Boltzmann Statistics REDOR (BS-REDOR) Data Analysis Guide

BS-REDOR manual version 1.0,
pertaining to BS-REDOR version 1.0
12 July 2007 Melbourne, Australia

BS-REDOR has been published: John D. Gehman, Frances Separovic, Kun Lu, and Anil K. Mehta; "Boltzmann Statistics Rotational-Echo Double-Resonance Analysis", *The Journal of Physical Chemistry B* **111** 7802-7811 (2007). Please reference this paper if you use BS-REDOR. This guide contains instructions for running the current implementation (version 1.0). We hope you like it.

The Distribution currently consists of the following, compressed into a bzipped tarball. You can run `tar -tvf bsredor01.tbz` to confirm, and `tar -xvf bsredor01.tbz` to uncompress and get started.

- `bsredorstatic`: Precompiled executable with static GSL (as discussed below)
- `Data`: Contains the sample analysis subdirectories featured in the paper: glycine and Aβ(16-22) nanotubes.
- `Doc`: Subdirectory for documentation
- `scaling.pl`: Simple Perl utility to help you determine scaling factors as per equation 13 in the paper
- `Src`: Source code for BS-REDOR.

Code currently runs only under linux (including OS X). To compile, the g++ compiler and GSL (the Gnu Scientific Library) *must* be installed (easily downloaded and installed from `www.gnu.org/software/gsl/`). The code also makes use of gnuplot (more than likely available in your linux distribution if not already installed, but also available at `www.gnuplot.info/`) to display graphical results of a run and automatically generate figures. To compile the code, simply enter the Src subdirectory and type `make` (you can run `make clean` to clean things up afterward). The executable is not installed globally, you must run it as an executable using this directory as the path. For example, move it to ~/BS-REDOR for easier command line specification.

Alternatively, a precompiled executable is provided in the BS-REDOR directory which includes the GSL library statically; this code was compiled on SuSE Linux 9.3 using g++ (gcc) version 3.3.5.

If you don't have or don't want gnuplot installed, you can still run the code and plot the data yourself after each run; you'll just have to ignore the error messages regarding the absence of gnuplot.

Input

To run, simply type on the command line (assuming you've moved the executable `bsredor` up one directory from `Src`): `~/BS-REDOR/bsredor bsredorinput`

`bsredorinput` is a file containing all the parameters for the run. The file must contain the following lines in the following order, with your options specified after the colon (and space) on each line:

```
MODE: fit
LAMBDA_D: none
XYDATA: REDOR_data_intensity.dat
DISTDIST: distadjust
PATH: /home/gehman/BS-REDOR/Data/Gly8k/
OUTPUT: gly8k_818
XN: C13
YN: N15
NEAR: 1.5
FAR: 6.0
GRAIN: 0.01
PER 2PI: 180
MID PULSE: 62.5
MAS: 8000
SCALING: 0.818
LAST POINT: 112
NTRIES: 4
ITERS AT TEMP: 200 200 300 500
MAX STEP SIZE: 0.50 0.5 0.2 0.1
BOLTZMANN: 0.30 0.3 0.1 0.005
INIT TEMP: 0.1 0.02 0.005 0.005
ATTEN TEMP: 1.002 1.002 1.001 1.001
MIN TEMP: 0.001 0.0002 0.0005 0.00001
ERR ITERS AT TEMP: 10
ERR MAX STEP SIZE: 0.01
ERR BOLTZMANN: 0.2
ERR INIT TEMP: 0.005
ERR ATTEN TEMP: 1.002
ERR MIN TEMP: 0.001
MIN STDEV2: 0.0005
PER MC BLOCK: 10
MAX MC BLOCKS: 50
MAX MC CHI2: 100
```

These parameters are:

`MODE:` Can be either `sim`, `fit`, or `err`.

- `sim` simply simulates a REDOR curve for the distance distribution specified in `DISTDIST` (below), and the REDOR curve gets written out and plotted together with data specified in `XYDATA` This allows you to have a play with different distance distributions relative to your data.
- `fit` runs a single BS-REDOR reconstruction.
- `err` runs a `fit` to start, but then also runs Monte Carlo (MC) iterations as specified. The statistics over MC iterations are governed by the Central Limit Theorem,

which indicates that an observable ξ of (very) roughly normal parent distribution (e.g. the mean ξ_1, standard deviation ξ_2, skewness ξ_3, kertosis ξ_4) is itself normally distributed. Hence MC iterations are performed in blocks of n (as prescribed by `MC PER BLOCK`), and $\xi_{m,b}$ for the average $m = 1$ and the standard deviation $m = 2$ is calculated for each new block b. The overall observables ξ_m of the parent distribution, and their associated standard errors ς_{ξ_m}, are then

$$\xi_m = \frac{1}{L}\sum_{b=1}^{L} \xi_{m,b} \tag{1}$$

$$\varsigma_m = \frac{1}{\sqrt{L}}\sqrt{\frac{L\sum_{b=1}^{L}\xi_{m,b}^2 - \left[\sum_{b=1}^{L}\xi_{m,b}\right]^2}{L(L-1)}}. \tag{2}$$

Convergence is reached when the maximum ς_2 over all points in the distance distribution falls below `MIN STDEV2`, or $L =$ `MAX MC BLOCKS`.

`LAMBDA_D:` Not Currently Implemented.

`XYDATA:` Full path to the data file. The data file must be a minimum of three columns, with another two optional:

1. N_c; number of rotor cycles for each data point
2. $(S_0)_j$; sum of sideband intensities (or integral) for reference spectrum
3. S_j; sum of sideband intensities (or integral) for dephased spectrum
4. $(S/S_0)_j$; not used, but included for easier reference
5. σ_j; error in the $(S/S_0)_j$ data (assumed = 1 if omitted)

`DISTDIST:` Full path to file holding list of discrete distances (one per line) and corresponding fraction of total distance distribution (space delimited). For example, a fit which wanted to account explicitly for interference from a natural abundance carbon at a peptide bond's distance would have one line:

```
1.33 0.011039
```

but an arbitrary simulation might look like:

```
2.5 0.333
3.2 0.167
4.0 0.500
```

`PATH:` Full path for output directory. MUST INCLUDE TRAILING / (forward slash)

`OUTPUT:` Name of the output directory which will be written, and the basename to be used for all output files

`XN:` The observe nucleus. Currently can be `C13`, `N15`, `D2`, `H2`, `F19`, `P31`, or `H1`

`YN:` Dephasing nucleus. Same options as `XN`

`NEAR:` The shortest distance to be considered in the distribution

`FAR:` The longest distance to be considered in the distribution

`GRAIN:` The resolution of distances in the distribution

`PER 2PI:` Ω; the number of discrete angles per 2π over which to integrate. Future improvements will speed things up by taking advantage of weighted α,β crystal files.

`MID PULSE:` $\phi \cdot T_r \cdot 10^6$; fraction ϕ of rotor period (eq 1) for the intra-rotor period π pulse, in μs

`MAS:` $1/T_r$; the Magic Angle Spinning rate, in Hz

`SCALING:` γ; the scaling factor; With reference to the paper (eq 14), give $\gamma_a \times \gamma_b$ here. The included scaling.pl perl code is included for convenience to help determine γ_a as per eq 13. γ_b of course depends on your rig.

`LAST POINT:` The largest number of rotor periods that the simulated or fit REDOR dephasing curve should be calculated for, irrespective of the data.

`NTRIES:` Number of simulated annealings to run

`ITERS AT TEMP:` List of iterations to perform in each respective simulated annealing run. If the number of `ITERS` supplied is less than `NTRIES`, the last value will be duplicated as many times as it needs to be.

`MAX STEP SIZE:` Maximum amplitude that each Lagrange multiplier can be adjusted during each iteration; also a list to be applied during each respective simulated annealing run, repeating the last supplied value as necessary for `NTRIES`

`BOLTZMANN:` k from eq. 12; Pseudo-Boltzmann constant, also a list as above.

`INIT TEMP:` List of initial temperatures to be used for each simulated annealing.

`ATTEN TEMP:` List of factors by which temperature is attenuated at each `ITER` during each simulated annealing run

`MIN TEMP:` Temperatures at which each simulated annealing run is finished

`ERR ITERS AT TEMP:` Same as `ITERS AT TEMP` above, but single value applied during the reconstructions run at each Monte Carlo iteration

`ERR MAX STEP SIZE:` Same as `MAX STEP SIZE` above, but single value applied during the reconstructions run at each Monte Carlo iteration

`ERR BOLTZMANN:` Same as `BOLTZMANN` above, but single value applied during the reconstructions run at each Monte Carlo iteration

`ERR INIT TEMP:` Same as `INIT TEMP` above, but single value applied during the reconstructions run at each Monte Carlo iteration

`ERR ATTEN TEMP:` Same as `ATTEN TEMP` above, but single value applied during the reconstructions run at each Monte Carlo iteration

`ERR MIN TEMP:` Same as `MIN TEMP` above, but single value applied during the reconstructions run at each Monte Carlo iteration

`MIN STDEV2:` If the maximum standard error in the standard deviation (i.e. the standard deviation of the Monte Carlo statistical standard deviation) at any point in the distance distribution is less than this value, stop the Monte Carlo process early

`PER MC BLOCK:` The number of Monte Carlo iterations to perform per block

`MAX MC BLOCKS:` The maximum number of Monte Carlo blocks to perform (will do a minimum of five)

`MAX MC CHI2:` It is impractical to examine the reconstruction from every Monte Carlo iteration. This parameter allows the rejection of any reconstructions over a given value. Typicaly set this to 3-4 times the χ^2 that you get in the single, careful reconstruction.

If the failure rate is more than a fraction of a percent, something's wrong, and you may want to explore different simulated annealing parameters (or collect better data).

Output

A number of output files are created within the specified directory, depending upon the `MODE` run. BEWARE: there is no over-write protection (yet) of output. Be certain to specify a new basename at the `OUTPUT` parameter of the input file, unless of course you want to over-write previous reconstructions.

`PATH/OUPUT.me:` The actual output data. Formatted columns are:

1: Distance (Angstroms)
2: Dipolar Coupling (Hz)
3: Distribution Density (normalized)
4: Cumulative Distribution Density Integral
5: Monte Carlo Distribution Density Mean
6: Monte Carlo Distribution Density Mean Error
7: Monte Carlo Distribution Density Standard Deviation
8: Monte Carlo Distribution Density Standard Deviation Error

9: Number of Cycles
10: S_0 Data
11: S Data
12: S/S_0 Data
13: S/S_0 Data Error
14: S/S_0 Reconstruction "Fit"
15: Forced Data Adjustment
16: Residual
17: Lagrange Multiplier
18: S/S_0 Data if Lagrange Multiplier < 0
19: S/S_0 Data if Lagrange Multiplier > 0

20: Number of Cycles (for high resolution reconstruction "fit")
21: S/S_0 Reconstruction (high resolution)
22: Forced Adjustment (high resolution)

`PATH/OUTPUT.sa:` The record of simulated annealing for the `fit`. Formatted columns are:

1: Temperature Step
2: Total Evaluations
3: Temperature
4: χ^2
5: Percentage of evaluations which resulted in lower χ^2
6: Percentage of evaluations which resulted in higher χ^2, but were randomly accepted
7: Percentage of evaluations which resulted in higher χ^2, but were randomly rejected
8+: Current configuration of Lagrange multipliers, one for each data point

`PATH/OUTPUT.log`: A log file of all details for the reconstruction

`PATH/OUTPUT.mx`: $\mathcal{R}$; row index is j, column index is i

`PATH/OUTPUT.mxT`: $\mathcal{R}$; row index is i, column index is j

`PATH/OUTPUT.gp`: Gnuplot script file to plot condensed results following run for quick and easy review. The same on-screen figure can be generated after the fact by `gnuplot -persist PATH/OUTPUT.gp`.

`PATH/OUTPUT-ps.gp`: Gnuplot script file to generate `PATH/OUTPUT-ps.eps`.

`PATH/OUTPUT-ps.eps`: Postscript file with reconstruction plotted on three panels: residuals, fit, and distance distribution.

The data file (specified in the inputs file) and the inputs file (specified on the command line when initiating a reconstruction) are also copied into the output directory.