Bayesian Analysis Users Guide
Release 4.00, Manual Version 1

G. Larry Bretthorst
Biomedical MR Laboratory
Washington University School Of Medicine,
Campus Box 8227
Room 2313, East Bldg.,
4525 Scott Ave.
St. Louis MO 63110
http://bayes.wustl.edu
Email: larry@bayes.wustl.edu

August 21, 2013
Contents

Manual Status 14

1 An Overview Of The Bayesian Analysis Software 17
   1.1 The Server Software ........................................... 17
   1.2 The Client Interface ........................................... 20
      1.2.1 The Global Pull Down Menus .............................. 22
      1.2.2 The Package Interface .................................... 22
      1.2.3 The Viewers .............................................. 25

2 Installing the Software 27

3 the Client Interface 29
   3.1 The Global Pull Down Menus ................................. 31
      3.1.1 the Files menu .......................................... 31
      3.1.2 the Packages menu ...................................... 36
      3.1.3 the WorkDir menu ...................................... 41
      3.1.4 the Settings menu ...................................... 42
      3.1.5 the Utilities menu ...................................... 46
      3.1.6 the Help menu .......................................... 47
   3.2 The Submit Job To Server area .............................. 47
   3.3 The Server area ............................................. 48
   3.4 Interface Viewers ........................................... 49
      3.4.1 the Ascii Data Viewer .................................. 49
      3.4.2 the fid Data Viewer .................................... 51
      3.4.3 Image Viewer ........................................... 56
         3.4.3.1 the Image List area ................................ 56
         3.4.3.2 the Set Image area ................................ 58
         3.4.3.3 the Image Viewing area ............................. 58
         3.4.3.4 the Grayscale area on the bottom .................. 60
         3.4.3.5 the Pixel Info area ................................ 60
         3.4.3.6 the Image Statistics area ......................... 60
      3.4.4 Prior Viewer ............................................ 62
      3.4.5 fid Model Viewer ....................................... 65
         3.4.5.1 The fid Model Format ............................... 65
### 3.4.5.2 The Fid Model Reports

### 3.4.6 Plot Results Viewer

#### 3.4.6.1 the Data, Model and Residuals Plots

#### 3.4.6.2 the Posterior Probabilities Plots

### 3.4.7 the Posterior Probability Vs Parameter Samples plot

#### 3.4.7.1 the Expected Log Likelihood Plot

#### 3.4.7.2 the Scatter Plots

#### 3.4.7.3 the Log Probability Plot

### 3.4.8 Text Results Viewer

### 3.4.9 Files Viewer

### 3.4.10 Fortran/C Code Viewer

#### 3.4.10.1 Fortran/C Model Viewer Popup Editor

### 4 An Introduction to Bayesian Probability Theory

#### 4.1 The Rules of Probability Theory

#### 4.2 Assigning Probabilities

#### 4.3 Example: Parameter Estimation

##### 4.3.1 Define The Problem

##### 4.3.1.1 The Discrete Fourier Transform

##### 4.3.1.2 Aliases

##### 4.3.2 State The Model—Single-Frequency Estimation

##### 4.3.3 Apply Probability Theory

##### 4.3.4 Assign The Probabilities

##### 4.3.5 Evaluate The Sums and Integrals

##### 4.3.6 How Probability Generalizes The Discrete Fourier Transform

##### 4.3.7 Aliasing

##### 4.3.8 Parameter Estimates

#### 4.4 Summary and Conclusions

### 5 Given Exponential Model

#### 5.1 The Bayesian Calculation

#### 5.2 Outputs From The Given Exponential Package

### 6 Unknown Number of Exponentials

#### 6.1 The Bayesian Calculations

#### 6.2 Outputs From The Unknown Number of Exponentials Package

### 7 Inversion Recovery

#### 7.1 The Bayesian Calculation

#### 7.2 Outputs From The Inversion Recovery Package

### 8 Bayes Analyze

#### 8.1 Bayes Model

#### 8.2 The Bayes Analyze Model Equation

#### 8.3 The Bayesian Calculations

#### 8.4 Levenberg-Marquardt And Newton-Raphson
8.5 Outputs From The Bayes Analyze Package ........................................... 167
  8.5.1 The “bayes.params.nnnn” and “bayes.model.nnnn” Files ............... 169
    8.5.1.1 The Bayes Analyze File Header ......................................... 169
    8.5.1.2 The Global Parameters .................................................. 174
    8.5.1.3 The Model Components .................................................. 175
  8.5.2 The “bayes.output.nnnn” File .................................................. 177
  8.5.3 The “bayes.probabilities.nnnn” File ......................................... 181
  8.5.4 The “bayes.log.nnnn” File ..................................................... 184
  8.5.5 The “bayes.status.nnnn” and “bayes.accepted.nnnn” Files ............... 187
    8.5.5.1 The “bayes.model.nnnn” File .......................................... 188
  8.5.6 The “bayes.summary1.nnnn” File ............................................. 189
  8.5.7 The “bayes.summary2.nnnn” File ............................................. 190
  8.5.8 The “bayes.summary3.nnnn” File ............................................. 191
  8.6 Bayes Analyze Error Messages ..................................................... 192

9 Big Peak/Little Peak ................................................................. 197
  9.1 The Bayesian Calculation .......................................................... 199
  9.2 Outputs From The Big Peak/Little Peak Package .............................. 206

10 Metabolic Analysis ................................................................. 209
  10.1 The Metabolic Model ............................................................... 213
  10.2 The Bayesian Calculation ........................................................ 215
  10.3 The Metabolite Models ............................................................ 218
    10.3.1 The IPGD.D2O Metabolite .................................................. 218
    10.3.2 The Glutamate.2.0 Metabolite .............................................. 222
    10.3.3 The Glutamate.3.0 Metabolite .............................................. 225
  10.4 The Example Metabolite ........................................................... 226
  10.5 Outputs From The Bayes Metabolite Package .................................. 228

11 Find Resonances ...................................................................... 229
  11.1 The Bayesian Calculations ....................................................... 231
  11.2 Outputs From The Bayes Find Resonances Package ............................ 236

12 Diffusion Tensor Analysis .......................................................... 237
  12.1 The Bayesian Calculation ........................................................ 239
  12.2 Using The Package ............................................................... 244

13 Big Magnetization Transfer ....................................................... 249
  13.1 The Bayesian Calculation ........................................................ 249
  13.2 Outputs From The Big Magnetization Transfer Package ...................... 252

14 Magnetization Transfer ............................................................. 255
  14.1 The Bayesian Calculation ........................................................ 257
  14.2 Using The Package ............................................................... 261
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 Magnetization Transfer Kinetics</td>
<td>267</td>
</tr>
<tr>
<td>15.1 The Bayesian Calculation</td>
<td>269</td>
</tr>
<tr>
<td>15.2 Using The Package</td>
<td>273</td>
</tr>
<tr>
<td>16 Given Polynomial Order</td>
<td>277</td>
</tr>
<tr>
<td>16.1 The Bayesian Calculation</td>
<td>279</td>
</tr>
<tr>
<td>16.1.1 Gram-Schmidt</td>
<td>279</td>
</tr>
<tr>
<td>16.1.2 The Bayesian Calculation</td>
<td>280</td>
</tr>
<tr>
<td>16.2 Outputs From the Given Polynomial Order Package</td>
<td>282</td>
</tr>
<tr>
<td>17 Unknown Polynomial Order</td>
<td>285</td>
</tr>
<tr>
<td>17.1 Bayesian Calculations</td>
<td>287</td>
</tr>
<tr>
<td>17.1.1 Assigning Priors</td>
<td>288</td>
</tr>
<tr>
<td>17.1.2 Assigning The Joint Posterior Probability</td>
<td>289</td>
</tr>
<tr>
<td>17.2 Outputs From the Unknown Polynomial Order Package</td>
<td>291</td>
</tr>
<tr>
<td>18 Errors In Variables</td>
<td>295</td>
</tr>
<tr>
<td>18.1 The Bayesian Calculation</td>
<td>297</td>
</tr>
<tr>
<td>18.2 Outputs From The Errors In Variables Package</td>
<td>300</td>
</tr>
<tr>
<td>19 Behrens-Fisher</td>
<td>303</td>
</tr>
<tr>
<td>19.1 Bayesian Calculation</td>
<td>303</td>
</tr>
<tr>
<td>19.1.1 The Four Model Selection Probabilities</td>
<td>306</td>
</tr>
<tr>
<td>19.1.1.1 The Means And Variances Are The Same</td>
<td>307</td>
</tr>
<tr>
<td>19.1.1.2 The Mean Are The Same And The Variances Differ</td>
<td>309</td>
</tr>
<tr>
<td>19.1.1.3 The Means Differ And The Variances Are The Same</td>
<td>310</td>
</tr>
<tr>
<td>19.1.1.4 The Means And Variances Differ</td>
<td>311</td>
</tr>
<tr>
<td>19.1.2 The Derived Probabilities</td>
<td>312</td>
</tr>
<tr>
<td>19.1.3 Parameter Estimation</td>
<td>313</td>
</tr>
<tr>
<td>19.2 Outputs From Behrens-Fisher Package</td>
<td>314</td>
</tr>
<tr>
<td>20 Enter Ascii Model</td>
<td>321</td>
</tr>
<tr>
<td>20.1 The Bayesian Calculation</td>
<td>323</td>
</tr>
<tr>
<td>20.1.1 The Bayesian Calculations Using Eq. (20.1)</td>
<td>323</td>
</tr>
<tr>
<td>20.1.2 The Bayesian Calculations Using Eq. (20.2)</td>
<td>324</td>
</tr>
<tr>
<td>20.2 Outputs From The Enter Ascii Model Package</td>
<td>327</td>
</tr>
<tr>
<td>21 Test Your Own ASCII Model</td>
<td>329</td>
</tr>
<tr>
<td>22 Ascii Model Selection</td>
<td>331</td>
</tr>
<tr>
<td>23 Phasing An Image</td>
<td>333</td>
</tr>
<tr>
<td>23.1 The Bayesian Calculation</td>
<td>334</td>
</tr>
<tr>
<td>23.2 Using The Package</td>
<td>340</td>
</tr>
</tbody>
</table>
List of Figures

1.1 The Start Up Window ........................................... 21
1.2 Example Package Interface .................................. 23

3.1 The Start Up Window ........................................... 30
3.2 The Files Menu ................................................. 31
3.3 The Load Image Selection Menu ............................... 33
3.4 The Packages Menu ............................................. 37
3.5 The Working Directory Pull Down Menu ..................... 42
3.6 The Working Directory Pop up ................................ 43
3.7 The Settings Pull Down Menu .................................. 44
3.8 The McMC Parameters Pop up ................................. 44
3.9 The Edit Server Popup ........................................... 45
3.10 The Submit Job Widget Group ............................... 48
3.11 The Server Widget Group ..................................... 49
3.12 The Ascii Data viewer ........................................... 50
3.13 The fid Data viewer .............................................. 52
3.14 The Fid Data Viewer Display Type ......................... 53
3.15 The Fid Data Viewer the Options Menu .................... 54
3.16 The Image Viewer ............................................... 57
3.17 The Image Viewer Right Mouse Menu ...................... 58
3.18 The Prior Viewer ............................................... 63
3.19 The Fid Model Viewer .......................................... 66
3.20 The Data Model and Residuals ................................ 69
3.21 The Plot Information popup .................................. 70
3.22 The Posterior Probabilities ................................... 71
3.23 The Posterior Probabilities Vs Parameter Value ........... 73
3.24 The Posterior Probabilities Vs Parameter Value a Skewed Example ................................. 74
3.25 The Expected Log Likelihood .................................. 76
3.26 The Scatter Plots ................................................ 77
3.27 The Log Probability Plot ...................................... 79
3.28 The Text Results Viewer ...................................... 81
3.29 The Bayes Condensed File ..................................... 84
3.30 Fortran/C Model Viewer ....................................... 87
3.31 Fortran/C Model Viewer ....................................... 88
11.1 the Find Resonances interface .................................................. 230
12.1 Diffusion Tensor Interface .......................................................... 238
12.2 Diffusion Tensor Parameter Estimates .......................................... 246
12.3 Diffusion Tensor Posterior Probability For The Model ................. 246
13.1 The Big Magnetization Package Interface .................................... 250
13.2 Big Magnetization Transfer Example Fid ...................................... 252
13.3 Big Magnetization Transfer Expansion ........................................ 253
13.4 Big Magnetization Transfer Peak Pick ........................................ 254
14.1 Magnetization Transfer Interface ............................................... 256
14.2 Magnetization Transfer Peak Pick ............................................. 262
14.3 Magnetization Transfer Example Data ........................................ 263
14.4 Magnetization Transfer Example Spectrum ................................... 264
15.1 Magnetization Transfer Kinetics Interface ................................... 268
15.2 Magnetization Transfer Kinetics Arrhernius Plot .......................... 274
15.3 Magnetization Transfer Kinetics Water Viscosity Table ................ 275
16.1 Given Polynomial Order Package Interface .................................. 278
16.2 Given Polynomial Order Scatter Plot ......................................... 284
17.1 Unknown Polynomial Order Interface ......................................... 286
17.2 The Distribution of Models ...................................................... 290
17.3 Unknown Polynomial Order Package Posterior Probability ............. 292
18.1 Errors In Variables Interface ..................................................... 296
18.2 Errors In Variables McMC Values File ...................................... 302
19.1 the Behrens-Fisher interface .................................................... 304
19.2 Behrens-Fisher Hypotheses Tested ............................................ 305
19.3 Behrens-Fisher Console Log ..................................................... 315
19.4 Behrens-Fisher Status Listing .................................................. 316
19.5 Behrens-Fisher McMC Values File, The Preamble ........................ 317
19.6 Behrens-Fisher McMC Values File, The Middle .......................... 318
19.7 Behrens-Fisher McMC Values File, The End ............................... 319
20.1 Enter Ascii Model Interface ...................................................... 322
21.1 Test Your Own Ascii Model Interface ........................................ 330
22.1 Ascii Model Selection Interface ................................................. 332
23.1 Absorption Model Images ......................................................... 334
23.2 Bayes Phase Interface ............................................................ 335
23.3 Bayes Phase Listing ............................................................... 341
List of Tables

8.1  Multiplet Relative Amplitudes ........................................ 157
8.2  Bayes Analyze Models ................................................... 173
8.3  Bayes Analyze Short Descriptions ................................. 186
Appendix B

Markov chain Monte Carlo With Simulated Annealing

Most of the packages in the Bayesian Analysis software use Markov chain Monte Carlo simulations to approximate the Bayesian posterior probability. To understand how a Markov chain can be used to do this, suppose there is some quantity called $M$. This quantity could be a set of parameters or it could be a selection of models and the object of the Bayesian calculation is to estimate the parameters or to determine which model best characterizes the data. While these two problems sound very different, they are really one and the same problem. To see this suppose $M$ is just the decay rate constant in a simple exponential decay, then discrete values of $M$, just specify a set of models $\{M_1, M_2, \ldots, M_n\}$ and in parameter estimation we compute the posterior probability for each model, just as we do in model selection. So model selection and parameter estimation are fundamentally the same problem. The major difference is that in parameter estimation problems, the functional form of the models is the same, while it can be different in model selection.

The estimation problems addressed by the various packages are all structurally similar, suppose $M$ consists of a set of parameters or model indicators, $M \in \{M_1, M_2, \ldots, M_n\}$, and we wish to compute the posterior probability for an individual parameters $P(M_j|DI)$ where $M_j$ is the hypothesis of interest, $D$ represents all of the data and $I$ is the prior information. Applying the rules of Bayesian probability theory, the joint posterior probability for all of the parameters is given by

$$P(M_1 \ldots M_n|DI) = \frac{P(M_1 \ldots M_n|I)P(D|M_1 \ldots M_n|I)}{P(D|I)} \quad (B.1)$$

which is Bayes’ theorem [1]. For those unfamiliar with the rules of Bayesian probability theory, see Chapter 4 for a tutorial on probability theory or consult [30, 3, 11, 32, 55, 31] for more detailed descriptions of probability theory when treated as extended logic.

If we normalize this posterior probability at the end of the calculation, then $P(D|I)$ can be dropped and one obtains:

$$P(M_1 \ldots M_n|DI) \propto P(M_1 \ldots M_n|I)P(D|M_1 \ldots M_n|I). \quad (B.2)$$

Using logical independence and the product rule, the joint prior probability, $P(M_1 \ldots M_n|DI)$, can
be factored to obtain

\[ P(M_1 \ldots M_n | DI) \propto P(M_1 | I) \cdots P(M_n | I) P(D | M_1 \ldots M_n I) \]  

(B.3)

and it is this joint posterior probability that most of the Markov chain Monte Carlo simulation approximate. To determine exactly how an individual package implements this calculation, see the Chapter describing that package.

If all a Markov chain Monte Carlo simulation did was to approximate this joint posterior probability, it would not be very useful because what is really needed in the Bayesian calculation is not \( P(M_1 \ldots M_n | DI) \), but \( P(M_j | DI) \) where \( P(M_j | DI) \) is computed using the sum rule of probability theory:

\[ P(M_j | DI) = \int_{M_1} P(M_1 | I) \cdots \int_{M_n} P(M_n | I) P(D | M_1 \ldots M_n I) dM_1 \cdots dM_n \]  

(B.4)

where the integrals are over all \( M_l \) except \( j \). In the Markov chain Monte Carlo simulation this part of the calculation is done using Monte Carlo integration which consists of sampling the joint posterior probability, the integrand, and then using the samples for each parameter as samples from the marginal posterior probability for each parameter separately.

### B.1 Metropolis-Hastings Algorithm

All of the packages that implement their calculations using Markov chain Monte Carlo use the Metropolis-Hastings algorithm and you can read more about the Metropolis-Hastings algorithm at the previous WikipediA link. You can consult the original 1953 paper [43] paper, Gilks et al. have written extensively on Markov chain Monte Carlo as used in Bayesian probability theory [23] and Radford Neal did his dissertation on that subject [44]. Here we will briefly summarize how the Metropolis-Hastings algorithm is used to approximate the Bayesian posterior probabilities.

At their heart all Markov chain simulations are random number generators in which you, the author, can choose the distribution of the random numbers. In the Bayesian calculations done in this software the chosen distribution is the joint posterior probability for all of the parameters and model indicators given the data and the prior information. By running the chain one can sample the joint posterior probability built into the Markov chain simulation. To run a Markov chain Monte Carlo simulation, one must be able to compute Eq. (B.3) for a given set of parameters and model indicators. Here is a very toy version of how one runs a Markov chain Monte Carlo simulation to sample the joint posterior probability:

1. One begins the process of generating a Markov chain by simply sampling the parameter from their valid range. We are going to call these parameters \( M_0 \) and the joint posterior probability computed using \( M_0 \) will be designated as \( P_0 \).

2. Next propose a new value for one or more of the parameter. Call this new proposed set of parameters \( M_1 \), and compute the joint posterior probability, \( P_1 \) using the proposed values.

3. Accept the proposal if \( P_1 \) is greater than \( P_0 \). Here accepting the proposal means that you replace \( M_0 \) and \( P_0 \) by \( M_1 \) and \( P_1 \) respectively and go back to Step 2.
4. If $P_1$ is less than or equal to $P_0$, then draw a random number, $r$, from a uniform $(0-1)$ random number generator, and if the ratio $P_1/P_0$ is greater than $r$, accept the proposed value of $P_1$ and $M_1$, i.e., replace $M_0$ and $P_0$ with $M_1$ and $P_1$ and go to Step 2.

5. Otherwise, reject the proposed values. Here rejecting the proposed values simply means going back to Step 2 without replacing $M_0$ or $P_0$.

This simple 5 step procedure is all it takes to generate a Markov chain Monte Carlo simulation. Unfortunately, implementing the calculation in practice, is more of an art than a science and, shortly, we will describe a few of the tricks used in ensuring the calculations work correctly.

There are several major problems with the Markov chain Monte Carlo simulation as described so far: First, it is possible for the simulation to become stuck in local maxima and one would never know it. So one needs a mechanism for detecting simulations that are trapped in local maxima. Second, even if the chain converges correctly its very difficult to tell if the Markov chain has reached a stationary point. One can run a single chain over multiple steps and then look at the path of the simulation, but this is a very unreliable method of testing whether or not a simulation has converged because simulations often deviate from the maximum. And third, with a single chain it is very difficult to adjust the acceptance rate, the number of times a change to a simulation is accepted divided by the total number of times one changed the simulation.

In the following sections we are going to describe how multiple simulations, simulated annealing, killing simulations and adjusting the rate of acceptance in the simulation can be used to generate Markov chains that are highly robust, and almost impossible to get stuck in a local maxima.

### B.2 Multiple Simulations

We do not run a single Markov chain Monte Carlo simulation; rather we run an ensemble of simulations in parallel. Typically, the ensemble is on the order of a few 10’s, for example the defaults number of simulations used in the interface is 50, 50 because experience with running multiple simulations indicates that most of the time 50 simulations is enough to explore most parameter spaces; while running fewer increases the risk of nonconvergence and running more usually make things run longer without improving convergence.

For reasons that will become apparent shortly, we initialize the simulations from the prior probability for the parameters. We then run the simulations through a fixed number of steps. Here running a simulation means that we vary the parameters in one simulation, and then either accept or reject the modified simulation based on the prescription given above, Section B.1. This procedure is repeated for each parameter in each simulation and we repeat this procedure at least 25 times for each parameter. So for example if there are 50 simulations, 20 parameters, and each parameter is varied 25 times, the operations count is about $50 \times 20 \times 25 = 25,000$ operations to bring the ensemble of simulations to equilibrium.

Between annealing steps various statistics are computed from the multiple simulations and these statistics are used to aid in judging convergence. Additionally, the expected value of the logarithm of the likelihood is used in thermodynamic integration, Section C, and the trajectories of each simulation are good visual aids in determining convergence.
B.3 Simulated Annealing

These simulations are run using simulated annealing. In simulated annealing one introduces an annealing parameter, which we call $\beta$, into the calculation of the joint posterior probability, Eq. (B.3). This annealing parameter is introduced by raising the direct probability for the data to the $\beta$ power:

$$P(M_1 \ldots M_n | \beta D I) = P(M_1 | I) \cdots P(M_n | I) P(D | M_1 \ldots M_n I)^\beta \quad (B.5)$$

where we have modified the notation to indicate that the joint posterior probability is a function of $\beta$. It is this modified joint probability density function that is used in the Markov chain Monte Carlo simulation. When $\beta = 0$, the likelihood is raised to the power of zero and the data completely disappears from the problem, one is sampling the prior. Consequently, when the Markov chain Monte Carlo simulations are initialized they are initialized from the prior probability for the parameters. Also note, that when $\beta = 1$ we are sampling the full joint posterior probability for the parameters and model indicators. The annealing parameter, $\beta$, is varied from zero to one according to some annealing schedule, discussed shortly.

Typically one starts the simulations with $\beta = 0$ and runs the simulations until they reach equilibrium. Running the simulations means changing the parameters in a simulation and then accepting or rejecting the change according to the simple perception given earlier, Chapter B.1. For a given value of $\beta$, when running the simulations, the posterior probability will increase to an equilibrium point. That is to say, the posterior probability will quit increasing and simply fluctuate about the peak in the posterior probability.

Ounce the simulations are in equilibrium, we increase the annealing parameter by small amount. This has the effect of knocking the ensemble of simulations out of equilibrium, so we again run the simulations until they reach equilibrium at this new value of $\beta$. When the annealing parameter is increased, the likelihood becomes more important and the simulations will begin to cluster around the high likelihood regions. However, because the annealing parameter is still small, the simulations will explore a much larger part of parameter space simply because the likelihood is not let strongly constraining them. As the annealing parameter is increased the likelihood becomes increasingly important and the simulations begin to cluster around increasingly high and higher likelihood regions.

B.4 The Annealing Schedule

The annealing schedule, the way the annealing parameter $\beta$ is varied from zero to one, can be something as simple as varying the annealing parameter linearly to something much more elaborate. In earlier versions of the software, a linear annealing schedule was used. In a linear annealing schedule the annealing parameter was given by:

$$\beta = \frac{j}{n} \quad (0 \leq j \leq n) \quad (B.6)$$

where $n$ is the number of nonzero steps taken in the annealing. This worked well for many problems, but sometimes ran into difficulty when the logarithm of the likelihood is very rapidly changing, because the first tentative steps in simulated annealing can raise the likelihood many hundreds of orders of magnitude and consequently the simulations can fail to local the global maximum of the posterior probability.
In the current version of the software the annealing parameter is adjusted dynamically as follows. The annealing parameter starts at zero, and the simulations are run until they reach equilibrium. Call this step $n$. For the next step, the $n + 1$ step, the annealing parameter is given by

$$\beta_{n+1} = \text{Min}(1, \beta_n + d\beta_n)$$

(B.7)

where $\beta_{n+1}$ is the value of $\beta$ to be used in the next annealing step. If the minimum number of annealing steps is $N$, then $d\beta$ is given by:

$$d\beta = \text{Min}\left(\frac{1}{\sigma + N}, 1 - \beta\right)$$

(B.8)

where $\sigma$ is the standard deviation of the logarithm of the likelihood computed from the ensemble of simulations. Note that if the standard deviation of the logarithm likelihood is small, then this method of computing $\beta$ just reduces to Eq. (B.6), i.e., a linear annealing schedule. However, when the simulations are first initialized by sampling the prior probability for the parameters, the standard deviation of the logarithm of the likelihood is usually large, and consequently, the simulations initially move slowly, gaining speed as the simulations converge on the global maxima.

It is this initial slow annealing that allows the multiple Markov chain simulations to explore the parameter space and locate the global maximum of the posterior probability. However, slowing the annealing down at small values of $\beta$ is not enough to ensure that the simulations reach the global maximum; it is still possible for simulations to become stuck in local maxima.

B.5 Killing Simulations

As noted, slowing down the annealing for small values of $\beta$ works very well for giving the simulations time to find the global maxima. However, it is not enough; it is still possible for simulations to become stuck in local maxima. These trapped simulations must be found and fixed as quickly as possible if the simulations are to reach a stationary point.

Up to now the Markov chain simulations have been described as having multiple chains running in parallel using simulated annealing with an annealing parameter set dynamically based on the standard deviation of the logarithm of the likelihood. Each step in the Markov chain proceeds roughly as follows, the value of the annealing parameter is computed and set for this step in the simulated annealing cycle. Prior to setting to the annealing parameter, simulations should be in equilibrium. However, increasing the annealing parameter throws the simulations out of equilibrium and because the simulations are out of equilibrium, we make a number of other modifications to the simulations, we adjust the rate of acceptance, and we kill off a number of simulations.

Between annealing steps, the algorithm doing the Markov chain simulation kills off low probability simulations. To do this the algorithm computes the logarithm of the posterior probability for each simulation. This table of logarithms is then indexed, sorted, and used to replace low probability simulations. In this step typically 10% of the simulations are replaced by higher probability simulations. The program simply takes the lowest probability simulation and then replaces the simulation by one of the simulations having higher probability. The higher probability replacement is chosen by drawing a random number from a Gaussian having a standard deviation that is roughly one third of the total simulations. So when a simulation is replaced, it is replaced by a higher probability simulation, but not necessarily the highest probability simulation.
B.6 the Proposal

When doing a Metropolis-Hastings Markov chain Monte Carlo simulation one must be careful in proposing new values of a parameter. If the current value of a parameter is $M_0$, one proposes a new parameter value $M_1$ as follows:

\[ M_1 = M_0 + \delta M \]  

(B.9)

where $\delta M$ is the change that is being made to the parameter. The exact method one obtains this $\delta M$ doesn’t matter except for one propriety that must be enforced. If the probability of moving from $M_0$ to $M_1$ is given by $P(M_1|M_0)$ then the probability of moving from $M_1$ to $M_0$ must be the same:

\[ P(M_0|M_1) = P(M_1|M_0) \]  

(B.10)

That is to say jumps in the proposal probability must be symmetric. There are countless modifications and addendum to this rule and you can look at the various references on Markov chain simulations on what these modifications are, but in the calculations implemented in this software package, Eq. (B.10) is the rule implemented using a simple Gaussian proposal. A Gaussian proposal has a number of advantages, for example it is symmetric in its argument and thus automatically satisfies Eq. (B.10). Additionally, a Gaussian has one additional parameter that is important, its standard deviation. By adjusting the size of the standard deviation of the proposal one can control how often the Markov chain transitions from one proposed value to another.

Now one might ask why this is important and the answer is simple, if the proposal it too small the simulation will not explore the parameter space and if the proposal is too large, the change in the posterior probability will be so great that the probability of accepting the change is zero and again the simulations do not explore the parameter space. Consequently, it is important to monitor the size of the proposal and to adjust it between annealing steps to ensure it is neither too small or too large.

The way the programs that implement the Markov chain simulations control the proposal is by keeping track of the acceptance rate for a given parameter. The acceptance rate is simply the ratio of the number of times a proposed parameter was accepted divided by the total number of times one proposed a new parameter value. There is not hard and fast rule on how often one should accept a parameter, but too often or too little are both bad. Additionally, if one is to error, then err on the side of more exploration of the parameter space is probably a good thing. Consequently, the programs that implement these calculations try and keep the acceptance rate between 20 and 30%. If the acceptance rate falls below 20% the proposal is decreased and if the acceptance rate is above 30% the proposal is increased. If it is between 20 and 30% no change is made. There are many addendum that could be added to this description, but it captures what the program actually does. Indeed, there is an output report generated by most of the packages in the Bayesian Analysis Software called an accepted report and that report is available while a package is running and needless to say, its primary output is the current acceptance rate for the various parameters.
Bibliography


[43] Metropolis, Nicholas, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller (1953), “Equation of State Calculations by Fast Computing Machines,” Journal of Chemical Physics. The previous link is to the American Institute of Physics and if you do not have access to Science Sitations you many not be able to retrieve this paper.


