

dtscalemerge

SBC01-????-00

February 16, 1996

Prepared by: J.W. Pflugrath
Molecular Structure Corporation
3200 Research Forest Drive
The Woodlands, Texas 77381
(713) 363-1033
(713) 364-3628 FAX
jwp@msc.com

Prepared for: Contract No. 943072401
Mary Westbrook
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439
(708) 252-8914
(708) 252-4021 FAX
westbrook@anl.el.anl.gov

Reviewed by:

Approved by:

Copyright © 1996, 1995, 1994 Molecular Structure Corporation
RESTRICTED RIGHT NOTICE SHORT FORM (JUNE 1987)

Use, reproduction, or disclosure is subject to restrictions set forth in Contract No. W-31
109-ENG-38 and Contract No. 943072401 with the University of Chicago, Operator of
Argonne National Laboratory

TABLE OF CONTENTS

1 SCOPE AND PURPOSE	3
1.1. Definitions and Abbreviations	3
1.2. References	4
2 BACKGROUND	4
3 MATHEMATICS OF SCALING	5
4 RUNNING DTSCALEMERGE	9
4.1. Command line options	9
4.2. Examples	11
4.3. Results	12
4.4. Further analysis	24
APPENDIX A EXAMPLE OUTPUT LISTING	26

1 Scope and Purpose

This document describes how to use `dtscalmerge` to calculate and apply scale factors to a d*TREK reflection list that contains multiple batches. In this process various statistics are calculated to help you evaluate the quality of the input reflections and the output averaged results. You use `dtscalmerge` to:

- calculate and apply scaling factors for reflections from multiple batches,
- average multiple observations of symmetry-related reflections into a single estimate of intensity and standard deviation; reject outliers in the process,
- calculate and display the multiplicity of observed reflections,
- calculate and display the number of overlapping reflections among the scaling batches,
- calculate and display *Rmerge vs Batch*,
- calculate and display *Rmerge vs Intensity/SigmaI*,
- calculate and display *Rmerge vs Resolution*, and
- calculate and display *Completeness vs Resolution*.

You need not refine any scaling parameters with `dtscalmerge`. The program will calculate statistics even if all reflections belong to a single scaling batch. You use the `dtreflnmerge` program to prepare input to `dtscalmerge` in a very flexible way.

1.1 Definitions and Abbreviations

The definitions described in the `dtreflnmerge`, `dtcollect`, `tCrefine`, and `dtpredict` documents are appropriate for `dtscalmerge`. Below are defined some new terms and equations.

I_{hj} is the j th measurement or observation of a reflection with Miller index h that has been reduced to an asymmetric unit. I_{hj} at this stage has been corrected for Lorentz-polarization effects, but **NOT** multiplied by any scaling factors.

I_{hi} is the intensity of the i th measurement or observation of a reflection with Miller index h that has been reduced to an asymmetric unit. I_{hi} at this stage has been corrected for Lorentz-polarization effects and multiplied by any scaling factors.

K_j is the scale factor to apply to reflections from scaling batch j , such that $I_{hi} = K_j I_{hj}$. K_j may be a function of other parameters. For example, for a radiation damage correction, $K_j = k_j e^{-2B_j(\sin\theta/\lambda)^2}$, where k_j is a scale factor and B_j is a “temperature” factor.

G_j is the inverse scale factor: K_j^{-1}

σ_{hj} is the standard deviation of the j th measurement or observation of a reflection with Miller index h . It has not been multiplied by any scaling factors.

σ_{hi} is the standard deviation of the i th measurement or observation of a reflection with Miller index h . It has been corrected for any scaling factors.

\bar{I}_h is the weighted average intensity of the unique reflection with Miller index h that has been computed from the scaled and corrected i symmetry-related observations of the reflection. It can also be computed from the scaling factors and the unscaled, but otherwise corrected, symmetry-related observations.

$$R_{merge} = \frac{\sum_h \sum_i |I_{hi} - \bar{I}_h|}{\sum_h \sum_i \bar{I}_h} \quad (1)$$

where the outer sum is over all unique reflections and the inner sum is over the symmetry-related observations for reflection with Miller index h . There must be at least 2 symmetry-related observations for an h to be included in R_{merge} .

1.2 References

The following are useful references:

- Fox, G.C. and Holmes, K.C. (1966) An Alternative Method of Solving the Layer Scaling Equations of Hamilton, Rollett and Sparks. *Acta Cryst.* **20**, 886-891.
Hamilton, W.C., Rollett, J.S. and Sparks, R.A. (1965) *Acta Cryst.* **18**, 129.
Hamilton, W.C. (1964) *Statistics in Physical Science*, The Ronald Press Company, New York.
Kabsch, W. (1988) Evaluation of Single-Crystal X-ray Diffraction Data from a Position-Sensitive Detector. *J. Appl. Cryst.* **21**, 916-924.
Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1988) *Numerical Recipes in C*, Cambridge University Press, Cambridge, pp. 517-565.
Rollett, J.S. (1965) *Computing Methods in Crystallography*, Pergamon Press, Oxford.

2 Background

In a single crystal x-ray diffraction experiment, you measure the intensities of many hundreds of Bragg reflections. As in any scientific experiment, you are not satisfied with a

single measurement of each reflection. Instead you make multiple measurements of all the reflections in order to assign for each Bragg reflection the best estimate of its intensity and standard deviation.

You also can expect certain constraints on your observations. Since x-ray diffraction is a Poissonian process, you expect the variance of intensity I measured in photons to equal the number of photons or I . That is, the standard deviation of I , σ_I , is at least \sqrt{I} . You cannot measure the intensity more precisely than \sqrt{I} . Furthermore, you expect all the measured intensities to be greater than or equal to 0. Any that are less than 0 are probably that way because of systematic errors introduced by your experimental method or because of *expected* statistical fluctuations.

When you scale together datasets of reflections collected at different times, on different instruments, or from different crystals or crystal volumes, you correct for a variety of effects. These may include

- different crystal volumes,
- radiation damage,
- different absorption due to different paths through the crystal,
- different detectors,
- wavelength dependent factors,
- different or fluctuating source intensities.

At the same time, you flag and exclude measurements which have a high probability of being mismeasured. Do not use scaling to massage your data so that you get a fantastic *Rmerge*. If your datasets are fantastic, there will be no need to massage or overfit the results. If your datasets do not meet the expectations defined by the physics of the experiment, then problems exist which need to be addressed and corrected. Rollett (1965) wrote on p. 105:

A moral of this example is that it is unwise to try to cover unknown systematic errors simply by increasing the estimates of random errors.

Dtscalemerge helps you decide if any problems exist and where they might be.

3 Mathematics of scaling

Hamilton, Rollett and Sparks (1965) published a least-squares method for determining the scale factors between crystallographic datasets. This method was made more robust by Fox and Holmes (1966) and serves as the basis for nearly all scaling algorithms used by crystallographers today.

Non-linear least squares

We will determine the scale factor K_j for a batch of reflections I_{hj} such that

$$K_j I_{hj} \approx \bar{I}_h \quad (2)$$

We could minimize the function

$$M = \sum_h \sum_j w_{hj} (K_j I_{hj} - \bar{I}_h)^2 \quad (3)$$

but this leads to problems, so we minimize instead the function

$$\chi^2 = \sum_h \sum_j w_{hj} (I_{hj} - G_j \bar{I}_h)^2 = \sum_h \sum_j \left(\frac{(I_{hj} - G_j \bar{I}_h)}{\sigma} \right)^2 \quad (4)$$

where the inverse scale factors G_j are applied to the weighted average intensities. The weights in this equation w_{hj} are derived from the standard deviations of the observations with possibly a couple of fudge factors applied:

$$w_{hj} = \frac{1}{(\sigma_{hj} E_{mul})^2 + (I_{hj} E_{add})^2} \quad (5)$$

where E_{mul} is an error factor (default = 1.0) to multiply all the observed standard deviations by and E_{add} is an error factor (default = 0.0) to multiply the observed intensity by and then add in quadrature to the standard deviations. The σ above without subscripts indicates a modified standard deviation derived from these error factors and is equal to $1/\sqrt{w_{hj}}$.

We also have an estimate for the weighted average intensity for use in least squares:

$$\bar{I}_h = \frac{\sum_j w_{hj} G_j I_{hj}}{\sum_j w_{hj} G_j^2} \quad (6)$$

We can also apply the scaling to the observed intensities ($I_{hi} = K_j I_{hj}$) and get the equivalent equation:

$$\bar{I}_h = \frac{\sum_i w_{hi} I_{hi}}{\sum_i w_{hi}} \quad (7)$$

were w_{hi} is the weight derived as before, but by first scaling the standard deviations appropriately:

$$w_{hi} = \frac{1}{(\sigma_{hj} K_j E_{mul})^2 + (\bar{I}_h E_{add})^2} \quad (8)$$

and

$$\sigma_h = \frac{1}{\sqrt{\sum_i w_{hi}}} \quad (9)$$

If our weights are correct and our errors are normally distributed, then we can estimate the value of χ^2 at its minimum. This serves as another useful check on the procedure and the weighting scheme.

Multiple observations

Recall that diffraction is essentially a Poissonian process. Thus, if we have 10 measurements of the same Bragg reflection, $I_{h1}, I_{h2}, \dots, I_{h10}$ each with intensity 100, we expect a standard deviation 10. We can scale and average these reflections together by hand. Clearly all the scale factors are equal to 1. What is the weighted average intensity? That is easy to calculate. What is the standard deviation?

$$\begin{aligned} \bar{I}_h &= 100 \\ \sigma_h &= \frac{1}{\sqrt{\sum_i w_{hi}}} = \frac{1}{\sqrt{\frac{1}{10^2} + \dots + \frac{1}{10^2}}} = \frac{1}{\sqrt{10}} = 3.16 \end{aligned} \quad (10)$$

Thus by making multiple observations we have increased the I/σ from 10 to almost 32. In other words, we have a more precise estimate of the intensity. Note that the same effect would have occurred had we measured just once, but counted 10 times longer. In this case, the expected intensity would have been $10 \times 100 = 1000$ and the standard deviation would have been $\sqrt{1000} = 31.62$ which gives the same I/σ as before. In a real experiment, the measurements would not all have been the same and would have exhibited systematic effects, so the above only serves as a qualitative example. We can expect however that the average I/σ of a dataset will increase by a factor of \sqrt{N} , where N is the average number of times a unique reflection is measured.

Rejecting outliers

In a carefully controlled experiment, you should expect very few, if any, outliers that should be rejected. In an x-ray diffraction experiment, the outliers might occur because of a cosmic ray impinging on the detector near the reflection in question. This presumes that you have excluded reflections that are blocked by the beamstop and controlled for other problems. Outliers can disturb the least squares scaling process, so an attempt should be made to exclude them during scaling.

Dtscalmerge does the following with respect to outliers. In the first two least squares minimization cycles, no reflections are rejected from the minimization. In later cycles, reflections are rejected as follows.

1. The average intensity and standard deviation are computed from all symmetry-equivalent reflections, *except* the one to be tested (the test reflection).
2. The difference between the average intensity and the intensity of the test reflection is computed for each observation in turn. The observation with the worst difference is flagged.
3. If the worst difference is larger than R times the standard deviation in magnitude, where R is set by the user, then that reflection is flagged as rejected and not used in the minimization of χ^2 . If R is set to 3, those reflections which have about a 99% probability of being outliers are rejected.
4. Steps 1-3 are repeated until no more observations are rejected.
5. If only 2 observations remain and one should be rejected, both are rejected.

With regard to step 5, there are some scenarios where you might expect the one with the smaller magnitude to be an outlier and others where you might expect the one with the larger magnitude to be an outlier. For instance, a CCD detector with a fiber-optic taper has zingers at some characteristic rate. The effect of a zinger is to have a large number of counts in a small area of pixels. If the zinger occurs directly on top of a Bragg reflection, then that reflection will have an intensity inconsistent with its symmetry-equivalent mates: it will be much higher than the others. If the zinger occurs in the area used to estimate the background for the reflection, then the background estimate will be too high and the intensity will be lower than expected. There are other obvious scenarios (ice diffraction, beamstop shadows) that will not be elaborated upon here. We think it is safest to reject both observations.

Expectations in the Statistics

If we have a proper error model, then we can calculate the R_{merge} that we expect to get from a dataset or subset thereof. We expect the standard deviations of our observations

to be no better than \sqrt{I} , or expressed as a fraction no better than \sqrt{I}/I or $1/\sqrt{I}$. We must adjust this by the effects of multiple measurements as discussed above. For example, if our average intensity is 100, there are no systematic errors and we measured each reflection twice, then the best we can expect is a *Rmerge* of $1/\sqrt{100} \div \sqrt{2} = 7.1\%$. On the other hand, if our average intensity is 1000 and we measured each reflection 6 times, then we might expect a *Rmerge* of 1.3%. If this is not the case, then either you are not counting photons or your error model (i.e. your σ 's) is wrong. It pays to keep this in mind when reviewing the results of the averaging procedure.

4 Running dtscalmerge

After `dtscalmerge` has been installed and placed in your `PATH`, just enter `dtscalmerge` along with command line options to run it. It parses the command line arguments, reads any required input reflection lists and image headers and writes a new reflection list. Messages are written to `stdout` and `stderr` as required. The syntax for running `dtscalmerge` is:

```
dtscalmerge input_header_file [ -fixB ]
                               [ -cycles nCycles ]
                               [ -reject fRejSigma ]
                               [ -sigma fExcludeSigma ]
                               [ -add fFudgeAdd ]
                               [ -mul fFudgeMul ]
                               [ -fix sBatchname ]
                               [ -scale fFixedScale ]
                               [ -bfac fFixedB ]
                               [ -ref scaled_refl_nlist_file ]
                               [ -anom ]
                               input_file [output_file]
```

Optional arguments appears within square brackets [...]. If an output file is not given on the command line, then no output file is written. You might do this to test options of `dtscalmerge`. Also note that `dtscalmerge` uses the standard d*TREK spacegroup file defined by the environment variable `DTREK_SPACEGROUP_FILE`. The crystal unit cell dimensions and spacegroup are determined from the `input_header_file`.

4.1 Command line options

You tell `dtscalmerge` what to do with options specified on the command line. The syntax of the command line options is described next.

`input_header_file`

Enter the name of a d*TREK image file. The header of this file is used to create a crystal and spacegroup object. The spacegroup is required in order to reduce reflection indices to the proper asymmetric unit. The crystal unit cell dimensions are required to calculate the proper resolution and completeness.

`-fixb` This fixes the so-called B-factors which are all set to 0, so that the scaling factors $K_j = k_j$. They are not refined. B-factor refinement is an attempt to model the decrease in diffraction intensities due to radiation damage over time. If you do not suspect radiation damage, you should not refine the B-factors. The default is to refine B-factors and not fix them to 0.

`-cycles nCycles`
Sets the maximum number of non-linear least squares cycles to perform. If the refinement converges, then the maximum number of cycles may not be reached. No shifts are applied the last cycle. In order to just calculate statistics, use `-cycles 1`. Default: 30.

`-reject fRejSigma`
Sets the rejection level for reflections. Reflections with scaled intensities that differ by more than `fRejSigma` from the weighted average intensity calculated by other symmetry-related reflections are flagged as rejected by setting their observed standard deviations to be negative. Default: 1×10^{11} .

`-sigma fExcludeSigma`
Input reflections with I/σ_I less than `fExcludeSigma` are excluded from the contributing to the scale factor refinement. However, these reflections are included in the final statistics. Default: 3.

`-mul Emul`

`-add Eadd` These error factors alter the error model and weights applied to the observed reflections according to the equation 5:

$$w_{hj} = \frac{1}{(\sigma_{hj} E_{mul})^2 + (I_{hj} E_{add})^2}$$

Defaults: $E_{mul} = 1$. $E_{add} = 0$.

`-fix sBatchname`

`sBatchname` specifies the name of the batch whose scale factors will remain fixed. The scale factors of all other batches will shift relative to this batch. Default: first batch in the input reflection list.

- `-scale fFixedScale`
`fFixedScale` specifies the scale factor k_j of the fixed batch. All shifted scale factors are rescaled so that this value remains fixed. Default: 1.
- `-bfac fFixedB`
`fFixedB` specifies the so-called B factor B_j of the fixed batch. All shifted B factors are rescaled so that this value remains fixed. Default: 0.
- `-ref scaled_refl_nlist_file`
This option writes the scaled, but unaveraged, reflection list to the specified file. Rejected reflections will have standard deviations less than 0. You can use `dtreflnmerge` to extract only rejected reflections from this file if desired (see section 4.4). The default is not to write such a file.
- `-anom` This option when present will calculate anomalous differences. The default is to average Bijvoets into a single intensity value.
- `input_file` The name of a d*TREK reflection list file. Only a single input file may be scaled and merged by `dtscalmerge`. If you wish to treat multiple reflection lists, first combine them with `dtreflnmerge`.
- `output_file`
The name of the output d*TREK reflection file. This file will contain the Miller index hkl , the average weight intensity I , its standard deviation σ_j for all unique reflections with contributions from the input reflection list. You can filter this reflection list with `dtreflnmerge` if you desire.

4.2 Examples

Once your input reflection list file has been prepared by `dtreflnmerge`, it is easy to run `dtscalmerge`. You also need an image file with the crystal and spacegroup information.

Example 1

```
1 # Example 1
2 dtscalmerge lyso.head lyso020.ref lyso020.ave
```

Line 2 The crystal and spacegroup information is found in the file `lyso.head`. The input reflection list is `lyso020.ref` and the output averaged reflection list is `lyso020.ave`. The defaults are used for all the options.

Example 2

```
1 # Example 2
2 dtscalemerge lyso.head -cycles 30 -reject 10e11 -sigma 3 -scale 1 \
3 -bfac 0 -mul 1.0 -add 0.0 lyso020.ref lyso020.ave
```

Lines 2-3 This is exactly the same as Example 1. The defaults for the options have been explicitly stated however.

Line 2 The backslash \ is the Unix continuation character. On VMS you would use a dash -.

Example 3

```
1 # Example 3
2 dtscalemerge lyso.head -cycles 30 -sigma 3 -mul 1.3 -add 0.03 \
3 -reject 3 lyso020.ref lyso020.ave
```

Lines 2-3 The error model has been changed so that the input standard deviations are adjusted to be larger. Only reflections with intensities greater than 3 standard deviations (before applying the E_{mul} and E_{add} factors) are used in refining the scale factors. Reflections are rejected which differ by more than 3 standard deviations from the weighted average.

4.3 Results

This section describes in detail a sample output produced by dtscalemerge. For this example, a scan of 20 images each 2.5 degrees in rotation width of a crystal of hen egg white lysozyme was integrated. The twenty reflection lists were merged with dtreflnmerge. The command script was as follows:

```
1 dtscalemerge lyso.head -mul 1.3 -add 0.02 -sigma 3 -reject 3.0 \
2 -fix L002 -cycles 10 -ref lys.una \
3 lysraxis20.ref lys.ave
```

This script changed the error model to multiply standard deviations by 1.3 and add in another 2% systematic error. Only reflections with $I/\sigma_I > 3$ were used in the scale factor refinement. Reflections which deviated by more than 3σ from the mean were rejected. The scaled, but unaveraged reflections, are written to lys.una, while the scaled and averaged unique reflections are written to lys.ave. The complete output without annotation is found in **Appendix A**. Below is the output with intervening annotations.

```
1 dtscalmerge: Copyright (c) 1996 Molecular Structure Corporation
2 Header of file lyso.head successfully read.
3 Creflnlist::nRead called.
4 Names of reflection fields:
5 nH (int)
6 nK (int)
7 nL (int)
8 nPartial (int)
9 nBatchIndex (int)
10 nPackedHKL (int)
11 nReducedH (int)
12 nReducedK (int)
13 nReducedL (int)
14 nAnomFlag (int)
15 nCentPhase (int)
16 fIntensity (float)
17 fSigmaI (float)
18 fObs_pixel0 (float)
19 fObs_pixel1 (float)
20 fObs_rot_mid (float)
21 f2STLsq (float)
22 fResolution (float)
23 sBatch (CString)
24 INFO in Creflnlist::nRead, EOF after 31914 reflections read in (31914 total now
25 in list).
```

Lines 1-25 The input reflection list is read in and the fieldnames in the reflection list are listed. Note that for efficiency, the fields that would normally be added by **dtscalmerge** are already present in the reflection list. If they had not been present, they would have been added, but this would take more time.

```
27 Cscalmerge listing:
28 Batch fixed: L002
29 Scale fixed: 1
30 Bvalue fixed: 0
31 Rej criteria: 3
32 Mul sig fact: 1.3
33 Add sig fact: 0.02
34 Max cycles: 10
35 FixB Flag: 0
36 Anom Flag: 0
```

Lines 27-36 The option settings used by this run of **dtscalmerge** are listed. These confirm what we set on the command line.

```
37 Crystal listing:
38 Unit cell lengths: 79.2, 79.2, 38.9
39 Unit cell angles: 90, 90, 90
40 Unit cell volume: 244006
41 Spacegroup number: 96
42 name: P43212
43 Num. equiv. posns: 8
```

Lines 37-43 Information about the crystal unit cell dimensions and spacegroup is listed next. This comes from the `input_header_file` which in this example is `lyso.head`.

```
44 There are 20 different batches in the input reflection list.
```

Line 44 20 different batch names were found in the input reflection list.

```
45 Observed position limits of the Batches
46 -----
47 Batch      Num      fObs_pixel0      fObs_pixell      fObs_rot_mid
48 name refs      Min      Max      Min      Max      Min      Max
49 -----
50 L001  1603      -96.0      95.2      -91.6      85.8      -28.8      -28.8
51 L002  1625      -95.2      95.0      -88.9      90.0      -26.3      -26.3
...
68 L019  1564      -96.0      94.9      -94.8      93.9      16.3      16.3
69 L020  1590      -95.7      94.8      -91.4      93.1      18.8      18.8
70 -----
```

Lines 45-70 A table lists for each batch the minimum and maximum value for the observed reflection positions found in the input reflection list file. In this example, the `fObs_pixel0` and `fObs_pixell` fields are in millimeters and not pixels. This table is a quick check that the input reflection list contains what you think it does.

```
71 Intensity and Resolution limits of the Batches
72 -----
73 Batch      Num      Intensity      Resolution      [2sinT/lam]^2
74 name refs      Min      Max      Min      Max      Min      Max
75 -----
76 L001  1603      -26.7      11964.6      39.60      1.94      0.0006      0.2646
77 L002  1625      -16.5      13621.4      79.20      1.95      0.0002      0.2642
...
94 L019  1564      -26.6      14997.3      27.75      1.94      0.0013      0.2653
95 L020  1590      -33.7      12513.4      27.75      1.94      0.0013      0.2647
96 -----
```

Lines 71-96 The minimum and maximum intensity, resolution and $|d^*|^2 \equiv (2\sin\theta/\lambda)^2$ for each batch is listed in this table. Check that the minimum intensity of each batch is not outrageously less than zero; this would indicate problems with integration. Confirm that the resolution limits are what you expected.

```
97  Sorting and reducing reflnlist to asymmetric unit ...
98  ... done sorting
```

Lines 97-98 The reflection list was reduced to the asymmetric unit and sorted. The input list need not be sorted as this step will ensure that symmetry-related reflections are adjacent in the file.

```
99  Selection string: -fIntensity/fSigmaI<3.000
100 Number of reflns which match above selection: 5294
```

Lines 99-100 This confirms that you excluded reflections with $I/\sigma < 3$ from contributing to the refinement of the scale factors. 5294 of the input reflections were thus excluded. These reflections are included in the final statistics however.

```
101 Last cycle: no shifts.
102 For cycle number 10
```

Lines 101-102 The last cycle was reached, so no shifts were applied. Dtscalemerge calculates and prints out statistics on the results.

```
103 Method 1      Expected Rmerge:    0.033
104 Method 2      Expected Rmerge:    0.034
105              Actual   Rmerge:    0.032
```

Lines 103-105 Two methods for approximating the *Rmerge* are given, followed by the actual *Rmerge* (Eqn 1) in the data set. The actual *Rmerge* should be close to the expected *Rmerge* or you have something wrong with your error model. The expected *Rmerge*'s are calculated as follows:

Method 1: $Rmerge = 1/\sqrt{\overline{I}_h}$, where \overline{I}_h is the average intensity for all averaged reflections.

Method 2: $Rmerge = \frac{\overline{\sigma}_h}{\overline{I}_h} \sqrt{\frac{N_h}{N_i}}$, where $\overline{\sigma}_h$ is the average standard deviation for averaged reflections, N_h is the number of unique reflections, and N_i is the number of overlaps.

106	Multiplicity of observed reflections									
107	-----									
108	Mult	1	2	3	4	5	6	7	8	>8
109	-----									
110	Refs	897	2720	3418	2628	2060	1492	500	142	20
111	-----									
112	*Reflections with a multiplicity of 1 are not used in									
113	scale factor refinement nor in Rmerge calculations.									

Lines 106-113 The multiplicity of observed reflections is listed in a table. The entire process of refining scale factors requires multiple observations for reflections. That is, reflections that are symmetry-related need to appear in multiple batches. Reflections with a multiplicity of 1 are used in scale factor refinement nor in the *Rmerge* calculations. A high redundancy will improve the I/σ in the merged and averaged results (see e.g. Eqn 10).

114	Overlaps among scaling batches										
115	-----										
116	Batch	L001	L002	L003	L004	L005	L006	L007	L008	L009	L010
117	-----										
118	L001	298	745	671	259	196	168	199	174	182	263 L001
119	L002	787	659	286	243	193	162	204	166	161	280 L002
120	L003	656	226	240	246	217	173	162	162	191	259 L003
...											
163	L020	137	172	151	166	178	166	198	169	191	134 L020
164	-----										
165	Batch	L011	L012	L013	L014	L015	L016	L017	L018	L019	L020
166	-----										

Lines 114-166 Overlaps among the scaling batches are listed next. As just stated, the entire refinement of scale factors depends on overlaps among the different batches. If a batch has no overlaps with the other batches, then there is no information present in the reflection list which can be used to refine the scale factor of that batch. In this case, the procedure used in the non-linear least squares algorithm will simply keep the scale factor for that batch unchanged. Note that this table is normally **NOT** symmetric. For example, there are 745 reflections in batch L001 that have at least 1 symmetry-related reflection in batch L002. But there are 787 reflections in batch L002 that have symmetry mates in L001. There are 298 reflections in L001 which overlap with a symmetry-related reflection in the same batch (each reflection is counted once). To make this excruciatingly clear, here is another example. Suppose batch x001 has 10 copies of a reflection with $hkl = (10\ 1\ 2)$ and batch x002 has just 1 copy. Then the table looks like:

	Batch	x001	x002

	x001	10	10
	x002	1	0

If a batch has few overlaps, you might use dtreflnmerge or an editor to combine that batch with another batch that you think will have the same scale factors (i.e. a batch that is adjacent in rotation angle or time scan; see section 4.4).

167	Reflections in input file					
168	-----					
169	Batch	Num	Num	Num	Num	Num
170	name	refs	excluded	rejs	ovlps	singles
171	-----					
172	L001	1603	206	53	1327	17
173	L002	1625	227	40	1341	17
174	L003	1609	238	34	1310	27
175	L004	1574	238	25	1280	31
...						
191	L020	1590	268	28	1152	142
192	-----					
193	All batches	31914	5294	585	25138	897

Lines 167-193 Some information about reflections in the batches is listed in a table.

Num refs Total number of reflections in this batch
Num excluded Number of reflection excluded because of the I/σ cutoff.
Num rejs Number of rejected reflections. There should be very few, if any rejected reflections. Any batch with a large number of rejected reflections needs investigation. Perhaps it is indexed differently from the other batches.
Num ovlps Number of reflections in this batch which overlap another reflection either in this batch or another batch.
Num singles Number of reflections with no overlaps in the entire dataset. You do not want many of these as you have only one observation, instead of many, for this datapoint and that is not very scientific.

194	Refined scale factors					
195	-----					
196	Batch	Num	Scale			
197	name	ovlps	K	*Shifts	B	*Shifts
198	-----					
199	L001	1327	1.0138	0.0000	0.0389	0.0000
200	L002	1341	1.0000	0.0000	0.0000	0.0000
201	L003	1310	0.9911	0.0000	-0.0036	0.0000
...						
216	L018	1203	1.0519	0.0000	-0.3011	0.0000
217	L019	1149	1.0593	0.0000	-0.3365	0.0000
218	L020	1152	1.0748	0.0000	-0.3388	0.0000
219	-----					
220	*Shifts are for previous cycle only!					

Lines 194-220 This table lists the refined scale factors for the batches and the number of reflections that contributed to the refinement. You should examine this to make sure that the scale factors “make sense”. In this example, a lysozyme crystal was used. We expect some radiation damage, but not alot. That is, we expect the later batches to need to be “scaled up” and have larger scale factors. Indeed this is the case. Remember that:

$$I_{corr} = I_{hi} = I_{hj} \times k_j e^{-2B_j \sin^2 \theta / \lambda^2}$$

Also, since the batches are from the same crystal and same experiment, we expect the scale to factors vary relatively smoothly. Indeed this is the case. However, if the crystal had been a plate, or larger than the beam, then we would expect larger variations due to more drastic volume changes. Finally, double check that refinement has converged. The shifts should be 0 on the last cycle.

221 In the tables below Rmerge is defined as:
222 $R_{merge} = \frac{\sum_h \sum_i |I_{hi} - \langle I_h \rangle|}{\sum_h \sum_i \langle I_h \rangle}$
223 where I_{hi} is the i th used observation for unique hkl h ,
224 and $\langle I_h \rangle$ is the mean intensity for unique hkl h .

Lines 221-225 You are reminded of the equation for *Rmerge*.

226	Rmerge vs Batch									
227	-----									
228	Batch	Average	Num	Num	Num	Num	<I/	ChiSq	Rmerge	Rmerge
229	name	counts	obs	rejs	ovlps	single	sig>	norm	batch	cumul
230	-----									
231	L001	865	1397	53	1327	17	12.4	0.387	0.036	0.036
232	L002	997	1398	40	1341	17	15.0	0.425	0.033	0.035
233	L003	865	1371	34	1310	27	11.7	0.314	0.035	0.035
234	L004	950	1336	25	1280	31	11.8	0.282	0.029	0.033
235	L005	887	1349	31	1286	32	11.7	0.292	0.032	0.033
236	L006	1024	1368	27	1310	31	12.7	0.309	0.032	0.033
237	L007	955	1382	27	1331	24	14.5	0.352	0.035	0.033
238	L008	921	1264	20	1216	28	13.3	0.293	0.031	0.033
239	L009	955	1310	25	1257	28	11.6	0.258	0.032	0.033
240	L010	989	1373	23	1319	31	12.4	0.256	0.027	0.032
241	L011	877	1229	26	1186	17	11.3	0.275	0.034	0.032
242	L012	951	1389	24	1338	27	11.9	0.258	0.030	0.032
243	L013	931	1245	21	1189	35	11.4	0.254	0.028	0.032
244	L014	933	1384	23	1308	53	11.9	0.245	0.030	0.032
245	L015	1069	1281	24	1210	47	11.3	0.279	0.032	0.032
246	L016	968	1248	31	1162	55	11.6	0.255	0.033	0.032
247	L017	912	1362	36	1264	62	11.1	0.246	0.032	0.032
248	L018	929	1314	36	1203	75	11.3	0.260	0.032	0.032
249	L019	977	1298	31	1149	118	11.0	0.283	0.032	0.032
250	L020	859	1322	28	1152	142	11.2	0.271	0.033	0.032
251	-----									
252	All batches	941	26620	585	25138	897	12.0	0.29	0.032	0.032

Lines 226-252

This table lists *Rmerge vs Batch*. You can easily tell if any batch does not match well with the other batches.

Average counts

The average counts of reflections in the batch. This includes reflections that overlap and those that do not. If you assume Poissonian counting statistics, then the expected *Rmerge* for this batch would be $1/\sqrt{\bar{I}}$. So for batch L001, $1/\sqrt{865} = 0.034$ is consistent with 0.036 found.

Num refs

Total number of reflections in this batch.

Num rejs

Number of rejected reflections. There should be very few, if any rejected reflections. Any batch with a large number of rejected reflections needs investigation. Perhaps the batch is indexed differently from the other batches.

Num ovlp

Number of overlapping reflections in this batch that have symmetry-related mates and that contributed to the *Rmerge*.

Num singles

Number of reflection with no overlaps in the entire dataset. You do not want many of these as you have only one observation, instead of many, for this datapoint and that is not very scientific.

<I/sig>

The average Intensity/SigmaI for reflections in the batch. This is for the corrected but unmerged reflections. We expected a higher <I/sig> for the averaged reflections shown in subsequent tables.

ChiSq norm

This column does not have correct values yet.

Rmerge batch

This is the *Rmerge* of the reflections which belong to this batch where \bar{I}_h is calculated from reflections from all batches.

Rmerge cumul

This is the *Rmerge* of the reflections which belong to this batch and all previous batches in the table where \bar{I}_h is calculated from reflections from all batches.

Rmerge vs Intensity/SigmaI											
Int/sigmaI	Average	Num	Num	Num	Num	<I/	ChiSq	Rmerge	Rmerge		
range	counts	obs	rejs	ovlps	mults	sig>	norm	shell	cumul		
18 - >20	1470	15510	326	15086	3407	30.9	1.225	0.029	0.029		
16 - 18	271	1567	49	1497	384	16.0	1.133	0.057	0.029		
14 - 16	206	1658	39	1584	421	14.0	1.032	0.065	0.030		
12 - 14	160	1713	49	1614	447	12.0	0.939	0.073	0.030		
10 - 12	120	1936	46	1857	536	10.0	0.852	0.088	0.031		
8 - 10	94	1648	26	1583	512	8.0	0.601	0.092	0.031		
6 - 8	70	1501	39	1370	520	6.0	0.416	0.114	0.032		
4 - 6	49	813	11	547	263	3.9	0.188	0.117	0.032		
2 - 4	---	274	0	0	0	---	---	---	0.032		
< 0 - 2	---	0	0	0	0	---	---	---	0.032		
< 0 - >20	941	26620	585	25138	6490	19.5	1.00	0.032	0.032		

Lines 253-269

This table lists *Rmerge vs Intensity/SigmaI*. You can tell at what resolution your crystal stopped diffracting.

Average counts

The average counts of reflections in the shell. This includes reflections that overlap and those that do not. If you assume Poissonian counting statistics, then the expected *Rmerge* for this shell would be $1/\sqrt{I}$. So for the shell of highest I/σ , $1/\sqrt{1470} = 0.026$ is consistent with 0.029 found. In the 4-6 shell, $1/\sqrt{49} = 0.142$ is consistent with 0.117 found.

- Num obs Number of reflections in this shell.
- Num rejs Number of rejected reflections. There should be very few, if any rejected reflections. Any shell with a large number of rejected reflections needs investigation. Perhaps there are alot of saturated reflections.
- Num ovlps Number of reflections in this shell that have symmetry-related mates and that contributed to the *Rmerge*.
- Num mults Number of unique reflections measured multiple times in the shell. The average multiplicity or redundancy for the shell is $\text{Num ovlps} / \text{Num mults}$.
- <I/sig> The average Intensity/SigmaI for averaged reflections in the shell. As expected, we have a larger <I/sig> for the averaged reflections than shown for unaveraged reflections in the *Rmerge vs Batch* table.
- ChiSq norm This is the normalized χ^2 popularized by Dr. Zbyszek Otwinowski calculated as

$$|\chi^2| = \frac{\sum_n \sum_i^{N_{hi}} w_{hi} (I_{hi} - \bar{I}_h)^2 \frac{N_{hi} - 1}{N_{hi}}}{N_h} \quad (11)$$

where N_h is the number of unique reflections and N_{hi} is the number of contributors to or multiplicity of averaged reflection \bar{I}_h . Values close to 1 suggest that your error model is valid.

Rmerge shell

This is the *Rmerge* for reflections which belong to this shell.

Rmerge cumul

This is the cumulative *Rmerge* for reflections which belong to this shell and all previous shells. You can see what your *Rmerge* would be if you cut off your data at different I/σ levels.

270	Rmerge vs Resolution										
271	-----										
272	Resolution	Average	Num	Num	Num	Num	<I/	ChiSq	Rmerge	Rmerge	
273	range	counts	obs	rejs	ovlps	mults	sig>	norm	shell	cumul	
274	-----										
275	79.2 - 6.12	2260	278	31	228	64	21.2	0.797	0.042	0.042	
276	6.12 - 4.34	2346	1564	44	1468	351	28.2	0.945	0.029	0.031	
277	4.34 - 3.54	3147	2196	37	2095	492	33.7	0.934	0.024	0.027	
278	3.54 - 3.07	2102	2586	39	2482	603	32.0	1.014	0.026	0.026	
279	3.07 - 2.74	1055	2963	55	2836	705	27.5	1.086	0.030	0.027	
280	2.74 - 2.51	597	3137	69	2982	747	22.7	1.125	0.036	0.028	
281	2.51 - 2.32	382	3321	74	3170	834	17.6	0.999	0.045	0.029	
282	2.32 - 2.17	278	3378	76	3199	832	14.0	1.028	0.057	0.030	
283	2.17 - 2.05	192	3204	72	3002	813	11.0	0.999	0.071	0.031	
284	2.05 - 1.94	118	3993	88	3676	1049	8.4	0.890	0.090	0.032	
285	-----										
286	79.2 - 1.94	941	26620	585	25138	6490	19.5	1.00	0.032	0.032	

Lines 270-286

This table lists *Rmerge vs Resolution*

Resolution range

The resolution range of the shell in Ångstroms.

Average counts

The average counts of reflections in the shell. This includes reflections that overlap and those that do not. If you assume Poissonian counting statistics, then the expected *Rmerge* for this shell would be $1/\sqrt{I}$. So for the shell of highest I/σ , $1/\sqrt{2260} = 0.021$ which is inconsistent with 0.042 found. It turns out that low resolution reflections that were obscured by the asymmetric beamstop were not

rejected, hence the higher *Rmerge* for this shell. In the 2.05-1.94 shell, $1/\sqrt{118} = 0.092$ is consistent with 0.090 found.

Num obs Number of reflections in this shell.

Num rejs Number of rejected reflections. There should be very few, if any rejected reflections. Any shell with a large number of rejected reflections needs investigation. Perhaps the shell contains extraneous diffraction from ice.

Num ovlps Number of reflections in this shell that have symmetry-related mates and that contributed to the *Rmerge*.

Num mults Number of unique reflections measured multiple times in the shell. The average multiplicity or redundancy for the shell is Num ovlps / Num mults.

<I/sig> The average Intensity/SigmaI for averaged reflections in the shell. As expected, we have a larger <I/sig> for the averaged reflections than shown for unaveraged reflections in the *Rmerge vs Batch* table.

ChiSq norm This is the normalized χ^2 popularized by Dr. Zbyszek Otwinowski calculated in Eqn 11. Values close to 1 suggest that your error model is valid.

Rmerge shell This is the *Rmerge* for reflections which belong to this shell.

Rmerge cumul This is the cumulative *Rmerge* for reflections which belong to this shell and all previous shells. You can see what your *Rmerge* would be if you cut off your data at different resolution levels.

287	Completeness vs Resolution										
288	-----										
289	Resolution	Calc	Num	Num	Num	Num	Num	Avg	%Comp	%Comp	
290	range	unique	obs	rejs	mults	single	unique	mult	shell	cumul	
291	-----										
292	79.2 - 6.12	363	278	31	64	19	83	2.98	22.9	22.9	
293	6.12 - 4.34	593	1564	44	351	52	403	3.77	68.0	50.8	
294	4.34 - 3.54	736	2196	37	492	64	556	3.88	75.5	61.6	
295	3.54 - 3.07	863	2586	39	603	65	668	3.81	77.4	66.9	
296	3.07 - 2.74	966	2963	55	705	72	777	3.74	80.4	70.6	
297	2.74 - 2.51	1061	3137	69	747	86	833	3.68	78.5	72.5	
298	2.51 - 2.32	1131	3321	74	834	77	911	3.56	80.5	74.1	
299	2.32 - 2.17	1226	3378	76	832	103	935	3.53	76.3	74.4	
300	2.17 - 2.05	1290	3204	72	813	130	943	3.32	73.1	74.2	
301	2.05 - 1.94	1362	3993	88	1049	229	1278	3.06	93.8	77.0	
302	-----										
303	79.2 - 1.94	9591	26620	585	6490	897	7387	3.52	77.0	77.0	

Lines 287-303

This table lists *Completeness vs Resolution*. In order to perform subsequent crystallographic analyses, you would like to have very complete data.

Resolution range	The resolution range of the shell in Ångstroms.
Calc unique	Number of calculated unique reflections in this shell. This is calculated from the unit cell dimensions and the space group in an exact manner (it is not an approximation).
Num obs	Number of observations for reflections in this shell. The more the better.
Num rejs	Number of rejected reflections. There should be very few, if any rejected reflections. Any shell with a large number of rejected reflections needs investigation. Perhaps the shell contains extraneous diffraction from ice or low resolution reflections are behind the beamstop. (Notice that more than 10% of the observations in the lowest shell are rejected. That's bad!)
Num mults	Number of unique reflections resulting from the overlaps in the shell. In other words, the number of unique reflections measured more than once.
Num single	Number of unique reflections measured just once.
Num unique	Number of unique reflections measured (Num mults + Num single).
Avg mult	The average multiplicity or redundancy in the reflections in the shell. Simply $(\text{Num obs} - \text{Num rejs}) / \text{Num unique}$.
%Comp shell	This is the percent completeness for reflections which belong to this resolution shell: $100 \times \text{Num unique} / \text{Calc unique}$.
%Comp cumul	This is the cumulative percent completeness for reflections which belong to this shell and all previous shells. You can see what your completeness would be if you cut off your data at different resolution levels.

4.4 Further analysis

In the previous example, we used `-ref lys.una` to write out the scaled but unaveraged reflections. You can filter this file with `dtreflnmerge` either for re-scaling or for additional analysis.

Q1: How do I view rejected reflections with `dtdisplay`?

A: Use `dtreflnmerge` to place the rejected reflections in their own reflection list. Then in `dtdisplay` use `File Load reflnlist...`

```
1a dtreflnmerge lys.una -fSigmaI>=0 rejects.ref
```

Line 1a All good reflections ($\sigma_I \geq 0$) are excluded from the output, leaving only rejects.

This next example gets rejected reflections from a particular batch: L001.

```
1b dtreflnmerge lys.una -fSigmaI>=0 -sBatch!=L001 1001.rej
```

Line 1b All good reflections ($\sigma_I \geq 0$) are excluded from the output, leaving only rejects. Then all reflections from batches except L001 are excluded.

Q2: What are the intensities of systematically absent reflections?

A: Use `dtreflnmerge` to copy only systematically absent reflections to their own reflection list.

```
2 dtreflnmerge lys.una -nCentPhase!=-1 absent.ref
```

Line 2 From the Appendix of the `dtreflnmerge` documentation, we know that if `nCentPhase` is -1, then the reflection is systematically absent. So this filter, excludes all but the absent reflections and writes them to `absent.ref`.

Q3: How do I combine batches?

A: Use `dtreflnmerge` (or an editor) to change the batch names to the same name.

```
3a dtreflnmerge lysraxis20.ref -sBatch!=L001 +sBatch==L002 \  
3b -sBatch=L001+2 \  
3c lysraxis20.ref -sBatch==L001 -sBatch==L002 \  
3d lysraxisnew.ref
```

Line 3a Read in the reflection list and exclude all except reflections from batch L001. Then include reflections from batch L002 which were just excluded.

- Line 3b Give remaining reflections the batch name L001+2.
Line 3c Read in the reflection list and exclude all except reflections from batches
 L001 and L002.
Line 3d Write the modified reflection list

Since the reflection list is an ASCII file, the above could also be done with a simple text editor.

Appendix A Example output listing

Below is the complete unabridged output listing of the example that was annotated in 4.3.

```
1 dtscalmerge: Copyright (c) 1996 Molecular Structure Corporation
2 Header of file lyso.head successfully read.
3 Creflnlist::nRead called.
4 Names of reflection fields:
5 nH (int)
6 nK (int)
7 nL (int)
8 nPartial (int)
9 nBatchIndex (int)
10 nPackedHKL (int)
11 nReducedH (int)
12 nReducedK (int)
13 nReducedL (int)
14 nAnomFlag (int)
15 nCentPhase (int)
16 fIntensity (float)
17 fSigmaI (float)
18 fObs_pixel0 (float)
19 fObs_pixell (float)
20 fObs_rot_mid (float)
21 f2STLsq (float)
22 fResolution (float)
23 sBatch (CString)
24 INFO in Creflnlist::nRead, EOF after 31914 reflections read in (31914 total now
25 in list).
26
27 Cscalmerge listing:
28 Batch fixed: L002
29 Scale fixed: 1
30 Bvalue fixed: 0
31 Rej criteria: 3
32 Mul sig fact: 1.3
33 Add sig fact: 0.02
34 Max cycles: 10
35 FixB Flag: 0
36 Anom Flag: 0
37 Crystal listing:
38
39 Unit cell lengths: 79.2, 79.2, 38.9
40 Unit cell angles: 90, 90, 90
41 Unit cell volume: 244006
42
43 Spacegroup number: 96
44 name: P43212
45 Num. equiv. posns: 8
46
47 There are 20 different batches in the input reflection list.
48
49 Observed position limits of the Batches
50 -----
51 Batch Num fObs_pixel0 fObs_pixell fObs_rot_mid
52 name refs Min Max Min Max Min Max
53 -----
54 L001 1603 -96.0 95.2 -91.6 85.8 -28.8 -28.8
55 L002 1625 -95.2 95.0 -88.9 90.0 -26.3 -26.3
56 L003 1609 -96.0 95.0 -90.4 93.7 -23.8 -23.8
57 L004 1574 -95.9 94.9 -91.9 90.6 -21.3 -21.3
58 L005 1591 -95.3 95.0 -90.6 93.1 -18.8 -18.8
59 L006 1641 -94.3 95.0 -92.5 89.4 -16.3 -16.3
60 L007 1640 -95.8 95.0 -93.6 94.4 -13.8 -13.8
```

57	L008	1556	-95.1	94.8	-93.9	91.8	-11.3	-11.3
58	L009	1606	-94.5	95.1	-89.6	92.9	-8.8	-8.8
59	L010	1639	-94.9	94.8	-89.3	93.7	-6.2	-6.2
60	L011	1504	-95.3	94.9	-93.5	94.7	-3.8	-3.8
61	L012	1665	-95.5	94.9	-93.7	94.0	-1.2	-1.2
62	L013	1511	-94.3	94.9	-89.9	90.6	1.2	1.2
63	L014	1642	-95.0	94.8	-91.9	91.4	3.8	3.8
64	L015	1575	-95.5	94.9	-94.0	92.3	6.2	6.2
65	L016	1546	-95.7	95.1	-91.0	94.1	8.8	8.8
66	L017	1634	-95.2	94.7	-94.2	88.9	11.3	11.3
67	L018	1599	-95.1	94.8	-91.4	94.6	13.8	13.8
68	L019	1564	-96.0	94.9	-94.8	93.9	16.3	16.3
69	L020	1590	-95.7	94.8	-91.4	93.1	18.8	18.8
70	-----							

71 Intensity and Resolution limits of the Batches

72	-----							
73	Batch	Num	Intensity		Resolution		[2sinT/lam]^2	
74	name	refs	Min	Max	Min	Max	Min	Max
75	-----							
76	L001	1603	-26.7	11964.6	39.60	1.94	0.0006	0.2646
77	L002	1625	-16.5	13621.4	79.20	1.95	0.0002	0.2642
78	L003	1609	-29.3	13773.7	19.80	1.94	0.0026	0.2647
79	L004	1574	-23.8	21598.9	35.42	1.95	0.0008	0.2639
80	L005	1591	-21.6	17330.9	35.42	1.94	0.0008	0.2650
81	L006	1641	-28.7	18304.0	28.00	1.94	0.0013	0.2650
82	L007	1640	-26.0	14175.1	56.00	1.94	0.0003	0.2650
83	L008	1556	-35.3	18142.3	56.00	1.94	0.0003	0.2650
84	L009	1606	-21.3	18305.0	21.97	1.94	0.0021	0.2650
85	L010	1639	-16.1	14213.1	17.22	1.94	0.0034	0.2647
86	L011	1504	-21.7	17216.5	19.13	1.94	0.0027	0.2650
87	L012	1665	-34.5	14162.0	19.13	1.95	0.0027	0.2642
88	L013	1511	-21.1	24077.9	35.42	1.94	0.0008	0.2653
89	L014	1642	-26.0	15128.2	17.65	1.94	0.0032	0.2650
90	L015	1575	-26.8	18961.3	21.06	1.94	0.0023	0.2648
91	L016	1546	-25.3	16982.7	22.73	1.94	0.0019	0.2650
92	L017	1634	-22.0	17017.0	26.19	1.94	0.0015	0.2650
93	L018	1599	-17.9	18772.2	26.19	1.94	0.0015	0.2648
94	L019	1564	-26.6	14997.3	27.75	1.94	0.0013	0.2653
95	L020	1590	-33.7	12513.4	27.75	1.94	0.0013	0.2647
96	-----							

97 Sorting and reducing reflnlist to asymmetric unit ...
98 ... done sorting

99 Selection string: -fIntensity/fSigmaI<3.000
100 Number of reflns which match above selection: 5294
101 Last cycle: no shifts.
102 For cycle number 10

103 Method 1 Expected Rmerge: 0.033
104 Method 2 Expected Rmerge: 0.034
105 Actual Rmerge: 0.032

106 Multiplicity of observed reflections

107	-----										
108	Mult		1	2	3	4	5	6	7	8	>8
109	-----										
110	Refs		897	2720	3418	2628	2060	1492	500	142	20
111	-----										

112 *Reflections with a multiplicity of 1 are not used in
113 scale factor refinement nor in Rmerge calculations.

114 Overlaps among scaling batches

115	-----											
116	Batch		L001	L002	L003	L004	L005	L006	L007	L008	L009	L010

117												
118	L001	298	745	671	259	196	168	199	174	182	263	L001
119	L002	787	659	286	243	193	162	204	166	161	280	L002
120	L003	656	226	240	246	217	173	162	162	191	259	L003
121	L004	256	190	246	188	225	197	124	191	215	236	L004
122	L005	196	161	217	225	228	250	122	193	266	212	L005
123	L006	184	136	185	220	268	389	224	230	293	235	L006
124	L007	264	230	206	150	149	232	719	227	172	247	L007
125	L008	182	144	172	220	199	214	208	433	255	234	L008
126	L009	168	137	185	216	284	284	146	256	257	304	L009
127	L010	262	234	257	246	205	202	196	216	299	246	L010
128	L011	170	146	197	223	248	193	145	163	224	253	L011
129	L012	289	226	262	239	221	236	198	183	226	267	L012
130	L013	180	151	200	224	214	172	128	144	171	228	L013
131	L014	285	247	264	246	219	197	165	173	186	214	L014
132	L015	240	194	225	211	214	197	144	133	161	179	L015
133	L016	228	184	240	219	208	158	141	130	146	172	L016
134	L017	287	229	248	230	229	190	154	153	152	161	L017
135	L018	211	175	225	209	199	179	138	125	144	152	L018
136	L019	195	172	197	206	209	167	129	151	163	178	L019
137	L020	228	185	208	202	182	173	159	151	157	156	L020
138												
139	Batch	L001	L002	L003	L004	L005	L006	L007	L008	L009	L010	
140												

141												
142	Batch	L011	L012	L013	L014	L015	L016	L017	L018	L019	L020	
143												
144	L001	192	299	197	306	249	241	301	214	208	245	L001
145	L002	180	293	206	321	252	236	302	211	222	238	L002
146	L003	202	259	213	284	225	245	261	232	210	221	L003
147	L004	223	244	228	255	219	224	237	206	214	207	L004
148	L005	255	222	228	226	226	220	232	208	219	190	L005
149	L006	212	248	206	220	231	183	223	201	198	191	L006
150	L007	195	257	177	214	189	187	208	179	176	207	L007
151	L008	182	211	164	196	162	152	173	149	174	186	L008
152	L009	214	225	175	193	171	158	162	155	185	179	L009
153	L010	260	255	238	212	190	179	188	163	188	167	L010
154	L011	190	226	188	189	180	176	170	180	168	139	L011
155	L012	238	219	218	205	215	203	211	195	154	176	L012
156	L013	186	210	150	217	176	169	176	151	166	154	L013
157	L014	199	208	217	156	199	180	201	205	170	173	L014
158	L015	181	223	175	202	143	161	167	165	165	178	L015
159	L016	166	210	173	182	163	129	205	176	154	166	L016
160	L017	152	211	179	212	168	201	132	143	185	193	L017
161	L018	163	185	154	206	171	181	143	146	187	163	L018
162	L019	151	142	150	172	170	160	187	189	107	191	L019
163	L020	137	172	151	166	178	166	198	169	191	134	L020
164												
165	Batch	L011	L012	L013	L014	L015	L016	L017	L018	L019	L020	
166												

167 Reflections in input file

168	Batch	Num	Num	Num	Num	Num
169	name	refs	excluded	rejs	ovlps	singles
170						
171						
172	L001	1603	206	53	1327	17
173	L002	1625	227	40	1341	17
174	L003	1609	238	34	1310	27
175	L004	1574	238	25	1280	31
176	L005	1591	242	31	1286	32
177	L006	1641	273	27	1310	31
178	L007	1640	258	27	1331	24
179	L008	1556	292	20	1216	28
180	L009	1606	296	25	1257	28
181	L010	1639	266	23	1319	31
182	L011	1504	275	26	1186	17

183	L012	1665	276	24	1338	27
184	L013	1511	266	21	1189	35
185	L014	1642	258	23	1308	53
186	L015	1575	294	24	1210	47
187	L016	1546	298	31	1162	55
188	L017	1634	272	36	1264	62
189	L018	1599	285	36	1203	75
190	L019	1564	266	31	1149	118
191	L020	1590	268	28	1152	142
192	-----					
193	All batches	31914	5294	585	25138	897

194 Refined scale factors

195	-----					
196	Batch	Num	Scale			
197	name	ovlps	K	*Shifts	B	*Shifts
198	-----					
199	L001	1327	1.0138	0.0000	0.0389	0.0000
200	L002	1341	1.0000	0.0000	0.0000	0.0000
201	L003	1310	0.9911	0.0000	-0.0036	0.0000
202	L004	1280	0.9978	0.0000	-0.0517	0.0000
203	L005	1286	0.9906	0.0000	-0.0543	0.0000
204	L006	1310	1.0071	0.0000	-0.0754	0.0000
205	L007	1331	1.0101	0.0000	-0.0940	0.0000
206	L008	1216	1.0135	0.0000	-0.1218	0.0000
207	L009	1257	1.0124	0.0000	-0.1855	0.0000
208	L010	1319	1.0183	0.0000	-0.1781	0.0000
209	L011	1186	1.0122	0.0000	-0.2058	0.0000
210	L012	1338	1.0272	0.0000	-0.2248	0.0000
211	L013	1189	1.0333	0.0000	-0.2345	0.0000
212	L014	1308	1.0353	0.0000	-0.2046	0.0000
213	L015	1210	1.0393	0.0000	-0.3008	0.0000
214	L016	1162	1.0590	0.0000	-0.3258	0.0000
215	L017	1264	1.0438	0.0000	-0.2858	0.0000
216	L018	1203	1.0519	0.0000	-0.3011	0.0000
217	L019	1149	1.0593	0.0000	-0.3365	0.0000
218	L020	1152	1.0748	0.0000	-0.3388	0.0000
219	-----					

220 *Shifts are for previous cycle only!

221 In the tables below Rmerge is defined as:

$$222 \text{ Rmerge} = \frac{\sum_i |I_{hi} - \langle I_h \rangle|}{\sum_i I_{hi}}$$

223 where I_{hi} is the i th used observation for unique hkl h ,
224 and $\langle I_h \rangle$ is the mean intensity for unique hkl h .

226 Rmerge vs Batch

227	-----									
228	Batch	Average	Num	Num	Num	Num	<I/	ChiSq	Rmerge	Rmerge
229	name	counts	obs	rejs	ovlps	single	sig>	norm	batch	cumul
230	-----									
231	L001	865	1397	53	1327	17	12.4	0.387	0.036	0.036
232	L002	997	1398	40	1341	17	15.0	0.425	0.033	0.035
233	L003	865	1371	34	1310	27	11.7	0.314	0.035	0.035
234	L004	950	1336	25	1280	31	11.8	0.282	0.029	0.033
235	L005	887	1349	31	1286	32	11.7	0.292	0.032	0.033
236	L006	1024	1368	27	1310	31	12.7	0.309	0.032	0.033
237	L007	955	1382	27	1331	24	14.5	0.352	0.035	0.033
238	L008	921	1264	20	1216	28	13.3	0.293	0.031	0.033
239	L009	955	1310	25	1257	28	11.6	0.258	0.032	0.033
240	L010	989	1373	23	1319	31	12.4	0.256	0.027	0.032
241	L011	877	1229	26	1186	17	11.3	0.275	0.034	0.032
242	L012	951	1389	24	1338	27	11.9	0.258	0.030	0.032
243	L013	931	1245	21	1189	35	11.4	0.254	0.028	0.032
244	L014	933	1384	23	1308	53	11.9	0.245	0.030	0.032

245	L015	1069	1281	24	1210	47	11.3	0.279	0.032	0.032
246	L016	968	1248	31	1162	55	11.6	0.255	0.033	0.032
247	L017	912	1362	36	1264	62	11.1	0.246	0.032	0.032
248	L018	929	1314	36	1203	75	11.3	0.260	0.032	0.032
249	L019	977	1298	31	1149	118	11.0	0.283	0.032	0.032
250	L020	859	1322	28	1152	142	11.2	0.271	0.033	0.032
251	-----									
252	All batches	941	26620	585	25138	897	12.0	0.29	0.032	0.032

253 Rmerge vs Intensity/SigmaI

254	-----										
255	Int/sigmaI	Average	Num	Num	Num	Num	<I/	ChiSq	Rmerge	Rmerge	
256	range	counts	obs	rejs	ovlps	mults	sig>	norm	shell	cumul	
257	-----										
258	18 - >20	1470	15510	326	15086	3407	30.9	1.225	0.029	0.029	
259	16 - 18	271	1567	49	1497	384	16.0	1.133	0.057	0.029	
260	14 - 16	206	1658	39	1584	421	14.0	1.032	0.065	0.030	
261	12 - 14	160	1713	49	1614	447	12.0	0.939	0.073	0.030	
262	10 - 12	120	1936	46	1857	536	10.0	0.852	0.088	0.031	
263	8 - 10	94	1648	26	1583	512	8.0	0.601	0.092	0.031	
264	6 - 8	70	1501	39	1370	520	6.0	0.416	0.114	0.032	
265	4 - 6	49	813	11	547	263	3.9	0.188	0.117	0.032	
266	2 - 4	---	274	0	0	0	---	---	---	0.032	
267	< 0 - 2	---	0	0	0	0	---	---	---	0.032	
268	-----										
269	< 0 - >20	941	26620	585	25138	6490	19.5	1.00	0.032	0.032	

270 Rmerge vs Resolution

271	-----										
272	Resolution	Average	Num	Num	Num	Num	<I/	ChiSq	Rmerge	Rmerge	
273	range	counts	obs	rejs	ovlps	mults	sig>	norm	shell	cumul	
274	-----										
275	79.2 - 6.12	2260	278	31	228	64	21.2	0.797	0.042	0.042	
276	6.12 - 4.34	2346	1564	44	1468	351	28.2	0.945	0.029	0.031	
277	4.34 - 3.54	3147	2196	37	2095	492	33.7	0.934	0.024	0.027	
278	3.54 - 3.07	2102	2586	39	2482	603	32.0	1.014	0.026	0.026	
279	3.07 - 2.74	1055	2963	55	2836	705	27.5	1.086	0.030	0.027	
280	2.74 - 2.51	597	3137	69	2982	747	22.7	1.125	0.036	0.028	
281	2.51 - 2.32	382	3321	74	3170	834	17.6	0.999	0.045	0.029	
282	2.32 - 2.17	278	3378	76	3199	832	14.0	1.028	0.057	0.030	
283	2.17 - 2.05	192	3204	72	3002	813	11.0	0.999	0.071	0.031	
284	2.05 - 1.94	118	3993	88	3676	1049	8.4	0.890	0.090	0.032	
285	-----										
286	79.2 - 1.94	941	26620	585	25138	6490	19.5	1.00	0.032	0.032	

287 Completeness vs Resolution

288	-----										
289	Resolution	Calc	Num	Num	Num	Num	Num	Avg	%Comp	%Comp	
290	range	unique	obs	rejs	mults	single	unique	mult	shell	cumul	
291	-----										
292	79.2 - 6.12	363	278	31	64	19	83	2.98	22.9	22.9	
293	6.12 - 4.34	593	1564	44	351	52	403	3.77	68.0	50.8	
294	4.34 - 3.54	736	2196	37	492	64	556	3.88	75.5	61.6	
295	3.54 - 3.07	863	2586	39	603	65	668	3.81	77.4	66.9	
296	3.07 - 2.74	966	2963	55	705	72	777	3.74	80.4	70.6	
297	2.74 - 2.51	1061	3137	69	747	86	833	3.68	78.5	72.5	
298	2.51 - 2.32	1131	3321	74	834	77	911	3.56	80.5	74.1	
299	2.32 - 2.17	1226	3378	76	832	103	935	3.53	76.3	74.4	
300	2.17 - 2.05	1290	3204	72	813	130	943	3.32	73.1	74.2	
301	2.05 - 1.94	1362	3993	88	1049	229	1278	3.06	93.8	77.0	
302	-----										
303	79.2 - 1.94	9591	26620	585	6490	897	7387	3.52	77.0	77.0	