Challenging spots - brief processing notes for iMosflm

Before starting, note the following.

It is assumed that all students will have gone through the online tutorial for iMosflm before even thinking about processing these data! The tutorial is available at

```
http://www.mrc-lmb.cam.ac.uk/harry/imosflm/ver105
```

(select "documentation" from the left hand menu, and the tutorial specifically for iMosflm, not the "Mosflm User Guide")

There are three datasets that the students have been requested to download , forming a three-wavelength MAD experiment - high energy remote wavelength (hrem), inflection point (ip) and peak (pk). If we look at the first image in the "hrem" dataset, we can see that the reason the spots are "challenging" is because we have a double crystal of some kind - it looks like a non-merohedral twin, where the two components are at right-angles to each other. Look at spots around the backstop and you can see that they are overlapped (giving the impression of a single lattice), but if you look at spots at ~11Å resolution you can see that two lattices can be seen quite clearly. I'd guess that the crystal is orthorhombic, but would want to try to index to confirm this. If you look very carefully, you can also see that each component is also split into at least two major components – these should be able to be integrated together.

For reasons that are not immediately apparent, the first dataset collected (hrem) has only 269 images, but the others each have 360.

Note that each dataset also has an image "0", which is distinguished from the rest by being called "Sk..." (lower case k) rather than "SK...". These extra images contain a direct beam shot - which is useful, because the beam position recorded in the image headers is not in the expected frame of reference for Mosflm!

Have a look at image 180 (which is 90º away in phi) to see if the spots from the two lattices are separate there too - it seems that they are.

There are two ways forward with these images - either try to process as a twin crystal and expand the measurement boxes to include both lattices (and try to deal with the twinning later), or pick one lattice and ignore the low resolution overlapped spots. Since the separation of the two lattices is quite clear at higher resolution, the former is not really an option. I would not bother processing the low resolution data that cannot be separated – so I'd set the lower resolution limit high enough to prevent this – so, to maybe 18Å.

A better way forward would be to try to collect data on a single crystal! If all the samples grown are similarly twinned, it may be possible to collect data on a microfocus beamline (e.g. I24 at Diamond) and just "shoot" one component, provided the crystal has clearly identifiable regions and they are not completely intermingled.

The route to process these data is therefore -

read images for hrem
set the spot finding region for indexing to exclude the low resolution region
index & estimate mosaicity
refine the cell

integrate
quickscale
read images for ip
refine cell
integrate
read images for pk
refine cell
integrate
scale all three datasets more carefully

I processed in this order because that's their alphabetical order.

In more detail

Read in the 269 "hrem" images using the normal iMosflm method.

Increase the minimum radius for the spot search to ~25mm using the tool in the image display, or in "Processing options->spot finding".

Go to the indexing pane - unless you have unchecked the options to automatically find spots and automatically index & estimate mosaic spread, the indexing will proceed and you will get a list of "good" solutions;

```
       Lat. Pen.   a      b      c      α      β      γ   σ(x,y)  σ(f)   δ(beam)
  1    aP     0  113.2  113.0  307.8 100.0  100.1  92.3  0.23  0.51   0.23 (0.1)
  2    mC     1  156.7  163.3  307.9  90.0  104.6  90.0  0.23  0.51   0.23 (0.1)
  3    mC     2  163.2  156.7  308.7  90.0  105.2  90.0  0.34  0.99   0.15(0.1)
  4    oF     3  156.6  163.3  595.8  79.6   90.0  90.0  0.35  0.93   0.16(0.1)
  5    aP     4  113.2  113.0  308.3  90.0   79.8  92.3  0.57  0.87   0.15(0.1)
  6    mI     4  156.5  595.9  113.3  90.0  133.9  90.0  0.54  1.05   0.07(0.1)
  7    mC     5  156.7  163.3  307.9  90.0  104.6  90.0  0.54  1.05   0.09(0.1)
  8    mC     9  604.2  112.2  112.9  90.0  100.1  90.0  0.45  0.86   0.09(0.1)
  9    mC     9  605.9  113.3  112.8  90.0  100.9  90.0  0.42  0.93   0.07(0.1)
 10    tI    13  112.5  112.5  595.4  90.0   90.0  90.0  0.50  1.05   0.11(0.1)
 11    oI    13  112.2  112.9  595.3  90.0   90.0  90.0  0.52  1.06   0.11(0.1)
 12    mP    60  113.1  113.1  307.8  90.0  100.2  93.8   -     -        -
```

Mosflm chooses the tetragonal solution, but its σ(x,y) is substantially larger than that for the triclinic basis solution. The face centred orthorhombic solution (#4) looks better. Choose this and check the predictions. If they look about right, estimate the mosaic spread.

Note that only one of the two lattices is predicted.

I would set the lower resolution limit to ~18Å or so to avoid trying to integrate those reflections with the worst overlap. At present the MTZ file format that Mosflm writes cannot deal with reflections with multiple components.

Try integrating the first 20 images (10°) and running "Quicksymm" - Pointless should indicate F222 as the space group, with fits in with what we chose from indexing above.

Now try refining the cell - Mosflm will give you a warning that it may be unstable; in this case it seems to work reliably and give better values for parameters at the start of integration. I used 4 segments (images 1-6, 91-96, 181-186 and 261-266) instead of the default from iMosflm. The mosaic spread will probably refine up to ~1.5 – 2°, so you may want to fix it at a sensible starting value, e.g. ~1.2°.

Select the "Integration task". Select all images, fix the mosaic spread, and set the overall box size to 21x21 (in “advanced integration” settings – also un-check “Optimise overall box size” to fix it – this will stop the box expanding into spots from the other lattice) and click the "Process button". Integration will take a few minutes on a modern computer. The integration proceeds smoothly up to around image 170, where there are some sharp kinks in the plots, probably due to overlapping reflections from the two lattices. However, the integration proceeds to completion.

Check the warnings
You will probably find a few warnings (click on the flag on the bottom right of the GUI for a list). Some of these are "information", some are "warnings"; it may be possible to improve the integration by taking account of these. In this instance, I get 8 in total - 3 "information" and 5 "warning".

Double-click on the text for each warning for detailed information on the problem and how to improve the processing. You may notice that some are contradictory - if this happens, it should usually be possible to just make a mental note, but if you have time you should examine the full-length log file to see if you can determine if the problem happens only with a certain range of images, and displays differently in different parts of the dataset.

Quickscale
Open "Settings->Processing options->Advanced integration", and uncheck the box "Treat anomalous data in Quickscale". For this part of the data we have probably have insufficient information to generate good statistics on the anomalous contribution. After processing all the parts of the data, we may want to reconsider this.

Select the "Quickscale" task - this launches a Pointless and Scala job - when finished, it will open a web-browser so that you can see the overall statistics. These have been calculated quickly using a small subset of Scala's options, so that you can make sure the images have been processed more-or-less okay. It is probably worthwhile scaling again more carefully to see if you can improve the statistics (dealt with later after integrating all the images).

Checking the results of Quickscale
Check the result from Pointless - does the symmetry agree with what you thought at the start?

Check the plots from Scala - they should vary smoothly through the data processing, values should lie in the "expected" ranges.

Processing the other datasets
Since the three datasets were collected with very similar wavelengths and crystal settings, the same orientation can be used for all three. So, after integrating the hrem dataset, load the images from the pk dataset and go straight to the Refinement task (no need to index). Make sure the lower resolution limit is 18Å and refine with four segments, with a fixed value for the mosaic spread – use the same as for integrating the hrem dataset. Now integrate - choose a new MTZ filename, check the resolution, fix the mosaic spread and overall box size and integrate the data; things seem to go okay.

The inflection point dataset (ip) processes a bit more easily than the peak dataset, even though both datasets look very similar. However, it can be processed in exactly the same way without any real problems.

It is usually better to allow the mosaicity to refine during integration, but it is the worst defined residual in the post-refinement, so sometimes it is necessary to fix it to stop it refining to unrealistic values. This is a case in point. It's also usually best to let the overall box size refine, but here there are compelling reasons to fix it to a relatively small value.

Scaling
The only things I would change to begin with regarding the scaling are -

(i) use “intensities combine” or “intensities integrated” instead of the default “intensities profile” - it seems that for strong reflections, the measurement is a little better like this.

(ii) it can be better if you are looking for a weak anomalous signal to use “anomalous off” in the scaling – outlier rejection is based on Friedel opposites having the same intensity rather than on Bijvoet pairs being different.

(iii) if you want to run the SHELXC/D/E pipeline, use “output polish unmerged” to produce an unmerged file in scalepack format, that SHELXC will process optimally.

So, modelling my Scala input on the default one from ccp4i, I write a shell script like this (you could use bash instead of csh, or Python or whatever you like);

```
#!/bin/csh -f

source /Users/harry/ccp4-6.2.0/ccp4-6.2.0-intel/ccp4.setup

foreach dataset ( hrem ip pk  )

echo running pointless for ${dataset}
pointless HKLIN ${dataset}.mtz HKLOUT pointless-${dataset}.mtz > pointless-
${dataset}.log

scala:
echo running scala for ${dataset}
scala HKLIN pointless-${dataset}.mtz \
      HKLOUT scala-${dataset}.mtz \
      SCALEPACK ${dataset}.sca \
      SCALES ${dataset}.scala \
      ROGUES ${dataset}_rogues.log \
      NORMPLOT ${dataset}_normplot.xmgr \
      ANOMPLOT ${dataset}_anomplot.xmgr \
      PLOT ${dataset}_surface_plot.plt \
      CORRELPLOT ${dataset}_correlplot.xmgr \
      ROGUEPLOT ${dataset}_rogueplot.xmgr > scala-${dataset}.log <<EOF

title ${dataset}
name project Challenging crystal Mad dataset ${dataset}
exclude EMAX 10.0
partials  check test 0.95 1.05 nogap
intensities combine
final PARTIALS
scales rotation SPACING 5 secondary 6 bfactor ON BROTATION SPACING 20
UNFIX V
FIX A0
UNFIX A1
initial MEAN
tie surface 0.001
tie bfactor 0.3
cycles 10 converge 0.3 reject 2
anomalous off
output polish unmerged
print brief nooverlap
RSIZE 80
EOF

end
```