

Refinement of Macromolecular structures using REFMAC5

Garib N Murshudov
York Structural Laboratory
Chemistry Department
University of York

Contents

- 1) Introduction
- 2) Considerations for refinement
- 3) TWIN
- 4) TLS
- 5) Dictionary and alternative conformations
- 6) Conclusions

Available refinement programs

- SHELXL
- CNS
- REFMAC5
- TNT
- BUSTER/TNT
- Phenix.refine
- RESTRANT
- MOPRO

Considerations in refinement

- Function to optimise (link between data and model)
 - Should use experimental data
 - Should be able to handle chemical (e.g bonds) and other (e.g. NCS, structural) information
- Parameters
 - Depends on the stage of analysis
 - Depends on amount and quality of the experimental data
- Methods to optimise
 - Depends on stage of analysis: simulated annealing, conjugate gradient, second order (normal matrix, information matrix, second derivatives)
 - Some methods can give error estimate as a by-product. E.g second order.

Two components of target function

Crystallographic target functions have two components: one of them describes the fit of the model parameters into the experimental data and the second describes chemical integrity (restraints).

Currently used restraints are: bond lengths, angles, chirals, planes, ncs if available, some torsion angles

Various form of functions

- SAD function uses observed F^+ and F^- directly without any preprocessing by a phasing program (It is not available in the current version but will be available soon)
- MLHL - explicit use of phases with Hendrickson Lattman coefficients
- Rice - Maximum likelihood refinement without phase information

Shortcomings of using ABCD directly

- Dependent on where you obtained your Hendrickson-Lattman coefficients
- Assumes that your prior phase information is independent from your model phases!

Differences between SAD and RICE in wARP*+ Refmac

	Resol. (Å)	Anom. atoms	Experiment	Residues RICE/SAD/FINAL
MutS	3.0	46 Se	SAD (peak)	493/1093/1600
subtilisin	1.77	3 Ca, S	SAD	6/259/275
thioesterase	2.5	8 Se	SAD (infl)	300/542/572
gere	2.75	12 Se	MAD(p/i)	43/110/444
cyanase	2.41	40 Se	MAD (p/i)	71/669/1560
thioesterase I	1.81	20 Br	SAD(peak)	35/431/462

*10 wARP cycles.

These results are from Raj Pannu and Pavol Skubak from Leiden

Twinning

merohedral and pseudo-merohedral twinning

Crystal symmetry:

Constrain:

Lattice symmetry *: P622
(rotations only)

Possible twinning:

P3

-

P222

merohedral

P2

$\beta = 90^\circ$

pseudo-merohedral

P2

-

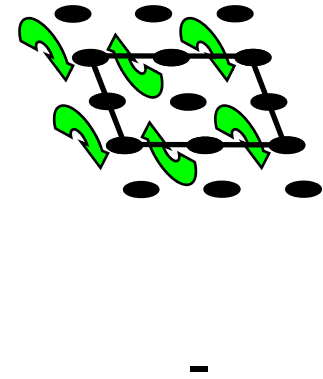
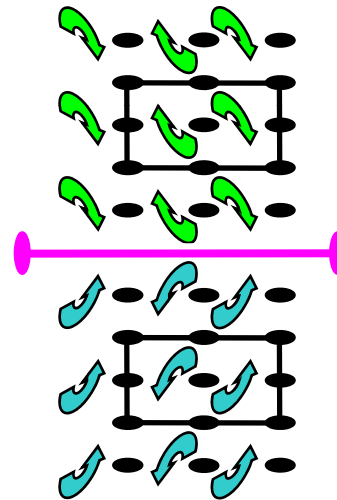
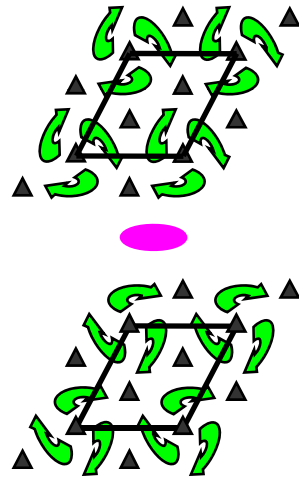
P2

-

Domain 1

Twinning operator

Domain 2

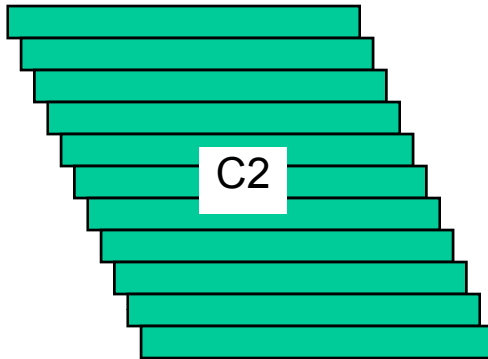


Crystal lattice is invariant with respect to twinning operator.

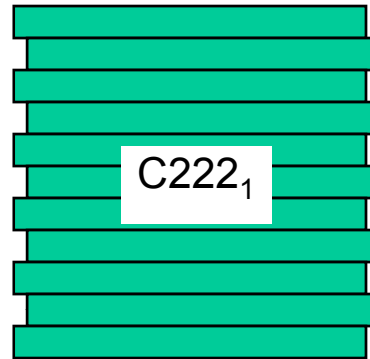
The crystal is NOT invariant with respect to twinning operator.

More than three layers, but less than the whole crystal.

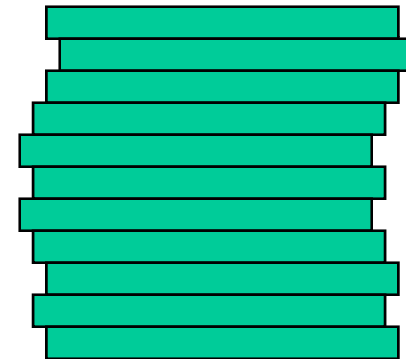
C2 single crystal



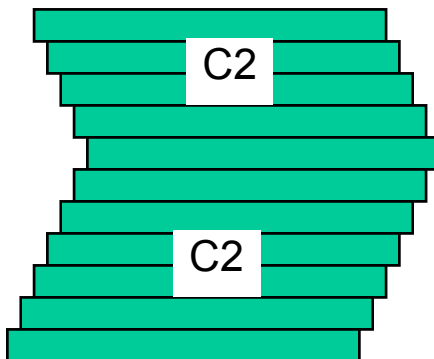
C222₁ single crystal



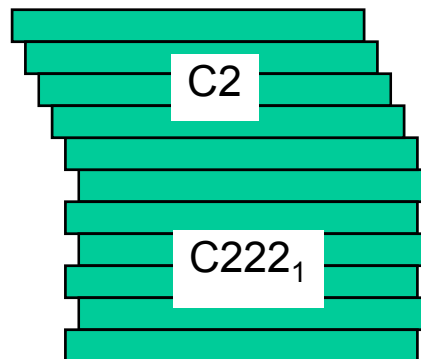
Disordered OD-structure



OD-twin

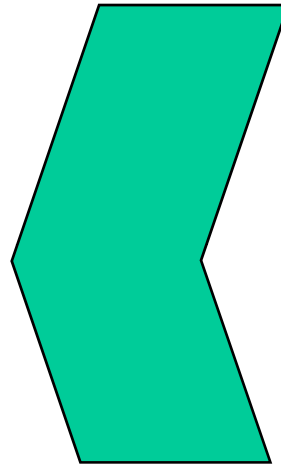


Allotwin

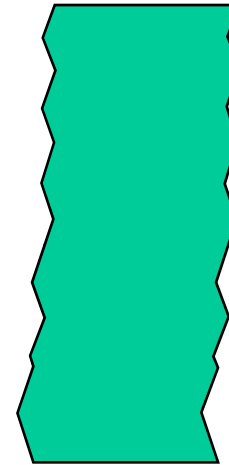


The whole crystal: twin or polysynthetic twin?

twin



polysynthetic
twin



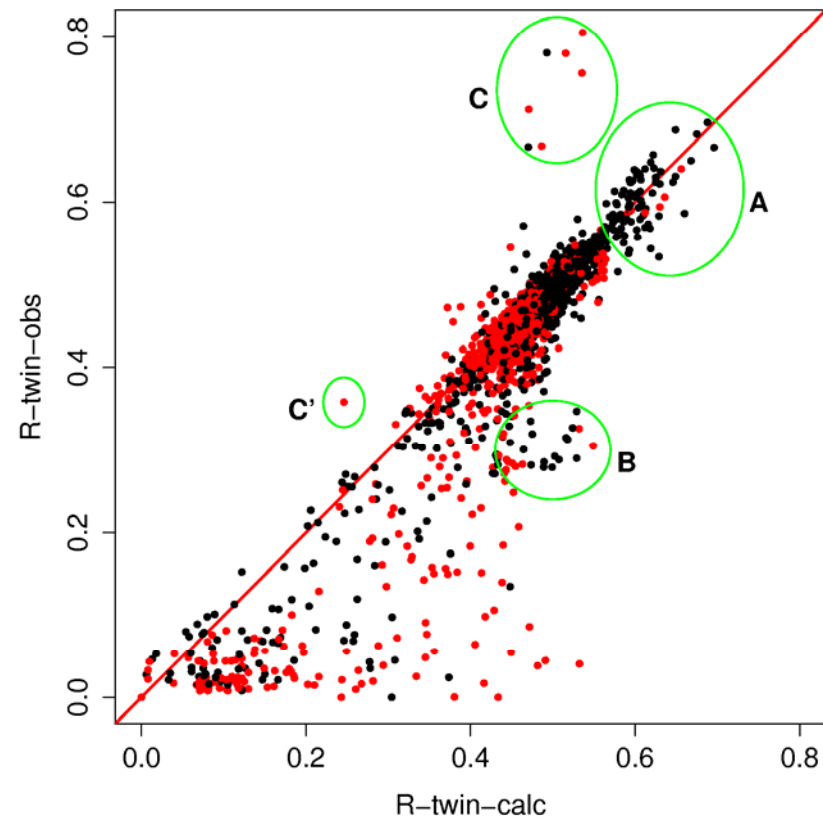
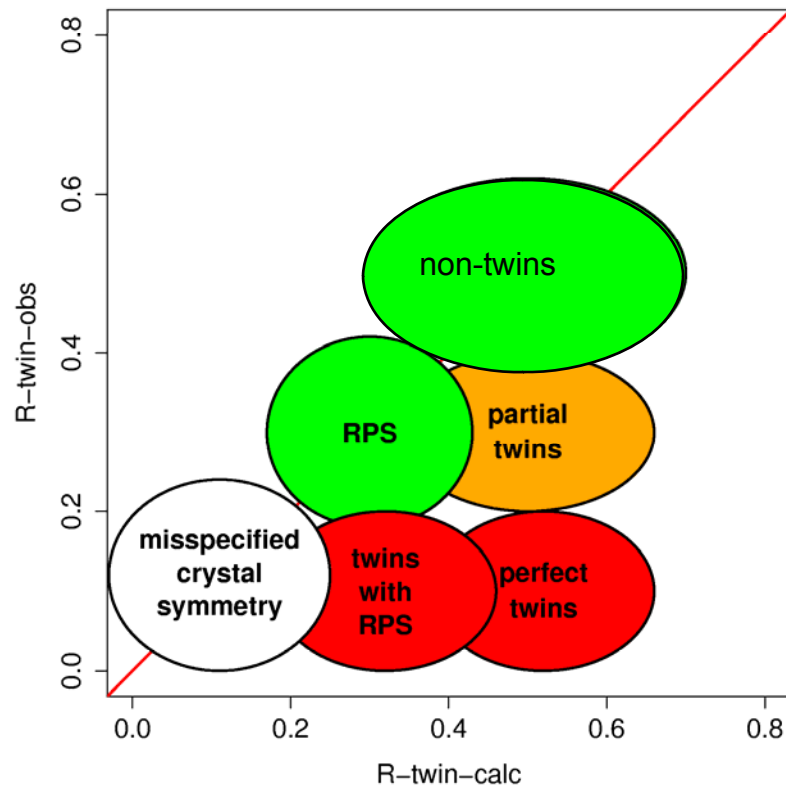
A single crystal can be
cut out of the twin:

yes

no

The shape of the crystal suggested that we dealt with polysynthetic OD-twin

RvR-plot



$$R_{\text{twin}} = \frac{\sum_h |I(h) - I(S_{\text{twin}} h)|}{2 \sum_h I(h)}$$

$$R_{\text{twin}}^{\text{obs}} :: I \rightarrow I^{\text{obs}}$$

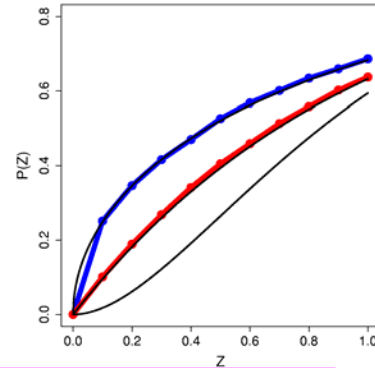
$$R_{\text{twin}}^{\text{calc}} :: I \rightarrow I^{\text{calc}}$$

- A: translational NCS
- B: mislabeling $F \square I$
- C, C': mislabeling $I \square F$

Red: (potential) merohedral twins

Black: (potential) pseudomerohedral twins

Perfect twinning test

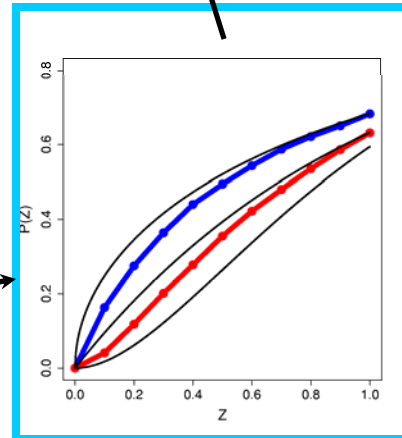
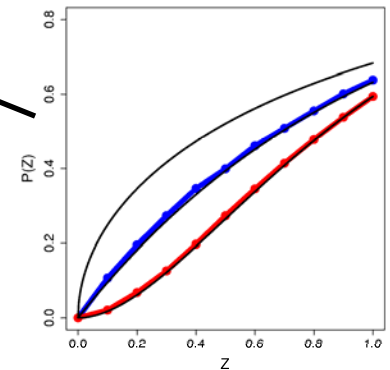
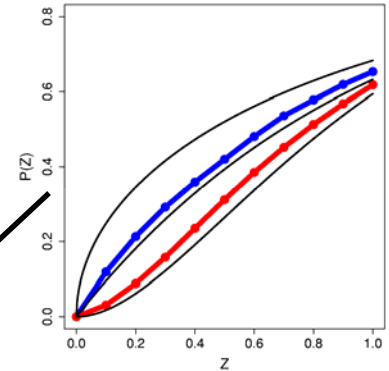
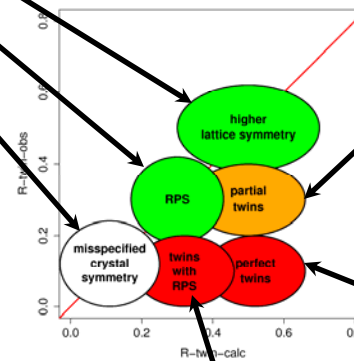


This test is implemented in TRUNCATE

Untwinned + pseudosymmetry:
test shows no twinning

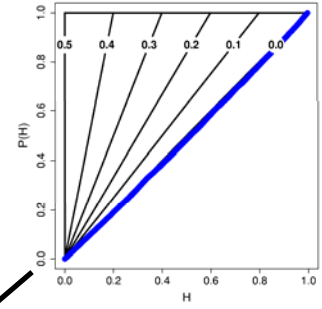
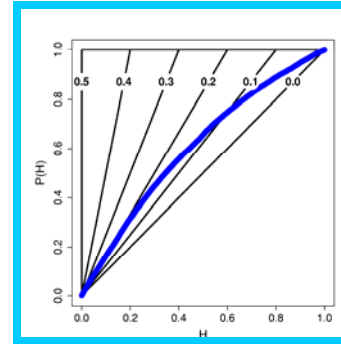
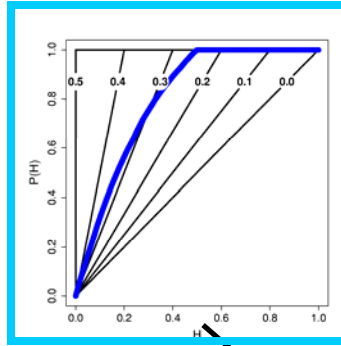
Twin + pseudosymmetry:
Test shows only partial
Twinning.

(decrease of contrast)



Partial twinning test

Non-linearity

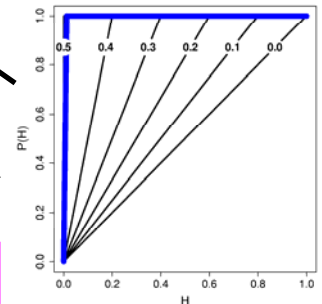
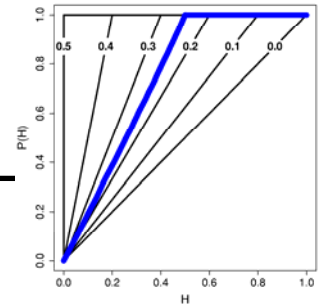
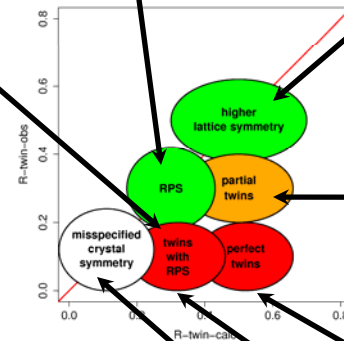


No pseudosymmetry: linear for both twins and non-twins.
Tilt shows twinning fraction.

The test is useless for perfect twins
(cannot distinguish it from higher symmetry)

Pseudosymmetry causes non-linearity.

Experimental errors + this non-linearity
makes the test hardly interpretable in
some cases.



This test is implemented in
SFCHECK

Twin refinement

Twin refinement in the new version of refmac is automatic.

- Twin operators are identified
- “Rmerge” for each operator is calculated and operators for which $R_{\text{merge}} < 0.50$ are kept: Twin plus crystal symmetry operators should form a group
- Twin fractions are refined and only domains with fraction above certain threshold are kept (default threshold is 0.05): Twin plus symmetry operators should form a group

Intensities can be used

Twin refinement is not possible together with SAD yet

Maximum likelihood refinement is used

Twinning can be used even if there is no twin indication

Likelihood

$$P(I_o; F) = \int_F P(I_o, F; F_c) dF = \int_F P(I_o; F) P(F; F_c) dF$$

$$P(I_o; F) = N_o e^{-\frac{(\sum |I_{o,j}| - \sum \alpha_{o,j} |I_{o,j}|)^2}{2\sigma_o^2}}$$

$$P(F; F_c) = N_c \prod e^{-\frac{|F - DF_c|^2}{\Sigma}}$$

The dimension of integration is in general twice the number of twin related domains. Since the phases do not contribute to the first part of the integrant the second part becomes Rice distribution.

The integration is carried out using Laplace approximation.

In principle these equations are general enough to account for: non-merohedral twinning (including allawtwin), unmerged data. A little bit modification should allow simultaneous twin and SAD/MAD phasing.

Electron density: likelihood based

Equation for map calculation:

$$\begin{aligned}FWT &= 2 \langle F \rangle - DF_c \\DELFWT &= \langle F \rangle - DF_c \\ \langle F \rangle &= \int_F F P(I_o, F; F_c) dF / \int_F P(I_o, F; F_c) dF\end{aligned}$$

It seems to be working reasonable well. For unbiased map it is necessary to integrate over errors in all parameters.

I hope it will be available in the next version of refmac

Test cases: Preliminary results

PDB ID	R in pdb	R after refmac**	R after twin	Twin fractions	Comments
1rxf	11.9	21.5	12.0	0.69 0.31	Refined with twin
1ap9*	25.8	31.7	27.6	0.65 0.35	Data between 5-2.35 were used
1gwy	21.6	22.1	18.4	0.74 0.26	Refined without twin
1jrg	21.1	23.5	16.7	0.73 0.27	Refined without twin

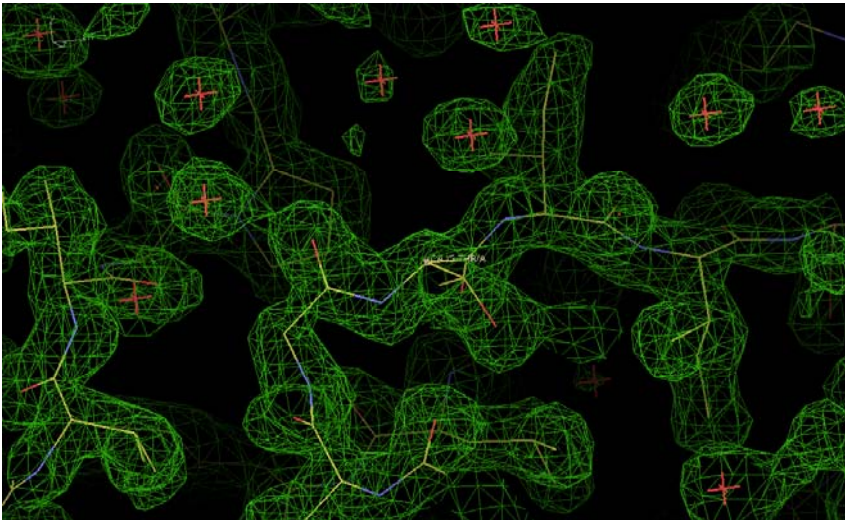
*Data could have been detwinned (bad idea)

**Zero cycle of “refinement” in REFMAC was used

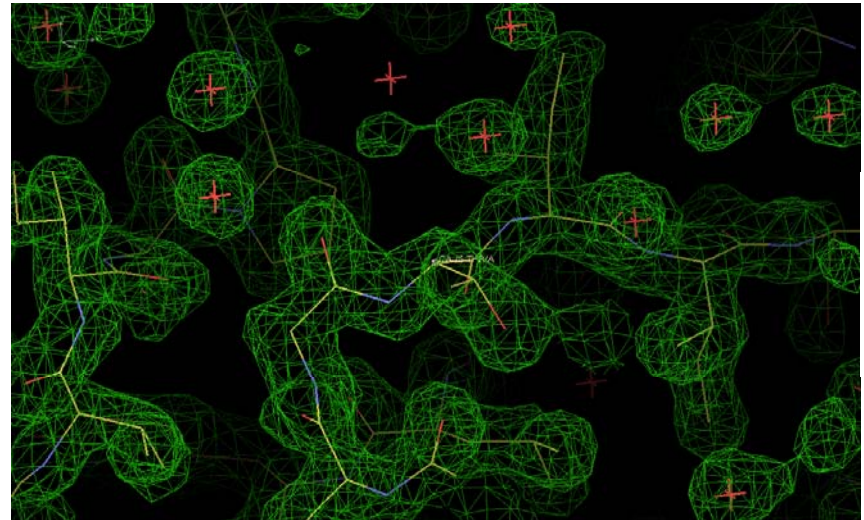
Electron density: 1gwy

What we will see

“refmac” map



“Twin” map

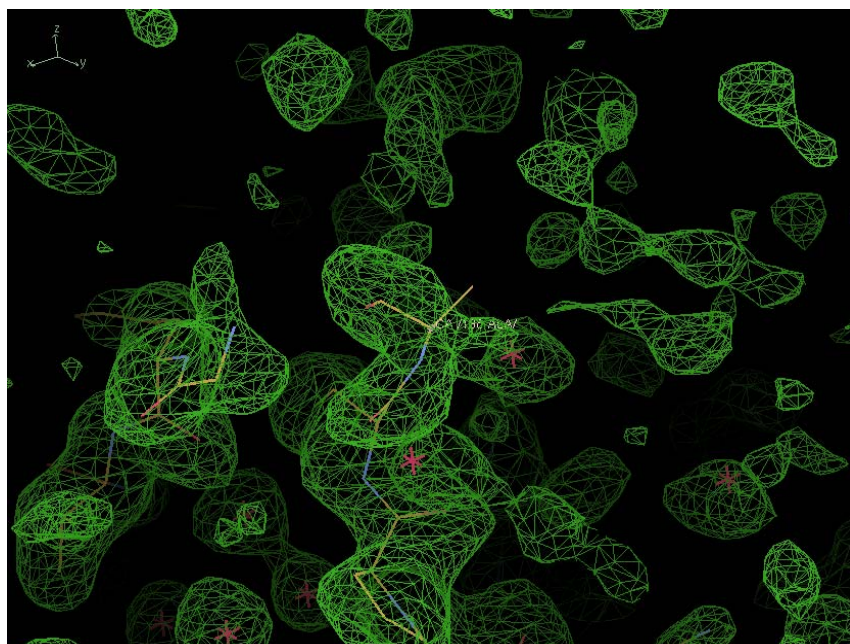


differences between electron densities are marginal. That is usual case especially when twin and NCS are almost parallel

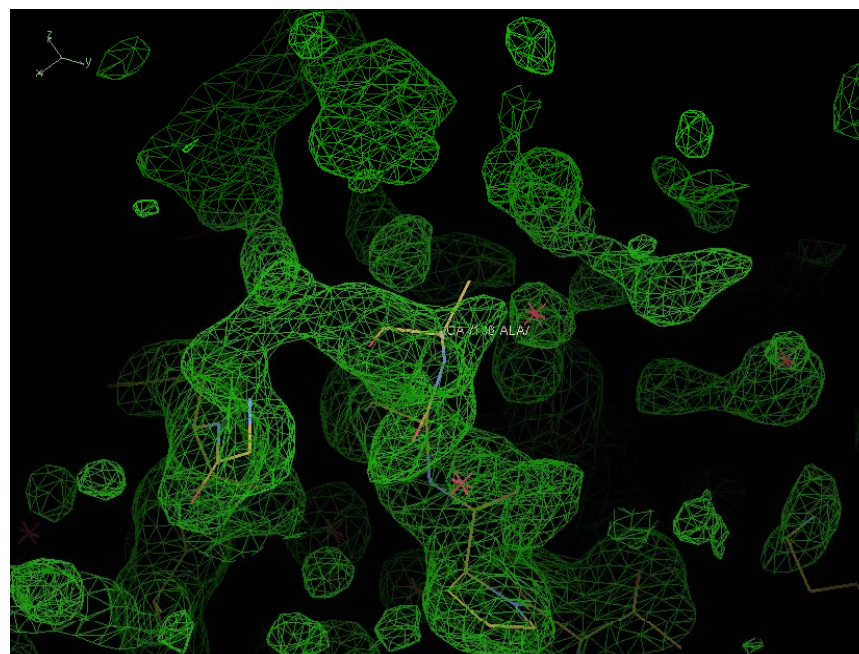
Electron density: 1rxf

We will see occasionally this

“refmac” map



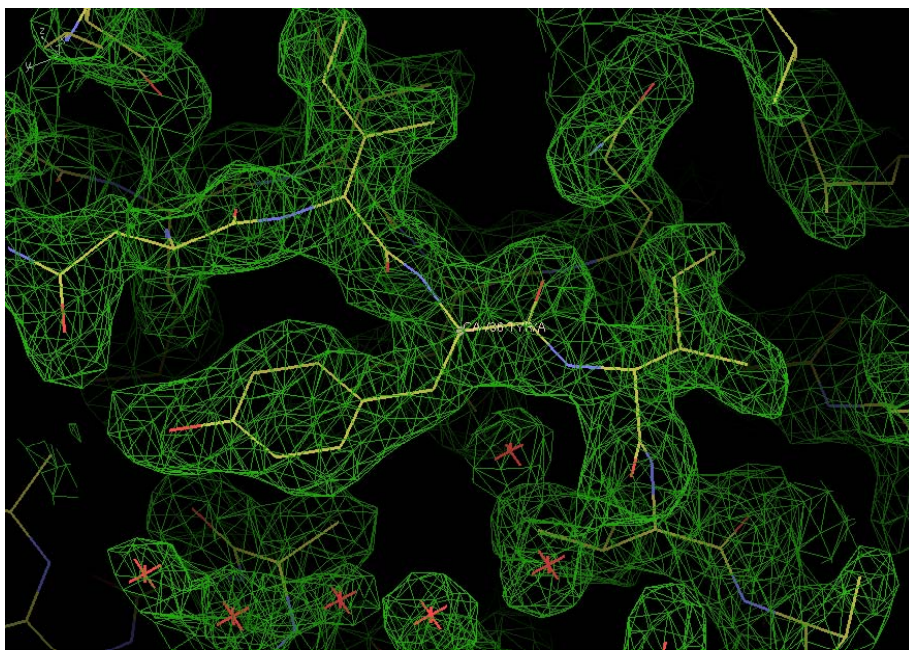
“twin” map



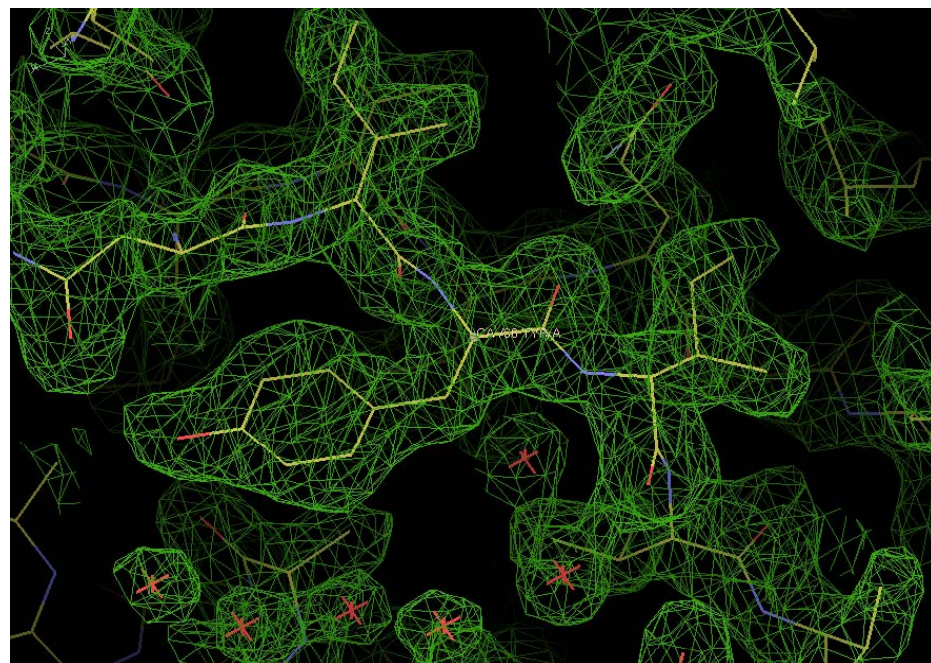
Electron density: 1jrg

More usual and boring case

“refmac” map



“twin” map



Effect of twin on electron density: Noise level. Very, very approximate

$$|F_t| e^{i\phi} \approx |F_R| e^{i\phi} + \alpha(|F_w| - |F_R|) e^{i\phi}$$

F_t - twinned structure factor

F_R - structure factor from “correct” crystal

F_w - structure factor from “wrong” crystal

The first term is correct electron density the second term corresponds to noise.

When twin and NCS are parallel then the second term is even smaller.

Conclusion

- Twinning occurs more often than we would like
- Twinning and rotational NCS occur very often together
- Twin refinement improves statistics and occasionally electron density
- PDB is a fantastic resource for testing and development

Map calculation

- After refinement programs usually give coefficients for two type of maps: 1) $2F_o - F_c$ type maps. They try to represent the content of the crystal. 2) $F_o - F_c$ type of maps. They try to represent difference between contents of the crystal and current atomic model. Both these maps should be inspected and model should be corrected if necessary.

- Refmac gives coefficients:

$2 m F_o - D F_c$ – to represent contents of the crystal

$m F_o - D F_c$ - to represent differences

m is the figure of merit (reliability) of the phase of the current reflection and D is related with model error. m depends on each reflection and D depends on resolution

If phase information is available then map coefficients correspond to the combined phases.

Parameters

Usual parameters (if programs allow it)

- 1) Positions x,y,z
- 2) B values – isotropic or anisotropic
- 3) Occupancy

Derived parameters

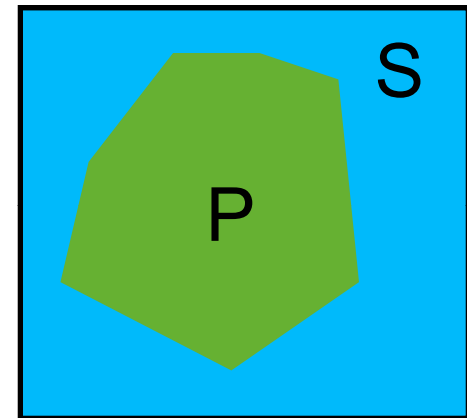
- 4) Rigid body positional
 - After molecular replacement
 - Isomorphous crystal (liganded, unliganded, different data)
- 5) Rigid body of B values – TLS
 - Useful at the medium and final stages
 - At low resolution when full anisotropy is impossible
- 6) Torsion angles

Bulk solvent

Method 1: Babinet's bulk solvent correction

At low resolution electron density is flat. Only difference between solvent and protein regions is that solvent has lower density than protein. If we would increase solvent just enough to make its density equal to that of protein then we would have flat density (constant). Fourier transformation of constant is zero (apart from F000). So contribution from solvent can be calculated using that of protein. And it means that total structure factor can be calculated using contribution from protein only

$$\begin{array}{ll} \rho_s + \rho_p = \rho_T & \Longleftrightarrow F_s + F_p = F_T \\ \rho_s + k\rho_p = c & \Longleftrightarrow F_s + kF_p = 0 \\ F_s = -kF_p & \implies F_T = F_p - kF_p = (1-k)F_p \end{array}$$



k is usually taken as $k_b \exp(-B_b s^2)$. k_b must be less than 1. k_b and B_b are adjustable parameters

Bulk solvent

Method 2: Mask based bulk solvent correction

Total structure factor is the sum of protein contribution and solvent contribution. Solvent region is flat. Protein contribution is calculated as usual. The region occupied by protein atoms is masked out. The remaining part of the cell is filled with constant values and corresponding structure factors are calculated. Finally total structure factor is calculated using

$$F_T = F_p + k_s F_s$$

k_s is adjustable parameter.



Mask based bulk solvent is a standard in all refinement programs. In reftmac it is default.

Overall parameters: Scaling

There are several options for scaling:

- 1) Babinet's bulk solvent assumes that at low resolution solvent and protein contributors are very similar and only difference is overall density and B value. It has the form: $k_b = 1 - k_s \exp(-B_s s^2/4)$
- 2) Mask bulk solvent: Part of the asymmetric unit not occupied by atoms are assigned constant value and Fourier transformation from this part is calculated. Then this contribution is added with scale value to "protein" structure factors. Total structure factor has a form: $F_{\text{tot}} = F_p + s_s \exp(-B_s s^2/4) F_s$.
- 3) The final total structure factor that is scaled has a form:

$$S_{\text{aniso}} S_{\text{protein}} k_b F_{\text{tot}}$$

TLS

TLS groups

Rigid groups should be defined as TLS groups. As starting point they could be: subunits or domains.

If you use script then default rigid groups are subunits or segments if defined.

In ccp4i you should define rigid groups (in the next version default will be subunits).

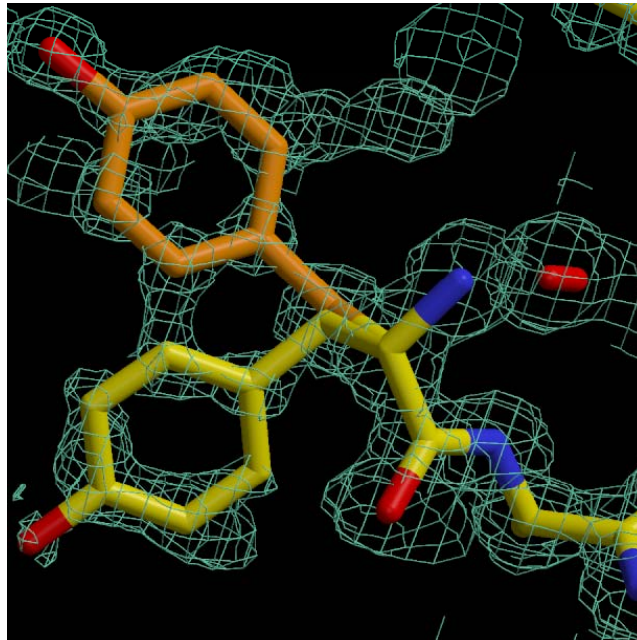
Rigid group could be defined using TLSMD webserver:

<http://skuld.bmsc.washington.edu/~tlsmd/>

Alternative conformations and links

Alternative conformations

Example from 0.88Å catalase structure: Two conformations of Tyrosine. Ring is clearly in two conformation. To refine it properly CB also needs to be split. It helps adding hydrogen atom on CB and improves restraints in anisotropic U values

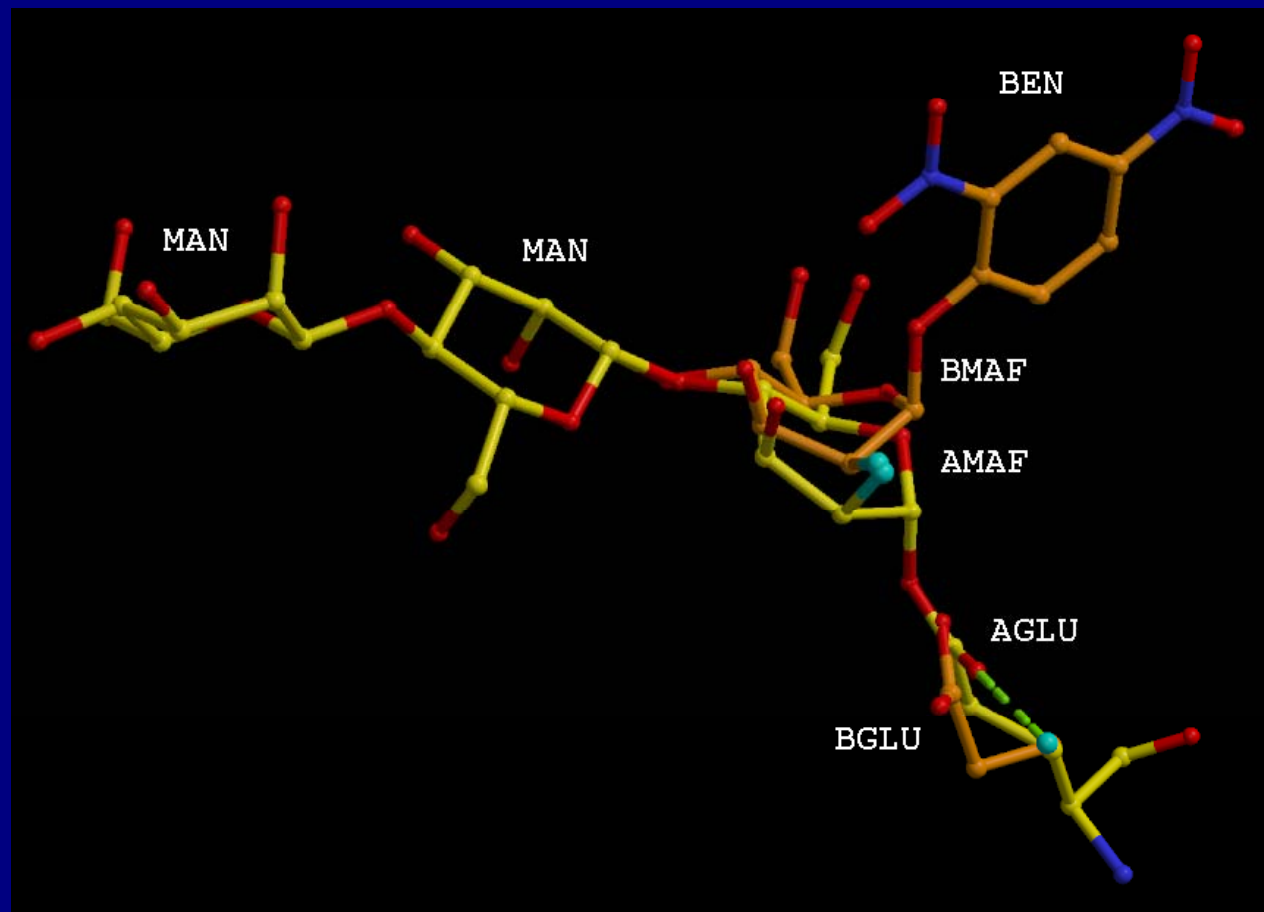


Alternative conformation: Example in pdb file

ATOM N	977	N	GLU	A	67	-11.870	9.060	4.949	1.00	12.89
ATOM C	978	CA	GLU	A	67	-12.166	10.353	4.354	1.00	14.00
ATOM C	980	CB	AGLU	A	67	-13.562	10.341	3.738	0.50	14.81
ATOM C	981	CB	BGLU	A	67	-13.526	10.285	3.654	0.50	14.35
ATOM C	986	CG	AGLU	A	67	-13.701	9.400	2.573	0.50	16.32
ATOM C	987	CG	BGLU	A	67	-13.876	11.476	2.777	0.50	14.00
ATOM C	992	CD	AGLU	A	67	-15.128	9.179	2.134	0.50	17.17
ATOM C	993	CD	BGLU	A	67	-15.237	11.332	2.110	0.50	15.68
ATOM (994	OE1	AGLU	A	67	-15.742	10.153	1.644	0.50	20.31

Link between residues in double conformation

Fluro-modified sugar MAF is in two conformation. One of them is bound to GLU and another one is bound to ligand BEN



Alternative conformation of links: how to handle

Description

Description of link(s) should be added to the library. When residues make link then each component is usually modified. Description of Link should contain it also

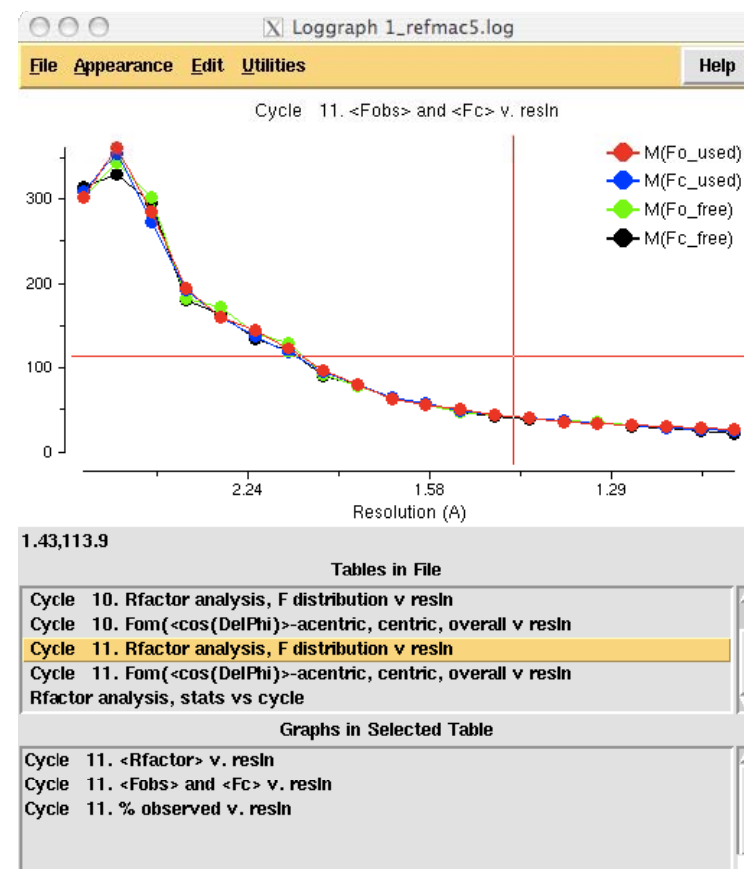
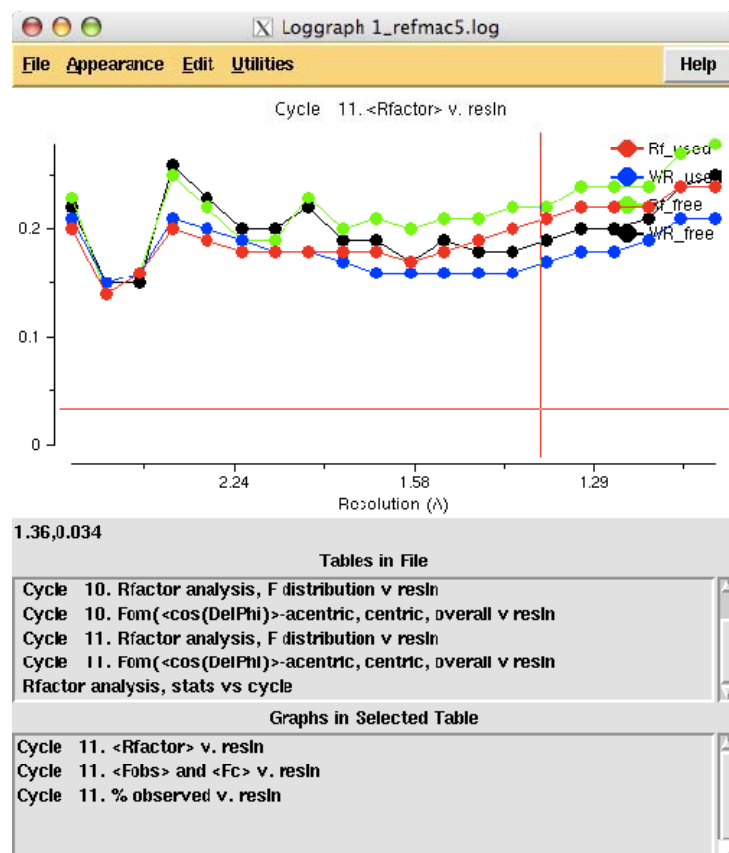
PDB

LINK	C6	BBEN	B	1		O1	BMAF	S	2	BEN-MAF
LINK	OE2	AGLU	A	320		C1	AMAF	S	2	GLU-MAF

Things to look at

- R factor/Rfree: They should go down during refinement
- Geometric parameters: rms bond and other. They should be reasonable. For example rms bond should be around 0.02
- Map and coordinates using coot
- Logggraph outputs. That is available on the cpp4i interface

Behaviour of R/Rfree, average Fobs vs resolution should be reasonable. If there is a bump or it has an irregular behaviour then either something is wrong with your data or refinement.



What and when

- Rigid body: At early stages - after molecular replacement or when refining against data from isomorphous crystals
- TLS - at medium and end stages of refinement at resolutions up to 1.7-1.6Å (roughly)
- Anisotropic - At higher resolution towards the end of refinement
- Adding hydrogens - Higher than 2Å but they could be added always
- Phased refinement - at early and medium stages of refinement
- SAD - at all stages(?)
- Twin - always (?)
- Ligands - as soon as you see them
- What else?

Conclusions

- If phases are available they should be used at least at the early and medium stages of refinement
- Unless there is very good reason not to all resolution should be used in refinement
- TLS describes overall motion and works well in practice
- Ligand and link description should be considered very carefully
- Although there is information about motion of molecule in the TLS parameters they should be used with care
- Twin seems to be more common than we would like