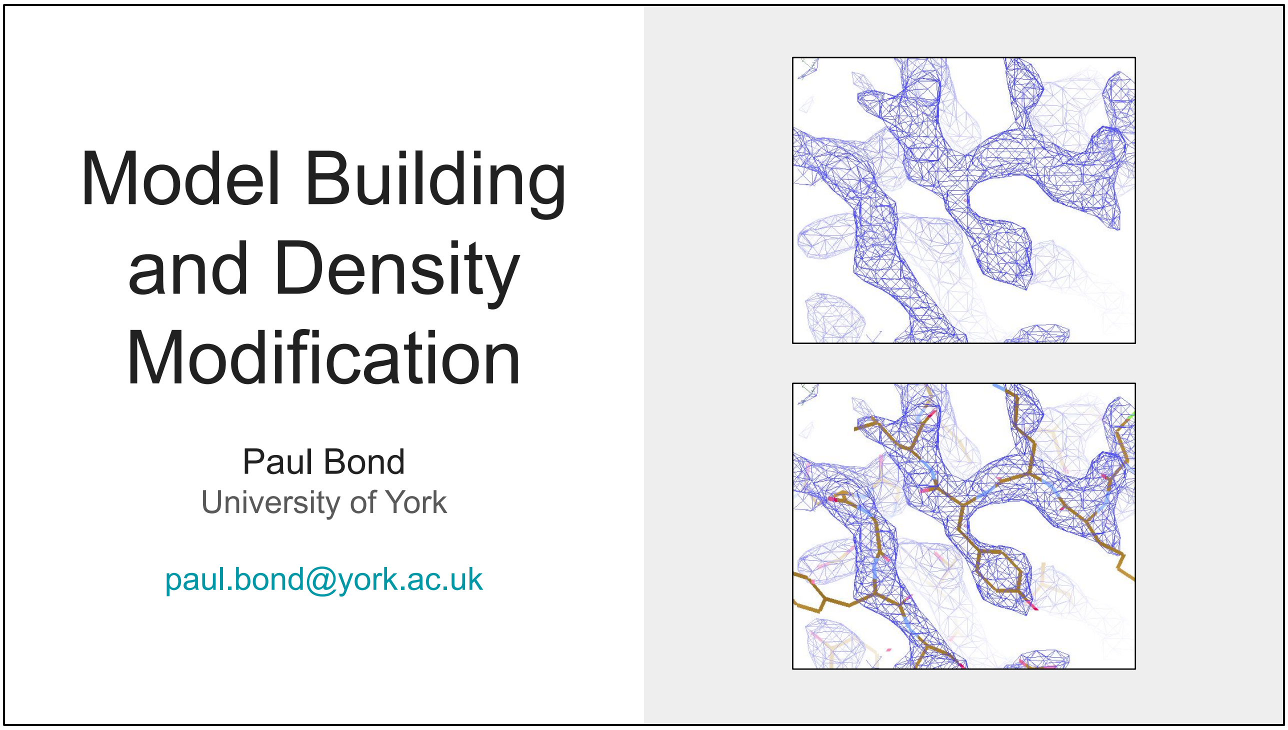
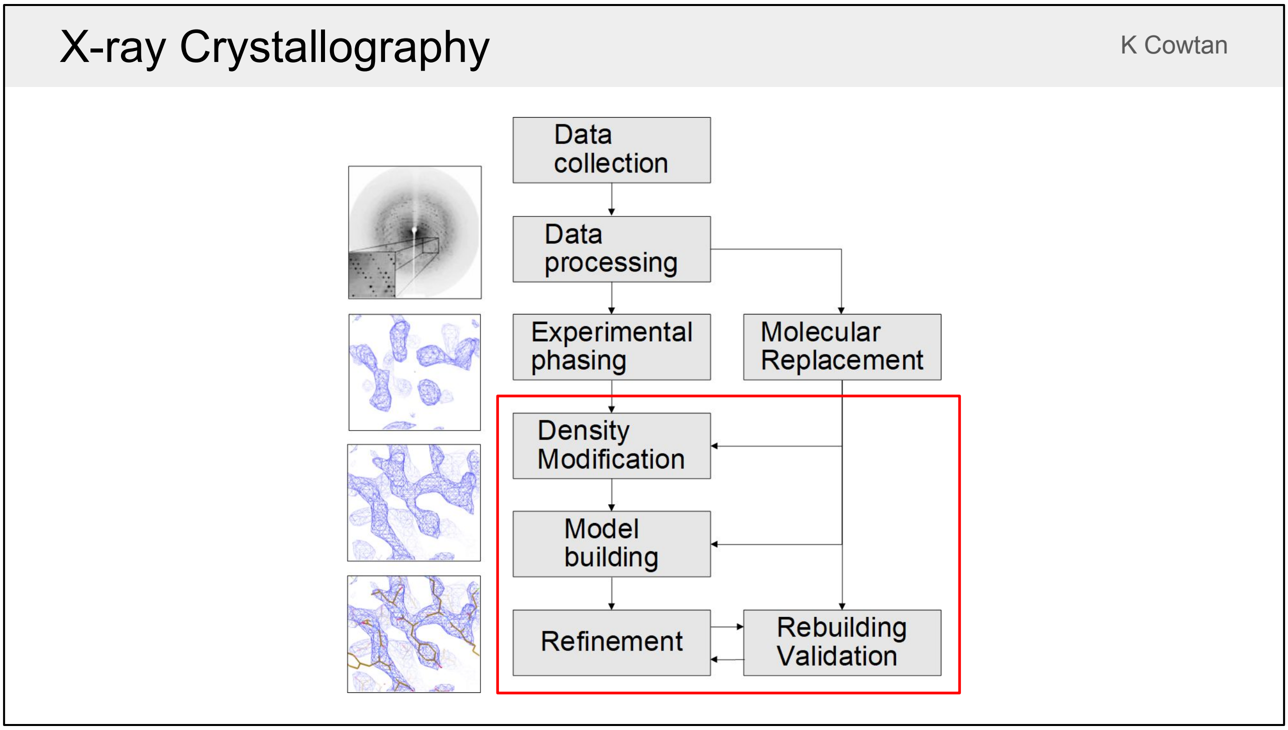


Model Building and Density Modification

Paul Bond
University of York

paul.bond@york.ac.uk

This presentation is on automated model building and density modification in CCP4. It focuses on the programs developed in the group of Kathryn Cowtan at the University of York: Parrot (density modification), Buccaneer (automated protein building), Nautilus (automated nucleotide building) and ModelCraft (a pipeline that includes all of these programs).



X-ray Crystallography

K Cowtan

Data collection

Data processing

Experimental phasing

Molecular Replacement

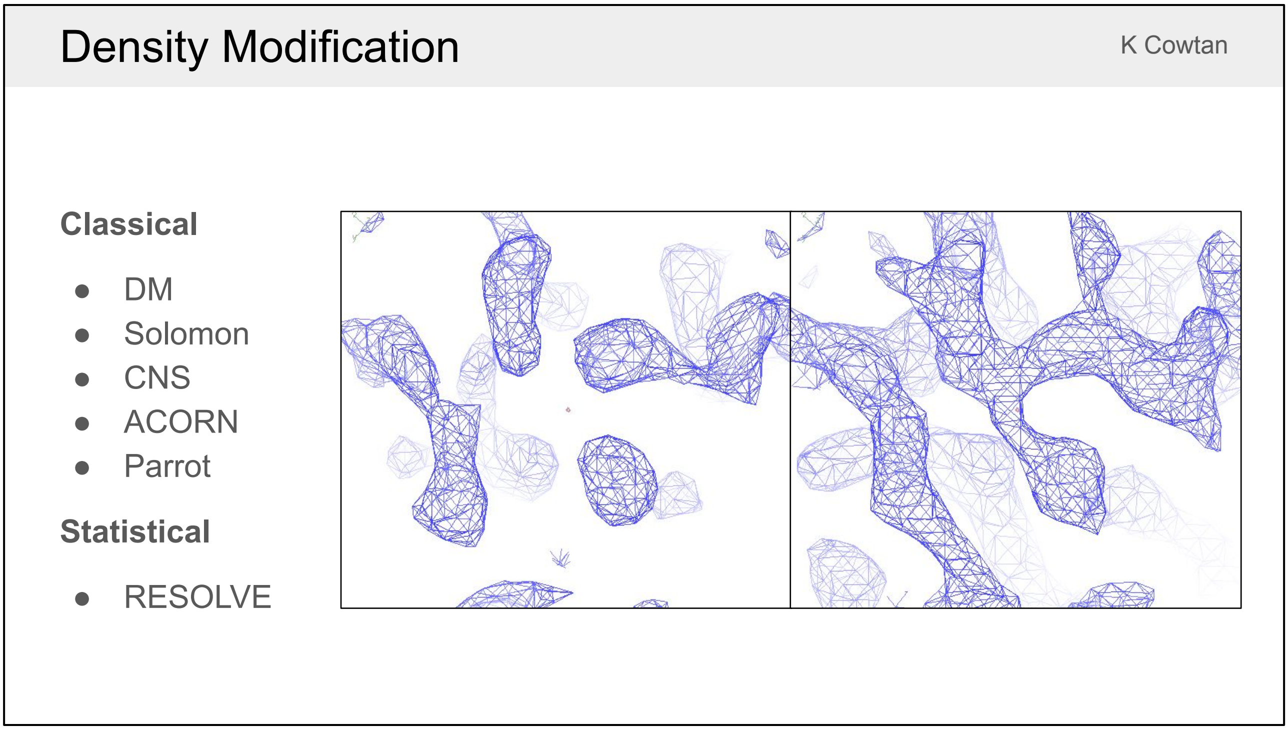
Density Modification

Model building

Refinement

Rebuilding Validation

This is a rough depiction of the structure solution process for X-ray crystallography. Once you've done the hard work of growing a crystal, you need to collect images and process them to get structure factor amplitudes. In order to get an electron density map you also need to get some phase estimates either through experimental phasing or molecular replacement. If you've used experimental phasing you will need to do density modification to improve the phases and hence the quality of the map. You then use automated model building to build an initial model into the improved map, followed by multiple rounds of refinement, rebuilding and validation. However, it's more likely that you'll get phases from molecular replacement (MR). If your MR model is poor then you might need to do density modification first before you can improve it with automated building. If the MR model is very good then you might even want to skip automated building and go straight to interactive rebuilding and validation. Automated model-building pipelines try to automate the steps in the red box as much as possible.



Density Modification

K Cowtan

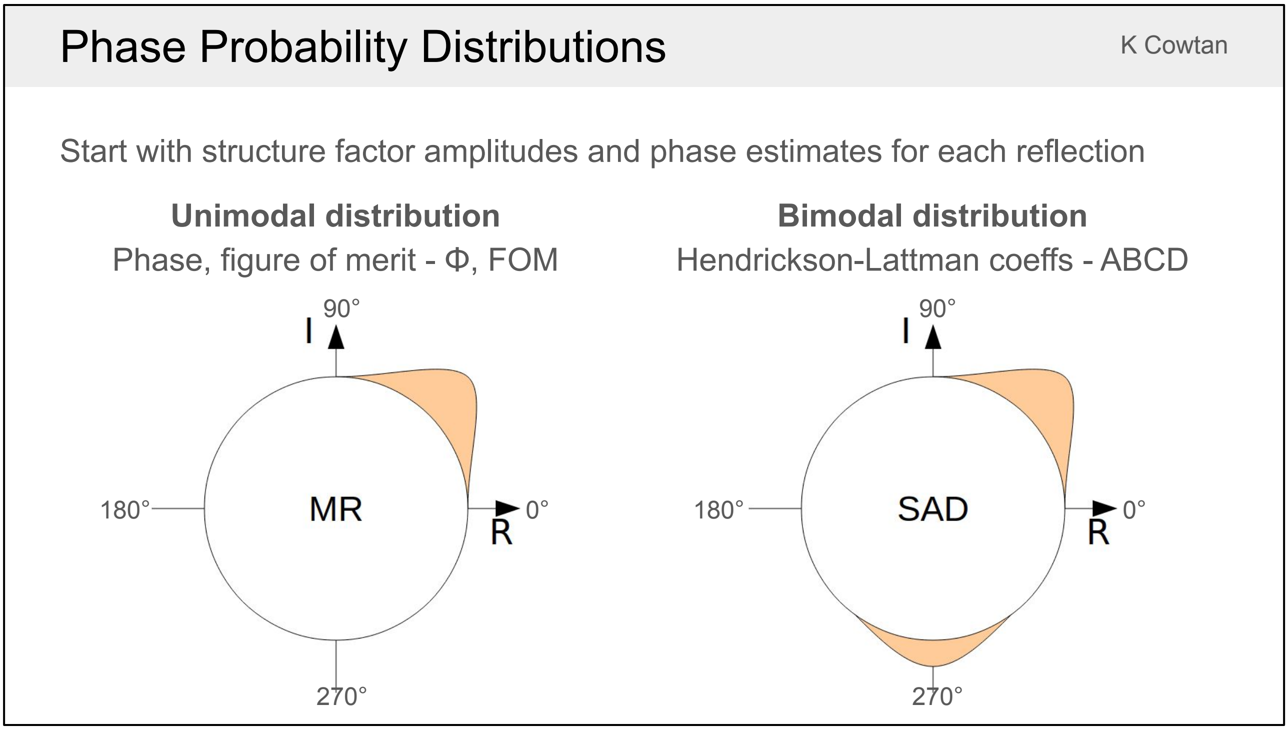
Classical

- DM
- Solomon
- CNS
- ACORN
- Parrot

Statistical

- RESOLVE

Density modification programs generally fall into one of two classes depending on how they work. A brief explanation is that classical density modification programs modify the density map to update the phases and statistical density modification programs create a map of density probability distributions to update the phases. ACORN and Parrot are the programs that are available in CCP4i2 and CCP4 Cloud. RESOLVE is the most widely used statistical density modification program and is available in Phenix.



Phase Probability Distributions

K Cowtan

Start with structure factor amplitudes and phase estimates for each reflection

Unimodal distribution

Phase, figure of merit - Φ, FOM

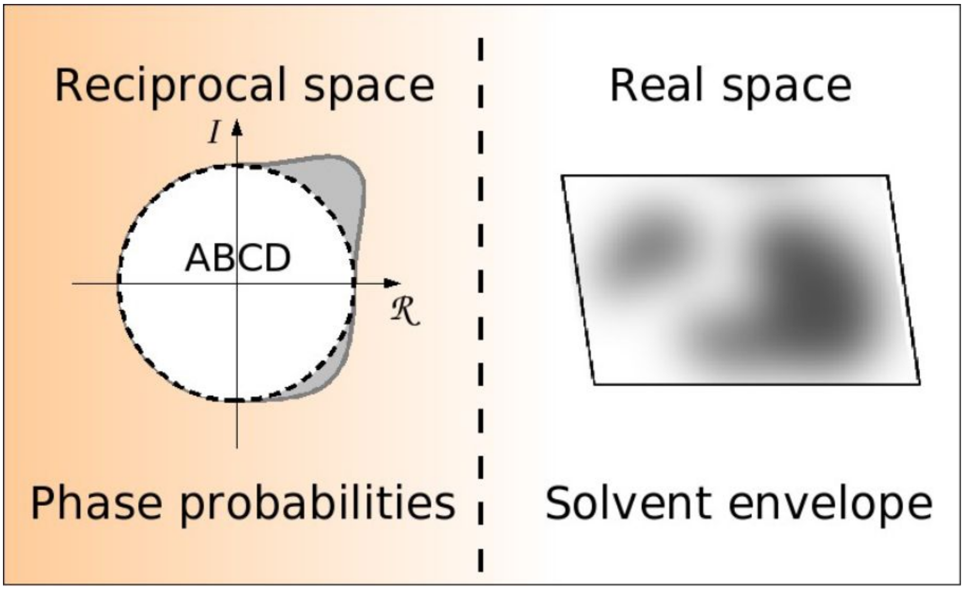
Bimodal distribution

Hendrickson-Lattman coeffs - ABCD

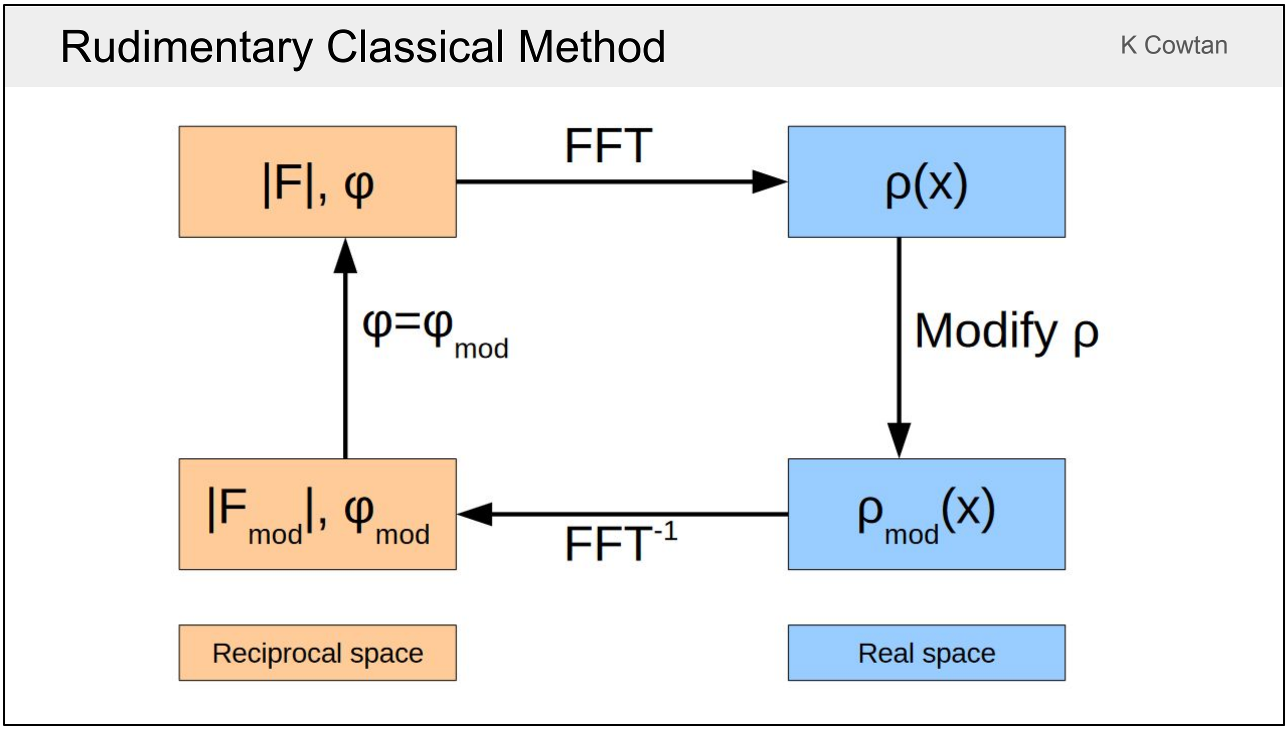
Firstly, it is important to understand phase probability distributions. Each reflection has an amplitude from data processing and a phase estimate from either molecular replacement or experimental phasing. Here the phase estimates are shown as probability distributions on Argand diagrams where the phase goes from 0 to 360 degrees around the circle. Each technique gives a different type of phase probability distribution. For molecular replacement, we know the model will contain errors so we can't be certain about the phase. It gives a unimodal distribution with the most likely phase (Φ) and a figure of merit (FOM) between 0 and 1 describing our confidence in that phase. The more errors we think the MR model contains the lower the FOM and the wider this distribution will be. Experimental SAD phasing gives a bimodal phase distribution because the two phasing circles overlap in two places, producing two possible phases with their respective uncertainties. Bimodal phase distributions are described by Hendrickson-Lattman coefficients, which are a set of four numbers: A, B, C and D. They are a more generic representation as they can also describe unimodal distributions (e.g. for MR) when both C and D are equal to 0. To calculate an electron density map we need an amplitude and a single phase (not a phase probability distribution). The map that is least noisy with respect to the uncertainty in the phase can be calculated by using the amplitude and phase at the center of mass of the phase distribution. In these Argand diagram representations the observed amplitude is the distance from the origin given by the radius of the circle. For a unimodal distribution the map phase is the same as the center of the distribution, but the amplitude gets reduced depending on the figure of merit. As the uncertainty increases and the distribution spreads out, the center of mass moves towards the center, and that structure factor becomes less important in calculating the map. For a bimodal distribution the map phase is somewhere between the two peaks and the center of mass is usually much closer to the center.

Reciprocal and Real Space

K Cowtan



The initial electron density map will be noisy, especially in the SAD case where there is an ambiguity between two possible phases, but density modification is able to improve the map by combining information in real and reciprocal space. We start from the phase probability distributions and use them to create an electron density map in real space, then we add some information in real space in order to narrow down the possible phases.



Rudimentary Classical Method

K Cowtan

$|F|, \varphi$

FFT

$\rho(x)$

Modify ρ

$\rho_{mod}(x)$

FFT^{-1}

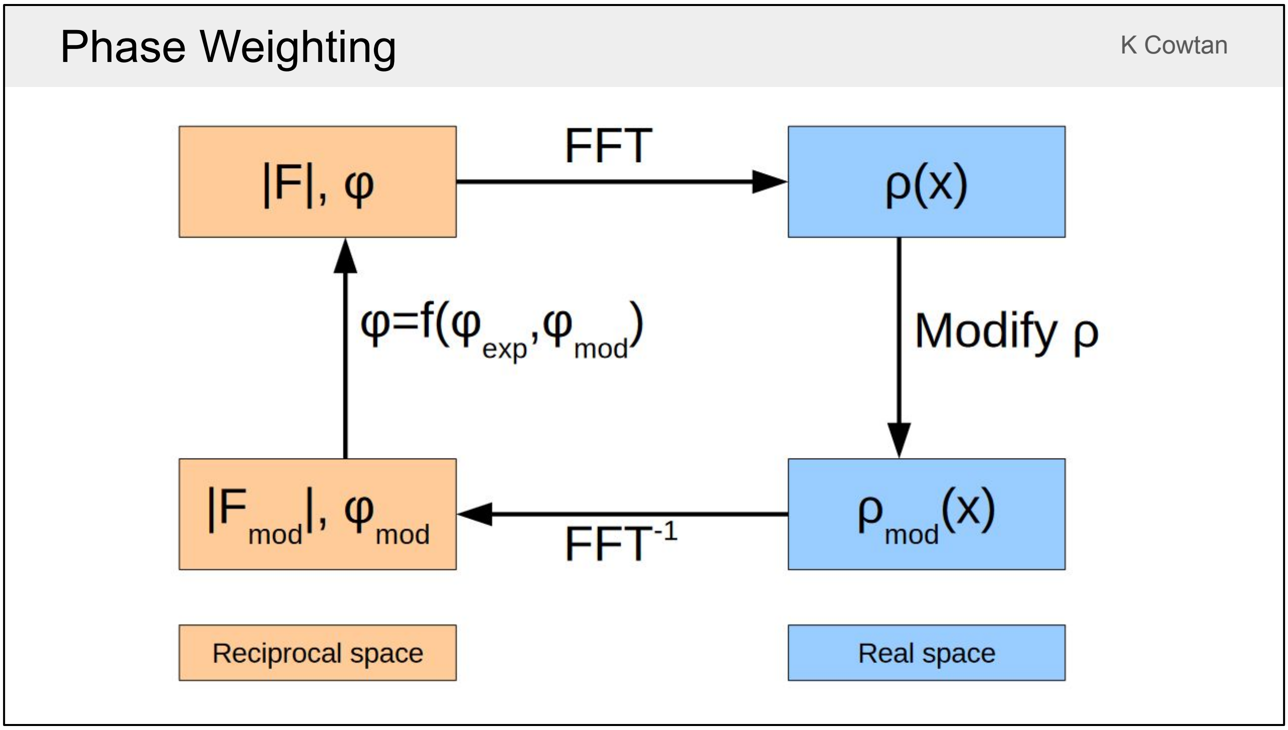
$|F_{mod}|, \varphi_{mod}$

$\varphi=\varphi_{mod}$

Reciprocal space

Real space

This is a classical density modification calculation in a very rudimentary form. It starts with a set of structure factor amplitudes (|F|) and phases (φ). It then calculates an electron density (ρ) map using a fast Fourier transform (FFT) and modifies the density in real space. For example, basic image processing algorithms could be used to reduce the noise in the map. An inverse Fourier transform is then used to get amplitudes and phases for the modified map, where both the amplitudes and phases will have changed. This information could be used by just replacing the original phases with the modified ones. By combining the updated phases with the original (experimentally determined) amplitudes the calculation can be repeated cyclically to improve the map.



Phase Weighting

K Cowtan

$|F|, \varphi$

FFT

$\rho(x)$

Modify ρ

$\rho_{mod}(x)$

FFT^{-1}

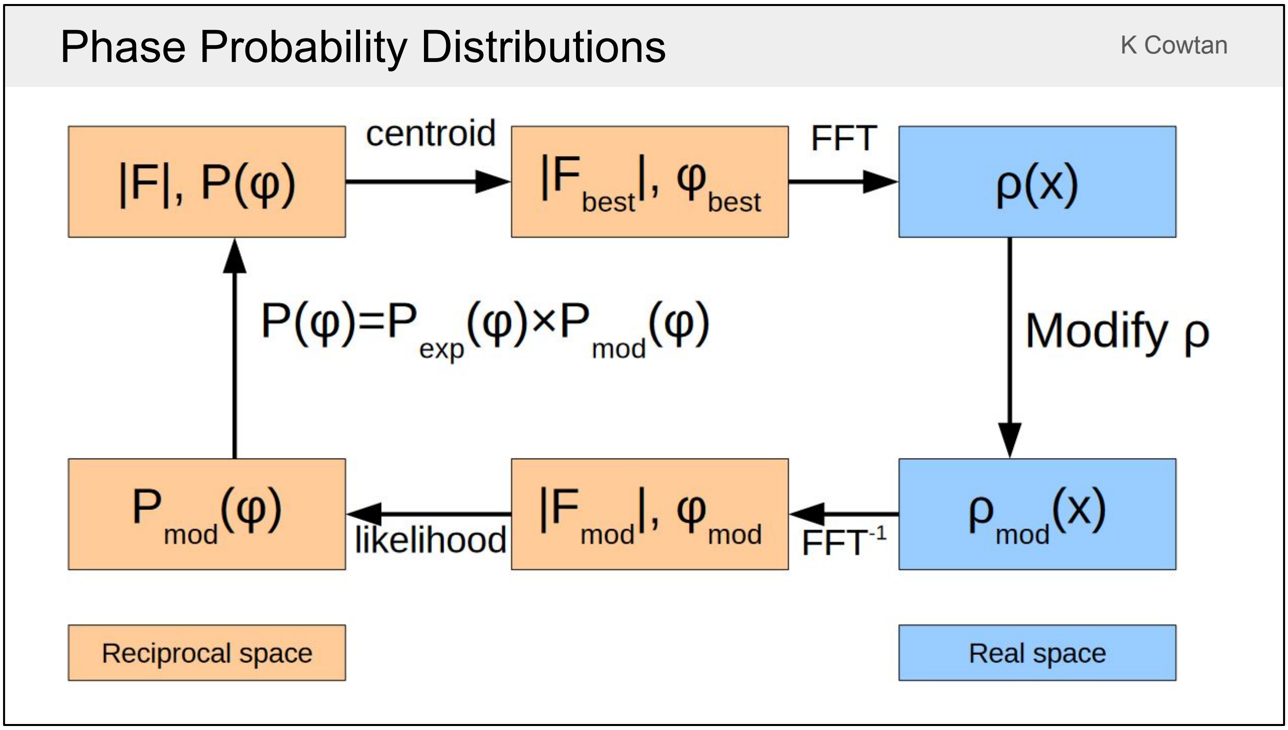
$|F_{mod}|, \varphi_{mod}$

$\varphi = f(\varphi_{exp}, \varphi_{mod})$

Reciprocal space

Real space

In practice, instead of the original phases being simply replaced, there is some combination of the original phases and the modified phases.



Phase Probability Distributions

K Cowtan

$|F|, P(\varphi)$

centroid

$|F_{best}|, \varphi_{best}$

FFT

$\rho(x)$

Modify ρ

$\rho_{mod}(x)$

FFT^{-1}

$|F_{mod}|, \varphi_{mod}$

likelihood

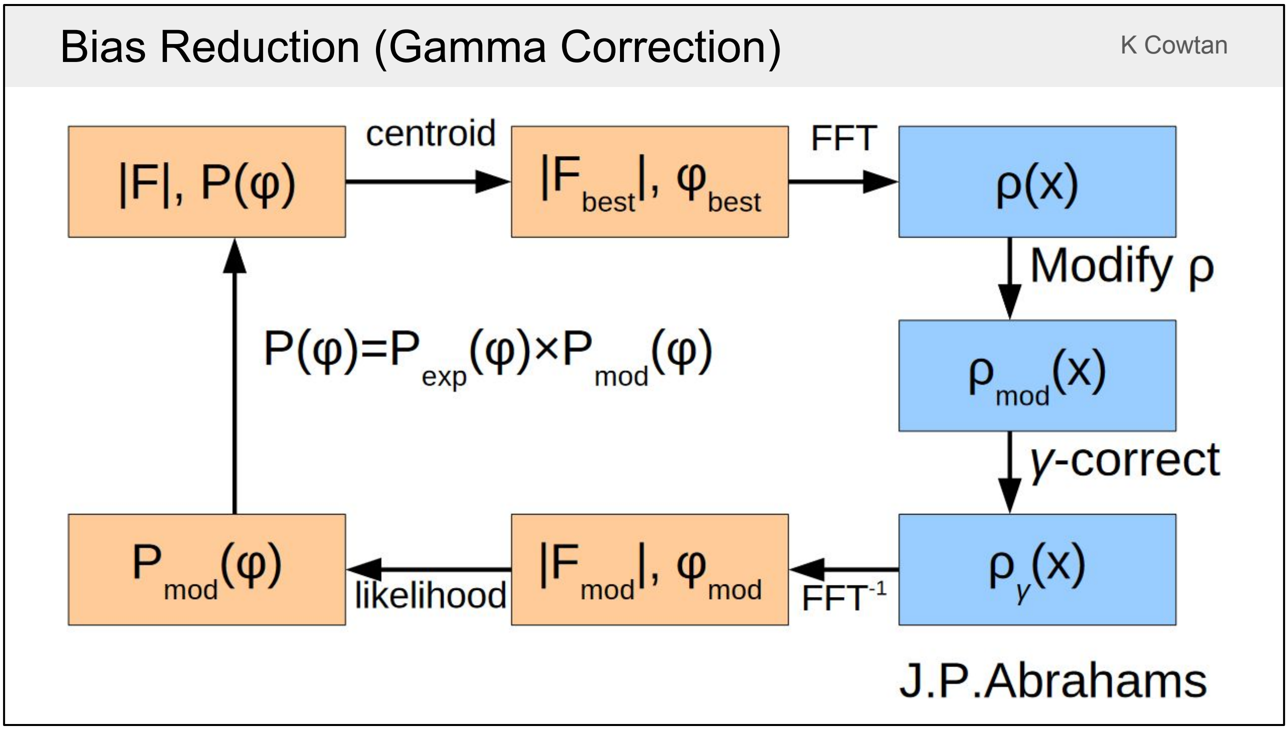
$P_{mod}(\varphi)$

$P(\varphi)=P_{exp}(\varphi)\times P_{mod}(\varphi)$

Reciprocal space

Real space

As discussed earlier, we are not dealing with single phases but with phase probability distributions. Starting from the experimental amplitudes and phase distributions in the top left, we find the center of mass to get best phases and weighted amplitudes that depend on the confidence in the best phases. As before, a Fourier transform is used to get a map, the map is modified, then an inverse Fourier transform produces modified amplitudes and modified single phases. These single phases need to be changed back into phase probability distributions, which is done by comparing the modified amplitudes with the experimental amplitudes. If there is a larger difference in amplitude then there is less confidence in the modified phase and the distribution is wider. The original distributions can then be updated through the product of the experimental distributions and the modified distributions. The main flaw in this approach is that you can only multiply probabilities together if they are independent. Unfortunately, the modified distributions are strongly dependent on the experimental distributions because they are derived from them. This leads to bias in the calculation. Repeated cycles increase the figures of merit to 1 to say we are certain about the phases, but the errors in the phases do not decrease to 0.



Bias Reduction (Gamma Correction)

K Cowtan

$|F|, P(\varphi)$

centroid

$|F_{best}|, \varphi_{best}$

FFT

$\rho(x)$

Modify ρ

$\rho_{mod}(x)$

γ-correct

$\rho_{\gamma}(x)$

FFT^{-1}

$|F_{mod}|, \varphi_{mod}$

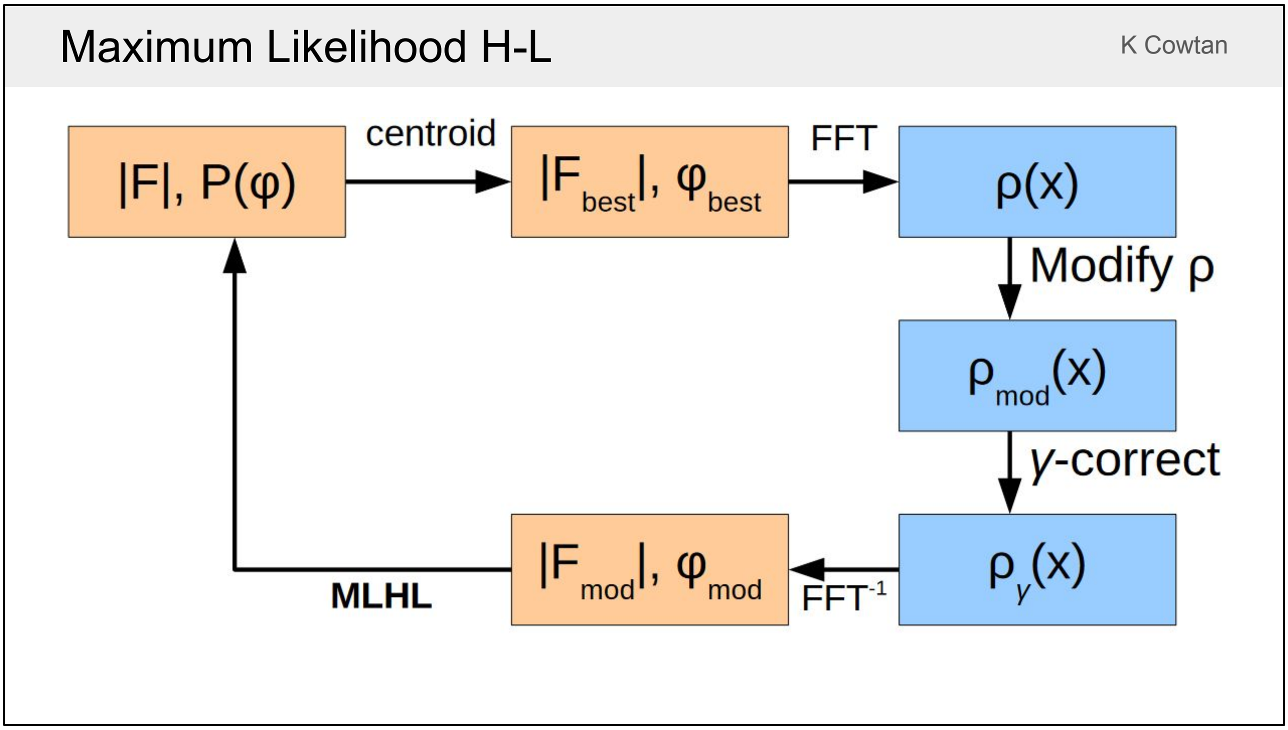
likelihood

$P_{mod}(\varphi)$

$P(\varphi)=P_{exp}(\varphi)\times P_{mod}(\varphi)$

J.P.Abrahams

A method called gamma correction was developed by Jan Pieter Abrahams in order to reduce the amount of bias in the calculation. This is an extra step after modifying the density to subtract out part of the initial density map. The amount of the initial map you subtract out has to be right to remove the bias. This is the classical density modification cycle used in the programs DM, SOLOMON and CNS.



Maximum Likelihood H-L

K Cowtan

$|F|, P(\varphi)$

centroid

$|F_{best}|, \varphi_{best}$

FFT

$\rho(x)$

Modify ρ

$\rho_{mod}(x)$

γ-correct

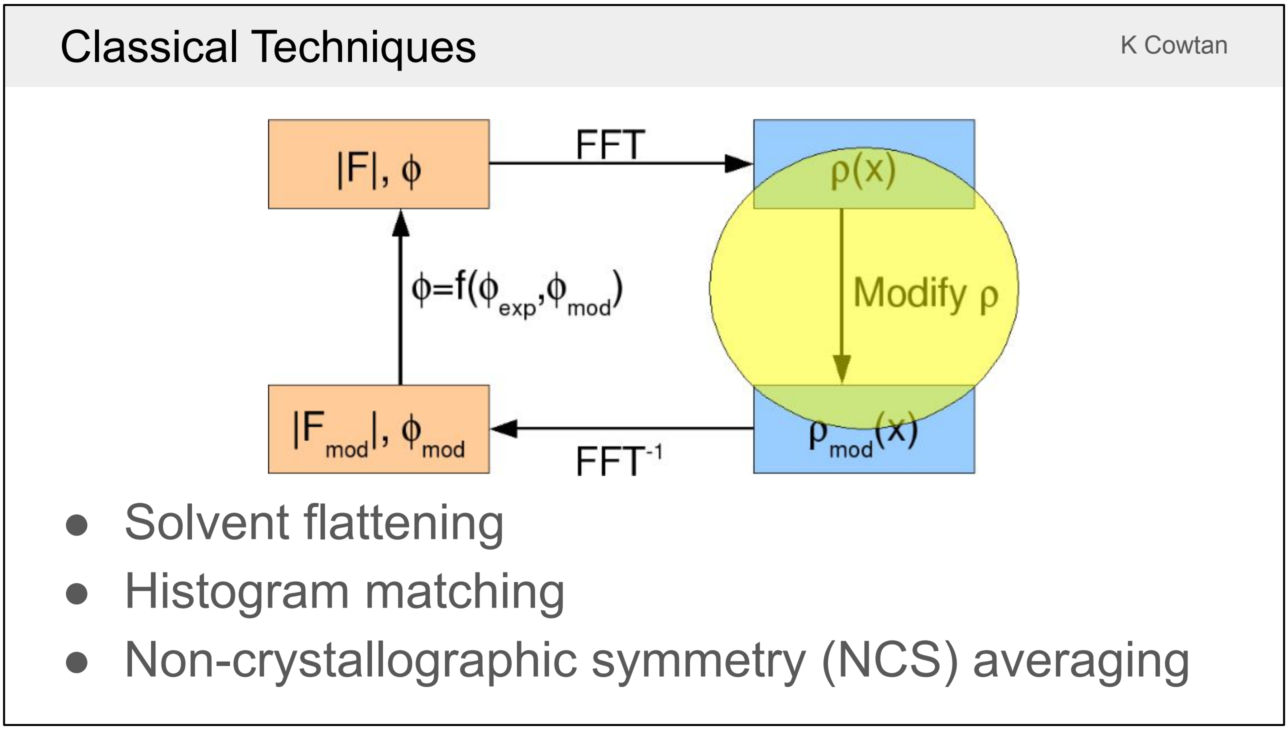
$\rho_{\gamma}(x)$

FFT^{-1}

$|F_{mod}|, \varphi_{mod}$

MLHL

A further development to this classical density modification cycle has been to merge the last two steps into a single likelihood calculation that includes the experimental phase distribution (i.e. Maximum Likelihood using Hendrickson-Lattman coefficients). This improves the method slightly more and is available in the program Parrot.



Classical Techniques

K Cowtan

$|F|, \phi$

FFT

$\rho(x)$

Modify ρ

$\phi=f(\phi_{exp},\phi_{mod})$

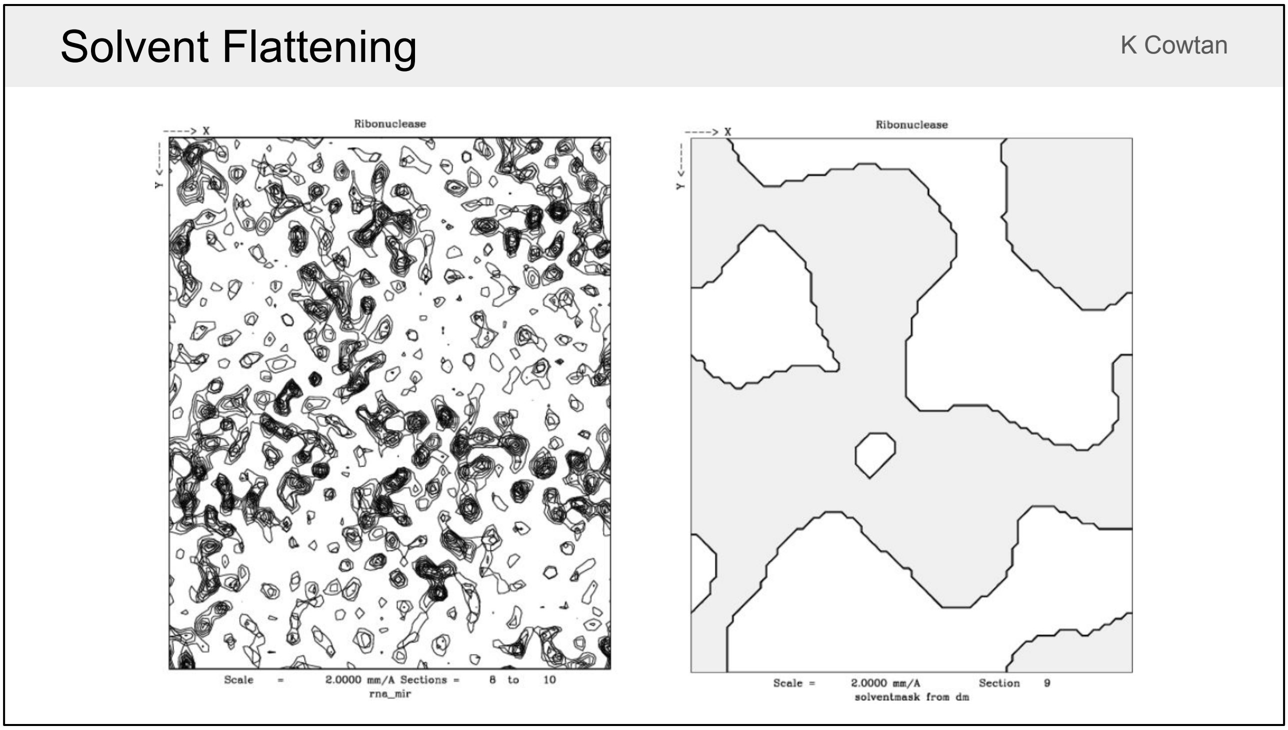
$|F_{mod}|, \phi_{mod}$

FFT^{-1}

$\rho_{mod}(x)$

- Solvent flattening
- Histogram matching
- Non-crystallographic symmetry (NCS) averaging

For classical density modification, the most important thing is how the map is modified in real space to improve it. It is a similar problem to standard image processing, e.g. when photo processing software is used to remove noise from an old photograph. In this case it is working in 3D instead of 2D and uses prior knowledge about well-phased electron density maps. There are three very widely used techniques that aim to make the density map more realistic: solvent flattening, histogram matching and NCS averaging.



Solvent Flattening

K Cowtan

Solvent flattening is the oldest technique and probably the most powerful one if they were used in isolation. The image on the left shows a few stacked 2D slices through a contoured electron density map. There are regions with lots of features that belong to the protein and sparse regions without many features that belong to the solvent. Using the knowledge that there are fewer features in the solvent regions (because they are disordered) and the fraction of solvent in the crystal, programs are able to draw a mask that divides the map up into protein regions and solvent regions. The solvent mask for the example above is shown in the image on the right. If the map has very good phases then the features in the solvent region will be real ordered features, i.e. mostly ordered waters close to the protein. However, early on in the calculation the phases are poor and there is more noise in in the solvent than real ordered features. Solvent flattening removes all the density in the solvent regions, which will remove both ordered features and noise. If it removes more noise than ordered features then the phases improve overall and the next cycle will produce a less noisy map.

Histogram Matching

K Cowtan

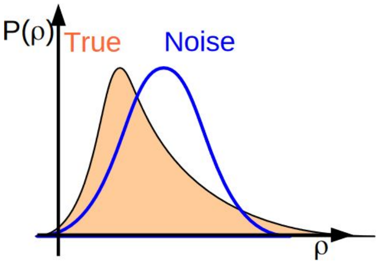
A technique from image processing for modifying the protein region.

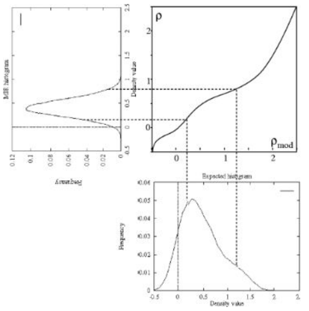
● Noise maps have Gaussian histogram.

● Well phased maps have a skewed distribution: sharper peaks and bigger gaps.

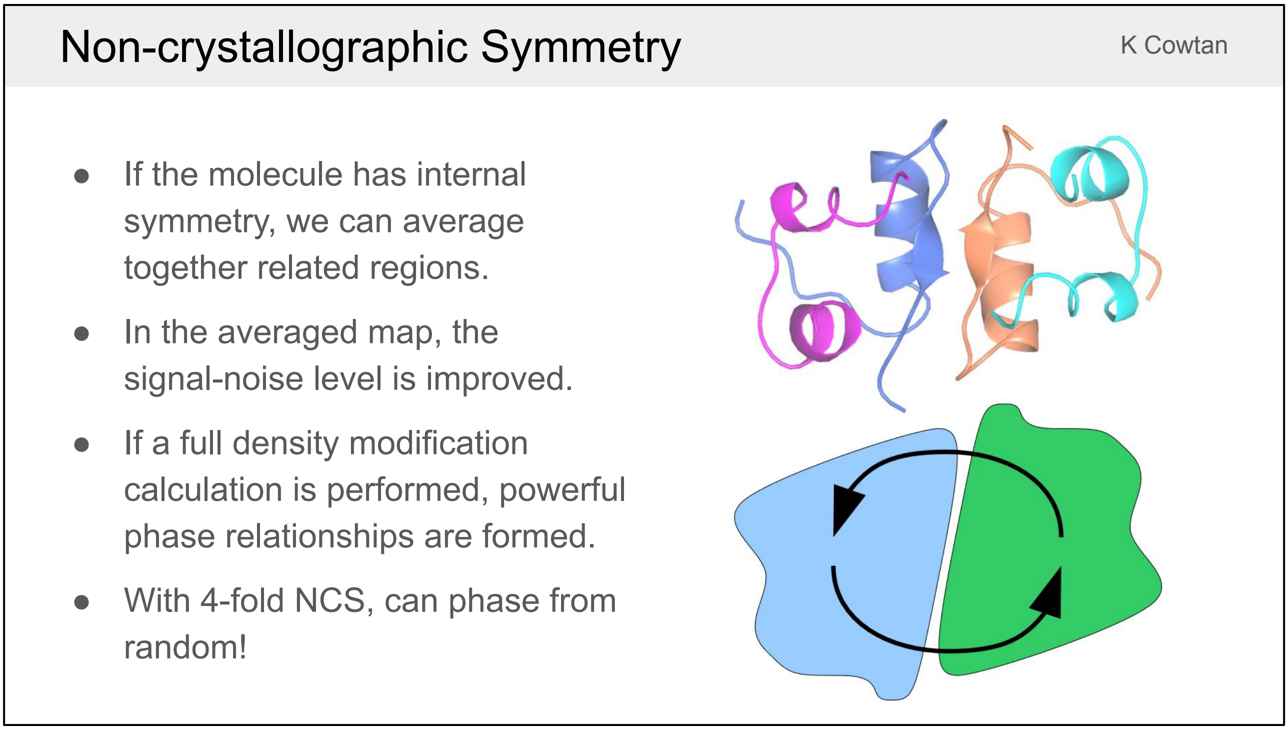
Sharpen the protein density by a transform which matches the histogram of a well phased map.

Useful at better than 4Å.





Histogram matching is a technique used to modify the density in the protein region which, even with a badly phased map, should have more real ordered features than noise. It uses the knowledge that pure noise maps have a gaussian distribution of density values but well-phased maps without noise have a more skewed distribution with fewer high density values (corresponding to atom positions) and more low density values (corresponding to the gaps between atoms). The protein density then gets modified using a monotonically increasing curve so that the density distribution looks like that of a well-phased map. This technique is generally useful at resolutions better than 4Å.



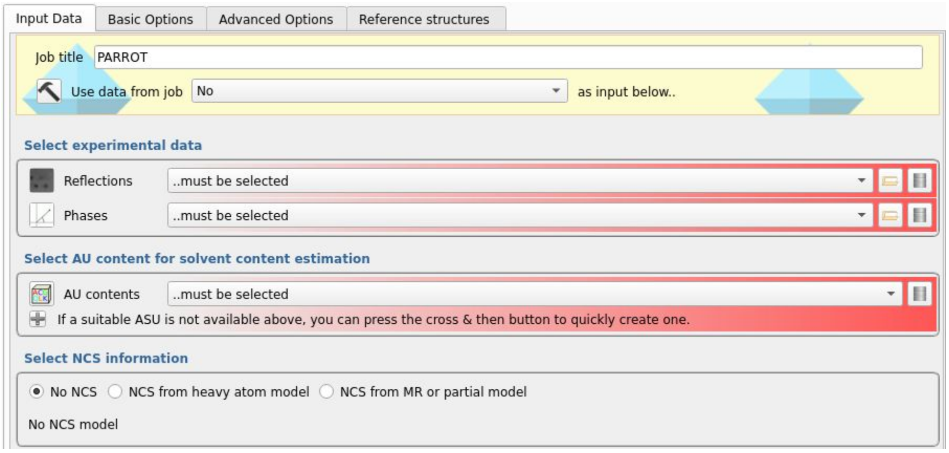
Non-crystallographic Symmetry

K Cowtan

- If the molecule has internal symmetry, we can average together related regions.
- In the averaged map, the signal-noise level is improved.
- If a full density modification calculation is performed, powerful phase relationships are formed.
- With 4-fold NCS, can phase from random!

Non-crystallographic symmetry (NCS) averaging is a more complicated technique than solvent flattening or histogram matching. If there are multiple copies of the same molecule within the asymmetric unit then it is possible to average together related regions of the density map. The example on the right shows an insulin dimer. It forms a hexamer in the crystal with three-fold crystallographic symmetry and two molecules in each asymmetric unit related by two-fold non-crystallographic symmetry. By averaging the density of the two molecules together, the real signal will be reinforced and the noise will sometimes reinforce and sometimes cancel out, which improves the signal-to-noise level. When the density modification calculation is performed cyclically, with the modified phases updating the previous ones for the next cycle, then relationships between the phases of different reflections are formed to give a big improvement. With 4-fold NCS, it is even theoretically possible to solve the phase problem starting from random phases, although this does not work in practice because we would not know which parts of the map to average.

CCP4i2 - Parrot



Job title PARROT

Use data from job No as input below..

Select experimental data

Reflections ..must be selected

Phases ..must be selected

Select AU content for solvent content estimation

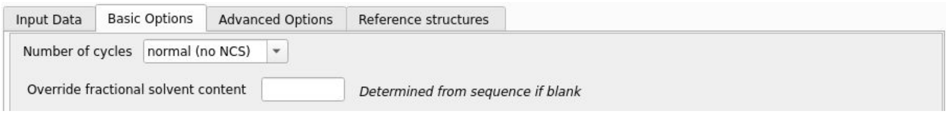
AU contents ..must be selected

If a suitable ASU is not available above, you can press the cross & then button to quickly create one.

Select NCS information

● No NCS ○ NCS from heavy atom model ○ NCS from MR or partial model

No NCS model

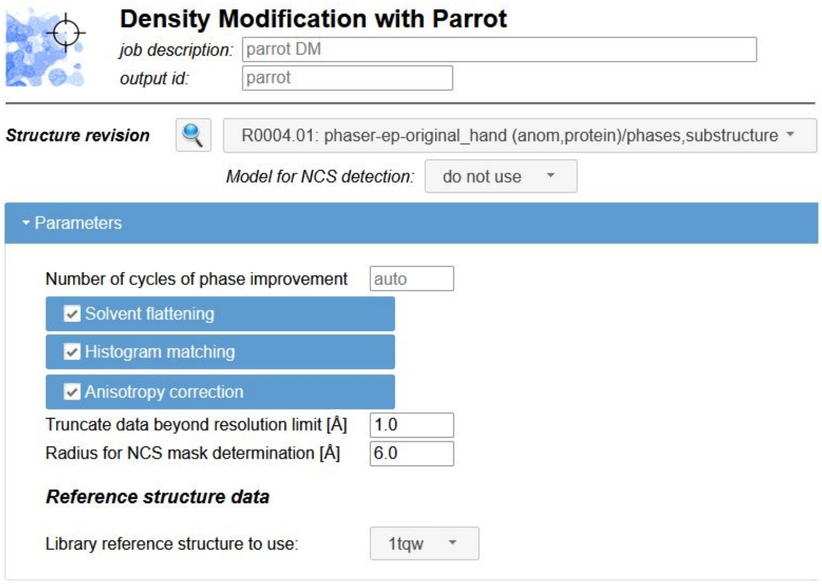


Number of cycles normal (no NCS)

Override fractional solvent content *Determined from sequence if blank*

This is the input page for Parrot in CCP4i2. It requires a set of reflections, starting phases that you want to improve and the asymmetric unit (AU) contents in order to estimate the solvent content. To determine the operators for NCS averaging, it must also be provided with either a heavy atom substructure or an atomic model (e.g. from MR or initial model building). The basic options tab allows you to specify the the number of cycles as either normal (without NCS), normal (with NCS) or many (with NCS), which correspond to 3, 10 or 100 respectively. Using NCS averaging means that more cycles can be performed without the resulting phases being too biased. The number of cycles option only changes the number of cycles and does not affect whether NCS averaging is performed. The tab also has the option of overriding the solvent content estimation. Note that the CCP4i2 Parrot task does not use the number of copies specified in the AU contents and will estimate the number of copies itself. If the number of copies is known and Parrot estimates it incorrectly then it is important to use this option as it affects the determination of the solvent mask. There are more advanced options in the other tabs that do not normally need to be changed.

CCP4 Cloud - Parrot



Density Modification with Parrot

job description: parrot DM

output id: parrot

Structure revision R0004.01: phaser-ep-original_hand (anom,protein)/phases,substructure

Model for NCS detection: do not use

Number of cycles of phase improvement: auto

☑ Solvent flattening

☑ Histogram matching

☑ Anisotropy correction

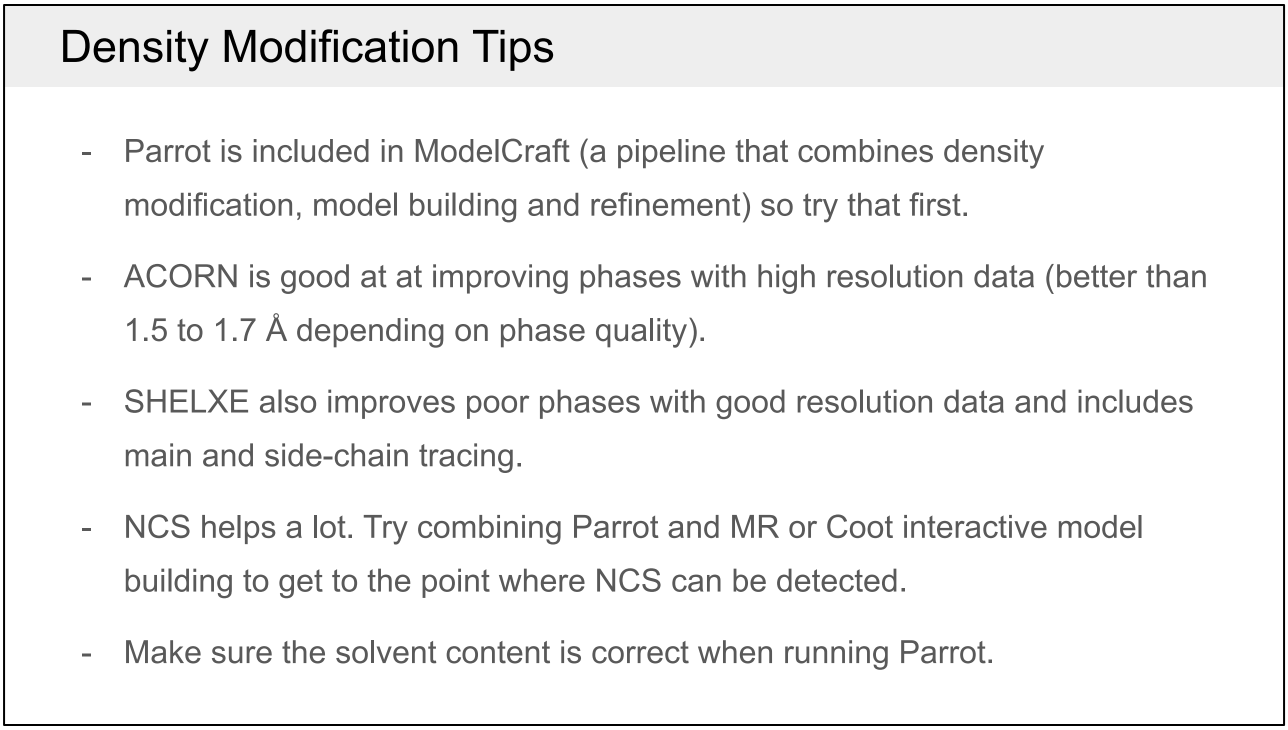
Truncate data beyond resolution limit [Å]: 1.0

Radius for NCS mask determination [Å]: 6.0

Reference structure data

Library reference structure to use: 1tqw

The is the interface to Parrot in CCP4 Cloud. It requires a structure revision that contains a set of starting phases. If the structure revision also has a model or a heavy atom substructure then that can be used for NCS detection. The number of cycles is chosen automatically by default, but it can be overridden. There are other advanced options that would not normally need to be changed, including options which are not in CCP4i2 to turn off the solvent flattening or histogram matching steps. Unlike the CCP4i2 task, CCP4 Cloud uses the number of copies specified in the asymmetric unit contents to calculate a solvent content that is given to Parrot. If the number of copies is not known it is recommended to create branches in the project tree with different contents, starting with the most likely.



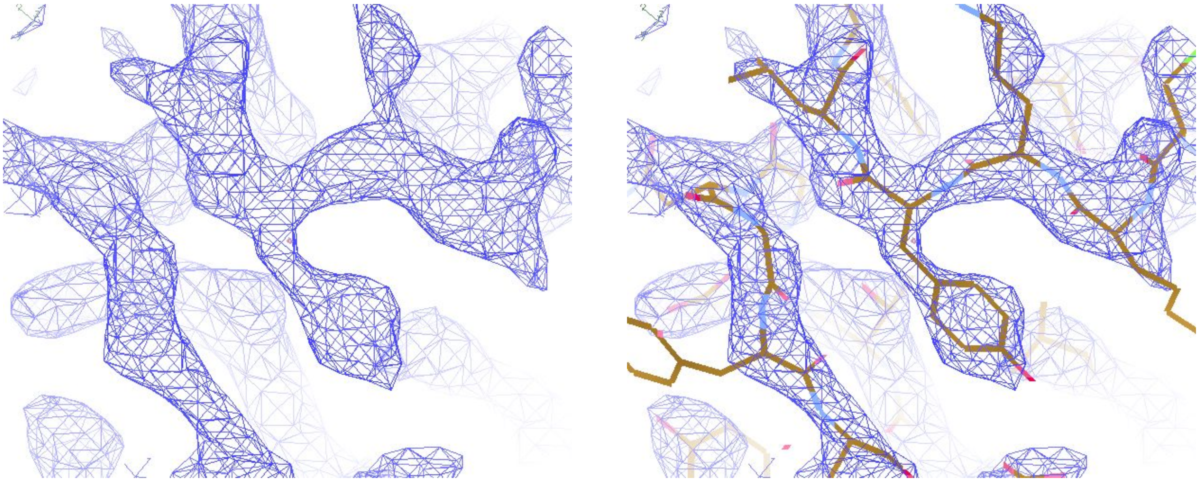
Density Modification Tips

- Parrot is included in ModelCraft (a pipeline that combines density modification, model building and refinement) so try that first.
- ACORN is good at at improving phases with high resolution data (better than 1.5 to 1.7 Å depending on phase quality).
- SHELXE also improves poor phases with good resolution data and includes main and side-chain tracing.
- NCS helps a lot. Try combining Parrot and MR or Coot interactive model building to get to the point where NCS can be detected.
- Make sure the solvent content is correct when running Parrot.

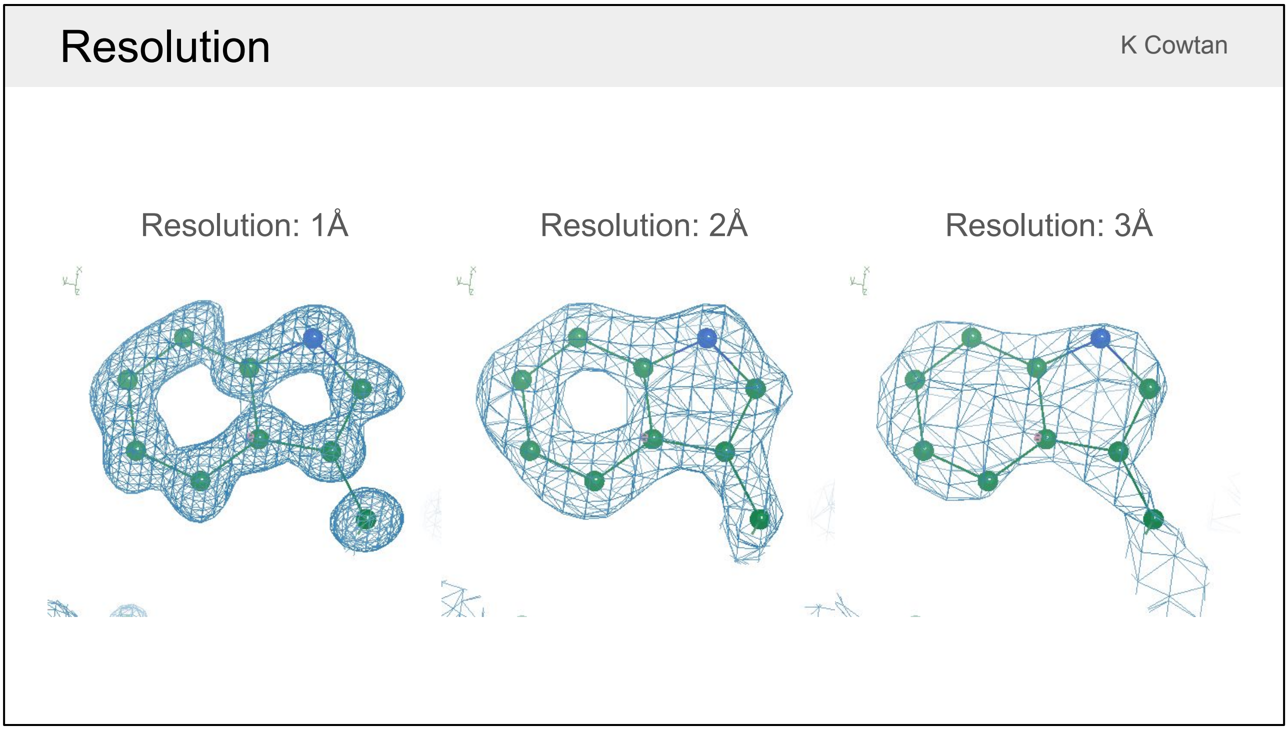
ModelCraft is an model-building pipeline that is available in both CCP4i2 and CCP4 Cloud. It includes density modification with Parrot, so it is worth first trying ModelCraft first as it may also build most of the model automatically. If ModelCraft does not work with the input data you give it then it is worth trying to improve the input through the techniques suggested in this slide.

Model Building

K Cowtan



Now we will look at automated model building. *Buccaneer* and *Nautilus* are programs for automated protein and nucleotide building, respectively. They are provided with a density map, target sequences and optionally an existing model, and they attempt to automatically build a model to fit the map.



Resolution

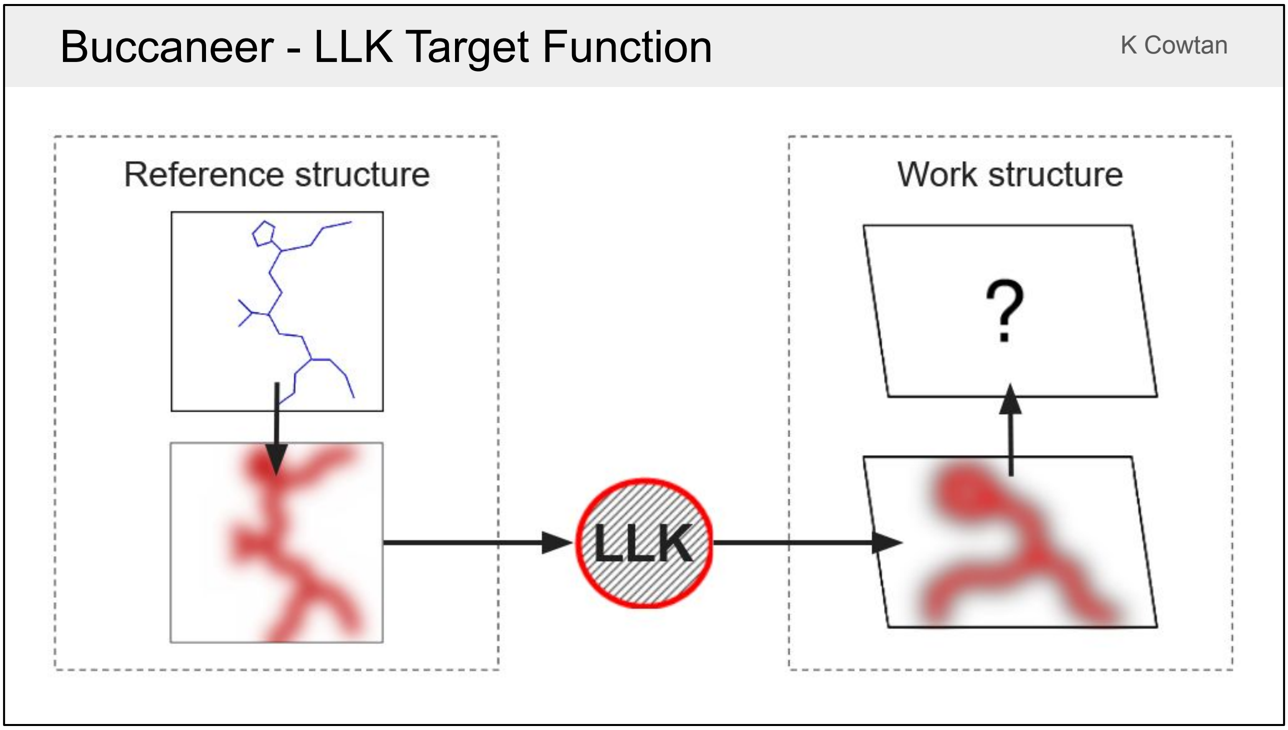
K Cowtan

Resolution: 1Å

Resolution: 2Å

Resolution: 3Å

A big problem with automated model building is that electron density maps look very different at different resolutions. At high resolution you can distinguish peaks for individual atoms, but if you write a program that looks for separate atomic peaks then it wont work when you have a low resolution map and you can't see them.



Buccaneer - LLK Target Function

K Cowtan

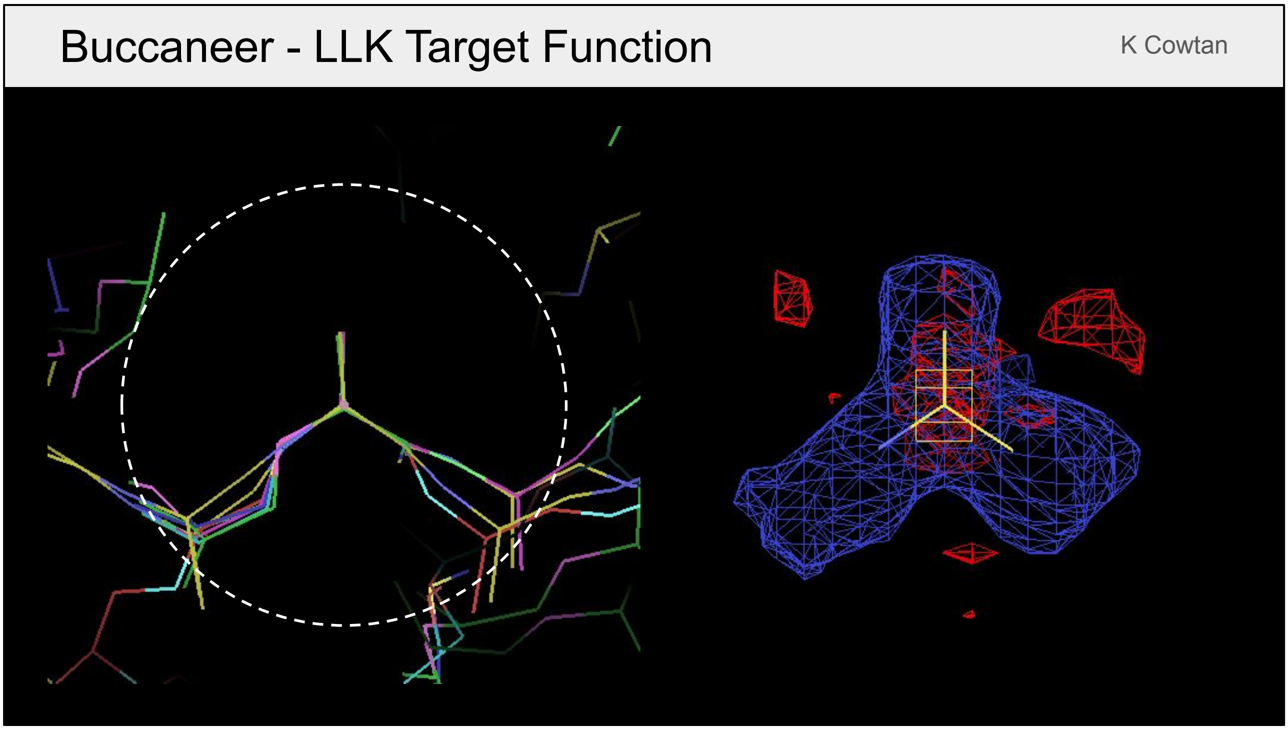
Reference structure

Work structure

?

LLK

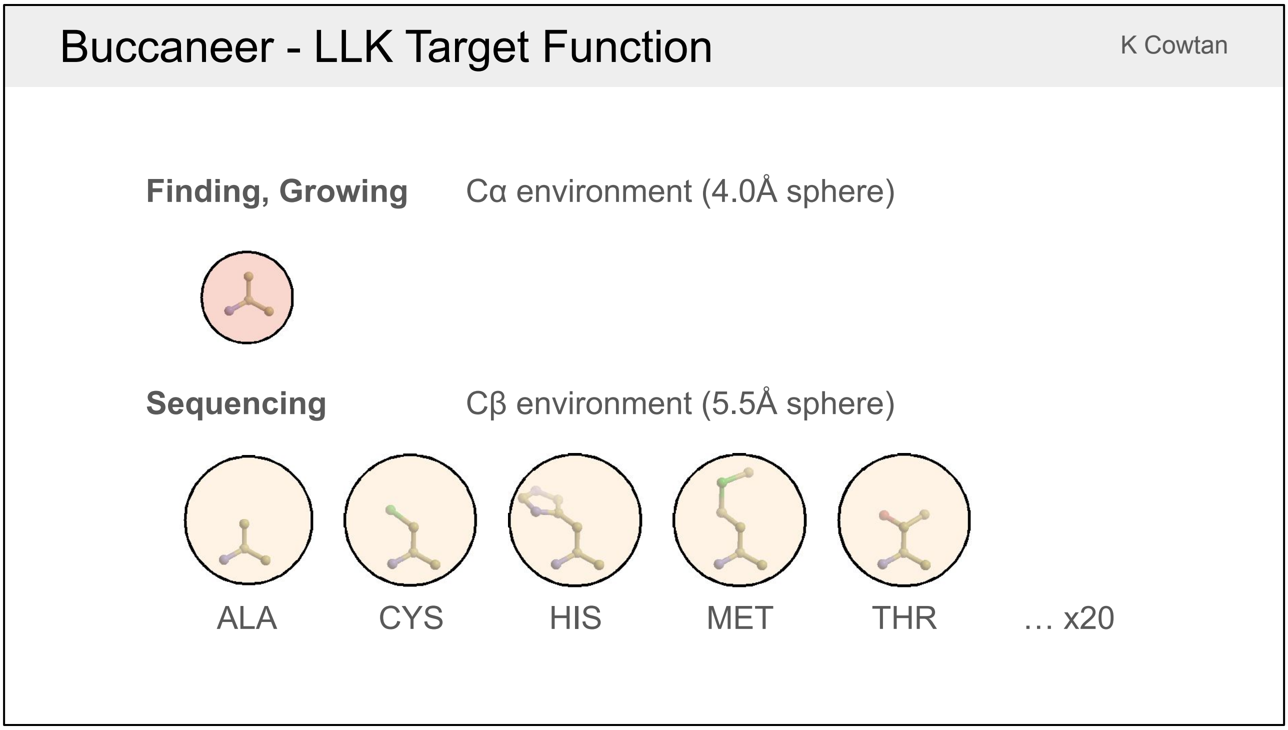
Buccaneer uses a known reference structure in order to help it build at both high and low resolution. You give B*uccaneer* the map you're trying to build a model into (the work map) and it generates a map for a reference structure with the same resolution and same level of noise as this map. Then it uses the reference structure to find out what features in the reference map look like by creating a log-likelihood (LLK) target. The LLK target is then used to search for the same features in the work map.



Buccaneer - LLK Target Function

K Cowtan

The target to search for individual residues is obtained by superimposing all the residues in the reference structure on top of each other. The mean and variance of the density is then used to construct a likelihood target within a 4 Å sphere. The blue density shows regions of conserved high density and the red density shows regions of conserved low density. Both are important when finding residues in the work map. If there is low density in the blue regions or high density in the red regions then it is likely not the correct position/orientation for a residue.



Buccaneer - LLK Target Function

K Cowtan

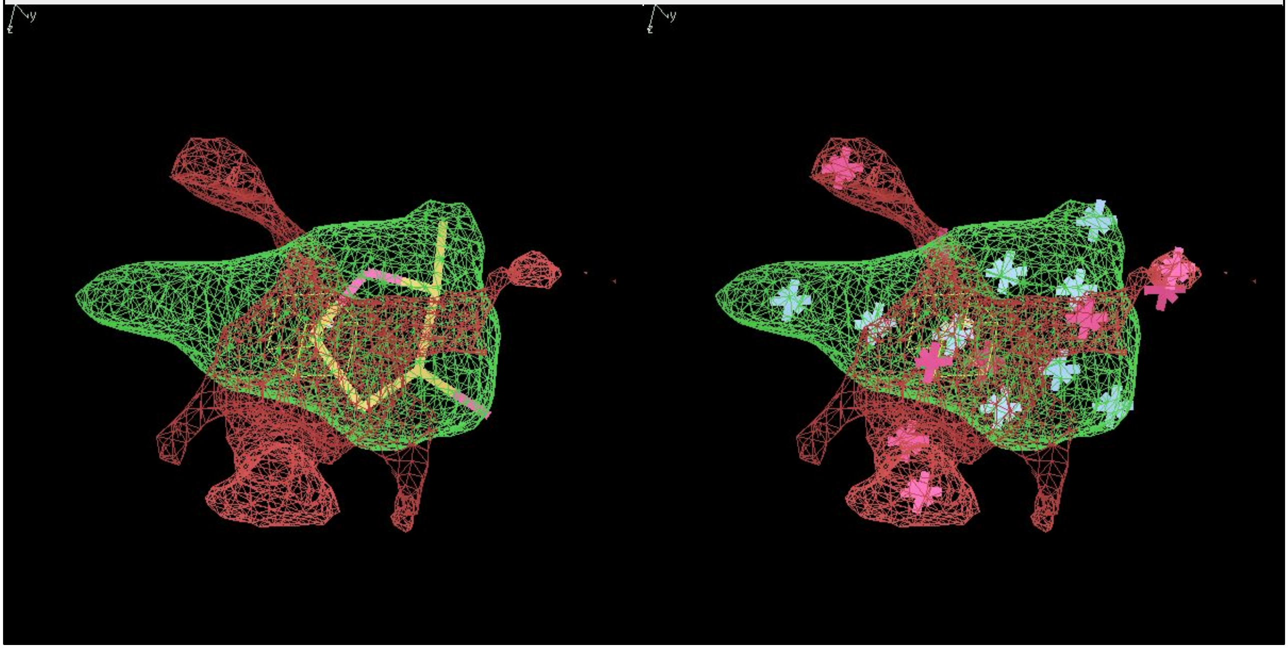
Finding, Growing Cα environment (4.0Å sphere)

Sequencing Cβ environment (5.5Å sphere)

The 4 Å C-alpha target is used in the first step of the program to find initial residue positions and the second step to grow those residues into chains. The program also uses 5.5 Å targets centred around C-beta atoms. A different target is made for each residue type and those are used for sequencing. Rotamers are not separated before making the targets, so the mean and variance of the density is calculated for side chains pointing in lots of different directions, but this still provides enough information to find the correct sequence.

Nautilus - Fingerprints

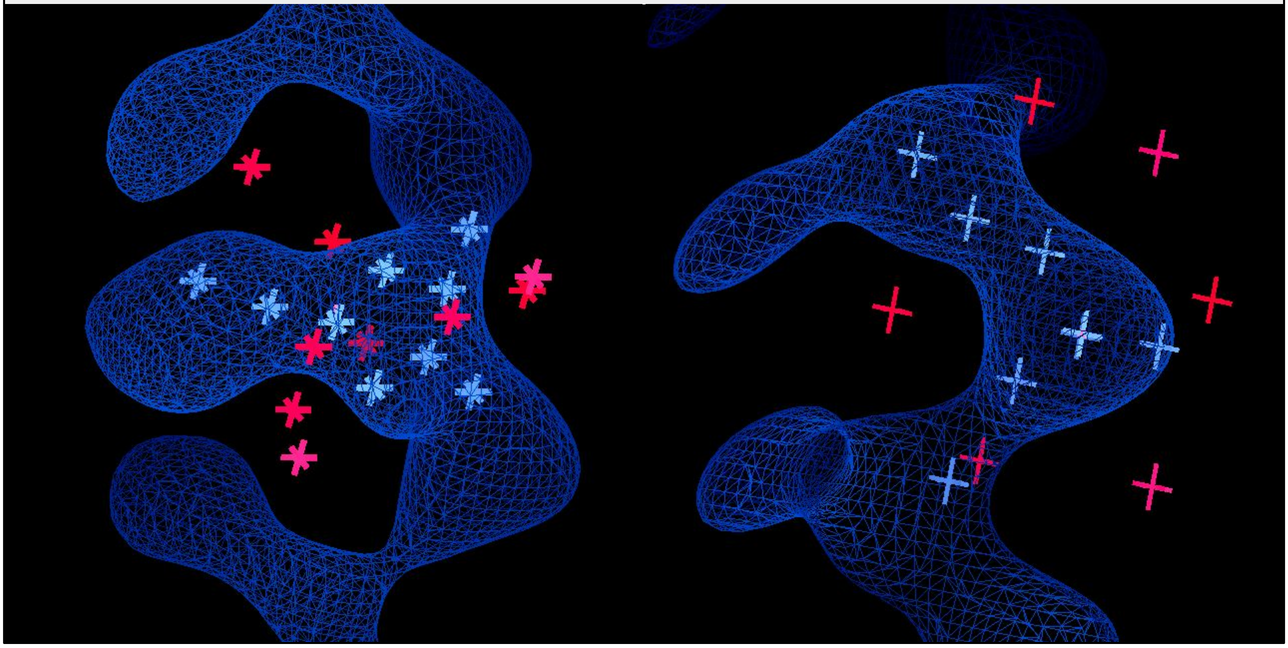
K Cowtan



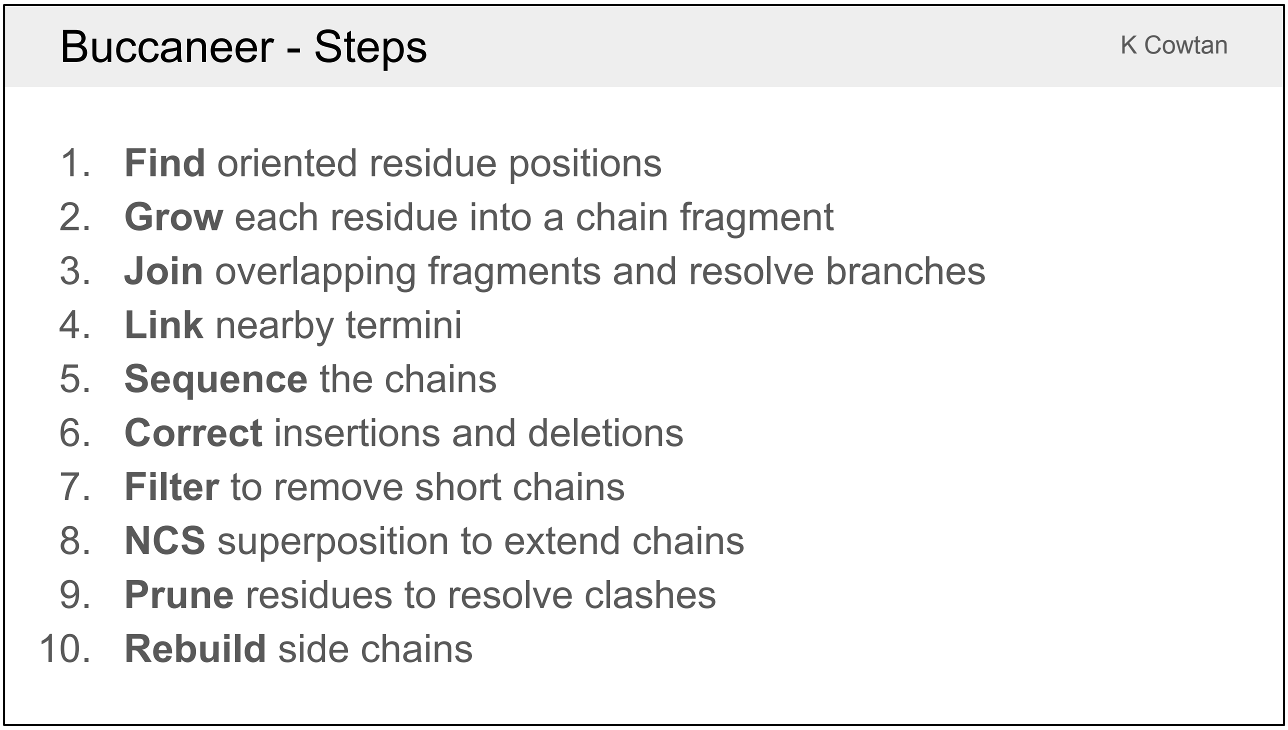
Like *Buccaneer*, *Nautilus* also identifies residue positions by looking for patterns of conserved high and low density, but it does so using a fingerprint detection method instead of simulating a map for a reference structure. The green map shows conserved high density for overlapped sugars. The the first part of the base is visible but phosphates aren't because they have multiple possible conformations. The red map shows conserved regions of low density. Probe points are placed at positions where high and low density are expected. This is a faster method than having a full LLK target function because it only looks at a few key points.

Nautilus - Fingerprints

K Cowtan



The fingerprints are translated and rotated in the map to find positions that have high and low density in the right places. The same process was followed to create a fingerprint for phosphate. It has high probe points on the phosphate atoms and in the neighbouring sugars and low probe points in the surrounding regions. This shows the sugar and phosphate fingerprints in place over some example density.



Buccaneer - Steps

K Cowtan

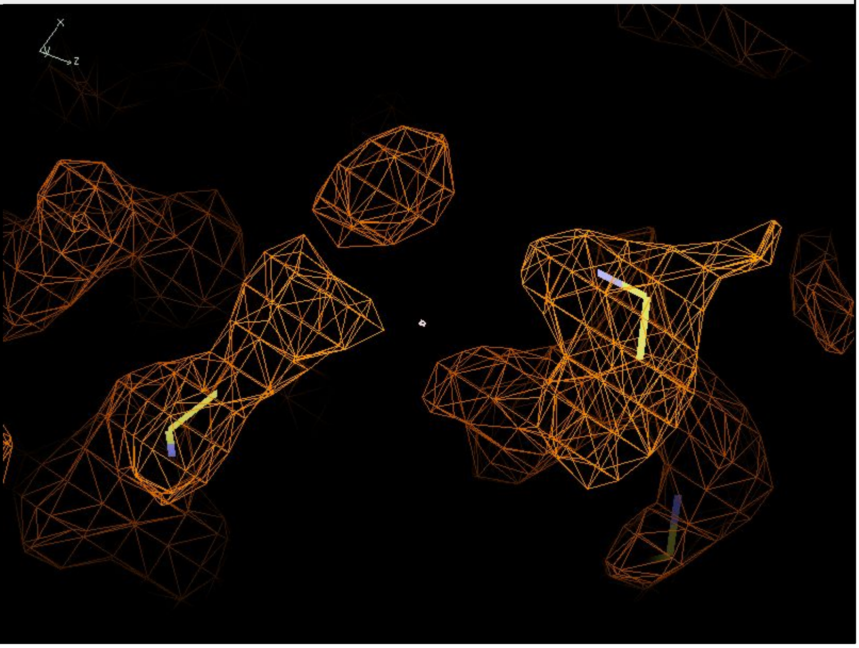
1. **Find** oriented residue positions
2. **Grow** each residue into a chain fragment
3. **Join** overlapping fragments and resolve branches
4. **Link** nearby termini
5. **Sequence** the chains
6. **Correct** insertions and deletions
7. **Filter** to remove short chains
8. **NCS** superposition to extend chains
9. **Prune** residues to resolve clashes
10. **Rebuild** side chains

Buccaneer and *Nautilus* both run as a cyclic calculation with multiple internal steps. These are the 10 steps in *Buccaneer*. The steps for *Nautilus* are similar.

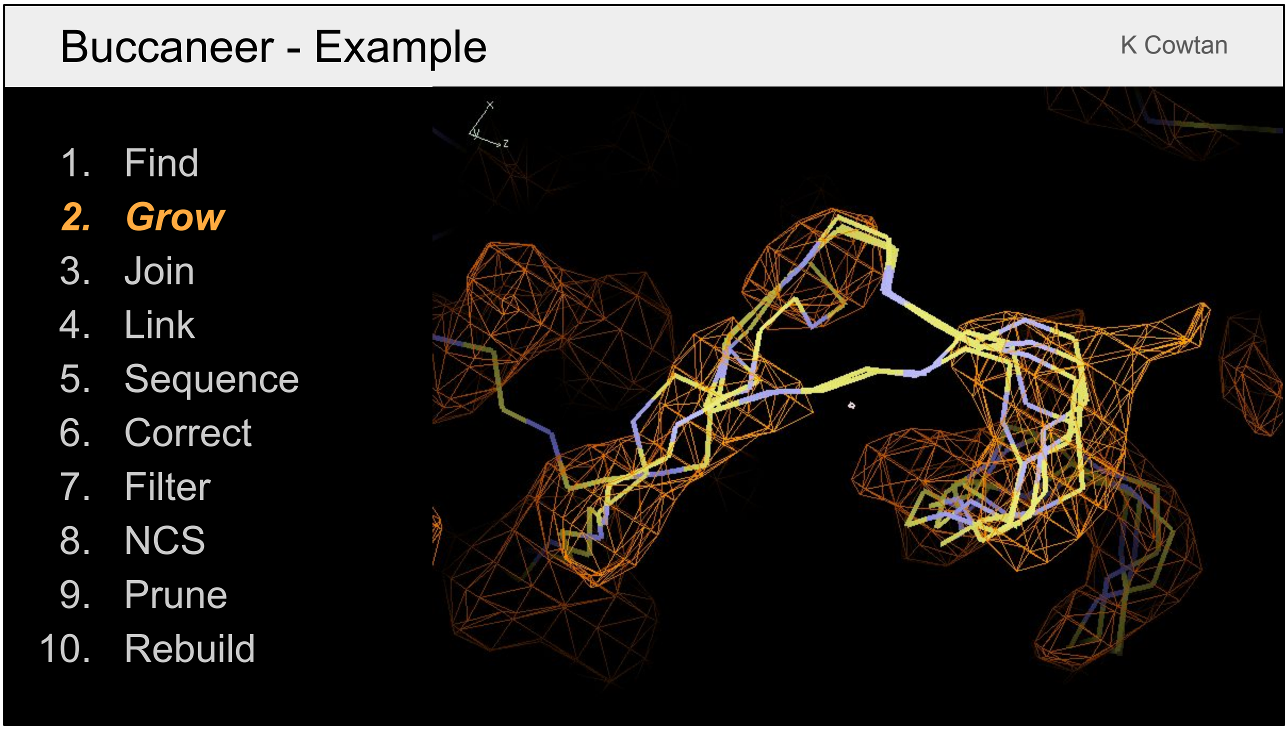
Buccaneer - Example

K Cowtan

1. ***Find***
2. Grow
3. Join
4. Link
5. Sequence
6. Correct
7. Filter
8. NCS
9. Prune
10. Rebuild



This is an example of *Buccaneer* building a difficult loop in a 2.9Å map. The first step is to find possible positions for new residues. It has placed residues at three positions in this figure (two in the foreground and one in the background) where the density has the right shape according to the C-alpha target function. There are also other residues placed out of this view.



Buccaneer - Example

K Cowtan

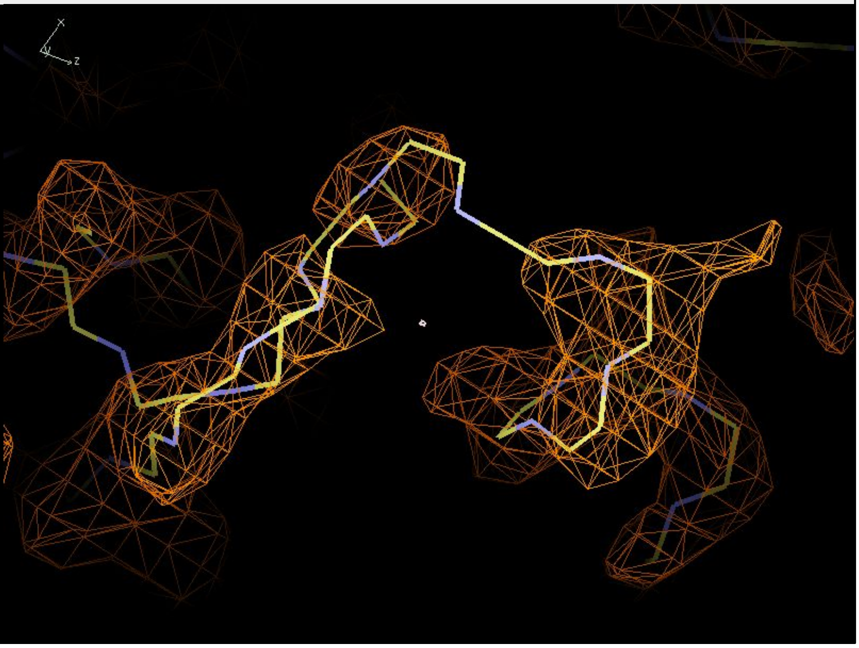
1. Find
2. ***Grow***
3. Join
4. Link
5. Sequence
6. Correct
7. Filter
8. NCS
9. Prune
10. Rebuild

The next step is to grow each residue into a long chain fragment. New residues get added at both ends using an exhaustive search over allowed Ramachandran angles and the same C-alpha target. Once the target score drops below a threshold value the growing stops. This step gives lots of fragments, many of which overlap. In this example it has built a helix on the right where the chains agree fairly well, two different paths to bridge the gap in the middle and a contradictory chain on the left that has been built in the opposite direction.

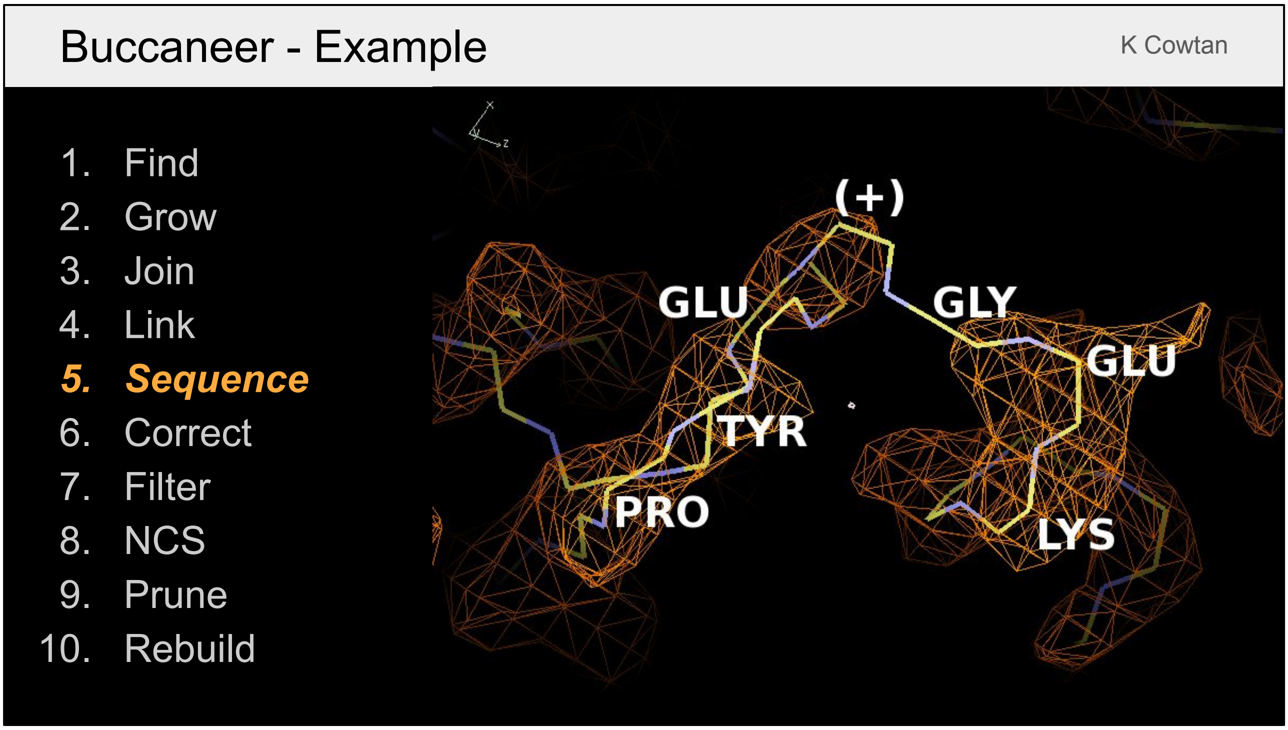
Buccaneer - Example

K Cowtan

1. Find
2. Grow
3. ***Join***
4. Link
5. Sequence
6. Correct
7. Filter
8. NCS
9. Prune
10. Rebuild



The joining step merges all the chain fragments that agree with each other into single chains. This step also has to make decisions about which routes to follow. It tries to build the longest chains (e.g. it chose the longest route over the middle loop), which helps to build helices correctly. The chain built in the reversed direction is still there because it can’t be merged and at this stage it hasn’t been decided which is correct.

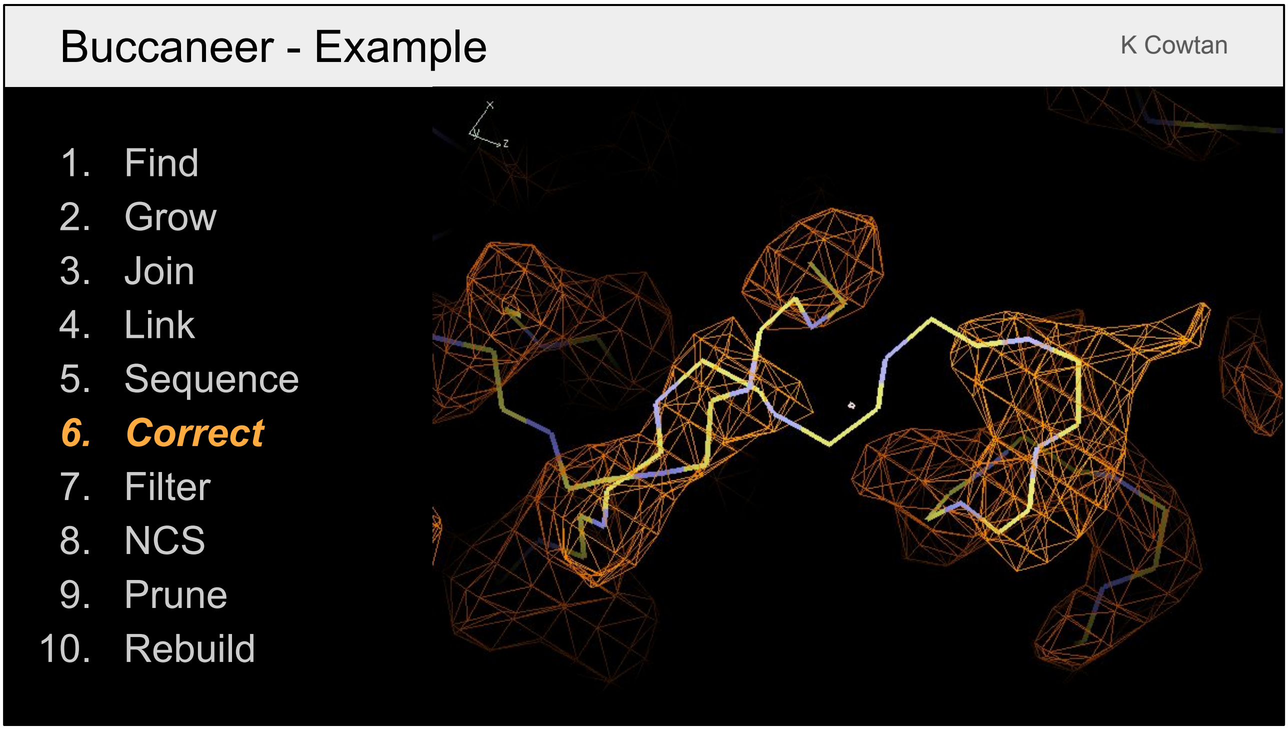


Buccaneer - Example

K Cowtan

1. Find
2. Grow
3. Join
4. Link
5. ***Sequence***
6. Correct
7. Filter
8. NCS
9. Prune
10. Rebuild

We have skipped the linking step because there are no termini that can be linked in this view. The sequencing step works by assigning each residue a probability that it is each of the 20 residue types using the C-beta targets. These probabilities are then compared to the known sequence(s) to look for continuous runs that stand out with a high probability. In this example, the sequence only fits the chain well if an extra symbol is added in the middle. This is called an insertion and (unless the sequence is wrong) it means the chain has been built incorrectly. The reversed chain on the left wasn't sequenced as there was no part of the sequence that fits well.

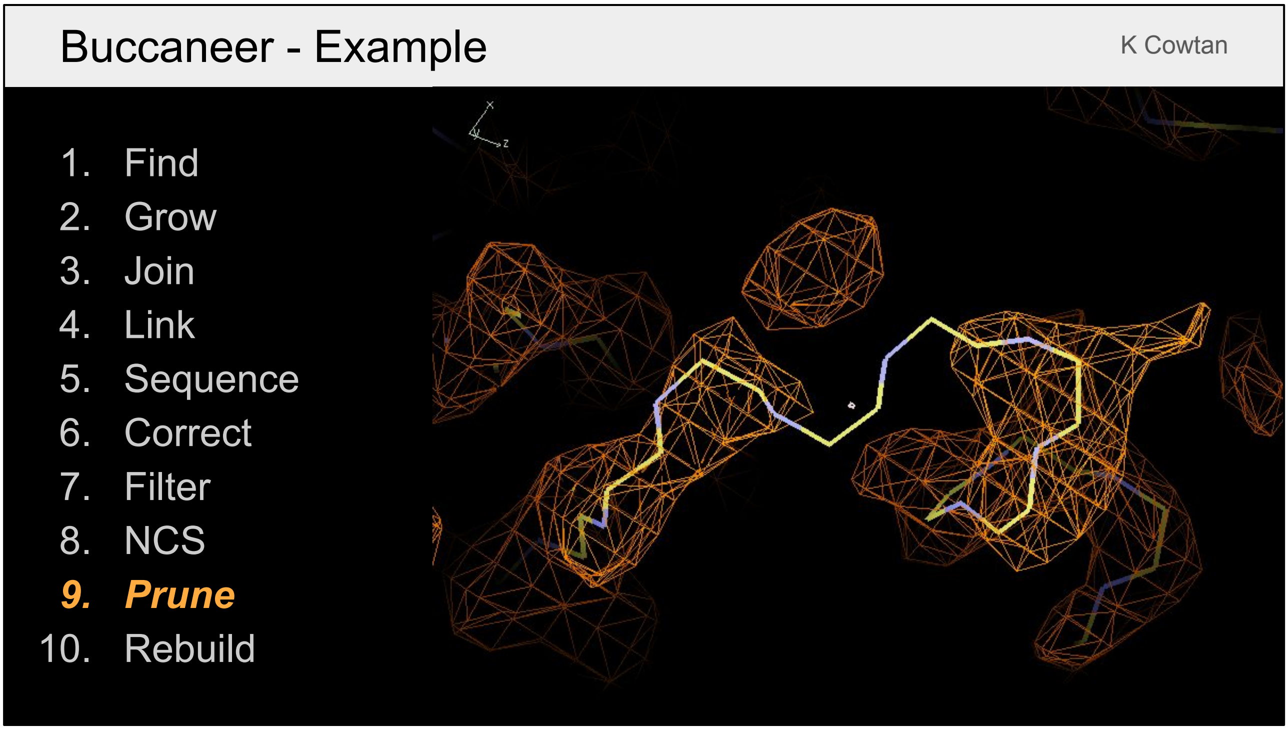


Buccaneer - Example

K Cowtan

1. Find
2. Grow
3. Join
4. Link
5. Sequence
6. ***Correct***
7. Filter
8. NCS
9. Prune
10. Rebuild

The next step is to correct any insertions (where it has built one extra residue) and deletions (where it has built one fewer residue) found during the sequencing step. The insertion that was over this loop as been corrected by removing three residues and rebuilding two.

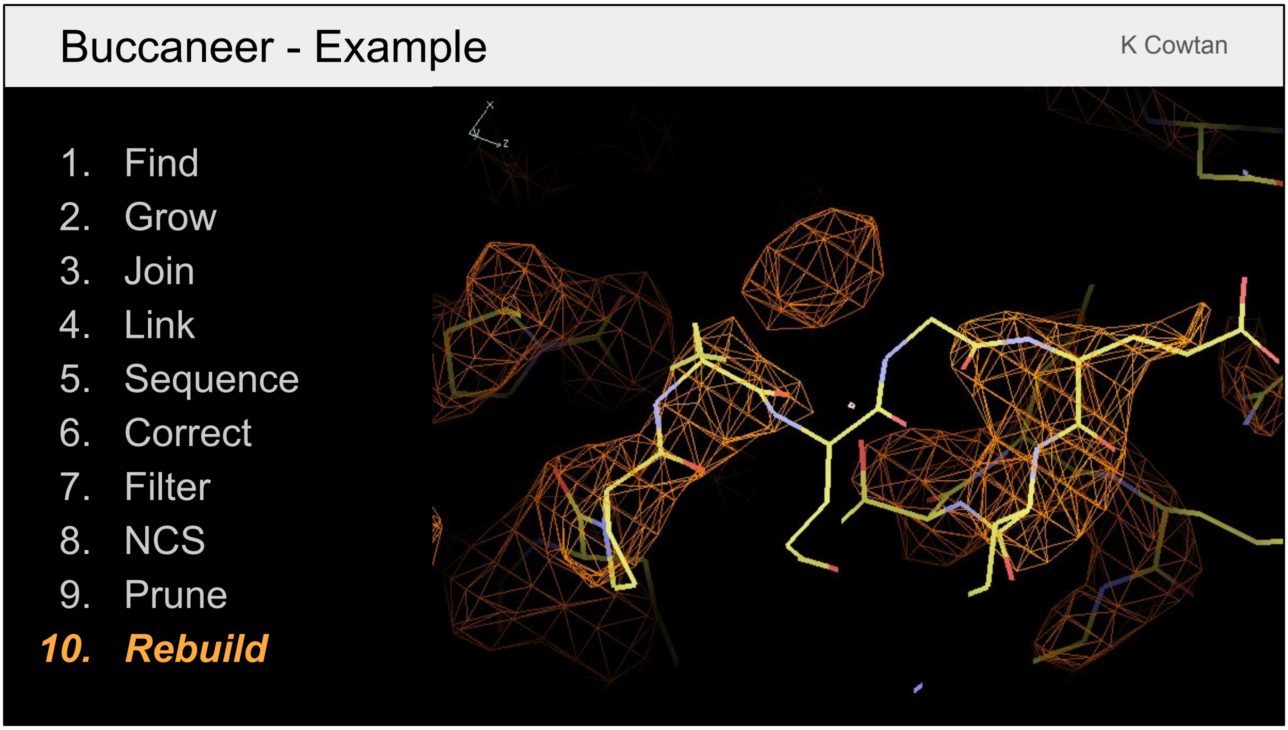


Buccaneer - Example

K Cowtan

1. Find
2. Grow
3. Join
4. Link
5. Sequence
6. Correct
7. Filter
8. NCS
9. ***Prune***
10. Rebuild

The pruning step removes residues from clashing chains to produce a single consistent model. If the pruned chains have less than 6 residues remaining they are removed too. Pruning is done at this late stage to have the most information about which chain is correct. Residues that are unsequenced or are from shorter chains with worse fit to the density will be removed preferentially. In this case the reversed chain on the left was removed.

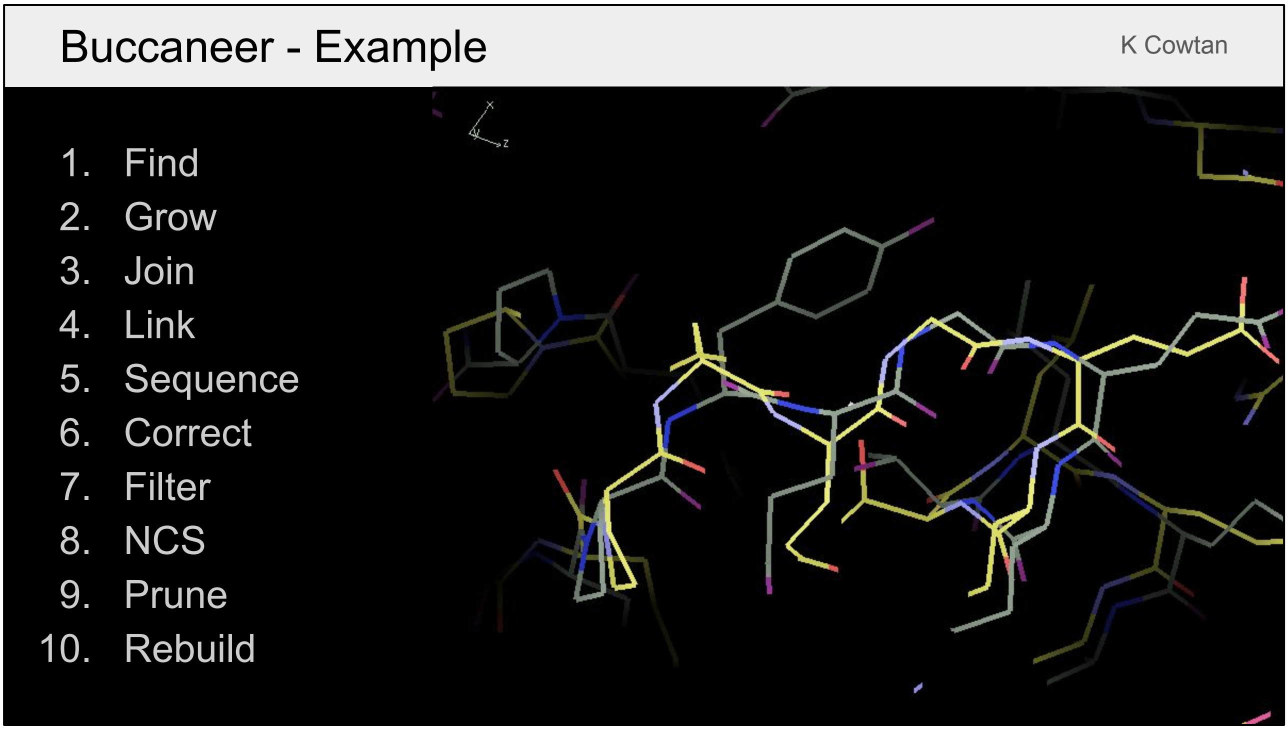


Buccaneer - Example

K Cowtan

1. Find
2. Grow
3. Join
4. Link
5. Sequence
6. Correct
7. Filter
8. NCS
9. Prune
10. ***Rebuild***

Finally, the last step is to build side chains for each residue using a rotamer library. The rotamer with the highest average density at the atomic positions is chosen. If two side chains end up clashing, *Buccaneer* will try and find a pair of rotamers that do not clash. If this fails both residues will be truncated to C-beta.

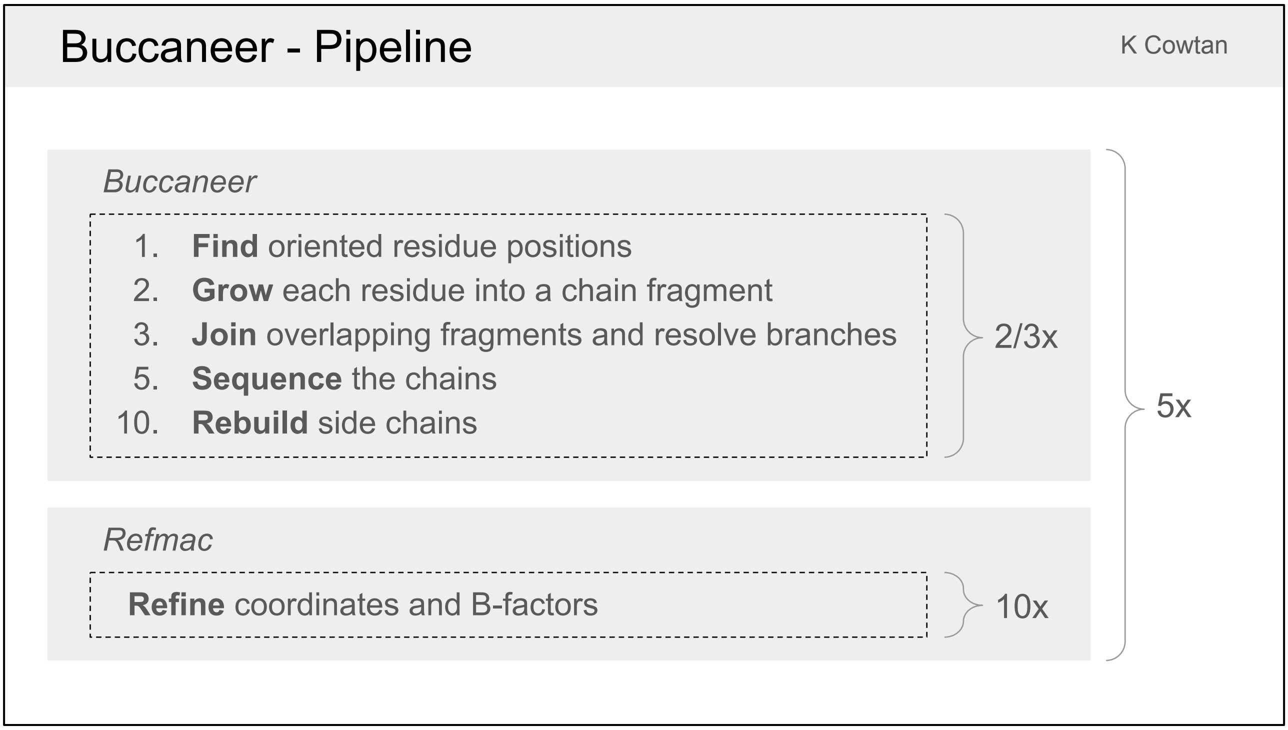


Buccaneer - Example

K Cowtan

1. Find
2. Grow
3. Join
4. Link
5. Sequence
6. Correct
7. Filter
8. NCS
9. Prune
10. Rebuild

This shows a comparison with the deposited model. *Buccaneer* built most of the model quite well but got some things wrong, for example the the tyrosine side chain. The model will improve after refinement with *Refmac* and further cycles of *Buccaneer*.



Buccaneer - Pipeline

K Cowtan

Buccaneer

1. **Find** oriented residue positions
2. **Grow** each residue into a chain fragment
3. **Join** overlapping fragments and resolve branches
5. **Sequence** the chains
10. **Rebuild** side chains

2/3x

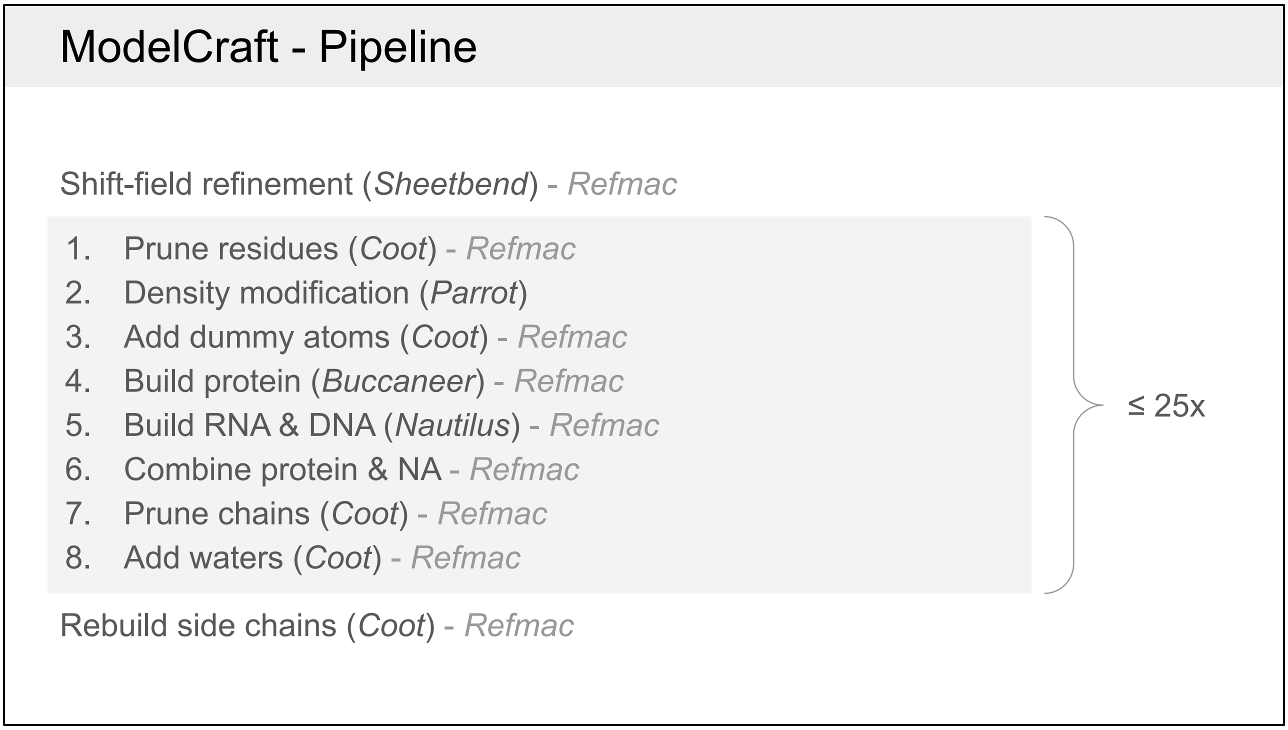
Refmac

Refine coordinates and B-factors

10x

5x

Buccaneer and *Nautilus* do not perform global refinement of the model, so the graphical interface actually runs a pipeline that combines them with *Refmac*, which refines the model coordinates and B-factors and produces an updated map for further building. The original pipeline in *CCP4i* runs 2 or 3 cycles of *Buccaneer* or *Nautilus* followed by 10 cycles of *Refmac*, repeats that 5 times, and outputs the final model.



ModelCraft - Pipeline

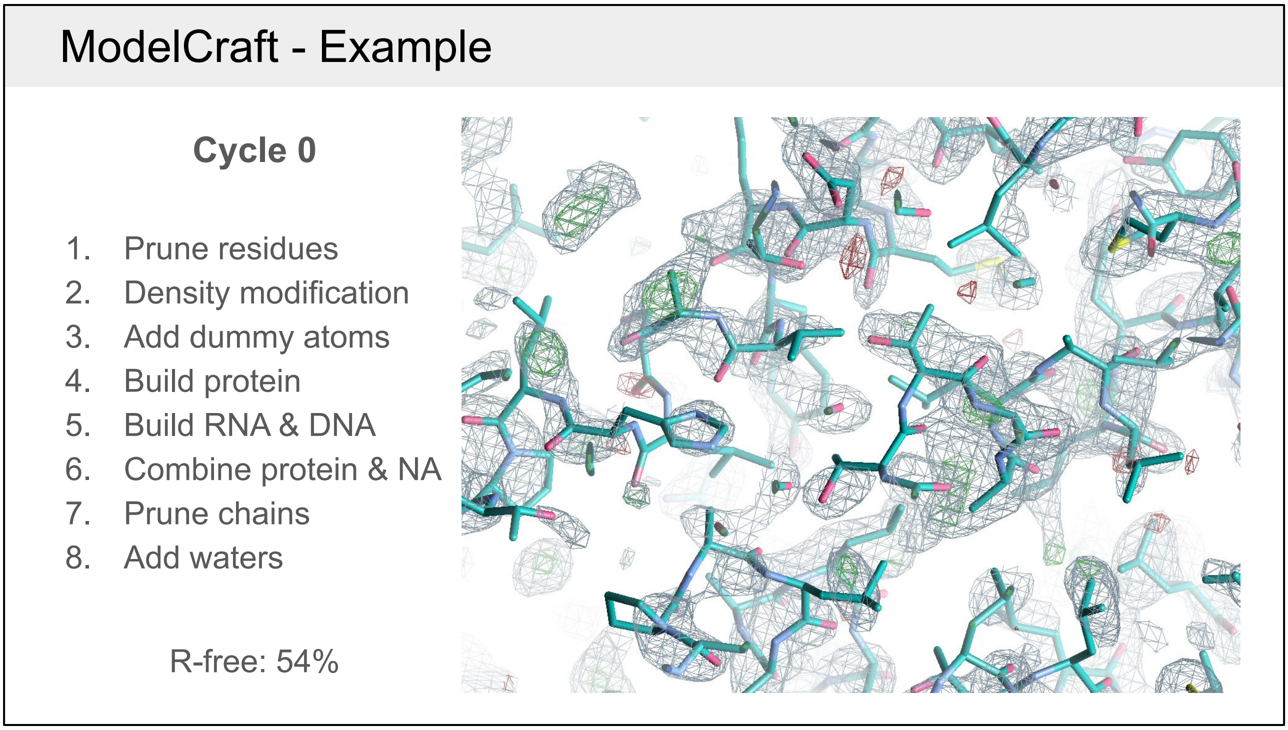
Shift-field refinement (*Sheetbend*) - *Refmac*

1. Prune residues (*Coot*) - *Refmac*
2. Density modification (*Parrot*)
3. Add dummy atoms (*Coot*) - *Refmac*
4. Build protein (*Buccaneer*) - *Refmac*
5. Build RNA & DNA (*Nautilus*) - *Refmac*
6. Combine protein & NA - *Refmac*
7. Prune chains (*Coot*) - *Refmac*
8. Add waters (*Coot*) - *Refmac*

≤ 25x

Rebuild side chains (*Coot*) - *Refmac*

ModelCraft is a new pipeline that was created to help build more structures when the initial phases are poor, such as after molecular replacement with a very incomplete or dissimilar model. It was also designed to build both protein and nucleic acids in the same pipeline. This is the overall *ModelCraft* pipeline as it stands. If you provide a starting model it is first refined using *Sheetbend* then *Refmac*. Then a single cycle of the pipeline has 8 steps, most of which are followed by refinement with *Refmac*. With high resolution data, it starts by pruning individual residues using *Coot*. Then it uses *Parrot* for density modification, which is especially powerful if there is non-crystallographic symmetry that can be determined from the current model. After *Parrot*, it adds dummy atoms using *Coot* to improve the phases, but the dummy atom structure is only accepted if R-free improves. The dummy atoms are removed from the model then *Buccaneer* builds the protein and *Nautilus* builds DNA and RNA. If the model contains both protein and DNA/RNA the results of *Buccaneer* and *Nautilus* are combined. *Coot* is used to remove incorrect chain fragments and to add waters, but again only accepting the result if R-free improves. Once the model stops improving (or it reaches 25 cycles) the cycle with the best model is used as the output. If the resolution is high then *Coot* is used to rebuild missing or incorrect side chains.



ModelCraft - Example

Cycle 0

1. Prune residues
2. Density modification
3. Add dummy atoms
4. Build protein
5. Build RNA & DNA
6. Combine protein & NA
7. Prune chains
8. Add waters

R-free: 54%

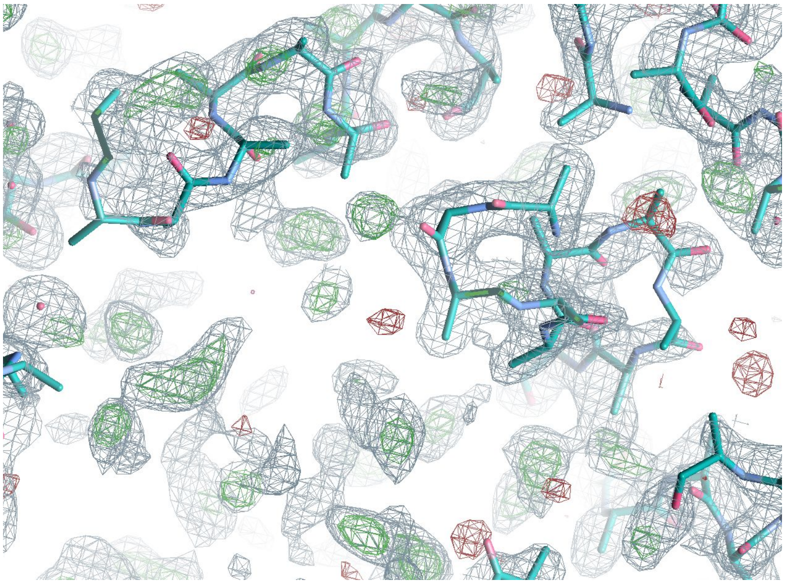
This is a difficult example where *ModelCraft* worked well. The target structure is 3GVP with 4 copies of a 435 residue protein and data to 2.25 Å resolution. The *AlphaFold* model was trimmed to remove residues with high pLDDT but it was not split into separate domains for molecular replacement. *MOLREP* placed four copies but only three of them were placed correctly. The F-map correlation after MR was 0.3, which is on the threshold of what would be considered a solution. The phases are made worse by the relative domain positions being slightly different in the true structure and the *AlphaFold* model. This is a view over the incorrectly placed copy after refinement with *Sheetbend* and *Refmac*, with the R-free value stuck around 54%.

ModelCraft - Example

Cycle 3

1. Prune residues
2. Density modification
3. Add dummy atoms
4. Build protein
5. Build RNA & DNA
6. Combine protein & NA
7. Prune chains
8. Add waters

R-free: 50%



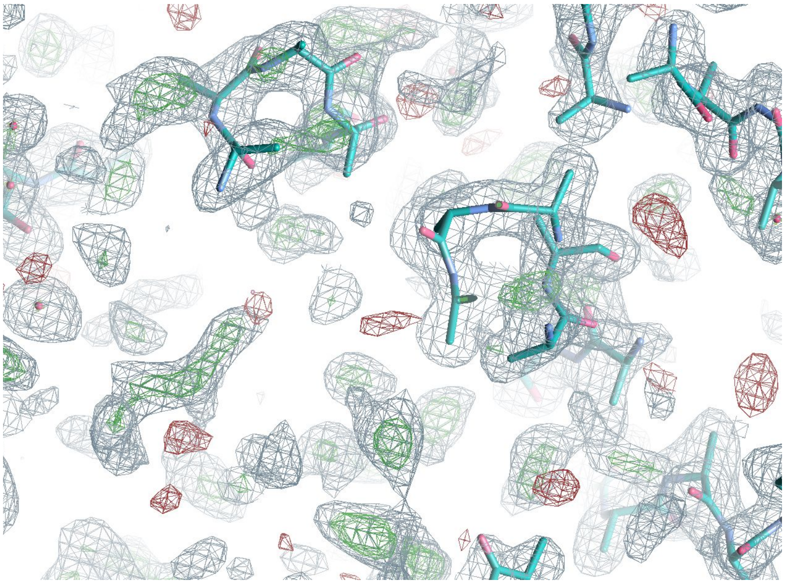
This is the same view but after 3 cycles of *ModelCraft*. As you can see, the model and map look completely different. The original structure in this region has been removed and now there are just a few small fragments built by *Buccaneer*. R-free has reduced slightly from 54% to 50%.

ModelCraft - Example

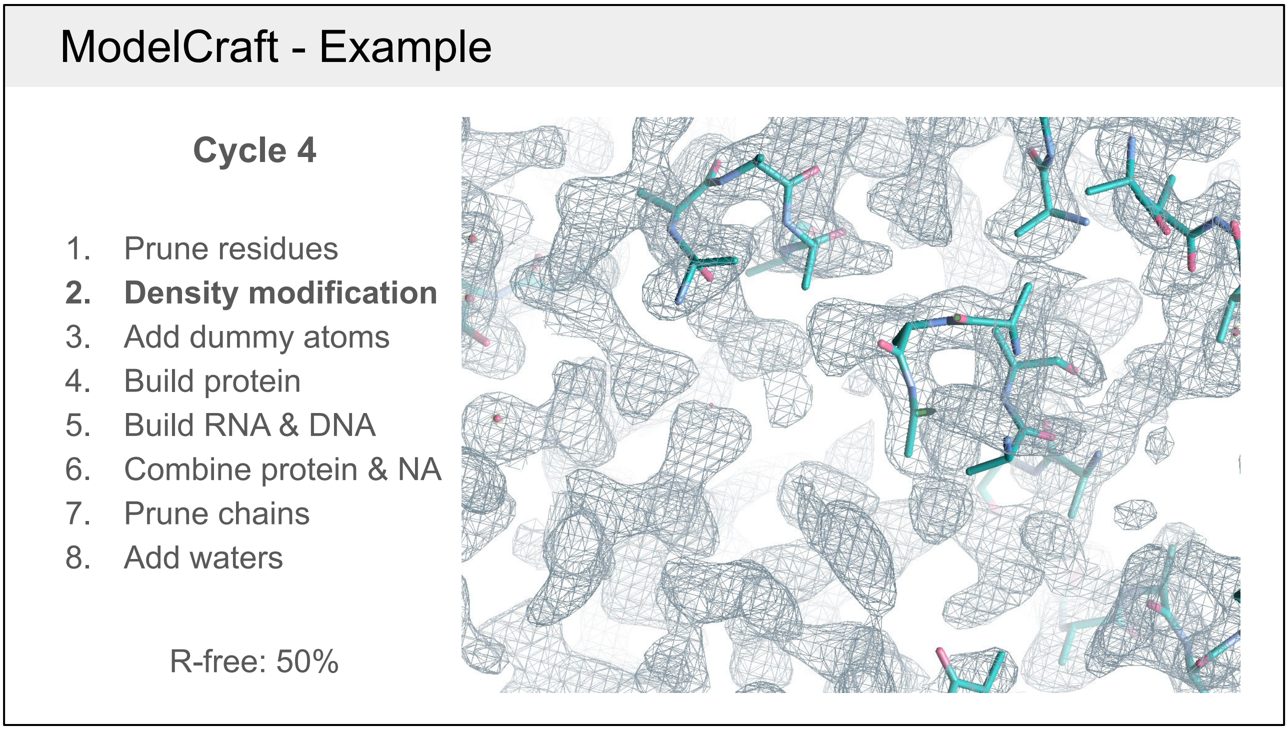
Cycle 4

1. **Prune residues**
2. Density modification
3. Add dummy atoms
4. Build protein
5. Build RNA & DNA
6. Combine protein & NA
7. Prune chains
8. Add waters

R-free: 50%



A large improvement is seen in the 4th cycle. The first step is to remove the worst residues using *Coot* and refine again. This step is only done if the resolution is better than 2.3 Å. The density for the removed residues in the top left disappeared after further refinement, suggesting that the density was biased.



ModelCraft - Example

Cycle 4

1. Prune residues
2. **Density modification**
3. Add dummy atoms
4. Build protein
5. Build RNA & DNA
6. Combine protein & NA
7. Prune chains
8. Add waters

R-free: 50%

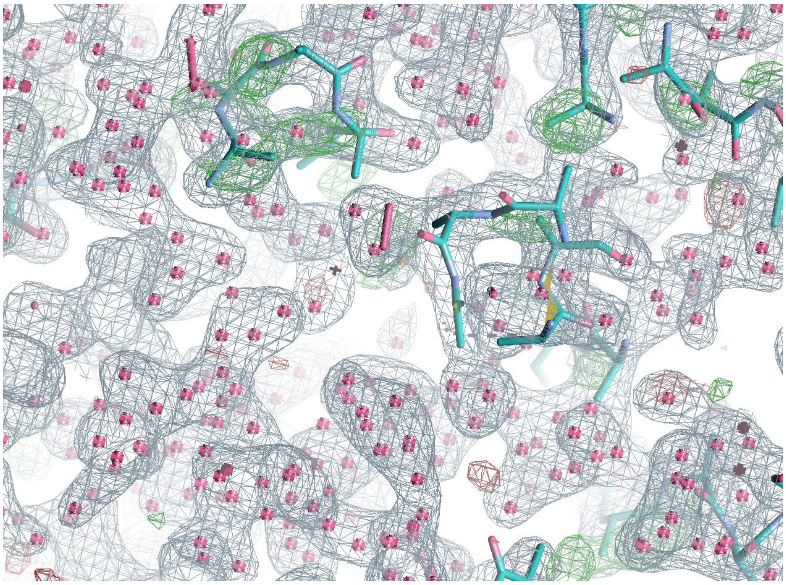
The second step is density modification using *Parrot*, which is especially powerful when there are multiple copies of a chain and the non-crystallographic symmetry (NCS) operators can be determined from the current model. NCS-related regions of the map will be averaged, weighted by how similar the density is. The area that gets averaged tends to grow as the phases improve. In this case, *Parrot* only determined symmetry operators between the three correctly-placed copies and not the copy in this region, but there is still a very big improvement to the density.

ModelCraft - Example

Cycle 4

1. Prune residues
2. Density modification
3. **Add dummy atoms**
4. Build protein
5. Build RNA & DNA
6. Combine protein & NA
7. Prune chains
8. Add waters

R-free: 32%



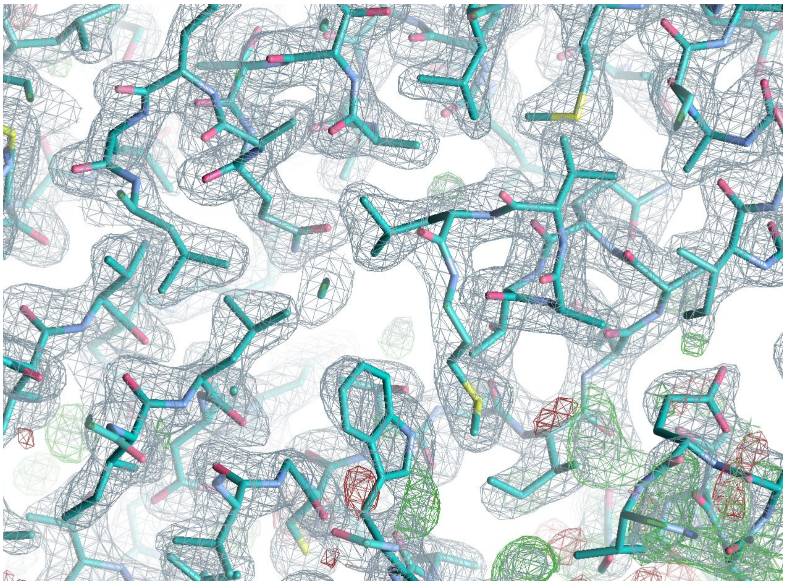
The third step adds dummy atoms into the *Parrot* map using *Coot*. The existing structure and the dummy atoms are then refined with *Refmac*. No attempt is made to interpret the dummy atoms as it is a difficult task and at lower resolutions it is unlikely that they correspond to atom positions. Instead, they only are used as a form of density modification. The resulting map is only accepted if R-free improves from the previous refinement of the protein model, which it did in this case from 50% to 32%.

ModelCraft - Example

Cycle 4

1. Prune residues
2. Density modification
3. Add dummy atoms
4. **Build protein**
5. Build RNA & DNA
6. Combine protein & NA
7. Prune chains
8. Add waters

R-free: 33%



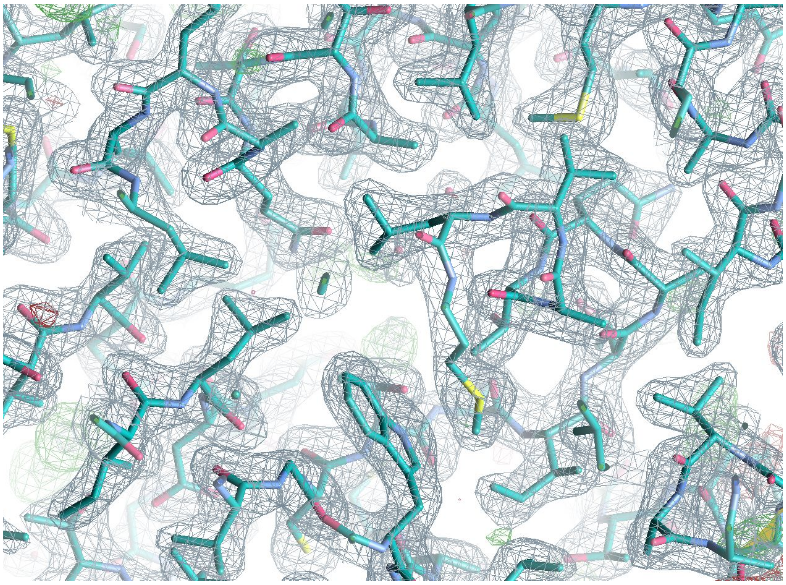
The dummy atoms are then discarded and *Buccaneer* is used to build a protein model. Because of the better phases, it has built most of this view correctly apart from some parts in the bottom right. R-free has increased slightly (from 32% to 33%) compared to the dummy atom model, but there will be much less overfitting as it is now a protein model with restraints applied.

ModelCraft - Example

Cycle 5

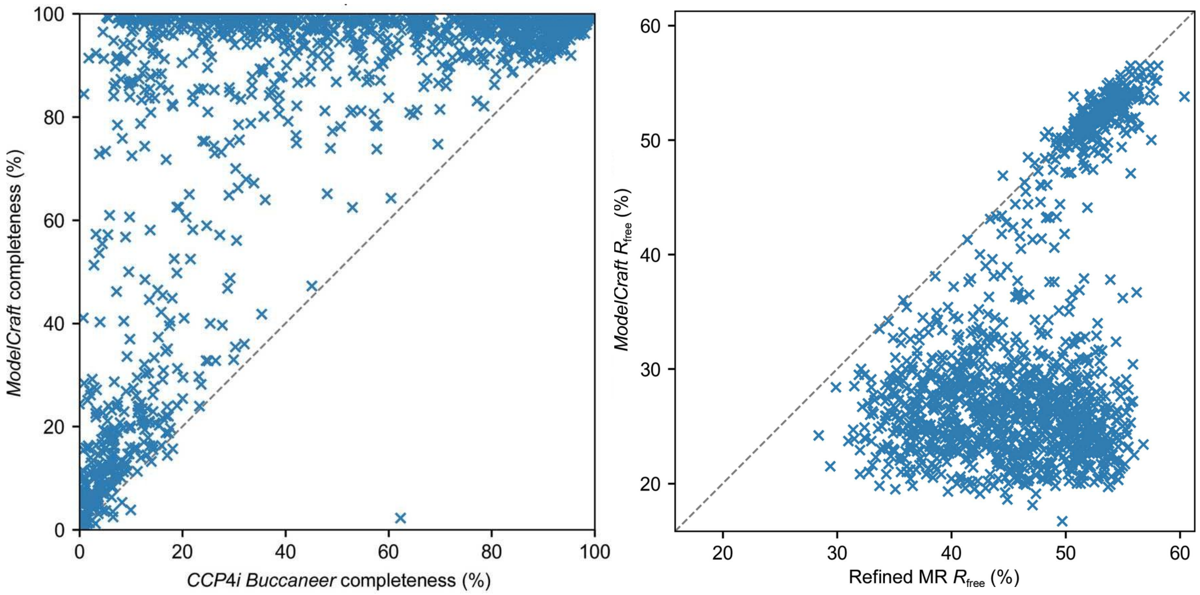
1. Prune residues
2. Density modification
3. Add dummy atoms
4. Build protein
5. Build RNA & DNA
6. Combine protein & NA
7. Prune chains
8. Add waters

R-free: 26%



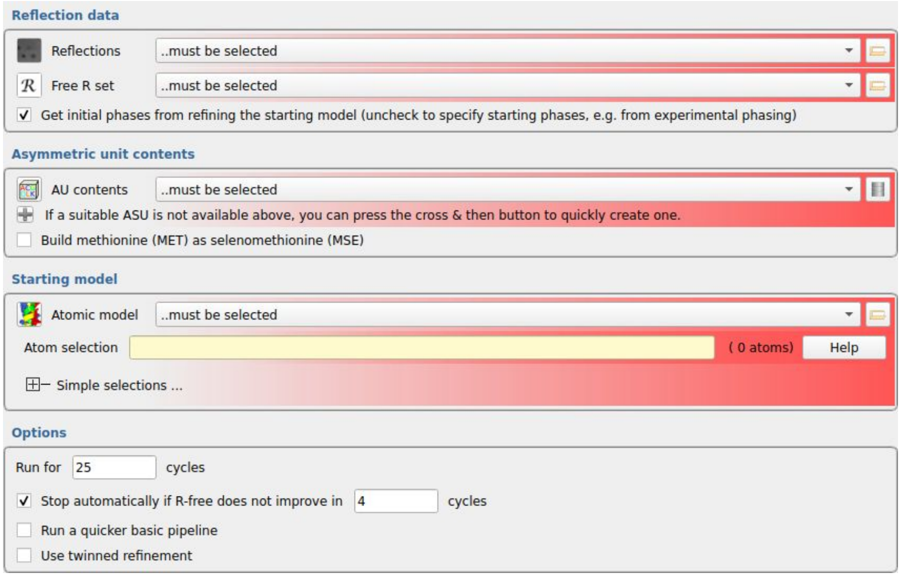
At the end of cycle 5 most of those issues have been fixed and R-free has improved to 26%. In this case, this is the best structure that ModelCraft produces. It still requires some interactive rebuilding and validation using in *Coot*. For example, there is a twisted peptide bond in the bottom right as indicated by the yellow triangle.

ModelCraft - Results



These charts show the performance of ModelCraft on a large molecular replacement test set with resolutions spread evenly between 1 and 3.5 Å and examples with both good and bad phases. On the left is a comparison of the completeness of protein models after the old *Buccaneer* pipeline from *CCP4i* and *ModelCraft*. There are a lot of cases at the top of the chart where ModelCraft produces a much more complete model. There is one case in the bottom of the chart where the old pipeline produces a much more complete model, but this issue was fixed in a later version of ModelCraft. On the right is a comparison of the free-R factor before and after ModelCraft. ModelCraft is sometimes able to build a good model even if R-free after MR is as high as 55%. If the MR model has an R-free below 50% then there is a good chance that ModelCraft will be able to improve it.

ModelCraft - CCP4i2



Reflection data

Reflections ..must be selected

Free R set ..must be selected

☑ Get initial phases from refining the starting model (uncheck to specify starting phases, e.g. from experimental phasing)

Asymmetric unit contents

AU contents ..must be selected

If a suitable ASU is not available above, you can press the cross & then button to quickly create one.

□ Build methionine (MET) as selenomethionine (MSE)

Starting model

Atomic model ..must be selected

Atom selection (0 atoms) Help

Simple selections ...

Options

Run for 25 cycles

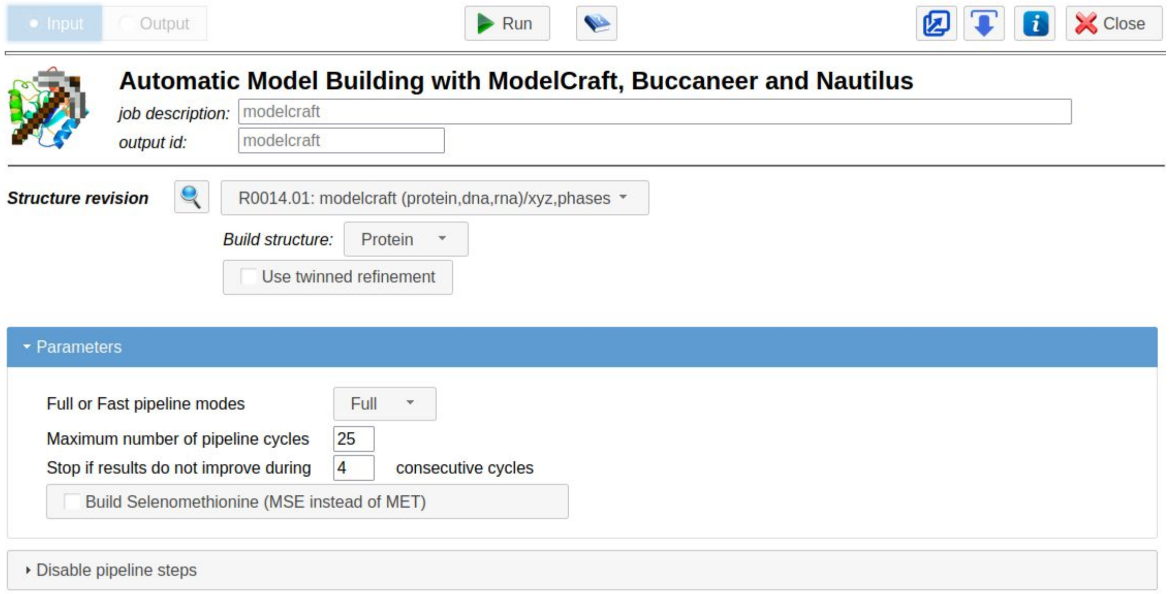
☑ Stop automatically if R-free does not improve in 4 cycles

□ Run a quicker basic pipeline

□ Use twinned refinement

This is the input page for ModelCraft in CCP4i2. You need to provide reflections, a free-R set and a description of the asymmetric unit contents. Initial phases can come from refining the starting model (e.g. after molecular replacement) or can be provided explicitly (e.g. after experimental phasing). In the latter case the starting model is optional. The interface provides a few options, such as how many cycles you want to run, whether you want to run a quicker basic pipeline that is more similar to the old *Buccaneer* pipeline and whether your crystal is twinned.

ModelCraft - CCP4 Cloud



Automatic Model Building with ModelCraft, Buccaneer and Nautilus

job description: modelcraft

output id: modelcraft

Structure revision R0014.01: modelcraft (protein,dna,rna)/xyz,phases

Build structure: Protein

□ Use twinned refinement

Parameters

Full or Fast pipeline modes: Full

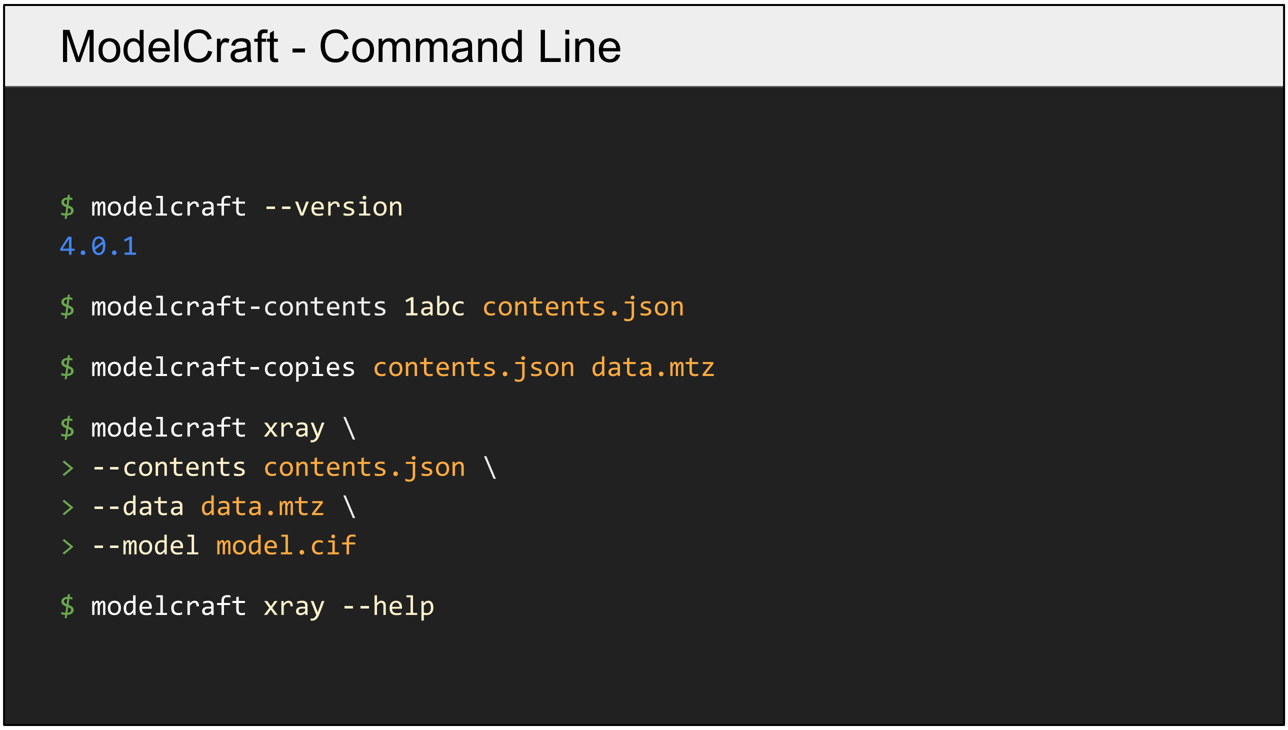
Maximum number of pipeline cycles: 25

Stop if results do not improve during 4 consecutive cycles

□ Build Selenomethionine (MSE instead of MET)

Disable pipeline steps

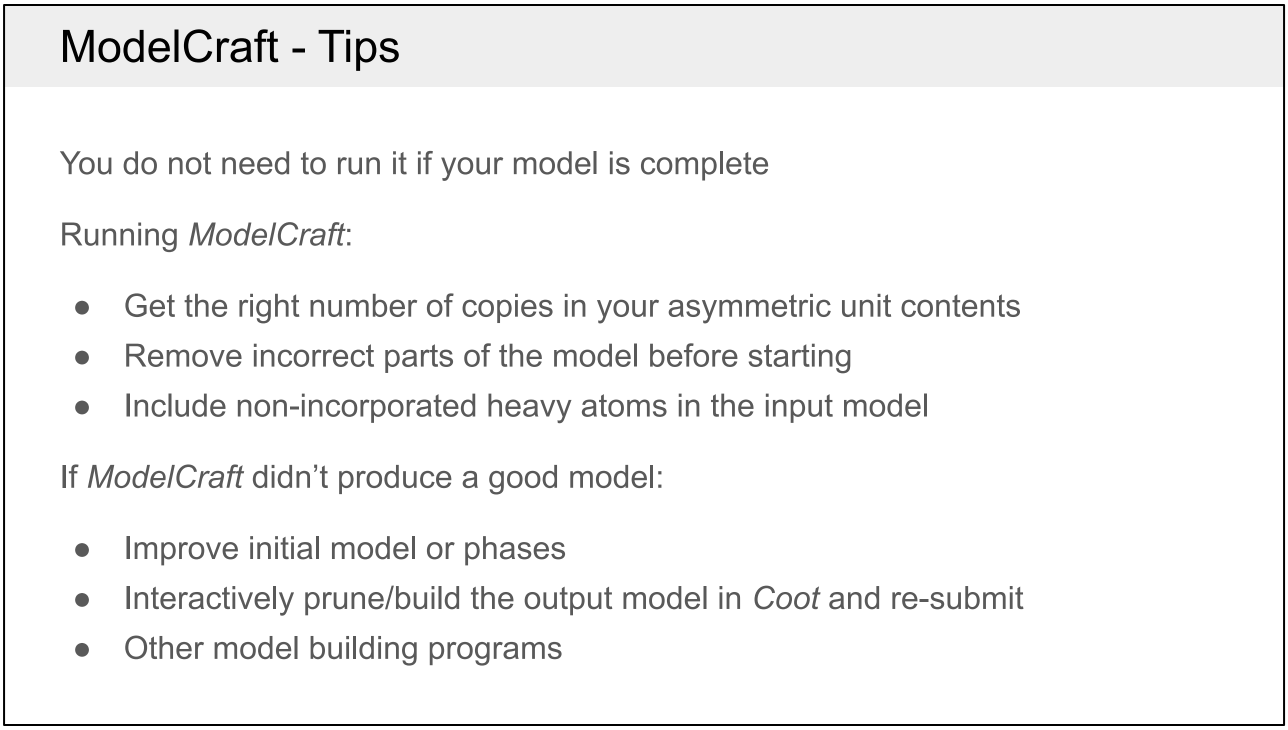
This is the interface to ModelCraft in CCP4 Cloud. There are fewer inputs to fill out here because CCP4 Cloud holds information together in a structure revision that comes from the task you are following on from. You can still choose the number of cycles, whether to run the full or the basic pipeline and whether to use twinned refinement.



ModelCraft - Command Line

```
$ modelcraft --version
4.0.1

$ modelcraft-contents 1abc contents.json

$ modelcraft-copies contents.json data.mtz

$ modelcraft xray \
> --contents contents.json \
> --data data.mtz \
> --model model.cif

$ modelcraft xray --help
```

ModelCraft can also be used on the command line. The JSON file holds a description of the asymmetric unit contents. There is a script to get the contents of a published PDB entry and a script to check the likely number of copies in your own data. If you want to run on the command line and not through *CCP4i2* or *CCP4 Cloud* there is more information at https://paulsbond.co.uk/modelcraft.



ModelCraft - Tips

You do not need to run it if your model is complete

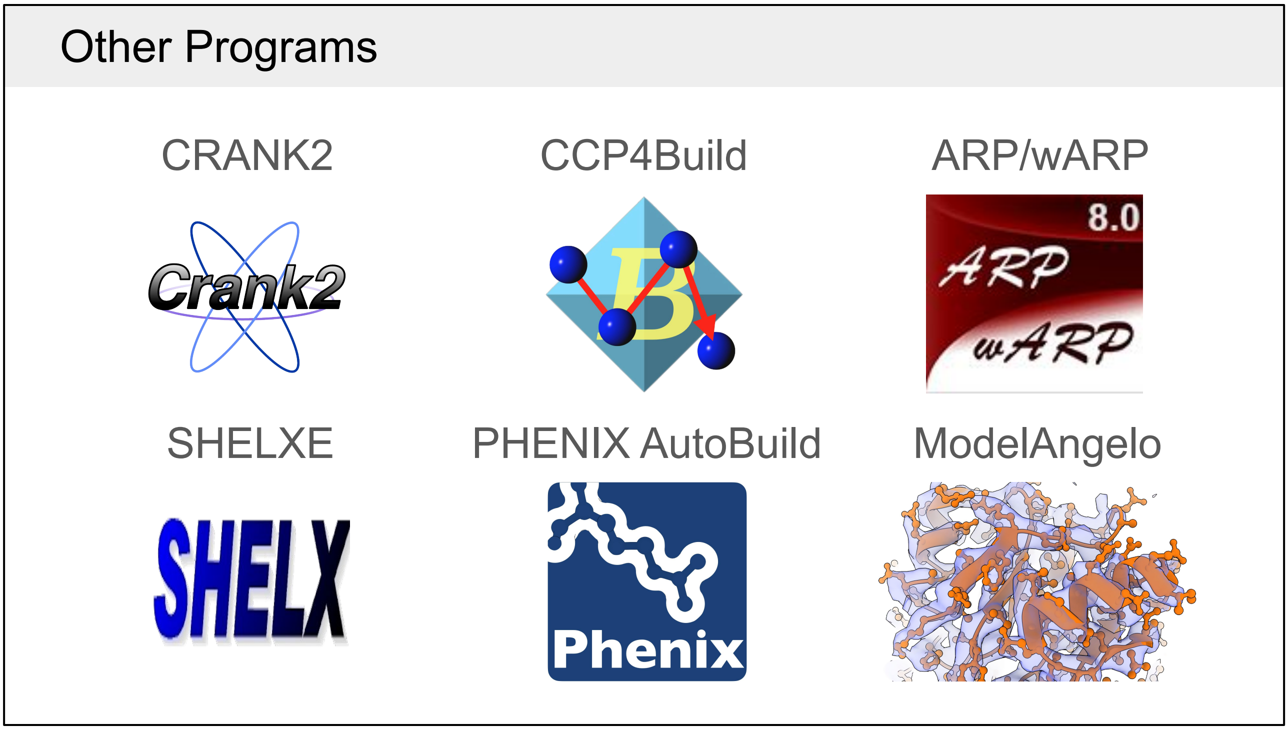
Running *ModelCraft*:

- Get the right number of copies in your asymmetric unit contents
- Remove incorrect parts of the model before starting
- Include non-incorporated heavy atoms in the input model

If *ModelCraft* didn’t produce a good model:

- Improve initial model or phases
- Interactively prune/build the output model in *Coot* and re-submit
- Other model building programs

When running *ModelCraft*, it is important to have the correct number of copies in order to calculate the solvent content for *Parrot*, otherwise density modification may flatten weak protein regions or amplify noise in the bulk solvent. If you know that part of your model is wrong, then it may be better to remove the incorrect residues first. It can also be useful to include non-incorporated heavy atoms (e.g. not Se from MSE) to improve the phases and to stop the model being built into that density. If *ModelCraft* hasn’t done a very good job at building there are a number of things to try. The starting phases might be improved through density modification or by trying a different molecular replacement model. You might also be able to do some interactive model building to improve the initial model. The partially-built output model can also be tidied up and used as input to another run. Other model-building programs could be used with the same input data or with the partially-built model.



Other Programs

CRANK2

CCP4Build

ARP/wARP

SHELXE

PHENIX AutoBuild

ModelAngelo

These are some of the other model-building programs that are available. *CRANK2* is full structure-solution pipeline for experimental phasing that uses Buccaneer for model building. *CCP4Build* is a model-building pipeline, similar to *ModelCraft*, available in *CCP4 Cloud* that includes *Parrot*, *Buccaneer*, *EDSTATS* and *Coot*. *ARP/wARP* and *SHELXE* are both good programs to try if you have high resolution data but poor phases. ARP/wARP has newer algorithms to help it work better at lower resolution and *SHELXE* has recently added an option to include side-chain building as well as main-chain tracing. *PHENIX AutoBuild* is a model building pipeline available in the PHENIX suite that is good for model completion. If you are building into a cryo-EM map and have access to a GPU, then ModelAngelo is able to build both protein and nucleotides with or without sequence information.

Acknowledgements

Supervisors

Kathryn Cowtan
Keith Wilson

Cowtan Group

Jordan Dialpuri
Soon Wen Hoh
Stuart McNicholas

Collaborators

Charles Ballard
Eleanor Dodson
Maria Fando
Ronan Keegan
Oleg Kovalevskiy
Eugene Krissinel
Andrey Lebedev
Rob Nicholls
Martin Noble
Marcin Wojdyr
Keitaro Yamashita

Program authors

Buccaneer
CCP4 Cloud
CCP4i2
CCP-EM
Coot
CTRUNCATE
EMDA
Gemmi
Nautilus
Parrot
Refmac
Servalcat
Sheetbend

UNIVERSITY of York

YSBL
York Structural Biology Laboratory

BBSRC
bioscience for the future

CCP4

CCP-EM