Summary of scaling algorithm development

James Beilsten-Edmands

September 1, 2017

1 General procedure

Evans (Acta D 62 pt 1, 72-82 (2006)) describes the general procedure for scaling symmetry-related observations. The inverse scaling factors g_{hl} are determined by minimising the function

$$\Phi = \sum_h \sum_l \frac{1}{\sigma^2_{hl}} (I_{hl} - g_{hl} \langle I_h \rangle)^2 + \text{parameter restraint terms.} \tag{1}$$

I_{hl} is the l^{th} observation of unique reflection h and $\langle I_h \rangle$ is the weighted average intensity for all observations of unique reflection h;

$$\langle I_h \rangle = (\sum_l g_{hl} I_{hl} / \sigma^2_{hl}) / (\sum_l g^2_{hl} / \sigma^2_{hl}). \tag{2}$$

1.1 Calculation of gradient

For a minimisation procedure, one needs to consider the derivative of the cost function with respect to model parameters p_i;

$$\frac{\partial \Phi}{\partial p_i} = \sum_h \sum_l 2 r_{hl} \frac{\partial r_{hl}}{\partial p_i}, \tag{3}$$

where

$$r_{hl} = \frac{1}{\sigma_{hl}} (I_{hl} - g_{hl} \langle I_h \rangle). \tag{4}$$

Calculating the derivative gives

$$\frac{\partial r_{hl}}{\partial p_i} = -\frac{1}{\sigma_{hl}} (\langle I_h \rangle \frac{\partial g_{hl}}{\partial p_i} + g_{hl} \frac{\partial \langle I_h \rangle}{\partial p_i}). \tag{5}$$

The partial derivative of $\langle I_h \rangle$ can be calculated by the quotient rule

$$\frac{\partial \langle I_h \rangle}{\partial p_i} = \frac{\partial}{\partial p_i} (\frac{u}{v}) = \frac{v u' - u v'}{v^2}, \tag{6}$$

hence

$$\frac{\partial\langle I_h\rangle}{\partial p_i} = \frac{(\sum_l g_{hl}^2/\sigma_{hl}^2)(\sum_l \frac{I_{hl}}{\sigma_{hl}^2}\frac{\partial g_{hl}}{\partial p_i}) - (\sum_l g_{hl}I_{hl}/\sigma_{hl}^2)(\sum_l \frac{2g_{hl}}{\sigma_{hl}^2}\frac{\partial g_{hl}}{\partial p_i})}{(\sum_l g_{hl}^2/\sigma_{hl}^2)^2} \tag{7}$$

which can be simplified by factoring out 'v' to

$$\frac{\partial\langle I_h\rangle}{\partial p_i} = \frac{(\sum_l \frac{I_{hl}}{\sigma_{hl}^2}\frac{\partial g_{hl}}{\partial p_i}) - \langle I_h\rangle(\sum_l \frac{2g_{hl}}{\sigma_{hl}^2}\frac{\partial g_{hl}}{\partial p_i})}{(\sum_l g_{hl}^2/\sigma_{hl}^2)}. \tag{8}$$

hence the derivative of the residual is given by [1]

$$\frac{\partial r_{hl}}{\partial p_i} = -\frac{\langle I_h\rangle}{\sigma_{hl}}\frac{\partial g_{hl}}{\partial p_i} - \frac{g_{hl}}{\sigma_{hl}}\frac{(\sum_l \frac{I_{hl}}{\sigma_{hl}^2}\frac{\partial g_{hl}}{\partial p_i}) - \langle I_h\rangle(\sum_l \frac{2g_{hl}}{\sigma_{hl}^2}\frac{\partial g_{hl}}{\partial p_i})}{(\sum_l g_{hl}^2/\sigma_{hl}^2)}. \tag{9}$$

Proceeding further is then dependent on the parameterisation of the scaling parameters.

In the Kabsch parameterisation, one determines a set of g-values g_j that are the scale factors assigned to a given resolution or phi 'bin', hence the model parameters p_i are just the g_j factors. Therefore $\frac{\partial g_{hl}}{\partial p_i} = \frac{\partial g_{hl}}{\partial g_j} = \delta(g_{hl}, g_j)$ i.e. the gradient is only non-zero if the g-parameter assigned to reflection hl is g_j (N.B. is this definitely true, is there some further dependence through $\langle I_h\rangle$?).

In a 'physical' scaling model (Evans 2006), the inverse scale factor is given by the multiplication of several factors, such as a scale factor for a given phi, a B-factor, an absorption factor etc which are themselves determined by a set of model parameters. Hence one will have to calculate how g_{hl} varies with these parameters, which will be more complex than in the Kabsch parameterisation.

1.2 Issue of degeneracy of the solution

The residual Φ is unchanged if one multiplies all of the scale factors by a constant, or by adding a constant to all of the B-factors (Evans 2006). As multiplying Φ by a constant is just changing the global scale parameter, this can easily be dealt with, for example by normalising the average g to 1, and does not affect the relative scale between reflections. It is less obvious how to deal with the B-factor term, as this changes the relative scale of reflections at different resolutions. Note that this B-factor dependence is in addition to the standard B-factor that causes a decrease in spot intensity for increasing resolution; at the start of the measurement, the standard B-factor already exists, as such one might expect that (if the true intensities were measured) the g-values as a function of resolution for the first time bin are all 1 - i.e. there is no further resolution-dependent correction encapsulated within the g factors due to radiation damage. Then, during the course of the

[1] N.B. I think this is slightly different than the expression derived by D. Waterman in 'scaling rotation datasets', where the standard deviations are only given through the weightings w_{hl} which appear to be inconsistent in the equation.

measurement, if radiation damage is present one would expect a resolution dependence to become apparent for increasing time bins. Therefore, after a minimum solution is found for the g-values, it seems that one should rescale all the g-values by a factor $\exp(B\sin^2\theta/\lambda^2)$ $= \exp(B/d^2)$, where B is chosen/fitted to bring the g-values of the first few time bins as close as possible to a constant, and d is a representative d-value for each resolution bin. N.B. There may be some degree of choice over how many time bins are chosen for this and uncertainty over a representative $1/d^2$ for each resolution bin. Then one can the divide out by this constant to normalise to one. This has the effect of setting the first scale factor to 1 and the first relative 'B'-factor to zero, similar to that described by Evans (Evans 2006).

In a paper where Kabsch describes the methodology behind the XDS algorithms (Acta D 66 pt 2, 133-144 (2010)), a parameter restraint term is used to weakly constrain the g-values to one. However, based on tests of my algorithm, including a weak parameter restraint does not find quite the same solution, and the solution depends on how tightly the values are constrained to one. Intuitively, one is imposing additional constraints on the solution, however I do not see that this is particularly necessary if one can use the 'invariant' scaling methods to bring the g-values back to physically reasonable values after they have been freely determined, even if this gives initial g-values that deviate significantly from one.

1.3 Implementation of Kabsch scaling

To implement the scaling algorithm, it is necessary to reorder and index the reflection table so that one can use a for loop to calculate the relevant quantities. This is contained within a 'data manager' object. This is the general method for reorganising the data:

- Filter out 'bad' refelctions - either using flags or selecting based on d-value, intensity etc. (currently not using flags).

- Sort reflections in the asymmetric unit, such that the data is grouped into groups of equivalent reflections.

- Bin reflections into resolution/d-bins, time/ϕ bins, detector positions etc. Create an index column in the reflection table for each type of correction (decay, modulation, absorption) and assign an index to each reflection.

- Assign a miller index 'h-index' to label each set of unique reflections. Also create an array containing information about how many reflections are in each group of equivalent miller indices.

Now that the data is ordered, calculations of the $\langle I_h \rangle$, g_l value arrays etc. can be quickly performed with a for loop over the reflection table, using the h-index/l-index to build up the relevant array. The steps of the algorithm are:

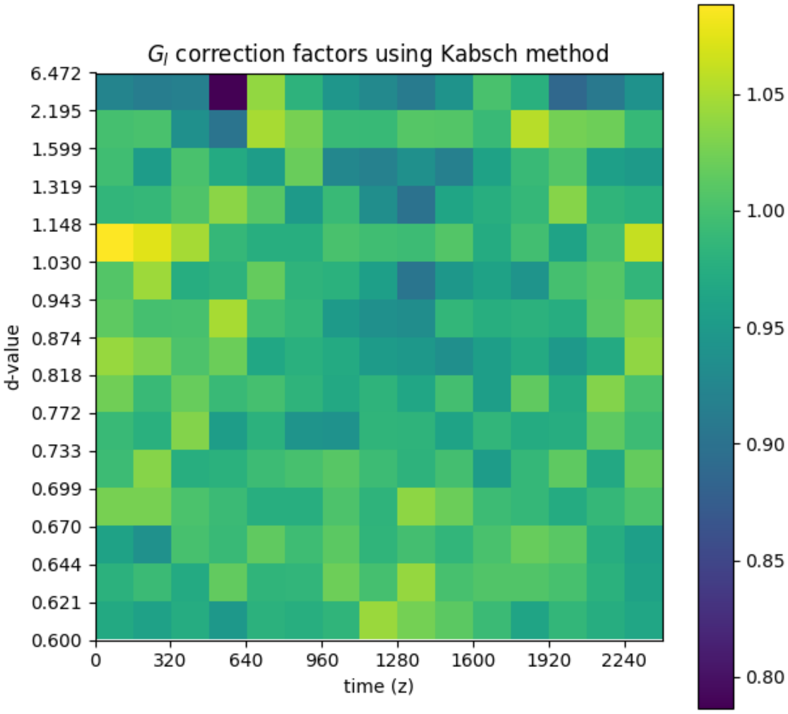

Figure 1: G_l correction factors for resolution as a function of time (z/ϕ) bin.

- Create a g-value array for each correction type (decay, modulation, absorption) and give initial values as 1.
- Use an LBFGS minimiser to minimise g-values, at each step calculating $\langle I_h \rangle$, the residuals and gradient function. Do this for the decay and modulation and absorption corrections in succession. There is potential choice here in the order- it seems to make sense to do absorption first if this is the major correction.
- Due to the redundancy in the decay g-value, do a renormalisation to an initial relative B value of 0 and scale of 1 after each 'decay' minimisation.
- The final correction factor for each reflection is then given by the product of the three correction factors.

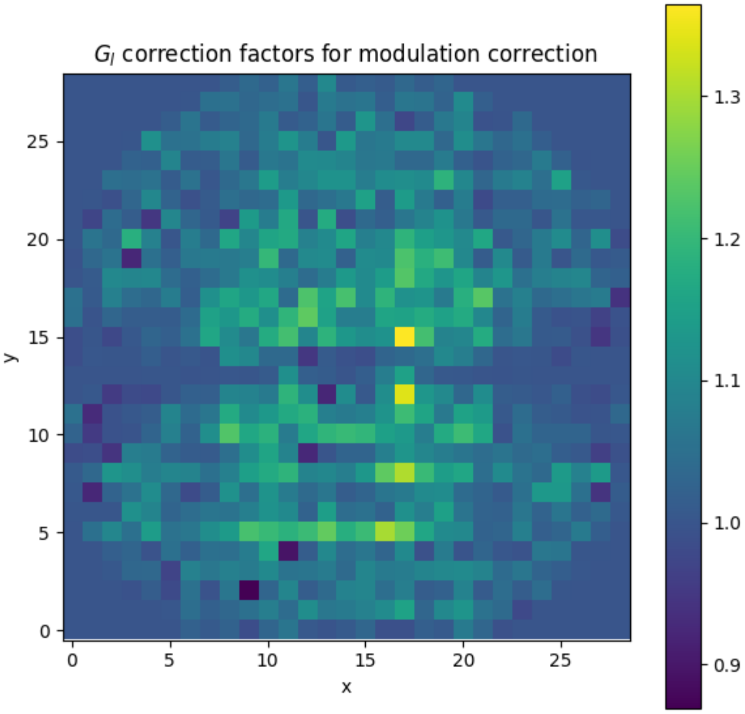

Figure 2: G_l correction factors for detector positions on a 29x29 grid.

I have now implemented basic decay, modulation and absorption corrections. The algorithm was run with three iteration cycles, giving an $R_{\text{p.i.m.}}$ of 0.082, and taking $\sim$ 50s to execute. The correction factors are shown in Figures 1, 2, 3 respectively. For the decay correction, it makes sense to bin the data into equal bins in $1/d^2$ between the max/min values. However this could leave a very wide bin at high d-value, so maybe a smart choice could be made based on the data, such as further subdivision of wide d-bins if there are sufficient reflections in the bins before and after subdivision. However there do not seems to be any obvious problems at least with this dataset. For the modulation correction, the result various significantly depending on whether one applies the LP and DQE corrections first. If not, these appear in this plot as high g factors near the centre, however if these are already accurately calculated in DIALS then maybe they should just be applied, leaving the plot shown in Figure 2.

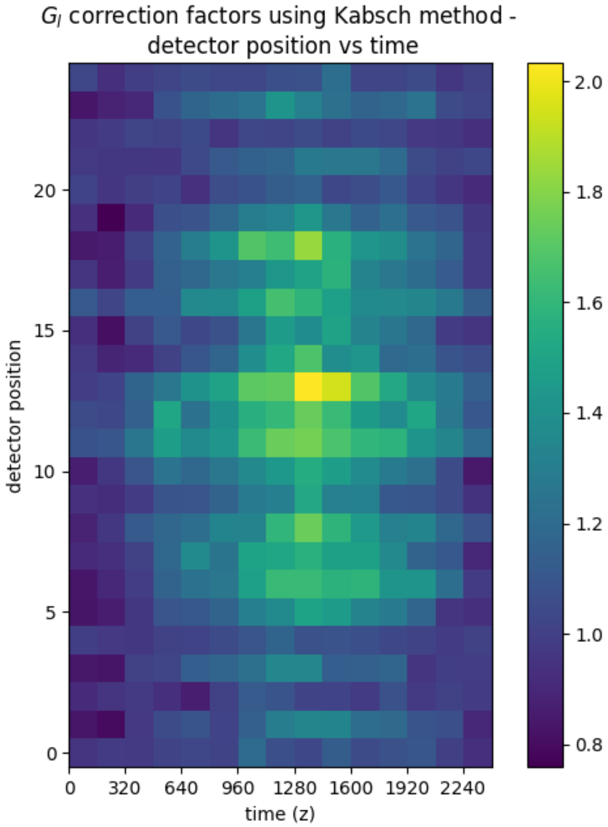

Figure 3: G_l correction factors for detector positions on a 5x5 grid as a function of time (z/ϕ) bin.

The absorption correction for a given time bin and detector position are shown in Figure 4. Kabsch recommends thirteen detector regions for the absorption correction, here I have used a 5x5 grid for simplicity. Maybe the thirteen detector regions is better as one may get less peaks at the edge of the detector? Optimum detector regions for this correction requires further investigation. I think Figure 4 is clearer than Figure 3 in showing the radial dependence of the correction factors (presumably larger at the centre as there is least absorption straight through the crystal for this data- although this could also be somewhat captured in the modulation correction?) and the general absorption modulation can be seen in the time dependence. For this dataset there seems to be significant time dependence, presumably due to absorption/illumination volume changing with rotation?

Further thoughts for consideration:

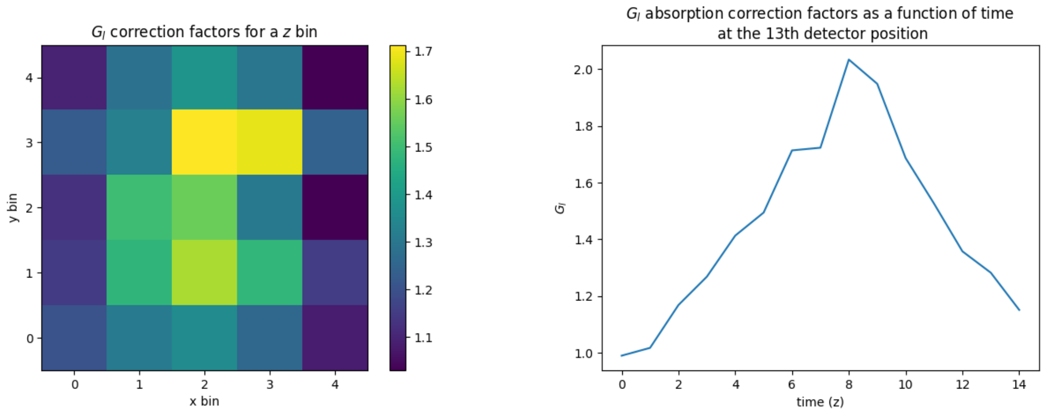

Figure 4: Absorption correction factors plotted as a function of detector position for the 7th time (z/ϕ) bin, and as a function of time for the central detector position bin.

- There is an error associated with the $\langle I_h \rangle$ values. When giving the g-values for correcting individual reflections (if the program shall provide the merged and unmerged data), should this also come with an error?
- In the code one can choose to use the summation intensities or the profile fitted intensities, however there are a large number of reflections with negative variances in some datasets which must be excluded. What is the best way to deal with these?
- In terms of the order of applying the corrections, it seems that the result is somewhat changed depending on the order, particularly if the LP correction has not already been applied. The order seems to allow the time dependence to switch between the resolution v time or detector positions v time factors, as experimentally there is correlation between resolution and detector position. maybe it is not important where it is, however I think makes more physical sense for the main time dependence to be contained within the absorption correction, thus leaving the resolution plot to show any radiation damage?

1.4 Aside: assessment of data quality

The traditional R_{merge} was replaced by the multiplicity-independent R-factor R_{meas}, which gives an indication of the agreement between symmetry equivalent reflections (i.e. is a precision indicator of the unmerged data). $R_{\text{p.i.m.}}$ indicates the quality of the data after averaging symmetry equivalent reflections (i.e. is a precision indicator of the merged data).

$$R_{\text{meas}} = \left(\sum_h \sqrt{\frac{n_h}{(n_h - 1)}} \sum_l |I_{hl} - \langle I_h \rangle| \right) / \sum_h \sum_l \langle I_h \rangle \tag{10}$$

$$R_{\text{p.i.m.}} = \left(\sum_h \sqrt{\frac{1}{(n_h - 1)}} \sum_l |I_{hl} - \langle I_h \rangle| \right) / \sum_h \sum_l \langle I_h \rangle \tag{11}$$

R_{meas} is useful for deciding between space groups, investigating radiation damage during measurement, whereas precision of merged intensities is useful for assessing quality for downstream calculations. In these calculations, $\langle I_h \rangle$ is the averaged (unscaled) intensity for each unique reflection. For scaled data, one presumably should take the scaled intensities and the best estimate of the weighted 'true' average intensity. As a scaling algorithm minimises the residuals between the scaled and weighted average, the improvement in the R-factors before and after scaling give an indication of the improvement in the precision due to scaling.

More recently, Karplus and Diederichs (Science 336 6084 (2012)) suggested that the use of Pearson correlation coefficients should be used to estimate the correlation of the observed dataset with the 'true' signal. Specifically, the dataset is split randomly into two parts, where each part contains half of the measurements of each unique reflection. The correlation coefficient ($\text{CC}_{1/2}$) is calculated between the average intensities in each subset for each unique reflection. This can be converted to the correlation of the dataset with the 'true' intensities CC* by the relation

$$CC* = \sqrt{\frac{2CC_{1/2}}{1 + CC_{1/2}}}. \tag{12}$$

This can be calculated for resolution bins and gives a statistical measure of the usefulness of the data for correct structural determination - the value of CC* is one at high d and drops off at low d as the intensity becomes weaker. This is suggested as being more useful than the R-metrics and cutoffs based on I/σ. Furthermore, this can be directly compared to CCs that assess crystallographic model quality (e.g CC_{work}, CC_{free}) and hence can be used to determine where data quality is limiting the model or if over/under-fitting is present.