Data integration and scaling

Harry Powell

MRC Laboratory of Molecular Biology

3rd February 2009

Abstract

Processing diffraction images involves three basic steps, which are indexing the images, refinement of the crystal and detector parameters, and integration itself. Each will be discussed with reference to its implementation in *Mosflm*. In addition to processing the images, *Mosflm* has a function to calculate an appropriate data collection strategy. *iMosflm*, the new graphical user interface, has real-time graphing tools to allow the user to monitor changes encountered in the data processing. In the latest version, it can also launch the CCP4 programs *Pointless* (to help identify the true symmetry of the crystal) and *Scala* (to scale and merge the data).

Scaling puts the observations measured in the integration step onto a common scale, taking account of differences between the images in the dataset. Scaling is followed by merging, when the components of the unique reflections (partial recorded reflections and symmetry equivalent reflections) are combined into unique observations.

It is extremely important to perform scaling and merging as soon as possible after data collection has finished, as it is the first real indication of data quality.

Optimization of Data Collection

Pre-process at least one image *before starting the full data collection* (preferably two at 90º to each other) to obtain:

- Cell parameters, crystal orientation and putative Laue group
- Estimate of mosaicity
- Effective resolution limit }
- Optimal crystal to detector distance } *e.g.* use *BEST*
- Exposure time }
- Strategy for data collection }

Remember! This is the last experimental stage - if you collect bad data now you are stuck with it. No data processing program can rescue the irredeemable!

Before starting to process

- Use the program tools to mask backstop, cryostream, other shadows.

- Set resolution limit to ~0.2Å higher than visible spots.

- Make sure beam position is more-or-less correct.

Data integration falls naturally into three parts:

(1) Determination of cell and orientation
- To give initial spot positions on detector
- to give an idea of cell dimensions and possible symmetry

(2) Refinement of these (and instrument) parameters - most accurately done *via* post-refinement

(3) Integration itself

Prerequisites for indexing (not only autoindexing!)

(1) Wavelength of radiation
(2) Beam position on image
(3) Crystal - detector distance

$n\lambda = 2d\sin\theta$ (Bragg's Law)

+ desirable

(4) detector mis-sets and deviations from ideal geometry

Indexing methods

Two methods in common use; both require mapping the two-dimensional detector co-ordinates onto a three-dimensional reciprocal lattice. Both give cell dimensions and the crystal orientation.

(1) Difference vectors - determine which vectors between RL points occur most often and reinforce each other.

(2) Fourier Transform methods; calculate either a 3D FFT from the RL points or 1D FFTs from the projections of the RL onto vectors around a hemisphere in reciprocal space.

In either case, apply operations to the primitive triclinic basis solution to generate Bravais Lattices to produce a putative list of solutions, and score these according to the distortion required.

Avoiding the wrong symmetry

- The unit cell dimensions are a consequence of the true crystal symmetry and do not define it.
- The true crystal symmetry cannot (in general) be determined from the unit cell dimensions, but only by comparing and analysing the intensities (or amplitudes) of those reflections which should be equivalent.
- Some polar space groups can be indexed in different ways
- Use tools like *Pointless* (CCP4) or *XPREP* (Bruker/Sheldrick) to perform this analysis and to convert to the correct setting

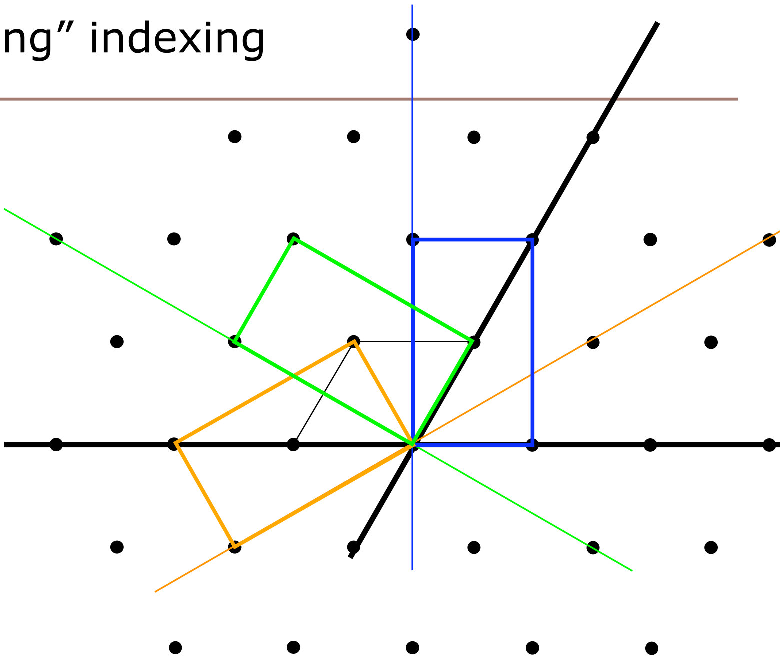
“equivalent but wrong” indexing

e.g. **a confusing case in C222:**

a=74.72, b=129.22
c=184.25 α=β=γ=90

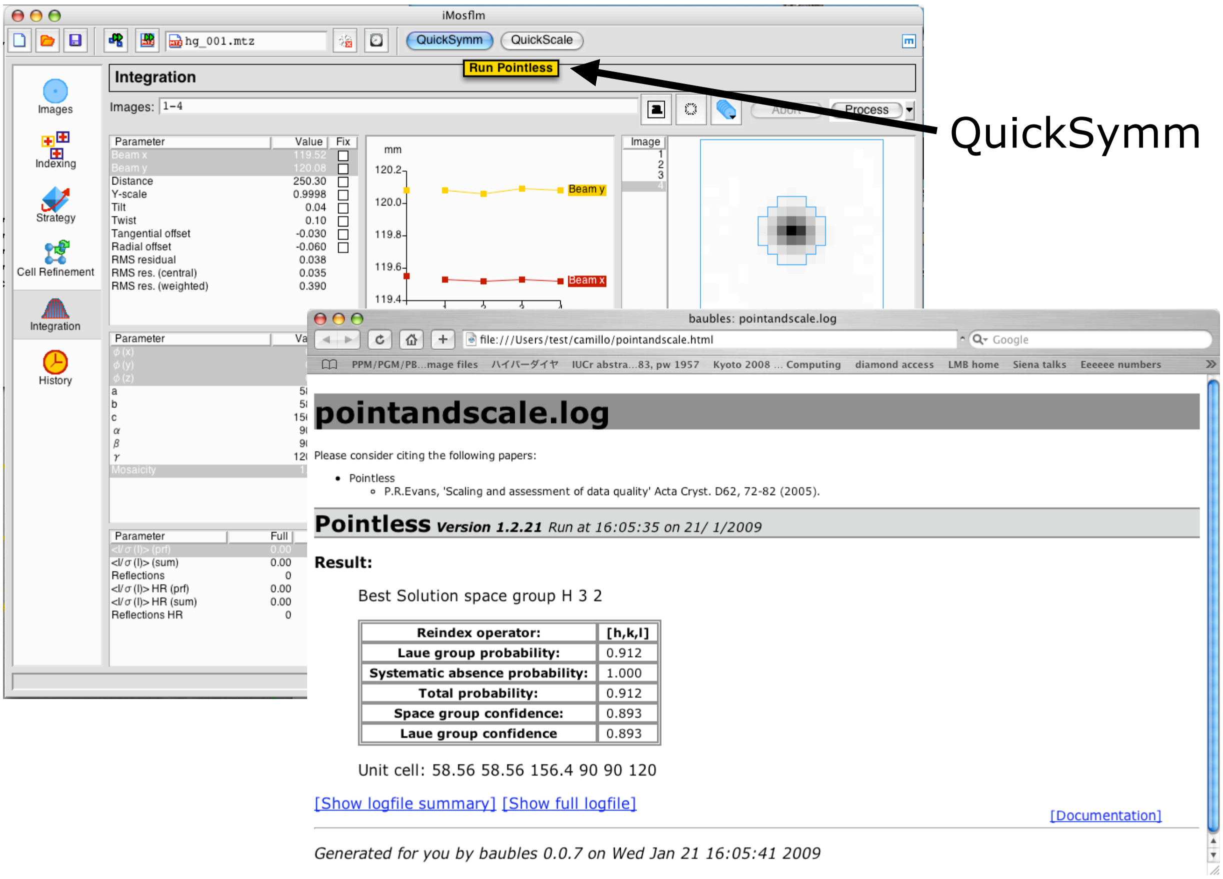
This has b ≈ √3 a   so can also be indexed on a hexagonal lattice

Hexagonal axes (black)



Three alternative C-centred orthorhombic lattices (coloured)

MRC | Medical Research Council



Parameter	Value	Fix
Beam x	119.52	□
Beam y	120.08	□
Distance	250.30	□
Y-scale	0.9998	□
Tilt	0.04	□
Twist	0.10	□
Tangential offset	-0.030	□
Radial offset	-0.060	□
RMS residual	0.038	
RMS res. (central)	0.035	
RMS res. (weighted)	0.390	

Parameter	Va
ϕ (x)	
ϕ (y)	
ϕ (z)	
a	5
b	5
c	15
α	9
β	9
γ	12
Mosaicity	1

Parameter	Full
<I/σ (I)> (prf)	0.00
<I/σ (I)> (sum)	0.00
Reflections	0
<I/σ (I)> HR (prf)	0.00
<I/σ (I)> HR (sum)	0.00
Reflections HR	0

pointandscale.log

Please consider citing the following papers:

- Pointless
  - P.R.Evans, 'Scaling and assessment of data quality' Acta Cryst. D62, 72-82 (2005).

Pointless *Version 1.2.21* *Run at 16:05:35 on 21/ 1/2009*

Result:

Best Solution space group H 3 2

Reindex operator:	**[h,k,l]**
Laue group probability:	0.912
Systematic absence probability:	1.000
Total probability:	0.912
Space group confidence:	0.893
Laue group confidence	0.893

Unit cell: 58.56 58.56 156.4 90 90 120

[Show logfile summary] [Show full logfile]

[Documentation]

Generated for you by baubles 0.0.7 on Wed Jan 21 16:05:41 2009

QuickSymm

Data integration falls naturally into three parts:

(1) Determination of cell and orientation

(2) Refinement of these (and instrument) parameters - most accurately done *via* post-refinement

- gives accurate cell dimensions
- allows accurate placement of measurement boxes on images

(3) Integration itself

Refinement of parameters

(1) Successful integration of an image requires accurate crystal and detector parameters. Cell dimensions can be refined by two complementary methods:

Spot position on detector, minimize:

$$\Omega_1 = \sum_{i=1}^{n} w_{ix}(X_i^{calc} - X_i^{obs}) + w_{iy}(Y_i^{calc} - Y_i^{obs})$$

n.b. i) rotation of crystal about phi axis has no effect on this residual so it can’t be refined.
ii) cell dimensions and other parameters (*e.g.* crystal to detector distance) may be strongly correlated.

Refinement of parameters

(2) Phi centroid method, minimize:

$$\Omega_2 = \sum_{i=1}^{n} w_i \left[\left(R_i^{calc} - R_i^{obs} \right) / d_i^* \right]^2$$

n.b. (i) need a reasonable knowledge of intensities for this, so it can only be done after integration - hence it is also called "*post-refinement*"

(ii) need a model for the rocking curve

(iii) can refine either mosaicity or beam divergence.

Some programs do this at the scaling stage rather than before integration

Data integration falls naturally into three parts:

(1) Determination of cell and orientation

(2) Refinement of these (and instrument) parameters - most accurately done *via* post-refinement

(3) Integration itself

- measures the intensity of the spots
- writes reflection file for further processing

Two basic types of integration:

(1) Summation integration (box-sum)

- add the counts in each pixel in the spot
- subtract the background underneath the spot

(2) Profile fitted integration

- improve measurements by applying a reflection profile calculated from many well-measured reflections

Summation integration (a)

Integration in the absence of X-ray background or detector noise

- Assign each pixel to the nearest reciprocal lattice point. Sum all pixels to obtain the integrated intensity.

- There is no penalty (in $I/\sigma(I)$) for including pixels beyond the physical extent of the diffraction spot.

- This situation is never realized in practice, even for very strong spots.

Summation integration (b)

Integration in the presence of X-ray background

- Need to subtract the X-ray background.

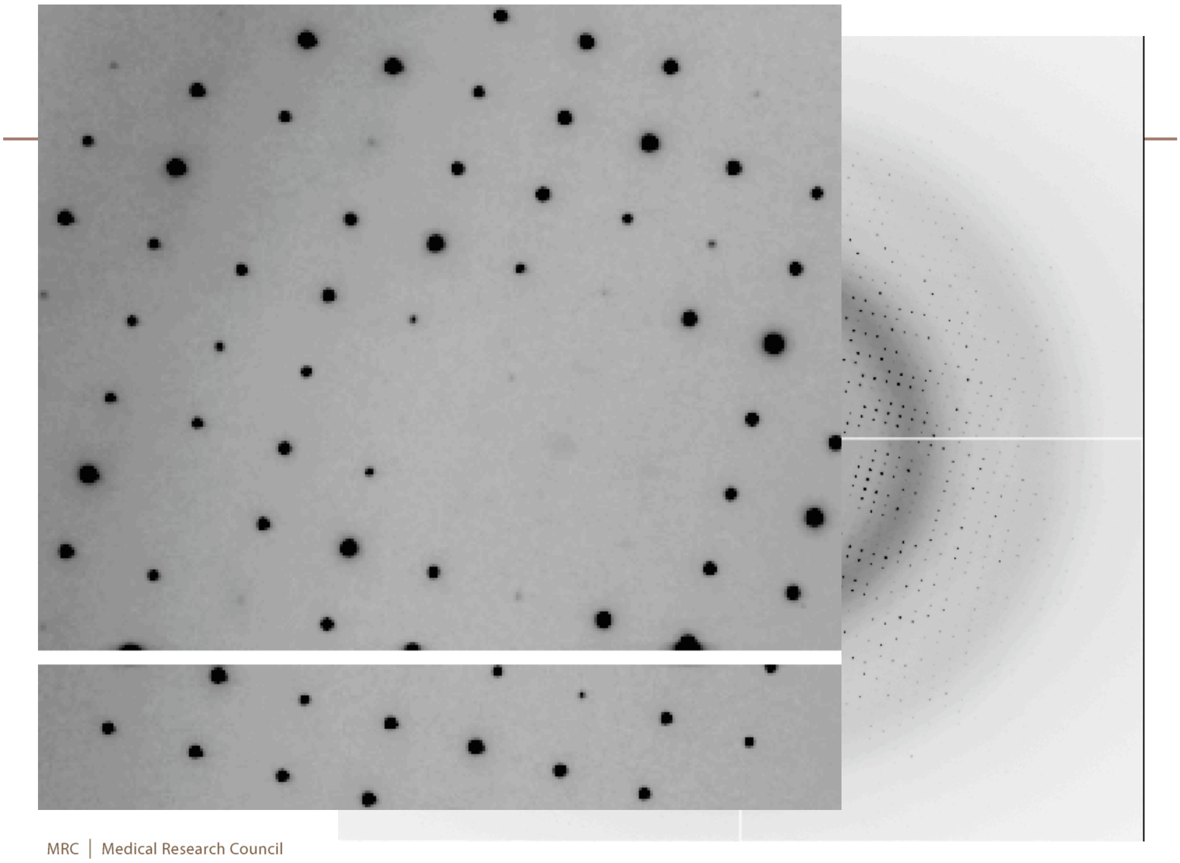
- This requires the definition of the peak and background regions; this definition may need to vary across the detector to allow for the variation in spot size.

- The size of a spot does not depend on its intensity.

Integration by profile fitting

Summation integration is unbiased (providing peak and background regions are correctly defined) but gives poor signal to noise ($I/\sigma(I)$) for weak reflections. Profile fitting can improve the estimation of weak intensities.

Based on the premise that the physical spot shape is independent of its intensity, and can be accurately modelled by analysing the profiles of strong, well-measured reflections.



Other improvements offered by profile fitting

- Measurement of incompletely resolved spots
- Elimination of outliers
- Estimation of overloaded reflections

Formation of standard profiles

In *Mosflm*, the standard profiles are usually accumulated over many images (10-20), so that most partials will be summed to give the equivalent fully recorded spot profile.

In practice, profiles based on both fully recorded and partially recorded reflections give very similar data quality to those based on fully recorded reflections only.

Variations in spot shape across the face of the detector need to be allowed for. In *Mosflm*, different profiles are evaluated for different regions of the detector, and a weighted sum of these standards used for evaluating each reflection.

Alternatively, a separate profile is evaluated for each reflection using other reflections within a specified distance of the reflection being integrated (*Denzo*, "profile fitting radius").

Two- and three-dimensional integration

2D integration

Measures spot intensity through profile fitting the shape of fully recorded reflections on individual images (*Mosflm*, *Denzo*, *etc...*). Partials *can* be included in this process. Historically considered more appropriate for coarse phi slicing.

3D integration

Builds up the profile by including the contributions of partials measured over successive images (*XDS*, *d*Trek*, *SAINT, etc...*). Historically recommended for fine phi slicing (but see *Pflugrath,* 1999).

Analysing the results of integration

- Check graphs - they should vary smoothly without obvious discontinuities.

- Large changes in parameters may indicate problems with the crystal or instrument.

- $I/\sigma(I)$ at (high resolution limit-0.2Å) should be >1

- Look at any images corresponding to discontinuities in the graphs.

- Check any warnings issued by the program; it may be best to re-process after following the advice given (all warnings given by *Mosflm* are accompanied by suggestions on how to improve the processing).

Scaling with *SCALA* in *iMosflm*

Scaling and merging the data is the next step following integration. It is important because:

- it attempts to put all observations on a common scale
- it provides the main diagnostics of data quality and whether the data collection is satisfactory

Because of this diagnostic role, it is important that data are scaled as soon as possible after collection, or during collection, preferably while the crystal is still on the camera.

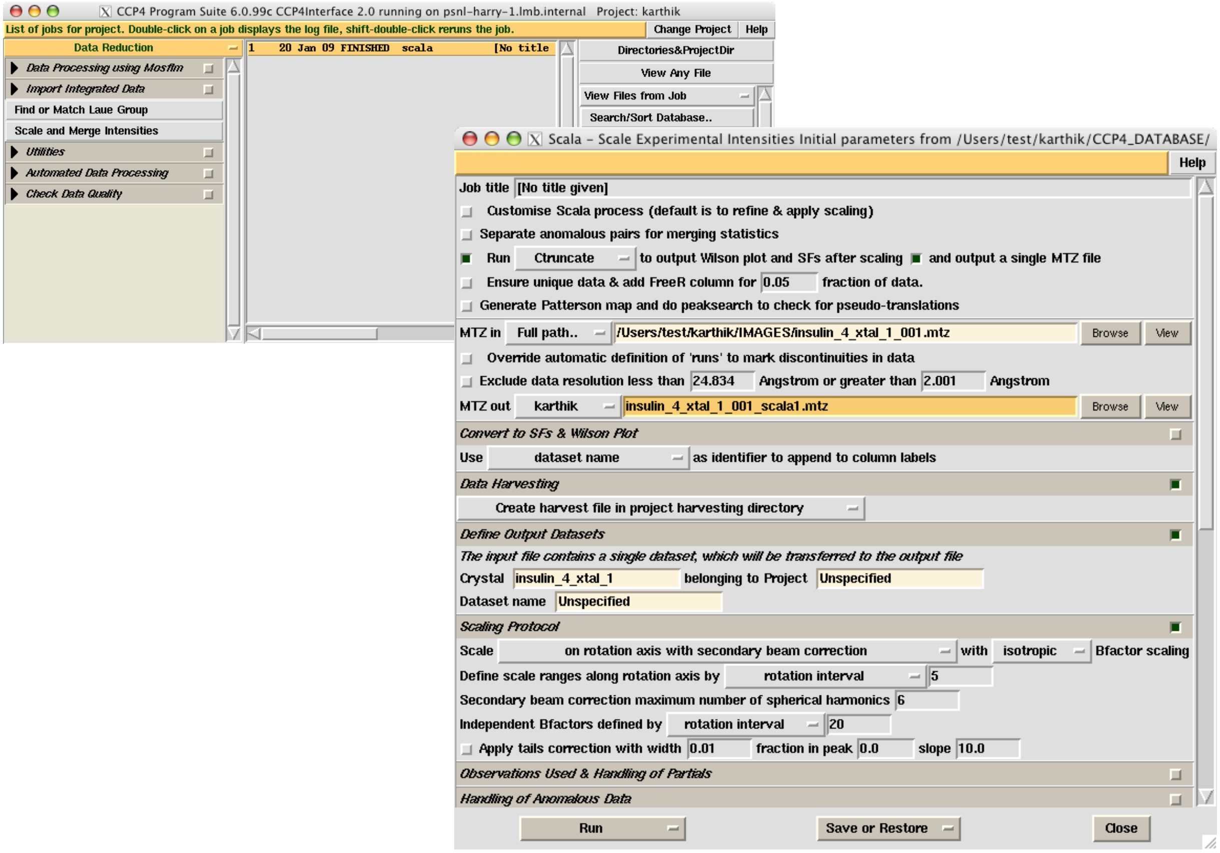
To aid this, *iMosflm* includes a “quick scaling” task in the integration pane to give an initial idea of the data quality.

Scaling with *SCALA* in *ccp4i*

The “quick scaling” task in *iMosflm* is not optimal; the mtz file produced is not intended for use in subsequent structural analysis.

Running *SCALA* from *ccp4i* is more flexible (*i.e.* it allows the run to be customised), so should give substantially improved results.

The *SCALA* task in *ccp4i* also runs *truncate* or *ctruncate*; these produce structure factors (which most other programs use) and statistics that can help to analyse twinning.


CCP4 Program Suite 6.0.99c CCP4Interface 2.0 running on psnl-harry-1.lmb.internal   Project: karthik
Change Project
Help
Data Reduction
Data Processing using Mosflm
Import Integrated Data
Find or Match Laue Group
Scale and Merge Intensities
Utilities
Automated Data Processing
Check Data Quality
Directories&ProjectDir
View Any File
View Files from Job
Search/Sort Database..
Scala - Scale Experimental Intensities Initial parameters from /Users/test/karthik/CCP4_DATABASE/
Help
Browse
View
Browse
View
Run
Save or Restore
Close

List of jobs for project. Double-click on a job displays the log file, shift-double-click reruns the job.

```
1    20 Jan 09 FINISHED  scala        [No title
```

Job title [No title given]

□ Customise Scala process (default is to refine & apply scaling)

□ Separate anomalous pairs for merging statistics

■ Run Ctruncate to output Wilson plot and SFs after scaling ■ and output a single MTZ file

□ Ensure unique data & add FreeR column for 0.05 fraction of data.

□ Generate Patterson map and do peaksearch to check for pseudo-translations

MTZ in Full path.. /Users/test/karthik/IMAGES/insulin_4_xtal_1_001.mtz

□ Override automatic definition of 'runs' to mark discontinuities in data

□ Exclude data resolution less than 24.834 Angstrom or greater than 2.001 Angstrom

MTZ out karthik insulin_4_xtal_1_001_scala1.mtz

Convert to SFs & Wilson Plot □

Use dataset name as identifier to append to column labels

Data Harvesting ■

Create harvest file in project harvesting directory

Define Output Datasets ■

The input file contains a single dataset, which will be transferred to the output file

Crystal insulin_4_xtal_1 belonging to Project Unspecified

Dataset name Unspecified

Scaling Protocol ■

Scale on rotation axis with secondary beam correction with isotropic Bfactor scaling

Define scale ranges along rotation axis by rotation interval 5

Secondary beam correction maximum number of spherical harmonics 6

Independent Bfactors defined by rotation interval 20

□ Apply tails correction with width 0.01 fraction in peak 0.0 slope 10.0

Observations Used & Handling of Partials □

Handling of Anomalous Data □

Checking the output of *SCALA*

Check these files/plots:

- ROGUES
- Normal probability plot(s)
- Correlation plot
- Surface plot
- *SCALA* log file
- loggraph output

What to check for in the output of *SCALA*

- the refinement should converge after a few cycles
- scale factor should not increase too much
- fractional partial bias - should be zero, never positive
- ideally no outliers in ROGUES file
- merging R factors (Rmeas, Rpim, etc.) should increase with
    - increasing resolution
    - decreasing intensity

- completeness
- multiplicity

Useful references

Acta Cryst. Section D, (55) 1999 pt 10 [*] & (62) 2006 pt 1[§]

Data collection:	*§	Zbigniew Dauter
	§	Gleb Bourenkov & Sasha Popov
Cryocooling:	§	Elspeth Garman
Autoindexing:	*	Harry Powell
2D integration:	*§	Andrew Leslie
3D integration:	*	Jim Pflugrath
Scaling:	§	Phil Evans

Acknowledgements

MRC | Laboratory of Molecular Biology

Andrew Leslie
Phil Evans
Luke Kontogiannis
Geoff Battye

ESRF
CCP4
MRC