

Simple REFMAC tutorials

Prerequisites: To use this tutorial you need to have ccp4. For “jelly” body, automatic and local ncs restraints, occupancy refinement you need to have the latest version of ccp4 - 6.2 or similar that has refmac 5.6. In addition to use restraints to homologous structures, secondary structure restraints you need to have ProSmart.

Instructions how to use new options are in:

www.ytbl.york.ac.uk/refmac/Presentations/Refmac_Erice_workshop.ppt
slides 45-50.

Data: Download the tutorial file from the website:

www.ytbl.york.ac.uk/refmac/Tutorials/ACA_tutorial.tar.gz

Create a subdirectory where you usually work and copy the tutorial file to this directory:

```
mkdir refmac
cd refmac
mv wherever_tutorial_file/refmac_tutorial.tar.gz .
```

Then untar and go to the refmac_tutorial directory.

```
tar xzvf refmac_tutorial.tar.gz
cd refmac_tutorial
```

There are several files to play around. Playing with these files can give you some idea about refinement in general and refmac5 in particular.

Start ccp4i. For each case you should create a project in the ccp4i using “Directories&ProjectDir” on the top right of the interface. It will help you to trace back what you have done and find files you need. There are five subdirectories. Each subdirectory is for one tutorial. These directories are: **rnase** – a simple refinement, **twin** – twin refinement, **lowres** – low resolution refinement and automatic NCS restraints, **1n5b** - tls refinement, , **sad** – sad refinement,.

In this tutorial we will use **rnase** – simple refinement, **twin** – twin refinement and **lowres** – low resolution refinement.

I) Simple refinement.

On the top of the ccp4i select “Refinement”. Click “Run Refmac5”. It will bring up refmac5 interface. Define the input mtz and coordinate files. They are rnase115_unique1.mtz and rnase.pdb. Press Run/Run now on the left bottom side.

While it is running you can have a look at the log file. When refmac finishes this job you should use coot (or another program) to view coordinates, maps and make corrections if you need to.

```
coot -pdb rnase_refmac1.pdb -data rnase115_refmac1.mtz
```

Current version of the program uses automatic weighting. For many cases it works sufficiently well. Sometimes (especially at high resolution) default weighting may give a relaxed geometry. In this case you can play with the weighting of X-ray and geometry terms. If you want to change weights then click “Refinement Parameters”, under this unclick “Use automatic weighting” and add an appropriate number into the field “Use weighting term”. For low resolution you may want to use very small values - 0.01 or even smaller. For higher resolution this number may need to be as high as 10. You may need to run refinement with different values to get it right. If after refinement run rms bond distances are more than 0.02 then you may want to reduce weighting, if rms bond value is less than 0.01 then you may need to increase. Smaller weighting value means tighter geometry. At every cycle REFMAC5 prints out currently used weight. This value could be found in the log file. For example:

```
Weight matrix      4.4438701
Actual weight      10.000000      is applied to the X-ray term
```

In this case rmsd for bond lengths after automatic refinement is 0.033. There might be reasons why rmsd is high (e.g. some amino acids or side chains may not be in electron density). It is a good idea to start model rebuilding after refinement and when you are sure that the model is correct then you can play with weights to adjust rmsd values.

If you want to “improve” rmsd of bond lengths then you can rerun refinement using half of the weight matrix value suggested by refmac. In this case it would be around 2.2. Weight matrix is under “Refinement parameters”. If you unclick “automatic weighting” then a new field – weight matrix will appear.

If rmsd bond lengths are too small (i.e. less than 0.01) then you may need to use larger weight matrix value.

After changing weight matrix to 2.2 rmsd of bond lengths after five cycles of refinement becomes more acceptable:

	Initial	Final
R factor	0.2783	0.1876
R free	0.2668	0.2052
Rms BondLength	0.0284	0.0201
Rms BondAngle	4.5704	1.6554
Rms ChirVolume	0.1696	0.1063

When automatic weighting was used these values were:

	Initial	Final
R factor	0.2783	0.1831
R free	0.2668	0.2030
Rms BondLength	0.0284	0.0327
Rms BondAngle	4.5704	2.3083
Rms ChirVolume	0.1696	0.1645

Note that initial R and Rfree are very close (in fact Rfree is smaller than Rfactor) in the beginning of refinement. The reason is that before this refinement session free R set was not assigned. In principle it is a good idea to select free R set in the beginning and keep them throughout refinement and model building.

II) Twin

First run usual refinement without twin option. Final statistics for non-twin refinement are:

	Initial	Final
R factor	0.3103	0.2779
R free	0.3184	0.3496
Rms BondLength	0.0088	0.0129
Rms BondAngle	1.4825	1.5648
Rms ChirVolume	0.1077	0.1034

Now run `refmac5` again as usual. This time click “Twin refinement” button on the interface. It will activate twin refinement. You can choose “Intensity based” or “Amplitude based” twin refinement. Only difference between them is that when amplitudes are given the program internally converts them to intensities. Theoretically using intensities would be better. But in practice it may be different. When you use “amplitude based” then Rfactors are usually smaller than when you use “Intensity based” refinement. It does not mean that derived models for amplitude based refinement are better.

Identification of twin operators and twin refinement are carried in several stages. In the first stage all potential twin operators are identified. They do not have to be actual twin operators. They are (in very simple terms) just elements of lattice symmetry that are not crystal symmetry.

```
The number of potential twin domains =      4
Twin domains ((pseudo)merohedral only):
Potential twin domain 1 with operator:  H,  K,  L, metric score 0.000
Potential twin domain 2 with operator: -K, -H, -L, metric score 0.000
Potential twin domain 3 with operator: -H, -K,  L, metric score 0.000
Potential twin domain 4 with operator:  K,  H, -L, metric score 0.000
```

There are four potential twin operators. It is understandable since the space group is $P3_2$ (point group symmetry is 3) and the lattice point group symmetry is 622. So there are potentially 4 twin operators (for simple relationship between point groups see Appendix 1).

In the next stage for each symmetry element the program calculates “Rmerge” – agreement between potentially twin related reflections.

**** Filtering out small twin domains, step 1

Twin operators with Rmerge > 0.44000000 will be removed

Symmetry operator= -K, -H, -L: R_merge =0.248: twin is probable
Symmetry operator= -H, -K, L: R_merge =0.237: twin is probable
Symmetry operator= K, H, -L: R_merge =0.027: twin or higher symmetry

When Rmerge is very small then program suggests that this operator could be element of higher space group. Perhaps crystal symmetry is higher than the data were processed with.

In the next stage the program calculates twin fractions for each domain that survived the first filtering stage. In this case all operators survive.

Twin domains with fraction < 7.00000003E-02 are removed

**** Twin operators with estimated twin fractions ****

Twin operator: H, K, L: Fraction = 0.391; Equivalent operators: K, -H-K, L; -H-K, H, L
Twin operator: -K, -H, -L: Fraction = 0.112; Equivalent operators: -H, H+K, -L; H+K, -K, -L
Twin operator: -H, -K, L: Fraction = 0.108; Equivalent operators: -K, H+K, L; H+K, -H, L
Twin operator: K, H, -L: Fraction = 0.390; Equivalent operators: H, -H-K, -L; -H-K, K, -L

Program prints out twin operators used and all their symmetry equivalents. It also prints out twin fraction for each domain. According to this table there are four domains. NB: When twin fractions are small then they may not be reliable. Especially at the early stages of refinement. To get some feeling about behaviour of Rfactors in the presence of twinning see: www.ytbl.york.ac.uk/refmac/papers/Rfactor.pdf

Refmac removes twin domains with small twin fractions. Default value is 0.07. In this case all values are larger than 0.07 and all "twin" operators survive.

Final statistics with twin refinement are:

	Initial	Final
R factor	0.1912	0.1566
R free	0.1796	0.2047
Rms BondLength	0.0088	0.0235
Rms BondAngle	1.4825	2.1812
Rms ChirVolume	0.1077	0.1336

Although Rfactors are substantially smaller with twin refinement than without twin refinement it does not mean that model also is substantially better. In most parts electron density will be very similar to non-twin case. However for weak parts of the electron density the signal may be more visible in twin refinement case. Again care should be exercised: Usually twin maps after twin refinement are more biased towards errors in the model. The reason is that when crystal is twinned then amount of information in the data about the model is decreased. It is an intrinsic property of data from twinned crystals.

Note 1: In some versions of ccp4i there was a bug and twin refinement was not activated. To check if it is activated check the log file. If it has TWIN keyword then everything is fine. If twin does not run automatically then use “Include keyword file”. Click on “Browse” and select file keyw.dat. This file contains a single keyword - TWIN. Adding this should activate twin refinement.

Note 2: In this case Rfactor and Rfree are diverging. It is because 1) This data set is from pdb and there was no twin flag there 2) Twin relates two (or more) sets of reflections. Usual free R selection does not take this fact into account. Refmac5 puts twin related reflections to the same set (i.e. either all related reflections belong to “free” or to “working” set). In general free reflection selection should be done before starting refinement accounting for potential twin operators. One way of achieving this is to select free reflections using higher group and expand them. It will ensure that throughout refinement and during all other treatment of the data all related reflections will belong to the same class.

Look at the maps and enjoy model building using coot.

III) Low resolution refinement

If you have older version of ccp4 then you can follow the instructions described in the presentation:

www.yybl.york.ac.uk/refmac/Presentations/Refmac_Erice_workshop.ppt
slides 45-50

Some descriptions of the options for low resolution refinement can be found in this paper:

Murshudov GN, Skubak P, Lebedev AA, Pannu NS, Steiner RA, Nicholls RA, Winn MD, Long F, Vagin AA “REFMAC5 for the Refinement of Macromolecular Crystal Structures” *Acta Cryst.* , **D67**, 355-367

This part of the tutorial may take very long time. Refinement with restraints to homologous structures takes longer time to run and convergence is slow.

In the new version of ccp4 there are options for jelly body, automatic NCS and map sharpening.

- a) Jelly body is under “Refinement Parameters”. You need to click “Use jelly-body refinement with sigma”. Change sigma to 0.01 or 0.02. This value defines “jelliness”. Smaller value means tighter restraints
- b) Automatic NCS restraints are under “Setup Non-Crystallographic Symmetry”. You need to click “use automatically generated local NCS restraints”. You can also use global NCS
- c) Map sharpening is under “Monitoring and Output Options”. You need to click “Perform map sharpening with B value 20.0”. B value should a little bit smaller than Wilson’s B value. The program can automatically identify sharpening parameters.

Here are some comparison of refinement at 4A resolution using various options:

Simple refinement:

	Initial	Final
R factor	0.3602	0.2218
R free	0.3563	0.3166
Rms BondLength	0.0098	0.0123
Rms BondAngle	1.6602	1.8073
Rms ChirVolume	0.2237	0.1477

NCS local

	Initial	Final
R factor	0.3602	0.2232
R free	0.3563	0.3124
Rms BondLength	0.0098	0.0127
Rms BondAngle	1.6602	1.8895
Rms ChirVolume	0.2237	0.1566

NCS local has little effect on refinement at this resolution.

Jelly body (all $d_{\text{model}} < 4.5$ are restrained, $\sigma = 0.01$):

	Initial	Final
R factor	0.3602	0.2533
R free	0.3563	0.2961
Rms BondLength	0.0098	0.0135
Rms BondAngle	1.6602	1.8347
Rms ChirVolume	0.2237	0.2155

Jelly body and NCS local:

	Initial	Final
R factor	0.3602	0.2535
R free	0.3563	0.2955
Rms BondLength	0.0098	0.0137
Rms BondAngle	1.6602	1.8408
Rms ChirVolume	0.2237	0.2157

Again improvement using NCS local restraints over “jelly” body refinement is marginal. However use of NCS may give more similar molecule that may make more sense biologically. Local NCS allows conformational changes to happen.

Using restraints to known homologous structures may improve behaviour of refinement further. With the current version of reftmac you need to use several runs to get restraints to homologous structures right. Restraints are generated by the program ProSmart (at the moment available from Rob Nicholls, ran105@york.ac.uk). The program will be available from the next version of ccp4).

Prosmart generates a list of restraints. These restraints can be used in refinement. However you need to play with several parameters to get optimal results (we still do not have optimal parameters, they will be available in future versions of reftmac/prosmart). These parameters include

Maximum interatomic distance to be used in refinement. Relevant keyword is:

external dmax <value>

Sigma used in restraints. Although prosmart generates sigmas in many cases changing them gives better results. Relevant keyword is

external weight scale <value>

If this keyword is used then all sigmas in the external restraints list will be divided by this value.

Filter out large deviations. Relevant keyword is:

external cut <value>

If this keyword is given then if the difference between model and reference structure interatomic distances is more than $value * sigma$ then this distance is ignored. It allows conformational changes to occur.

If we use sigma equal to 0.07, maximum distance 4.0 then refinement gives:

	Initial	Final
R factor	0.3602	0.2557
R free	0.3563	0.2905
Rms BondLength	0.0098	0.0117
Rms BondAngle	1.6602	1.8325
Rms ChirVolume	0.2237	0.1982

Free R is reduced and R factor is higher. It can be claimed that overfitting is reduced. Another tool that helps low resolution refinement is map sharpening. Refmac uses regularised map sharpening, i.e. it attempts to increase signal while reducing amplification of noise.

IV) TLS refinement

This example of TLS refinement in Refmac5 uses 1n5b, a molecular chaperone. The asymmetric unit contains 2 dimers, chains A + B and chains C + D. Data is in P212121 to 2.0A

To run Refmac5, we need to specify what TLS groups we wish to use. CCP4 wiki site has a page about different ways of creating TLS groups.

To create TLS group using ccp4i:

1. Go to ccp4i -> Refinement -> Model preparation -> Create/Edit TLS file.
3. Give "TLS out" a name such as 1n5b_in.tls
4. Define 4 groups (use "Add another TLS group" to create additional entries):
 - Chain A, residues 3 to 130
 - Chain B, residues 3 to 130
 - Chain C, residues 3 to 129
 - Chain D, residues 3 to 130
5. Click "Run now" to create the file.

We are now ready to do TLS refinement

6. Select the task Run Refmac5
7. Select mode "TLS & restrained refinement" from the protocol folder (top left button after the "Job title").

8. Set the files:

```
HKLIN 1n5b.mtz
XYZIN 1n5b.pdb
TLSIN 1n5b_in.tls (from above)
```

Note 1: If you do not give input tls file then each chain will be taken as a tls group.

Note 2: Waters close to tls groups will be added to those groups (it is a new feature and available from ccp4 6.0.99c onwards)

Note 2: If you do not give tls file then refmac will choose each chain as single tls group. So you can try just step 7 (Select mode "TLS & restrained refinement")

9. In the folder "TLS parameters" select "Set initial Bfactors"
10. For a quick run, use 5 TLS cycles and 5 restrained cycles

When Refmac5 has finished, check the logfile for the usual things: Rfree, geometry, warning messages. TLS section shows the raw TLS parameters at each cycle. These don't mean much on their own, but you can see if the TLS refinement is converging. Now we need to analyse the TLS parameters. It is available from ccp4 6.1.1 onwards.

11. In the "Model Completion & Analysis" section select "Analyse TLS parameters" task.

12. Set the files:

```
TLSIN is TLSOUT from previous Refmac job
XYZIN is XYZOUT from previous Refmac job
```

13. Click "Output file containing the axis .." and give the AXES file a name 1n5b_tls.vector

14. In the Axes folder, select mmCIF format

15. In Other Options, select "Analyse derived atomic"

Check the log file for "NON-POSITIVE DEFINITE WARNINGS" (I think there will be none!)

Output coordinates contain ANISOU records derived from the TLS plus residual B factors. Use coot to view them. Axis can be viewed using ccp4mg (I think?)

V) Dictionary

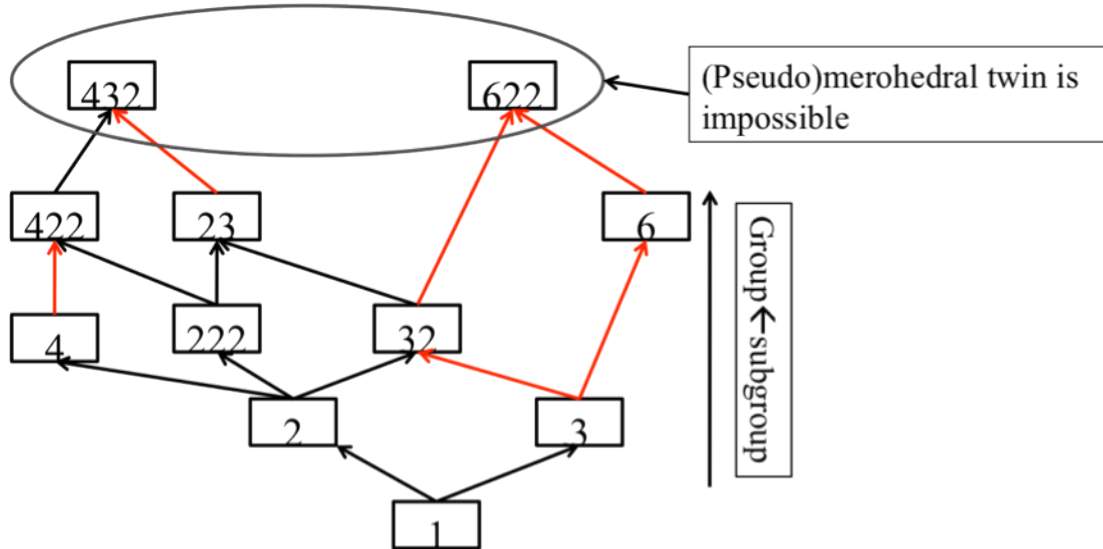
If you are interested you can find tutorials for the dictionary of ligand and link generation in:

<http://www.ytbl.york.ac.uk/mxstat/JLigand/index.html>

To run jligand you need to have java 1.5 or later versions. There are three tutorials on this site: 1) How to create ligand description; 2) Link description and 3) ligands with metal.

Appendix 1: Subgroup-group relationship and psuedo meroherdal twinning.

This graph shows space groups that allow (pseudo)merohedral twinning. In theory if the point group is 1 then maximum number of twin operators (under constraints on the lattice parameters) could be 24 (point group 432). Point groups 432 and 622 are maximal groups and there can be no (pseudo)merohedral twinning.



Red arrows: No constraints are needed, merohedral twin could happen
 Black arrow: Additional constraints on cell parameters are needed, pseudo merohedral twinning can happen

Appendix 2: Behaviour of Rfactors in the presence and absence of twinning in the absence of any other crystal growth peculiarity (no pseudo rotation, modulation, pseudo translation, OD-disorder)

α is actual twin fraction and β is modeled twin fraction. σ_x is standard deviation of errors in atomic positions.

