Automated phase improvement and model building

Kevin Cowtan
cowtan@ysbl.york.ac.uk
X-ray structure solution pipeline...

Data collection → Data processing → Experimental phasing → Molecular Replacement

Molecular Replacement → Density Modification → Model building → Refinement → Rebuilding Validation

Kevin Cowtan, cowtan@ysbl.york.ac.uk

2012
Density Modification

- Traditional density modification: e.g. 'dm', 'solomon', 'parrot', CNS
- Statistical density modification: e.g. 'resolve', 'pirate'
Density modification

Starting point:
• Structure factor amplitudes
• Phase estimates:
  – MR: Unimodel distribution
  – SAD: Biomodal distribution
Density modification

How do we represent phase probability distributions?

- Phase/figure of merit - Φ, FOM
  - (unimodel, MR only)
- Henrickson-Lattman coeffs – ABCD
  - (bimodal or unimodal, general)
Density modification

- Density modification is a problem in combining information:

![Diagram showing reciprocal and real space with phase probabilities and solvent envelope](image)
Density modification

1. Rudimentary calculation:

\[ |F|, \varphi \]
\[ \varphi = \varphi_{\text{mod}} \]
\[ \rho(x) \]
\[ \rho_{\text{mod}}(x) \]

Reciprocal space

Real space

FFT

FFT\(^{-1}\)

 Modify \(\rho\)
Density modification

2. Phase weighting:

\[ |F|, \phi \]

\[ \phi = f(\phi_{\text{exp}}, \phi_{\text{mod}}) \]

\[ |F_{\text{mod}}|, \phi_{\text{mod}} \]

\[ \rho_{\text{mod}}(x) \]

\[ \rho(x) \]

|F|, \phi \rightarrow FFT \rightarrow \rho(x) \rightarrow \rho_{\text{mod}}(x) \rightarrow \text{Real space} \rightarrow \text{Reciprocal space} \rightarrow |F_{\text{mod}}|, \phi_{\text{mod}} \rightarrow \text{ FFT}^{-1} \]

Modify \( \rho \)
Density modification

4. Bias reduction (gamma-correction):

\[
|F|, P(\phi) \rightarrow |F_{\text{best}}|, \phi_{\text{best}} \rightarrow \rho(x) \rightarrow \rho_{\text{mod}}(x) \rightarrow \rho_{\gamma}(x)
\]

- \(P(\phi) = P_{\text{exp}}(\phi) \times P_{\text{mod}}(\phi)\)
- \(P_{\text{mod}}(\phi)\) is the likelihood
- \(\rho_{\text{mod}}(x)\) is modified density
- \(\rho_{\gamma}(x)\) is \(\gamma\)-corrected density

J.P. Abrahams
Density modification

5. Maximum Likelihood H-L:

\[ |F|, P(\varphi) \rightarrow |F_{\text{best}}|, \varphi_{\text{best}} \rightarrow \rho(x) \rightarrow \text{Modify} \rho \rightarrow \rho_{\text{mod}}(x) \rightarrow \gamma\text{-correct} \rightarrow \rho_x(x) \]

\[ \text{centroid} \rightarrow \text{MLHL} \rightarrow |F_{\text{mod}}|, \varphi_{\text{mod}} \rightarrow \text{FFT}^{-1} \]
Density modification

6. Statistical density modification:

\[ |F|, P(\varphi) \]  \rightarrow  \[ |F_{\text{best}}|, \varphi_{\text{best}} \]  \rightarrow  \[ \rho(x) \]  \rightarrow  \[ P(\rho(x)) \]

\[ P(\varphi) = P_{\text{exp}}(\varphi) \times P_{\text{mod}}(\varphi) \]

Infer

Transform distribution

RESOLVE, PIRATE

Kevin Cowtan, cowtan@ysbl.york.ac.uk
Density modification

Traditional density modification techniques:
- Solvent flattening
- Histogram matching
- Non-crystallographic symmetry (NCS) averaging

\[ |F|, \phi \rightarrow \text{FFT} \rightarrow \rho(x) \rightarrow \text{Modify } \rho \rightarrow \rho_{\text{mod}}(x) \]

\[ |F_{\text{mod}}|, \phi_{\text{mod}} \leftarrow \text{FFT}^{-1} \]
Solvent flattening
Histogram matching

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.
Non-crystallographic symmetry

- 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

- How do you know if you have NCS?
  - Cell content analysis – how many monomers in ASU?
  - Self-rotation function.
  - Difference Pattersons (pseudo-translation only).
- How do you determine the NCS?
  - From heavy atoms.
  - From initial model building.
  - From molecular replacement.
  - From density MR (hard).
- Mask determined automatically.
Combining phase probabilities

Once we have an estimate for the error in $\phi_{\text{mod}}$, we can construct a probability distribution $P_{\text{mod}}(\phi)$. The next cycle can be started with

$$P_{\text{new}}(\phi) = P_{\text{exp}}(\phi)P_{\text{mod}}(\phi)$$

**Problem:** $P_{\text{exp}}(\phi)$ and $P_{\text{mod}}(\phi)$ are not independent. The result is bias, increasing with cycle.
Density modification in Parrot

Builds on existing ideas:

• DM:
  Solvent flattening
  Histogram matching
  NCS averaging
  Perturbation gamma

• Solomon:
  Gamma correction
  Local variance solvent mask
  Weighted averaging mask
Density modification in Parrot

New developments:

- MLHL phase combination (as used in refinement: refmac, phenix.refine)
- Anisotropy correction
- Problem-specific density histograms (rather than a standard library)
- Pairwise-weighted NCS averaging...
Estimating phase probabilities

Traditional approach: Rice likelihood function

1. Estimate the accuracy of the modified F/phase
2. Turn this into a phase probability distribution
3. Combine with the experimental phase probability

The estimate for the accuracy of the modified F/phase come from the agreement between the modified F and the observed F. **Source of bias.**
Estimating phase probabilities

Problem:

Error estimation does not take into account experimental phase information.

The experimental data tells us that the probable error is different in the two cases.

Using the additional information from the phases improves the error model and reduces bias.
Estimating phase probabilities

Solution:
MLHL-type likelihood target function.

Perform the error estimation and phase combination in a single step, using a likelihood function which incorporates the experimental phase information as a prior. This is the same MLHL-type like likelihood refinement target used in modern refinement software such as refmac or phenix.refine.
Recent Developments:

Pairwise-weighted NCS averaging:
- Average each pair of NCS related molecules separately with its own mask.
- Generalisation and automation of multi-domain averaging.
Parrot
Parrot: Rice vs MLHL

Map correlations

Comparing old and new likelihood functions.
Parrot: simple vs NCS averaged

Map correlations

Comparing with and without NCS averaging.
Model Building

Model building software:

• Proteins:
  - Buccaneer
  - ARP/wARP
  - Phenix autobuild

• Nucleic acids:
  - Nautilus/Coot
  - ARP/wARP
  - Phenix autobuild
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 $C_\alpha$ => LLK target.

- Use mean/variance.

4A sphere about Ca also used by 'CAPRA' Ioeger 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

Sequencing: Look for C-beta environment

ALA  CYS  HIS  MET  THR  ...  x20

(4.0Å sphere about Cα)

(5.5Å sphere about Cβ)
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 2.9A 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
Buccaneer: Results

Model completeness not very dependent on resolution:
Buccaneer: Results

Model completeness dependent on initial phases:

![Graph showing model completeness vs. initial map correlation. The x-axis represents initial map correlation ranging from 0 to 1, and the y-axis represents Buccaneer model completeness ranging from 0 to 1.]
Buccaneer
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: 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!
Nucleic Acid Building

Nautilus:

- A new tool for nucleic acid model building
- Automated (CCP4i) or interactive (Coot)
- Starting from:
  - Experimental phasing
  - Molecular replacement
  - Protein complexes
Nautilus

The task:

- To build continuous nucleic acid chains into electron density.
- To assign sequence to those chains.
- To allow addition of nucleotide chains to non-nucleotide structures.
Nautilus

'Fingerprint' detection:

• Identify high and low density features consistent with the presence of nucleic acid features.

• Very fast.

• Related to 'Essens' (Kleywegt and Jones), but with looks at both ridges and troughs.
Nautilus

Types of fingerprint:

- Sugar
- Phosphate
- Base type (A/C/G/U)
Nautilus

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. 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 $1 \times \text{sugar} + 2 \times \text{phosphate fingerprints}$
- Phosphate:
  - Build a pair of nucleic acids using the best matching equivalent from the database, scored by $1 \times \text{phosphate} + 2 \times \text{sugar fingerprints}$
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 seconds

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 chains:
• 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
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 of the 50 closest matches being of each type (+ an error term).

Google: k-NN (k-Nearest Neighbour)
Nautilus

Results:

- Good results on synthetic noisy data at 3.5Å and user reports on real data at 3.8Å.
  - Need more data

- Like 'buccaneer', phases are more important than resolution.

- Failed on a quadruplex structure with good phases.
  - Try a different database?
Acknowledgements

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