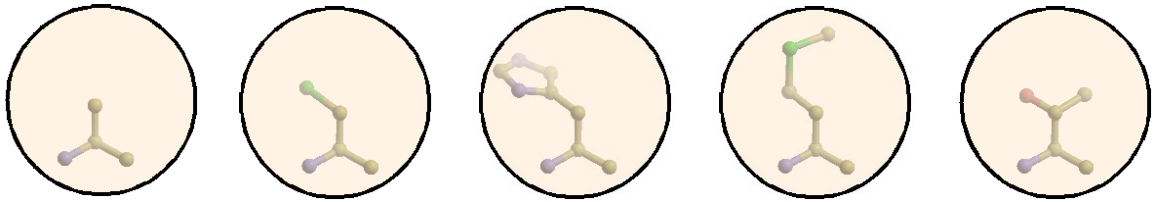
The same likelihood function is used several times. This makes the program very simple (<3000 lines), and the whole calculation works over a range of resolutions.

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

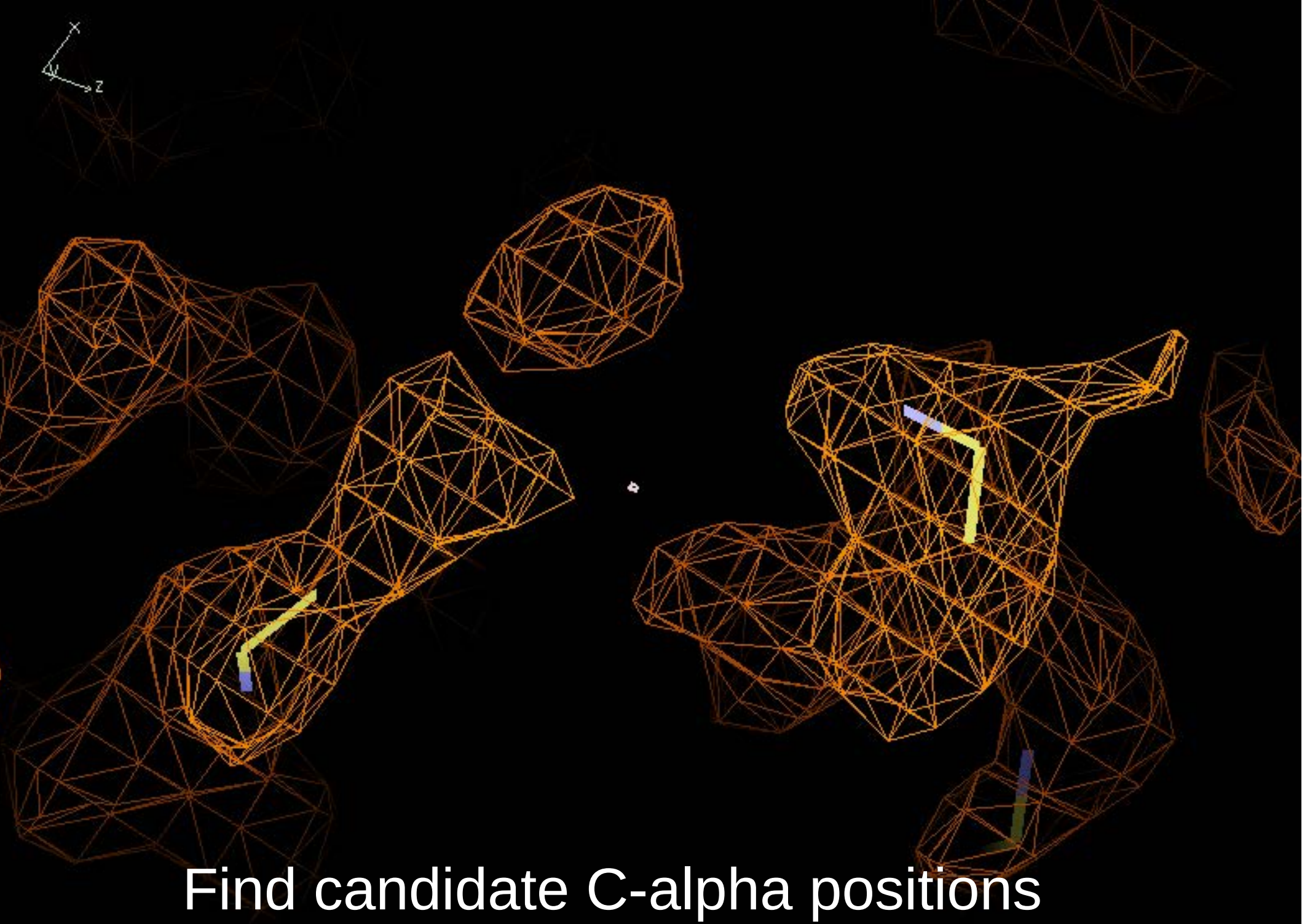
10 stages:

- **Find** candidate C-alpha positions
- **Grow** them into chain fragments
- **Join** and merge the fragments, resolving branches
- **Link** nearby N and C terminii (if possible)
- **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

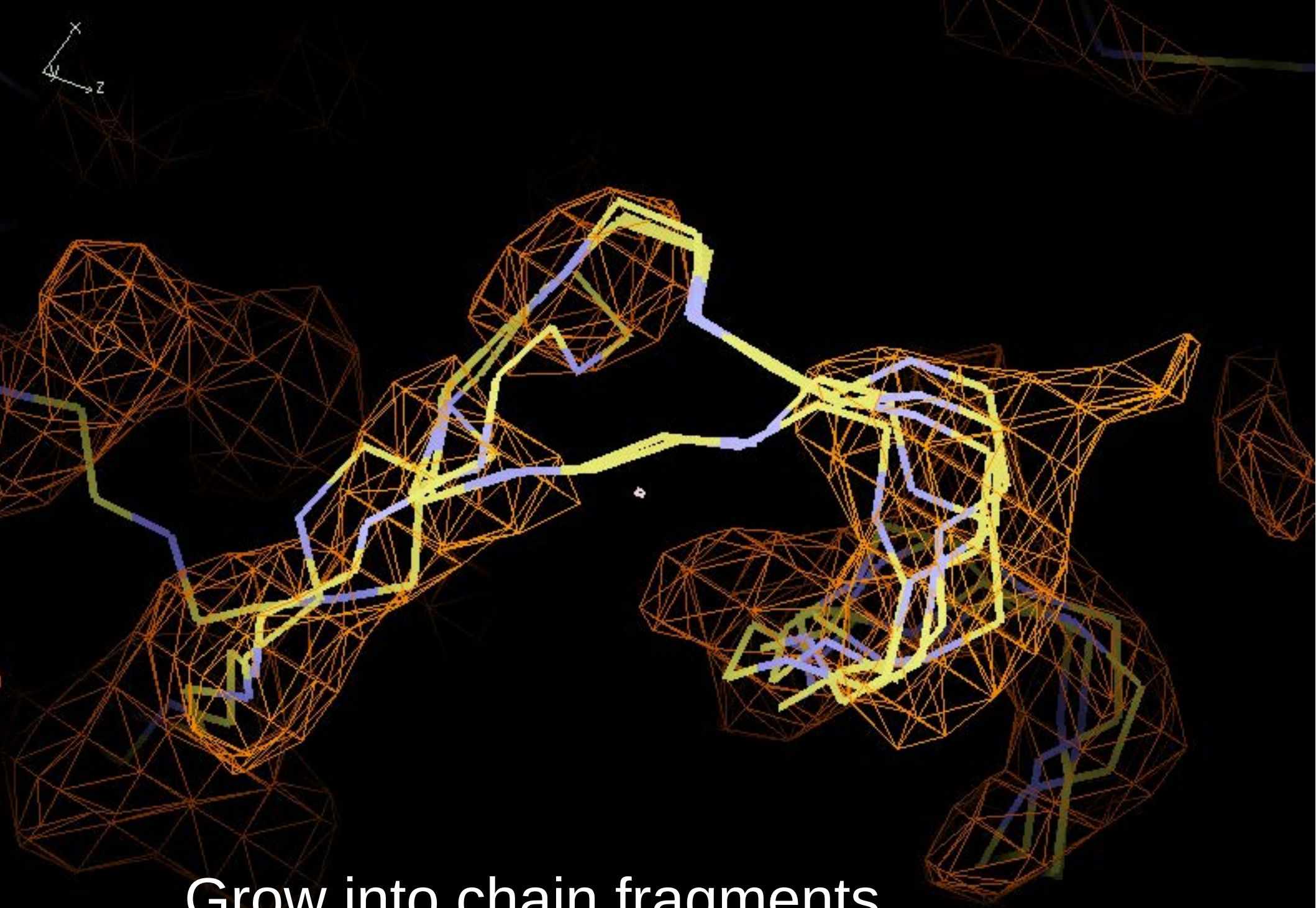
Buccaneer

Case Study:

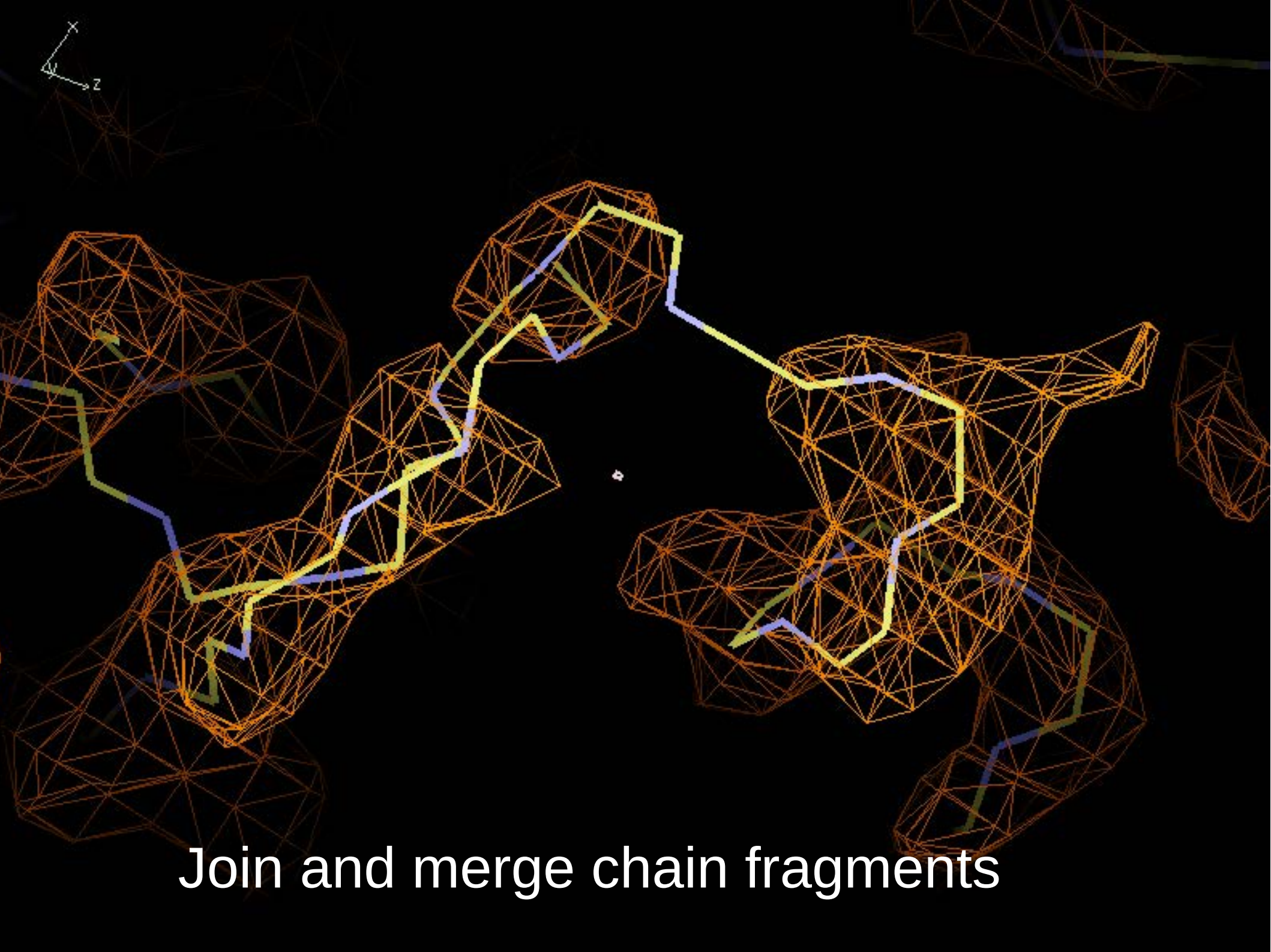
A difficult loop in a 2.9Å map, calculated using real data from the JCSG.

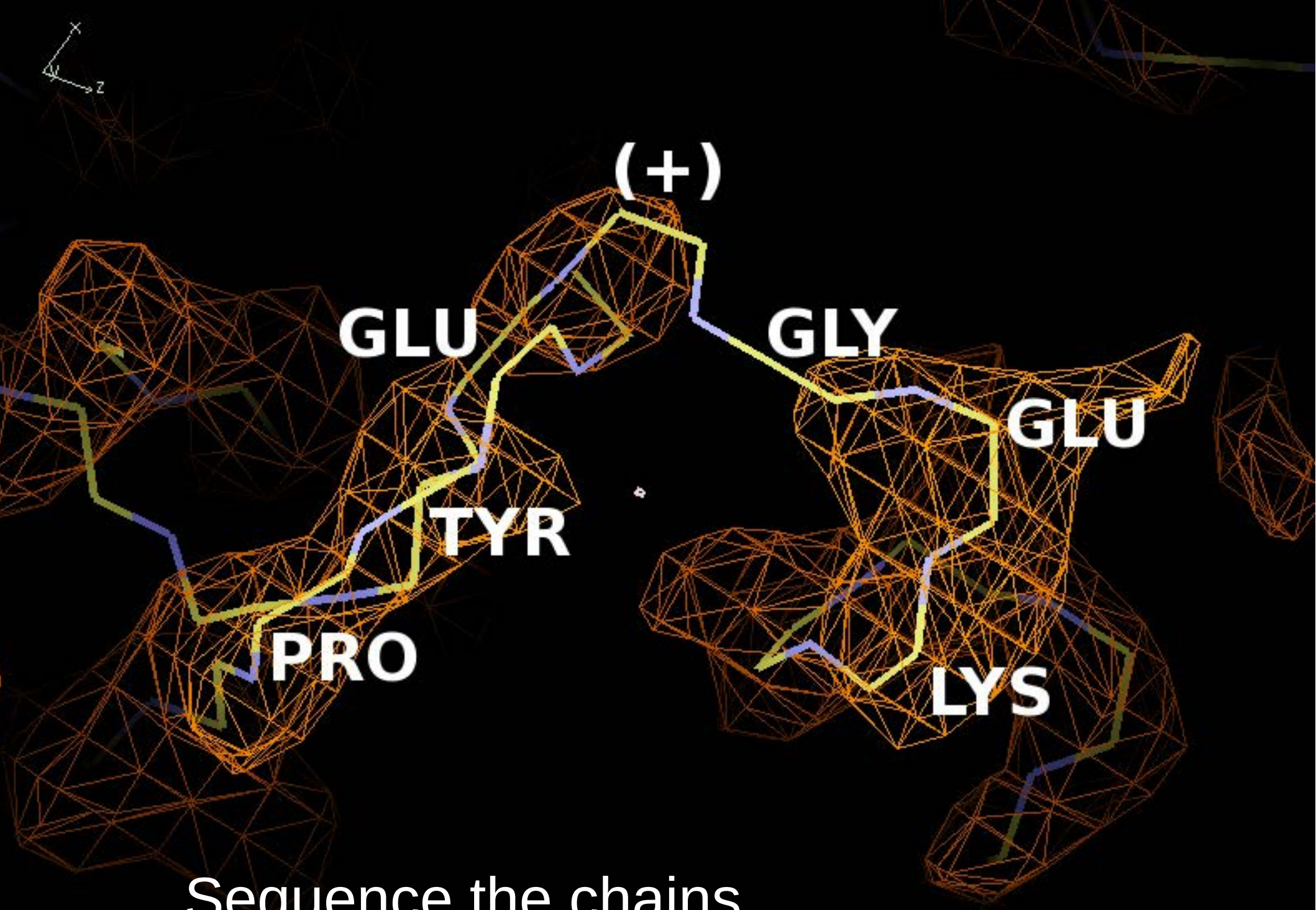


Find candidate C-alpha positions

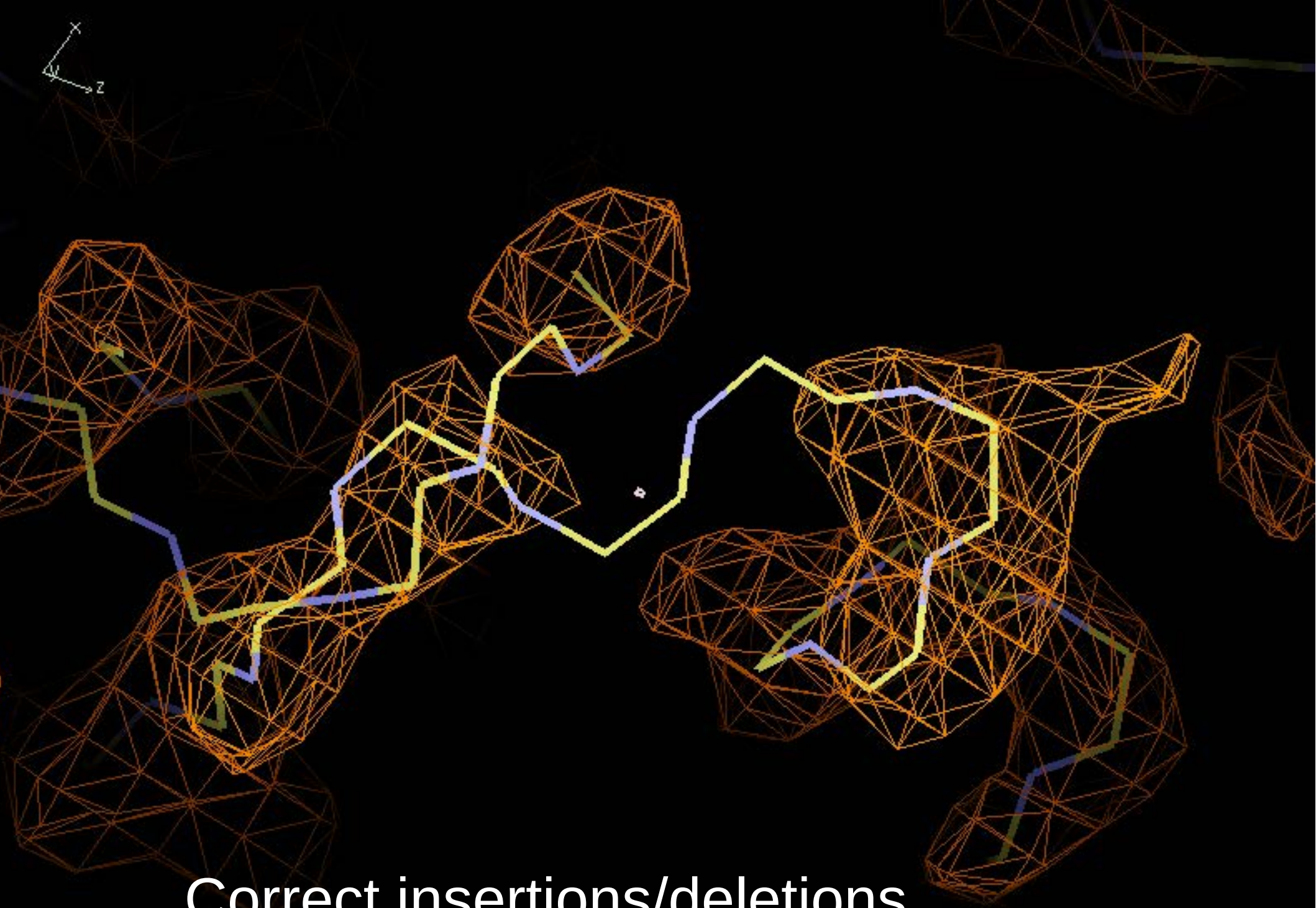


Grow into chain fragments


Join and merge chain fragments

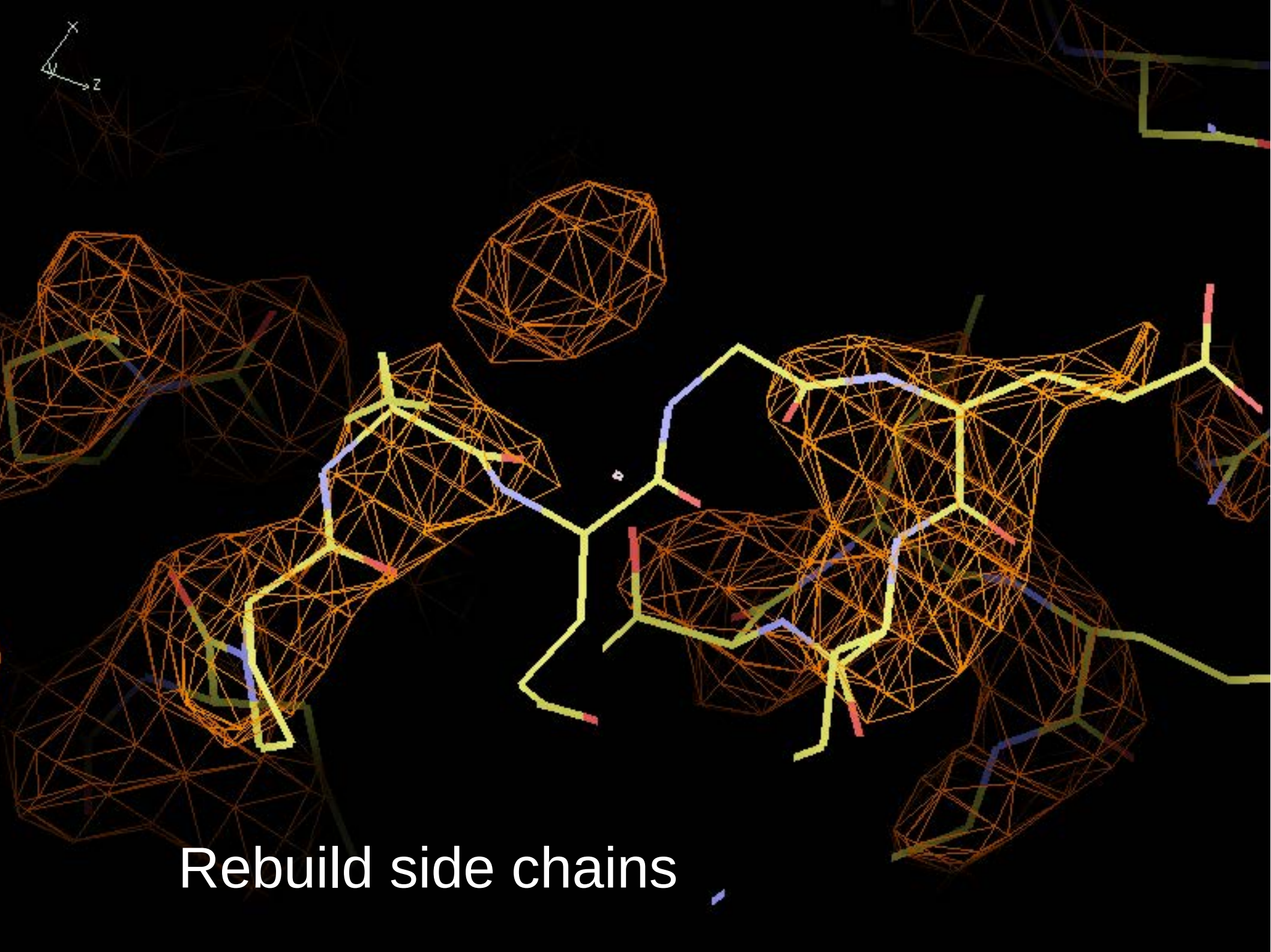


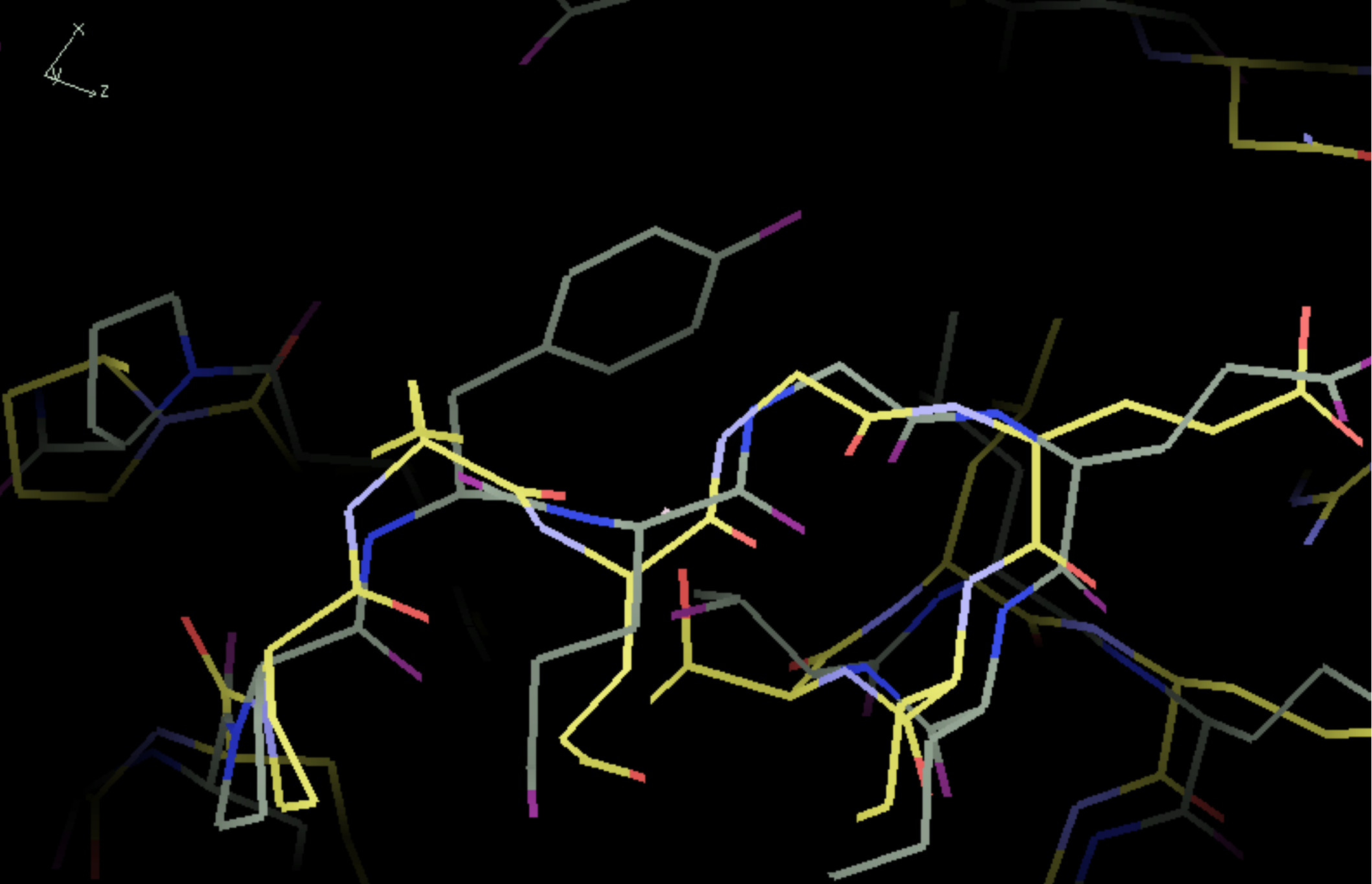
Sequence the chains



Correct insertions/deletions

Prune any remaining clashing chains


Rebuild side chains

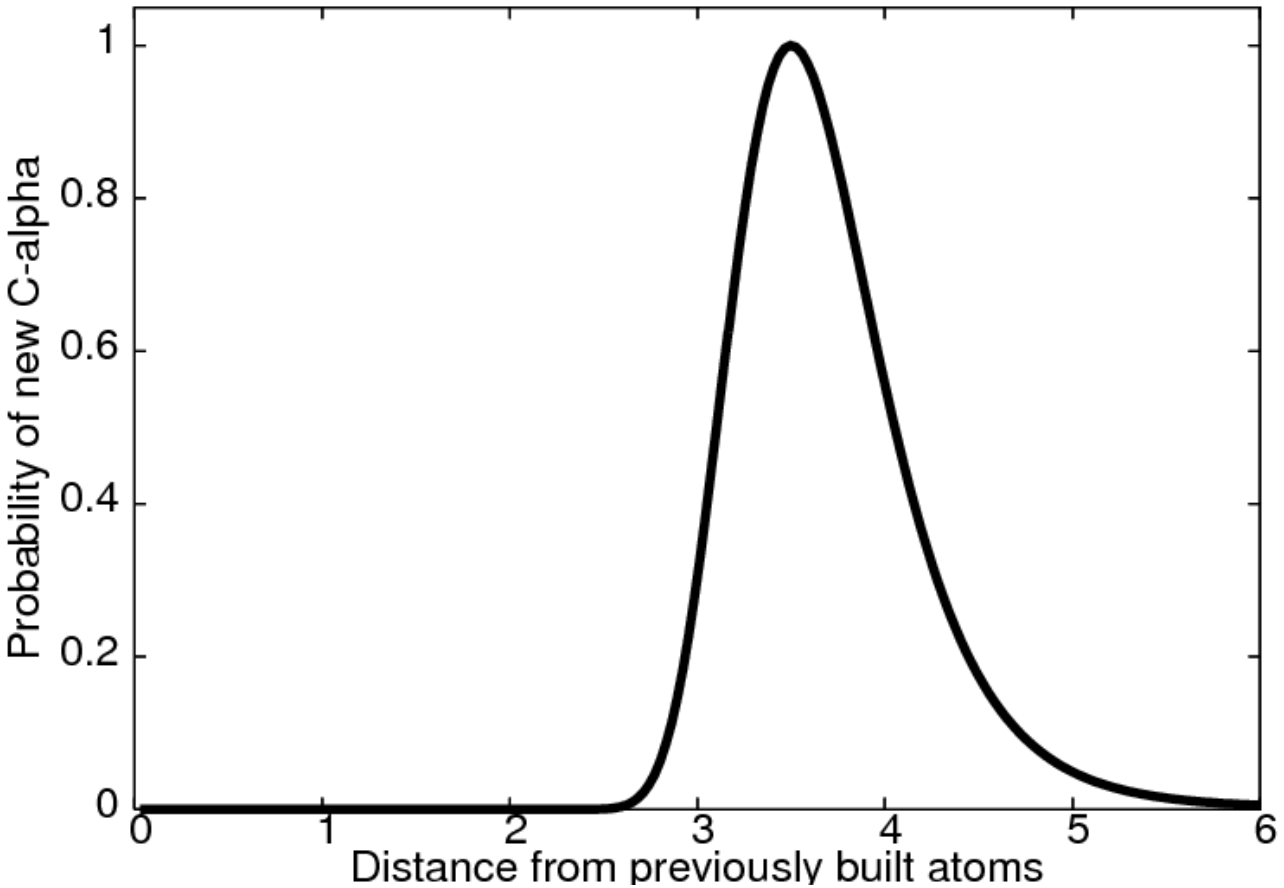


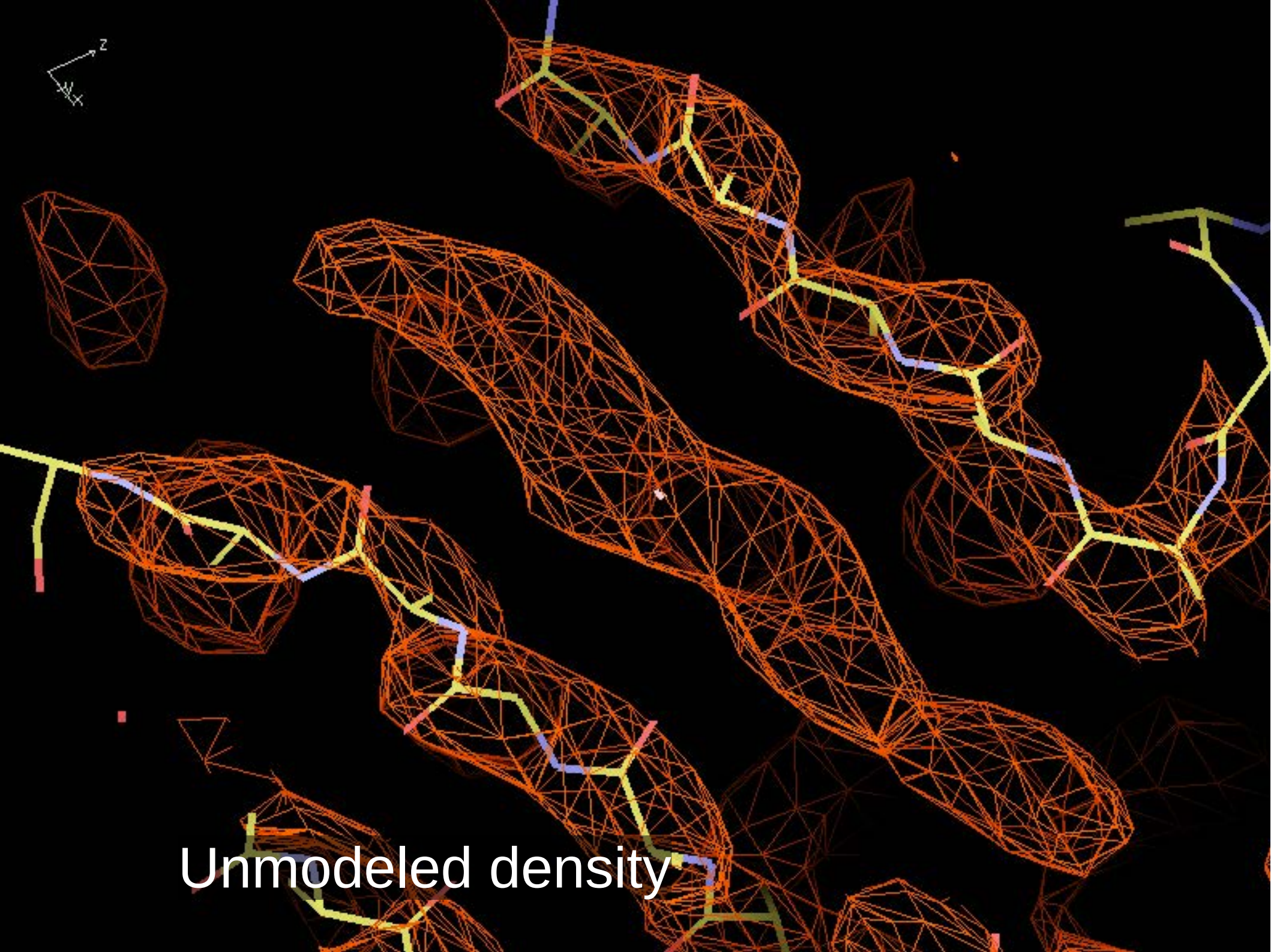
Comparison to the final model

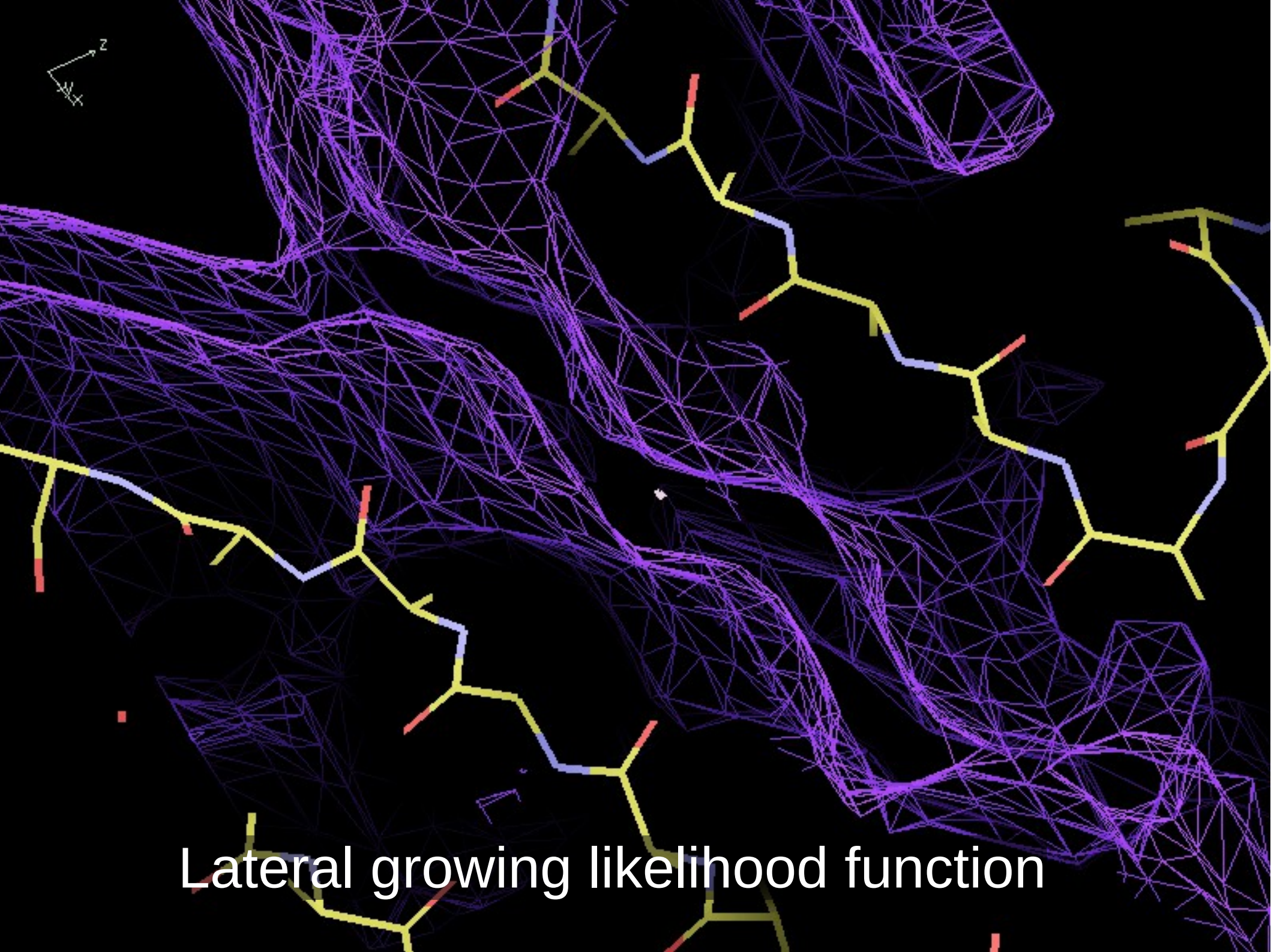
Buccaneer

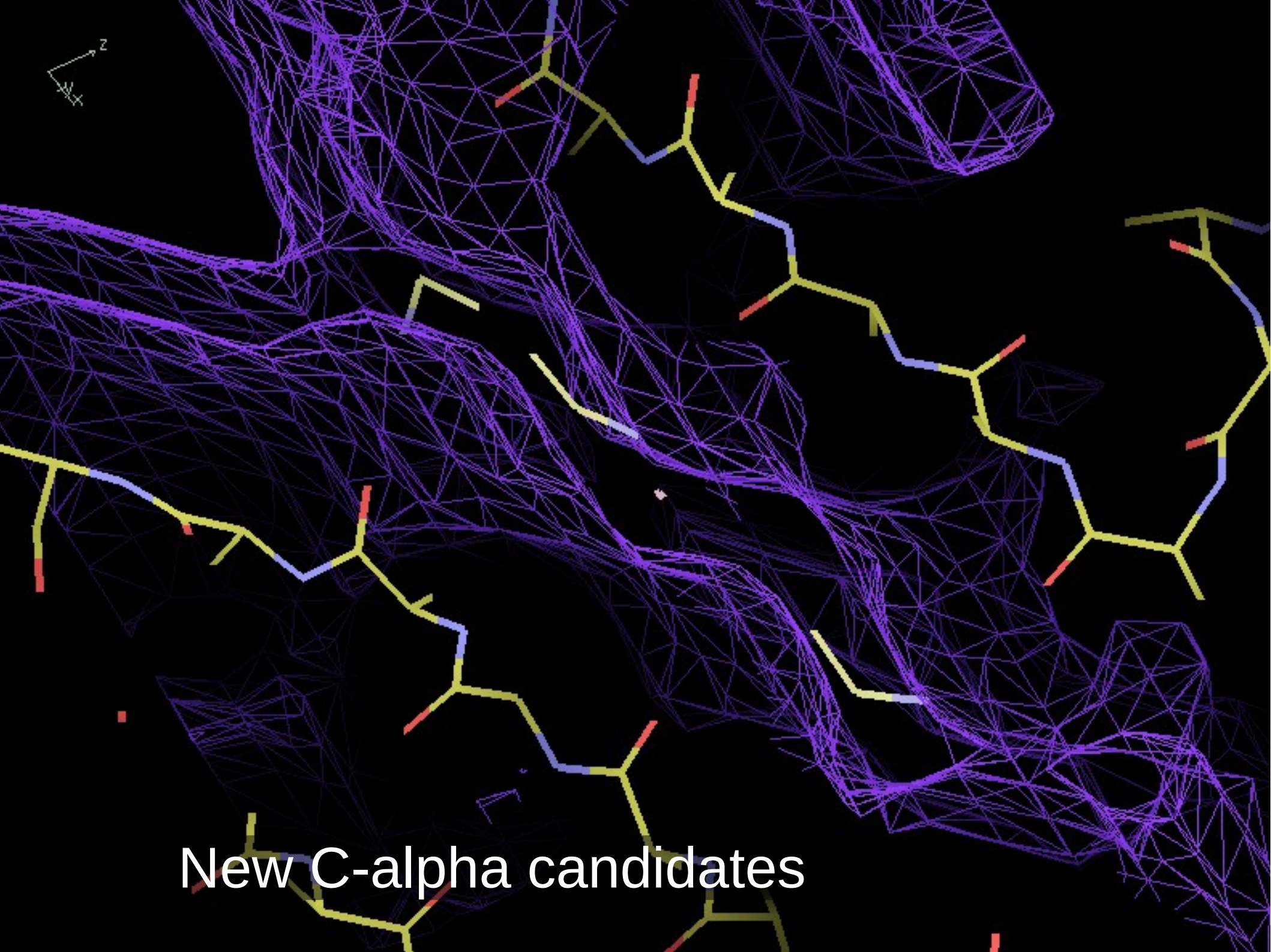
Model completion uses “**Lateral growing**”:

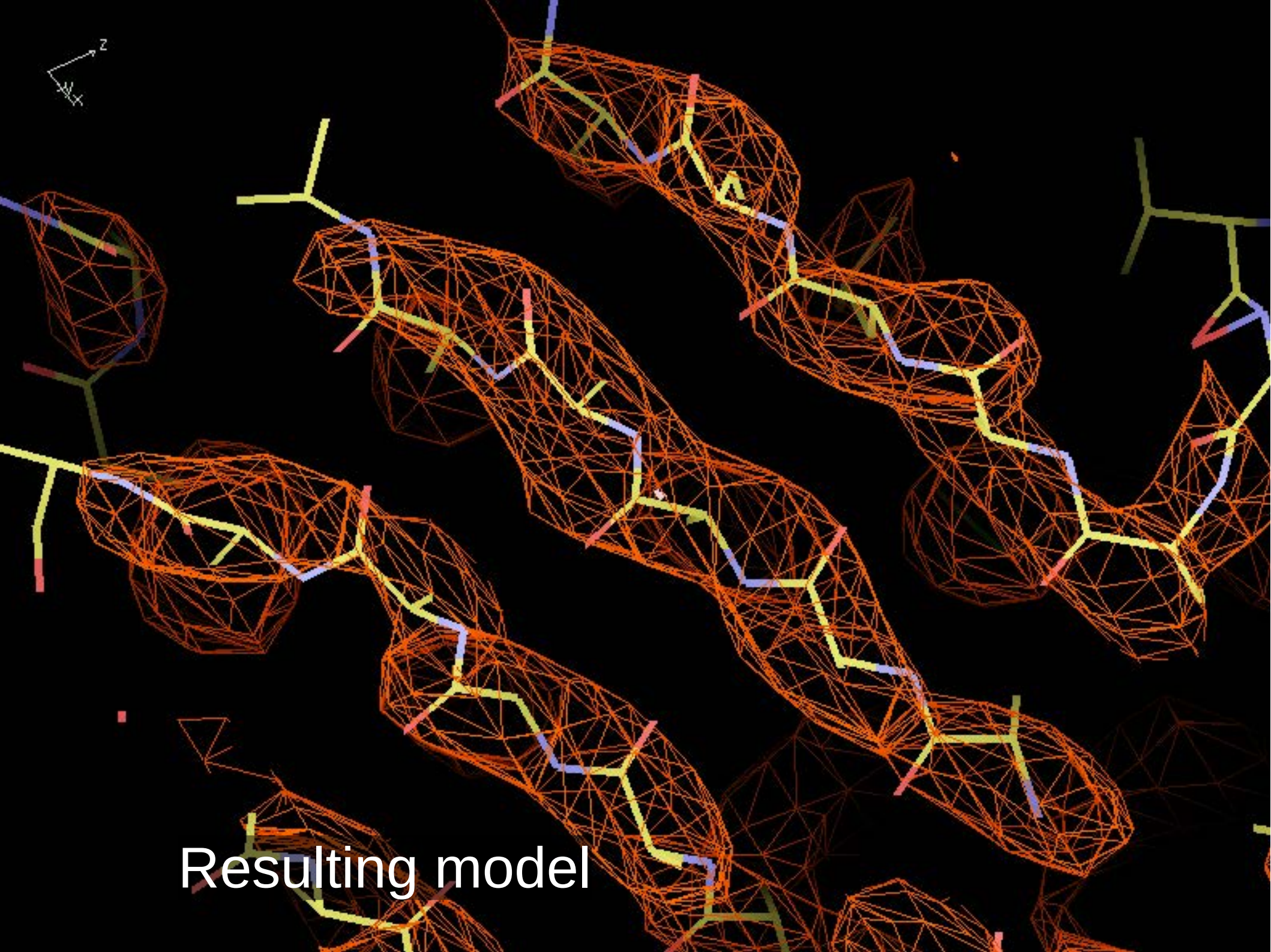
Grow sideways from existing chain fragments by looking for new C-alphas at an appropriate distance “sideways” from the existing chain:




Z
X
Unmodeled density

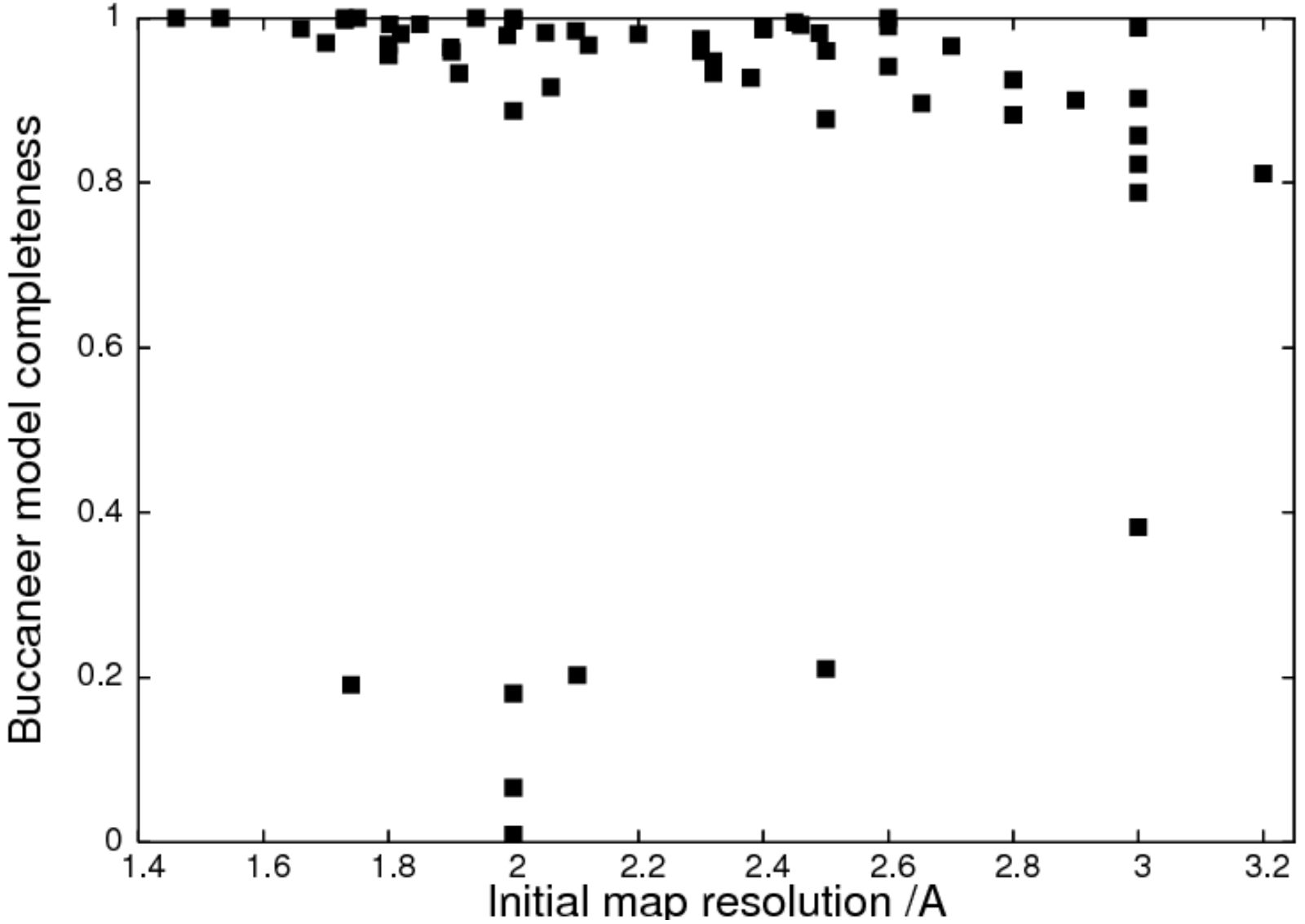

Lateral growing likelihood function


New C-alpha candidates


Z
X
Resulting model

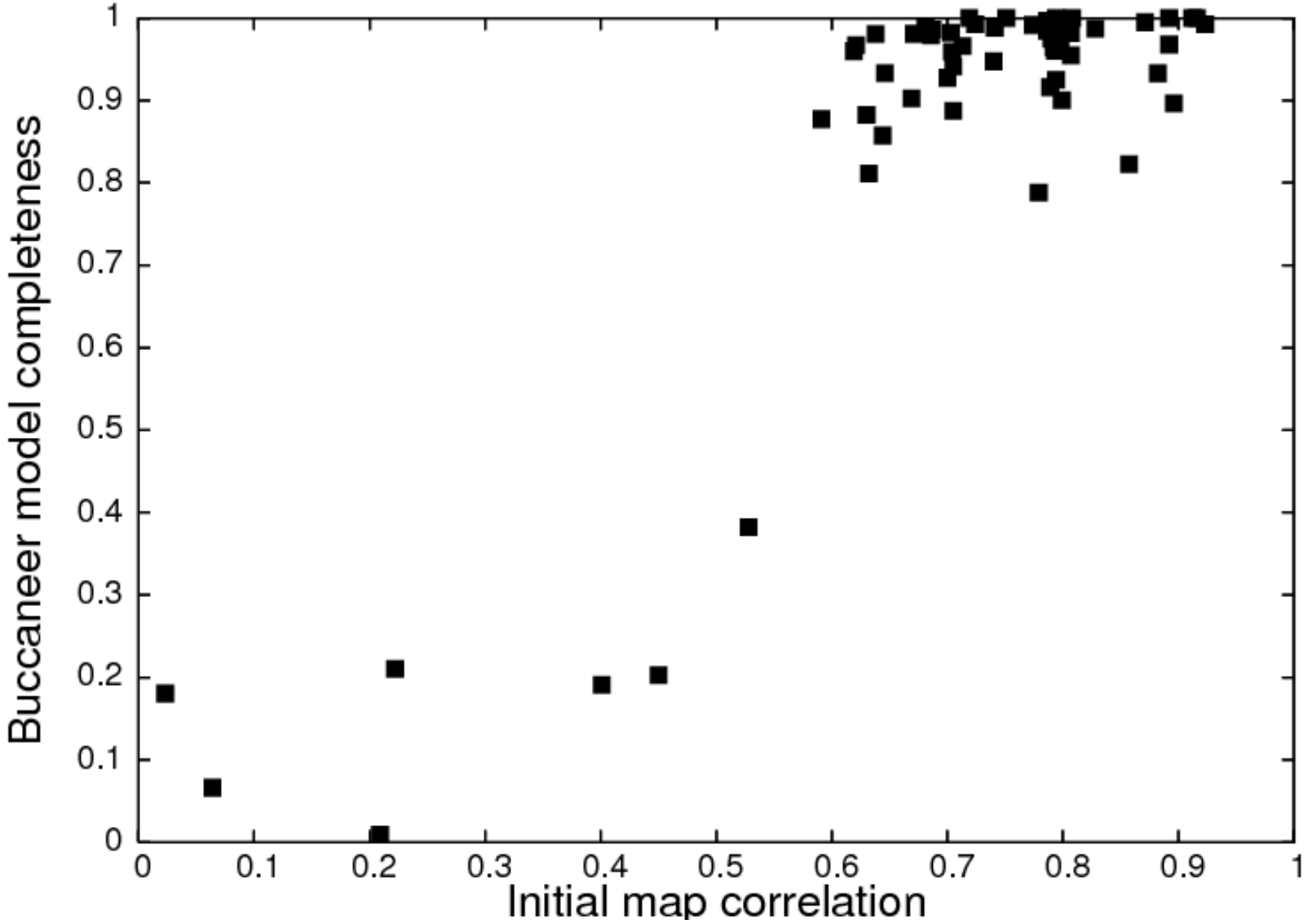
Buccaneer: Results

Model completeness not very dependent on resolution:

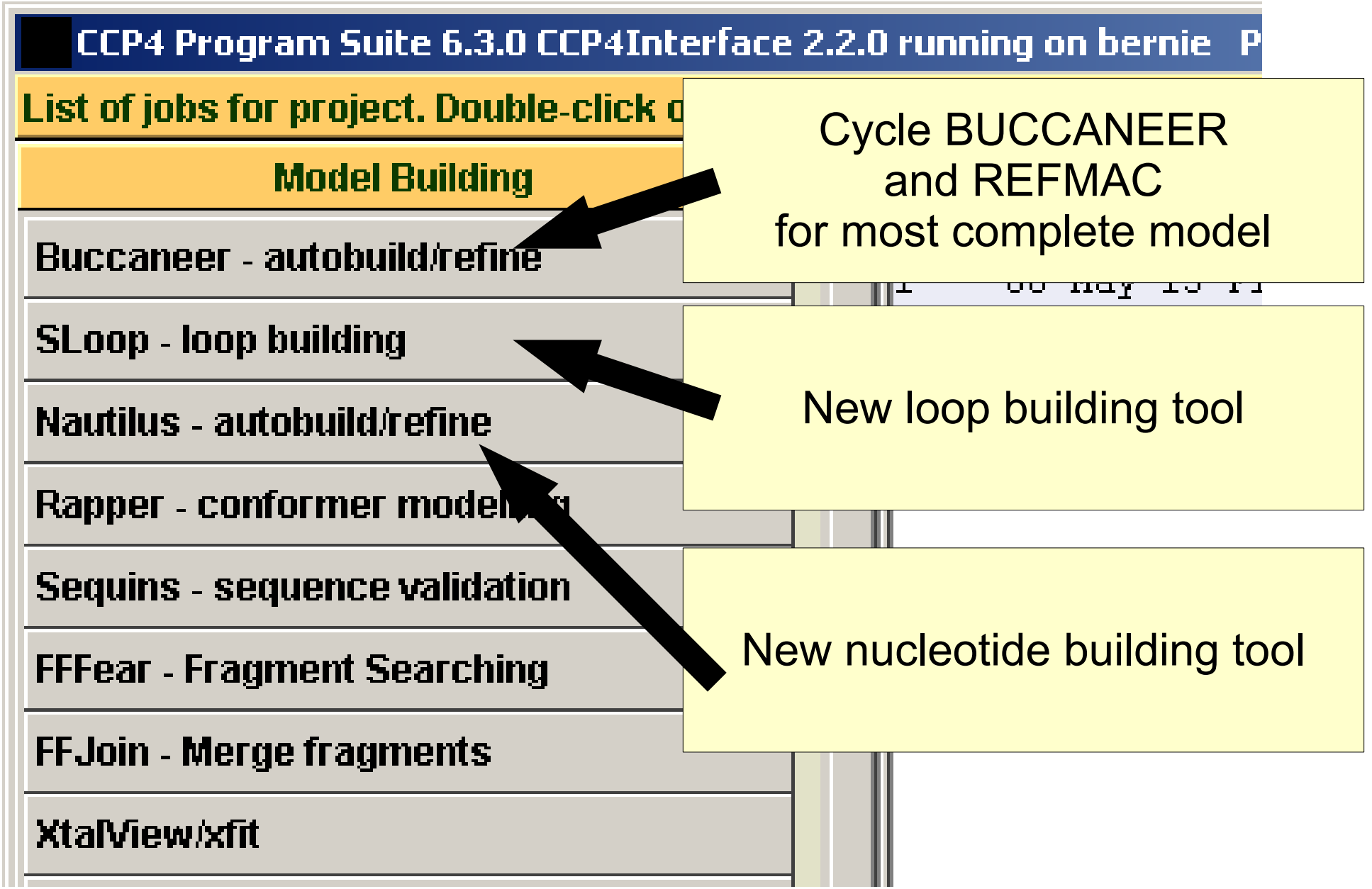


Buccaneer: Results

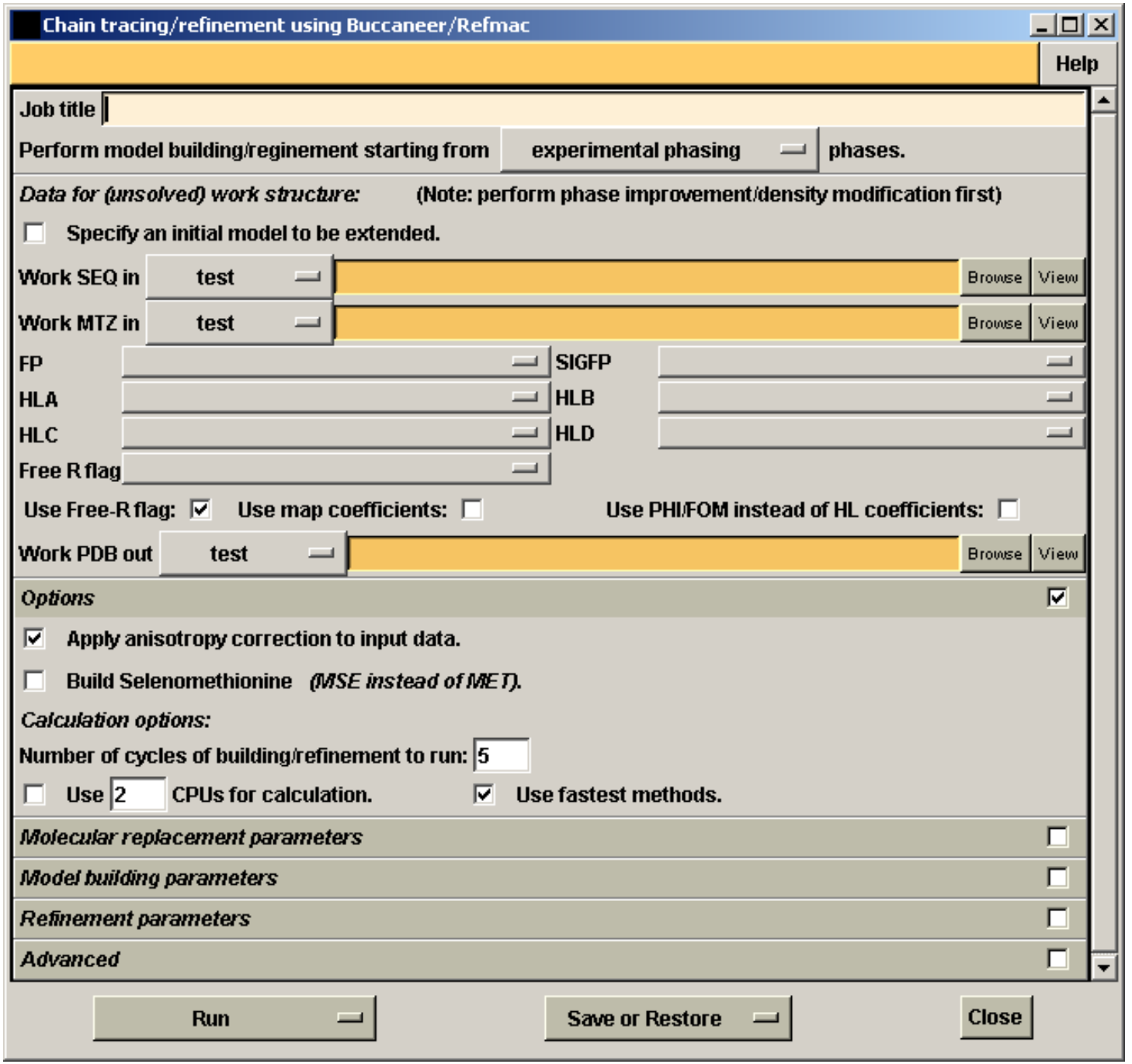
Model completeness dependent on initial phases:



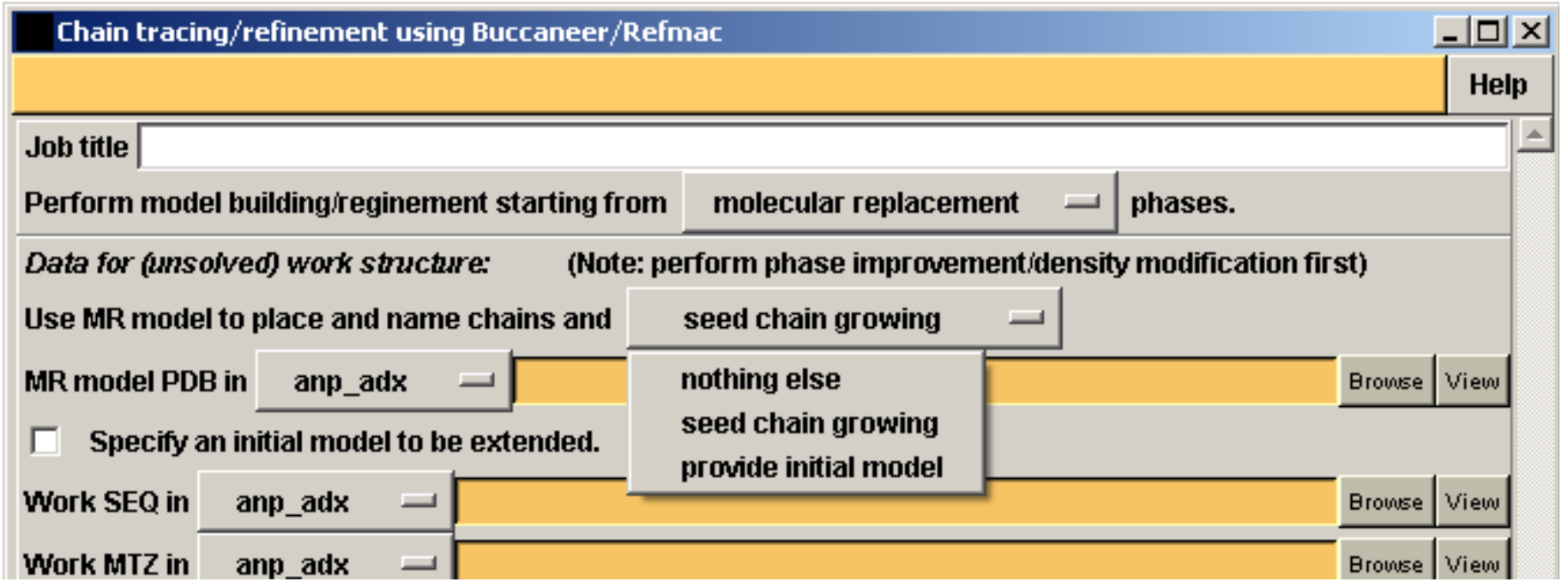
Buccaneer


CCP4 Program Suite 6.3.0 CCP4Interface 2.2.0 running on bernie
List of jobs for project. Double-click
Model Building
Buccaneer - autobuild/refine
SLoop - loop building
Nautilus - autobuild/refine
Rapper - conformer
Sequins - sequence validation
FFFear - Fragment Searching
FFJoin - Merge fragments
XtalView/xfit
Cycle BUCCANEER and REFMAC for most complete model
New loop building tool
New nucleotide building tool

Buccaneer


Chain tracing/refinement using Buccaneer/Refmac
Help
Job title
Perform model building/reginement starting from
experimental phasing
phases.
Data for (unsolved) work structure:
(Note: perform phase improvement/density modification first)
Specify an initial model to be extended.
Work SEQ in
test
Browse
View
Work MTZ in
test
Browse
View
FP
SIGFP
HLA
HLB
HLC
HLD
Free R flag
Use Free-R flag:
Use map coefficients:
Use PHI/FOM instead of HL coefficients:
Work PDB out
test
Browse
View
Options
Apply anisotropy correction to input data.
Build Selenomethionine
(MSE instead of MET).
Calculation options:
Number of cycles of building/refinement to run:
5
Use
2
CPUs for calculation.
Use fastest methods.
Molecular replacement parameters
Model building parameters
Refinement parameters
Advanced
Run
Save or Restore
Close

Buccaneer (MR)



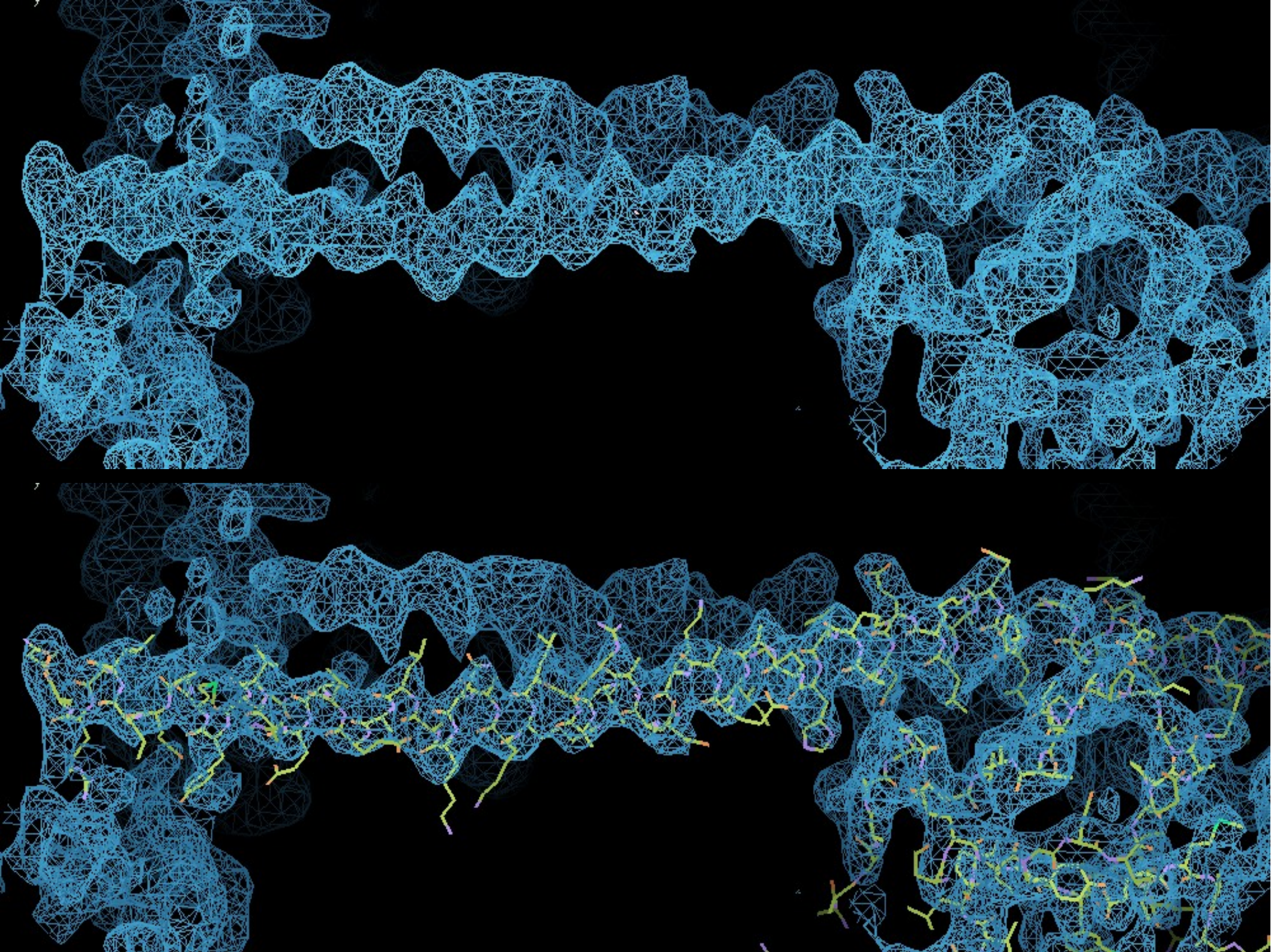
- different modes to balance trade-off between model bias and stabilizing of calculation with poor data

Buccaneer

What you need to do afterwards:

- Tidy up with Coot.
  - Or ARP/wARP when resolution is good.
  - Buccaneer+ARP/wARP better+faster than ARP/wARP.

- Typical Coot steps:
  - Connect up any broken chains.
  - Use density fit and rotamer analysis to check rotamers.
  - Check Ramachandran, molprobity, etc.
  - Add waters, ligands, check un-modeled blobs..
  - Re-refine, examine difference maps.



Buccaneer: Future

Buccaneer 1.7

- Loop building (sloop)

Buccaneer 1.8

- Reworked joining and correction code

Buccaneer 1.9

- CIS peptides? GLY conformations? NCS?

Buccaneer: Summary

A simple, (i.e. MTZ and sequence), very fast method of model building which is robust against resolution.

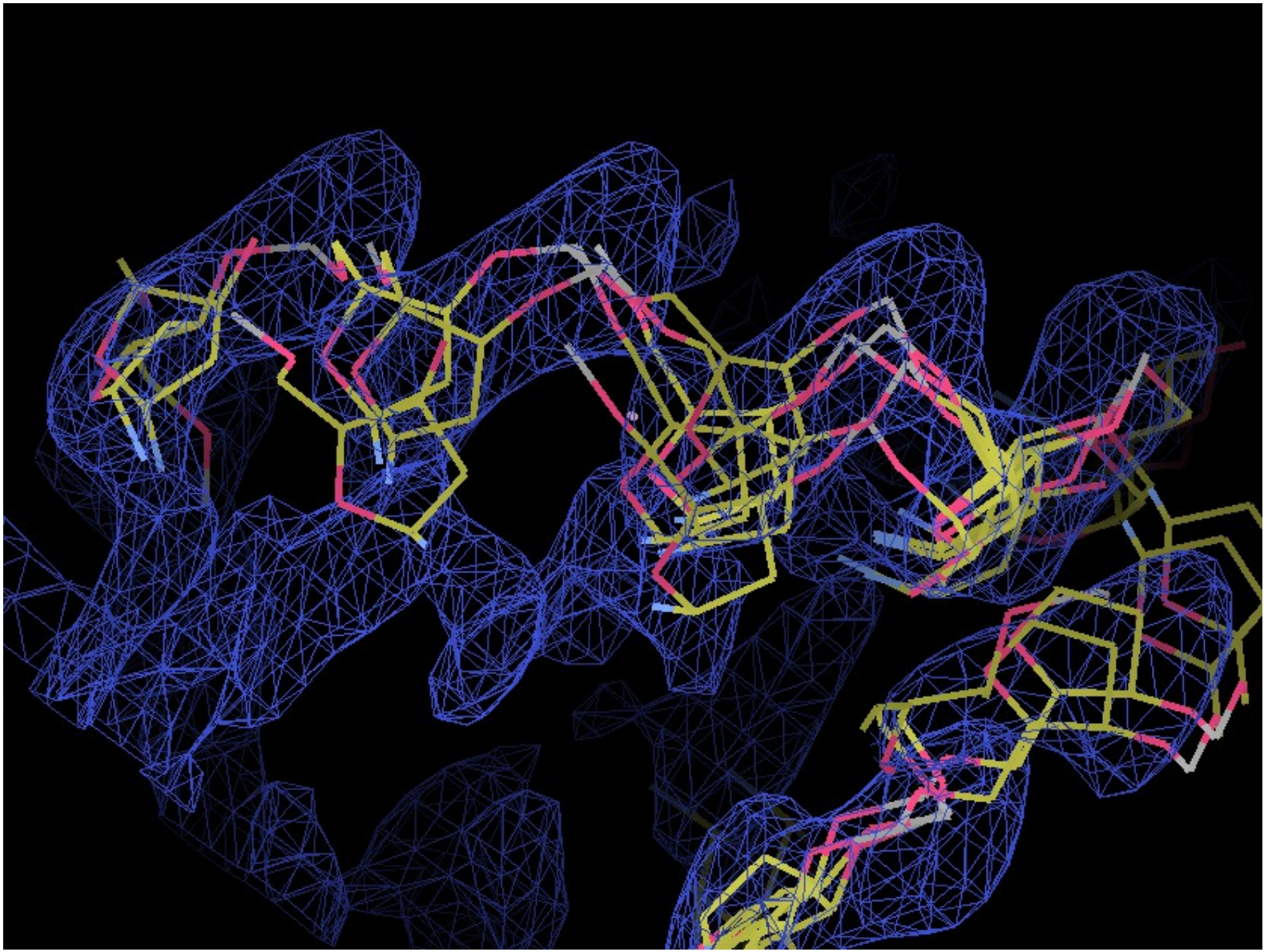
User reports for structures down to 3.7Å when phasing is good.

Results can be further improved by iterating with refinement in refmac (and in future, density modification).

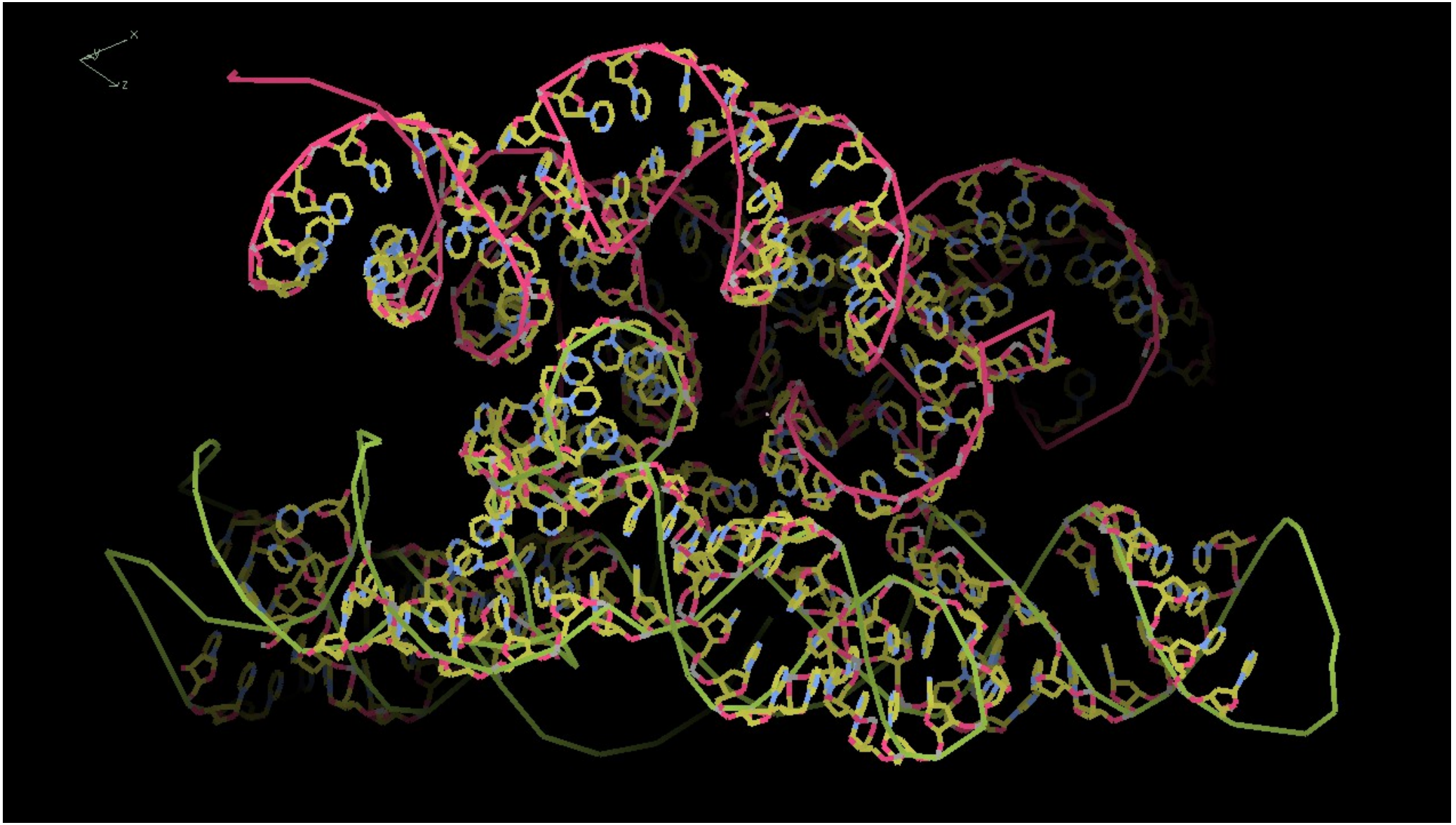
Proven on real world problems.

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 (poly-nucleotide building)



Nautilus (available in CCP4)



Coot integration

- Cootaneer: sequencing part of Buccaneer
- Fast secondary structure finding
- Cootilus: automated nucleic acid finding/building (https://www.youtube.com/watch?v=QGN6tF-zKOE)
- DB loop: loop/fragment building based on SLoop

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, CCP4