BUGS 0.6 *
Bayesian inference Using Gibbs Sampling
(Addendum to Manual)

David Spiegelhalter  Andrew Thomas  Nicky Best  Wally Gilks

MRC Biostatistics Unit, Institute of Public Health,
Robinson Way, Cambridge CB2 2SR

Tel: 44-1223-330300  Fax: 44-1223-330388
e-mail: bugs@mrc-bsu.cam.ac.uk  ftp: ftp.mrc-bsu.cam.ac.uk
http://www.mrc-bsu.cam.ac.uk/bugs

September 26, 1997

This Addendum specifies additional features of BUGS 0.6, and should be read in conjunction with the current manual for BUGS 0.5 (Spiegelhalter et al., 1996a).

Contents

1 Getting started  ................................................. 2
  1.1 Getting the software ........................................ 2
  1.2 The script file for ‘bugs’ (Sparc) .......................... 2
  1.3 The script file for ‘backbugs’ (Sparc) ..................... 2

2 New Facilities in 0.6  ........................................... 3
  2.1 Checkpoint command ......................................... 3
  2.2 Metropolis sampling ......................................... 3
  2.3 Minor changes ................................................. 3

3 Corrected bugs from Version 0.5  .................................. 3

4 Known restrictions still existing in Version 0.6 ................. 4

5 Examples  ......................................................... 4

*BUGS © copyright MRC Biostatistics Unit 1997. ALL RIGHTS RESERVED. The support of the Economic and Social Research Council (UK) is gratefully acknowledged. The work was funded in part by ESRC (UK) Award Number H519 25 5023.
1 Getting started

1.1 Getting the software

BUGS 0.6 may be obtained from World Wide Web page http://www.mrc-bsu.cam.ac.uk/bugs, or by anonymous ftp from ftp.mrc-bsu.cam.ac.uk in directory pub/methodology/bugs: login as anonymous and give your full e-mail address as the password. The message and README files will tell you how to obtain the program, examples and documentation.

1.2 The script file for ‘bugs’ (Sparc)

#!/bin/sh
# runs BUGS interactively
#
case $# in
   0) bugs06.sparc 32 bugs;;
   *) echo ’bugs6’;;
esac
rm bugs.buf

“32” refers to the number of bins in the metropolis algorithm (Section 2.2).
“bugs” refers to the header for filenames.
An appropriate path name should be added before “bugs06.sparc”.

1.3 The script file for ‘backbugs’ (Sparc)

#!/bin/sh
# runs BUGS taking commands from command file
#
case $# in
   0) echo ’usage backbugs command_file’
      exit 1;;
   1) bugs06.sparc 32 bugs $1;;
   *) echo ’usage backbugs command_file’ ;;
esac
rm bugs.buf

An appropriate path name should be added before “bugs06.sparc”.
So to submit a series of commands from a file `job.cmd`, use the command `backbugs job.cmd`.

## 2 New Facilities in 0.6

### 2.1 Checkpoint command

A command now exists to save people wasting a whole run if a crash occurs. If you type, say, `checkpoint(1000)`, then after every 1000 iterations `BUGS` will

- write to file `bugs.res` the current parameter values
- if anything is being monitored, write the simulated values sequentially to `bugs.sif` (an ‘ind’ file) and `bugs.sof` (an ‘out’ file). This format is readable by `CODA`.

Note that `bugs.res` can be used as an initial value file for restarting a run.

### 2.2 Metropolis sampling

A general Metropolis-within-Gibbs routine can now be used for non-log-concave sampling. This routine uses a simple histogram-based proposal distribution (Ritter and Tanner, 1992), and therefore any parameter that requires Metropolis sampling must have bounded range, which is best set up using the `I(,)` construct on the prior distribution, although this is not necessary if a uniform or beta prior distribution are assumed. This is best adapted to as narrow a range as possible to bound the posterior distribution. The default number of bins in the histogram is 32, but this can be changed within the `BUGS` script (see Section 1.2).

If a bounded range is not given, an error message will say that `BUGS` is Unable to choose update method for node.

Improved Metropolis routines are being written for future versions, and will be available in `BUGS` for Windows.

### 2.3 Minor changes

1. Non-integer $r$ and $n$ can be used in binomial sampling.
2. Nodes are updated in the reverse order to their specification.

## 3 Corrected bugs from Version 0.5

1. The adaptive rejection sampling routine has been changed in the hope of avoiding the crashes that have occurred. However, this may have made it slower.
2. Compilation has been speeded up.
3. The `stats` command no longer overwrites the contents of `bugs1.out`.
4. The `I(,)` construct works for all univariate distributions (previously it was ignored when using, for example, the logistic)
5. The sd function has been fixed.

6. Some of the more annoying error messages have been fixed: for example a missing data or initial value file no longer leads to a memory fault crash.

4 Known restrictions still existing in Version 0.6

1. It is still not possible to place any structure on a covariance matrix given an inverse Wishart distribution. For bivariate normal distributions, such as an intercept and slope of a growth curve model, we recommend modelling as two univariate normals with, say, intercept being a covariate of slope.

2. The COSMOS example does not fully work.

5 Examples

5.1 Dugongs with Metropolis

This problem is described in Spiegelhalter et al. (1996b)[page 4], and comprises a non-linear and non-conjugate model:

\[ Y_i \sim \text{Normal}(\mu_i, \tau), \quad i = 1, \ldots, 27 \]
\[ \mu_i = \alpha - \beta \gamma^x_i \quad \alpha, \beta > 1; 0 < \gamma < 1. \]

This gives a non-log-concave distribution for \( \gamma \). The problem was previously handled by discretizing \( \gamma \), and specifying equal prior probabilities for each discrete value. The BUGS 0.6 code is shown below.

```plaintext
model dugongs;
const
   N = 27;  # number of observations
var
   x[N], Y[N], mu[N], alpha, beta, gamma, tau, sigma, U1, U2, U3;
data x, Y in "dugongs.dat";
init in "dugongs.in";
{
   for (i in 1:N) {
      mu[i] <- alpha - beta*pow(gamma, x[i]);
      Y[i] ~ dnorm(mu[i], tau)
   }
alpha ~ dnorm(0.0, 1.0E-4);
beta ~ dnorm(0.0, 1.0E-4);
tau ~ dgamma(1.0E-3, 1.0E-3); sigma <- 1.0/sqrt(tau);
gamma ~ dunif(0.5, 1.0);
# Transform alpha, beta and gamma to scale used by Carlin and Gelfand
U1 <- log(alpha);
U2 <- log(beta);
```
U3 <- logit(gamma);
}

We note that \( \gamma \) has been given a bounded domain by using a uniform prior distribution. BUGS 0.6 detects that the Metropolis sampler is required and reports during compilation: Metropolis method chosen for node gamma.

**Analysis**

After a 500 iteration burn-in, 1000 iterations took 31 seconds using the default 32 bins for the Metropolis sampler.

<table>
<thead>
<tr>
<th>U1 (log ( \alpha ))</th>
<th>U2 (log ( \beta ))</th>
<th>U3 (logit ( \gamma ))</th>
<th>( \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>C &amp; G posterior mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ratkowsky least squares estimate</td>
<td>0.975</td>
<td>-0.014</td>
<td>1.902</td>
</tr>
<tr>
<td>BUGS posterior mode (95% interval)</td>
<td>0.979</td>
<td>-0.029</td>
<td>1.896</td>
</tr>
<tr>
<td>Discretisation (0.933, 1.032)</td>
<td>(-0.180, 0.117)</td>
<td>(1.366, 2.364)</td>
<td>(0.074, 0.129)</td>
</tr>
<tr>
<td>Metropolis (0.931, 1.032)</td>
<td>(-0.031, 0.114)</td>
<td>(1.319, 2.398)</td>
<td>(0.075, 0.132)</td>
</tr>
</tbody>
</table>

(The C & G posterior refers to that in Carlin and Gelfand (1991)).

We note that the Metropolis sampler is considerably faster than using discretisation, and the results are virtually indistinguishable.

### 5.2 Epilepsy with hierarchical centering

This example is described in Spiegelhalter et al. (1996c) (page 30), in which convergence problems are noted. Gelfand et al. (1995) and Gelfand et al. (1996) discuss the method of hierarchical centering for such models, in which each stochastic variable is, as far as possible, considered as arising from a stochastic mean. In effect, covariates are entered as ‘high’ in the model as possible. They argue this procedure should often improve convergence, and further evidence is provided by Roberts and Sahu (1997).

For Model III in the epilepsy example, rather than having both random effects entering into a single regression for the Poisson mean, we may separate out the random effects to create an additional level on the model. The model is thus given by:

\[
\begin{align*}
 y_{jk} & \sim \text{Poisson}(m_{jk}) \\
 \log m_{jk} & = b_{jk} \\
 b_{jk} & \sim \text{Normal}(\mu_{jk}, \tau_b) \\
 \log \mu_{jk} & = \alpha_0 + \alpha_{Base} \log(\text{Base}_{j}/4) + \alpha_{Trt} \text{Trt}_{j} + \alpha_{Age} \text{Age}_{j} + \alpha_{V4} V4_{k} + b_{1j} \\
 b_{1j} & \sim \text{Normal}(0, \tau_{b1})
\end{align*}
\]
Coefficients and precisions are given independent “noninformative” priors. The appropriate graph is shown in Figure 1.

Figure 1: Graphical model for epil example, using a hierarchically centered parameterisation

Model specification for epil example with hierarchical centering. The part indicated +++++ is identical to that given in Spiegelhalter et al. (1996c) [page 30].

```r
for(j in 1:N) {
  for(k in 1:T) {
    + alpha.Trt * (Trt[j] - Trt.bar)
    + alpha.BT * (BT[j] - BT.bar)
    + alpha.Age * (log.Age[j] - log.Age.bar)
    + alpha.V4 * (V4[k] - V4.bar)
    + b1[j];
    y[j,k] ~ dpois(m[j,k]);
    log(m[j,k]) <- b[j,k];
    b[j,k] ~ dnorm(mu[j,k],tau.b);  # subject*visit random effects
  }
  b1[j] ~ dnorm(0.0,tau.b1);  # subject random effects
  +++++
```

Analysis
A burn-in of 3000 iterations was followed by a further 7000 iterations. This took approximately 30 minutes.
We have generally found that hierarchical centering leads to both quicker sampling and earlier convergence.

### 5.3 Orange trees - non-linear hierarchical models

This example is analysed in Lindstrom and Bates (1990) as an example of a mixed non-linear growth curve model. The data describe the growth of each of five orange trees, with measurements at seven common times:

<table>
<thead>
<tr>
<th>Tree</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
<th>$y_5$</th>
<th>$y_6$</th>
<th>$y_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>58</td>
<td>87</td>
<td>115</td>
<td>120</td>
<td>142</td>
<td>145</td>
</tr>
<tr>
<td>2</td>
<td>33</td>
<td>69</td>
<td>111</td>
<td>156</td>
<td>172</td>
<td>203</td>
<td>203</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>51</td>
<td>75</td>
<td>108</td>
<td>115</td>
<td>139</td>
<td>140</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>62</td>
<td>112</td>
<td>167</td>
<td>179</td>
<td>209</td>
<td>214</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>49</td>
<td>81</td>
<td>125</td>
<td>142</td>
<td>174</td>
<td>177</td>
</tr>
</tbody>
</table>

A logistic growth curve model, with an unknown maximum, is assumed. We first standardise the covariate $x_j$ to $z_j = (x_j - \bar{x})/\text{sd}(\bar{x})$ in order to improve convergence and stability of estimates, and to make the random effects assumptions more reasonable.

$y_{ij} \sim \text{Norm}(m_{ij}, \tau_e)$

$$m_{ij} = \frac{e^{\theta_{1j}}}{1 + e^{\theta_{1j} + \theta_{3}z_j}}$$

$\theta_{ik} \sim \text{Norm}(\mu_k, \tau_k)$

Lindstrom and Bates (1990) only take $\phi_{i1} = e^{\theta_{1j}}$ as a random effect with a Gaussian population distribution. We shall allow all three growth parameters to vary between trees; means and precisions are given independent “noninformative” priors.
Figure 2: Graph of the orange tree example.

As mentioned in Section 2.2, we need to specify a range for each of the $\theta$ parameters to be sampled using the Metropolis algorithm. It is convenient to first run a fixed-effect model in which five independent growth curves are fitted to the data, specifying only that the $\theta$’s are all between -20 and +20. This gives rise to estimates for $\theta$ that suggest generous lower and upper bounds of (4,6) for $\theta_{ii}$’s, (-2,0) for $\theta_{i2}$’s and (-3,0) for $\theta_{i3}$’s. These are then placed in the data file as the lower and upper vectors.

Model specification for orange example

```plaintext
code
model otree;
const
n = 7,
K = 5;
var
tauC, mu[3], tau[3], Y[K,n], m[K,n], phi[K,3], theta[K,3],
lower[3], upper[3], sigmaC, sigma[3], x[n], x.bar, x.sd;
data in "otree.dat";
inits in "otree.in";
{
x.bar <- mean(x[]);
x.sd <- sd(x[]);
for (i in 1:K) {
  for (j in 1:n) {
    Y[i, j] ~ dnorm(m[i, j], tauC)
  }
}
```

m[i, j] <- exp(theta[i, 1]) / (1 + exp(theta[i, 2] + theta[i, 3] * (x[j] - x.bar)/x.sd));

for (k in 1:3) {
theta[i, k] ~ dnorm(mu[k], tau[k])I(lower[k], upper[k])
}

tauC ~ dgamma(1.0E-3, 1.0E-3)
sigmaC <- 1 / sqrt(tauC)
for (k in 1:3) {
mu[k] ~ dnorm(0, 1.0E-4)
tau[k] ~ dgamma(1.0E-3, 1.0E-3)
sigma[k] <- 1 / sqrt(tau[k])
}

Analysis
A burn-in of 500 iterations followed by a further 1000 iterations took approximately 2.25 minutes.

<table>
<thead>
<tr>
<th>variable (θ_k)</th>
<th>mean (μ_k) ± SE</th>
<th>sd (σ_k) ± SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>θ_1</td>
<td>5.23 ± .11</td>
<td>.24 ± .12</td>
</tr>
<tr>
<td>θ_2</td>
<td>-64 ± .15</td>
<td>.14 ± .12</td>
</tr>
<tr>
<td>θ_3</td>
<td>-1.44 ± .13</td>
<td>.12 ± .11</td>
</tr>
</tbody>
</table>

From the size and standard deviations of the random effects σ’s, the assumption of Lindstrom and Bates (1990) that a random effect is only required for θ_1 appears reasonable.

5.4 ddIdC: a longitudinal Laird-Ware mixed model

Consider the Gaussian linear mixed model (Laird and Ware, 1982),

\[
Y_i = X_i \alpha + W_i \beta_i + \epsilon_i \\
\beta_i \sim N_q(0, V^{-1})
\]

where the \( Y_i \) are vectors of length \( n_i \) containing the observations on the \( i^{th} \) unit, and the \( \epsilon_i \) are error vectors of the same length independently distributed as \( N_{n_i}(0, \sigma^{-2} I_{n_i}) \), \( i = 1, \ldots, k \): note that all our Normal parameterisations are in terms of precisions. In this mixed model, \( X_i \) is an \( n_i \times p \) design matrix of covariates and \( \alpha \) is a corresponding \( p \times 1 \) vector of fixed effects. In contrast, \( W_i \) is a \( n_i \times q \) design matrix (\( q \) typically less than \( p \)), and \( \beta_i \) is a \( q \times 1 \) vector of subject-specific random effects. The \( \beta_i \) model the subject-specific means, as well as enabling the model to capture marginal dependence among the observations on the \( i^{th} \) unit. The hierarchical specification of this model is completed by adding the prior distributions \( \Omega \equiv V^{-1} \sim \text{Wishart}(R, \rho) \), \( \tau \equiv \sigma^{-2} \sim \Gamma(a, b) \), and \( \alpha \sim N_p(c, D) \).

We apply this model to continuous longitudinal data from a clinical trial originally described by Abrams et al. (1994), which compared the effectiveness of two antiretroviral drugs (didanosine, ddI, and zalcitabine, ddC) in 467 persons with advanced HIV infection. The response variable \( Y_{ij} \) for
patient \( i \) at time \( j \) is the square root of the patient’s CD4 count, a seriological measure of immune system health and prognostic factor for AIDS-related illness and mortality. The dataset records patient CD4 counts at study entry and again at 2, 6, 12, and 18 months after entry, though a great many of these observations are missing for many patients (the sample sizes at the five time points for the two drug groups are (230, 182, 153, 102, 22) and (236, 186, 157, 123, 14), respectively).

Following a Bayesian reanalysis of these data (Carlin, 1996; Carlin and Louis, 1996), we seek to fit model (1) where the \( j^{th} \) row of the patient \( i \)’s design matrix \( W_i \) takes the form

\[
w_{ij} = (1, t_{ij}, (t_{ij} - 2)^+),
\]

where \( t_{ij} \in \{0, 2, 6, 12, 18\} \) and \( z^+ = \max(z, 0) \). Thus the three columns of \( W_i \) correspond to individual-level intercept, slope, and possible change in slope after the two month visit (by which time the drugs are expected to produce a detectable benefit). We further account for the effect of two covariates by including them in the fixed effect design matrix \( X_i \). These covariates are \( d_i \), a binary variable indicating whether patient \( i \) received ddI (\( d_i = 1 \)) or ddC (\( d_i = 0 \)), and \( a_i \), a binary variable telling whether the patient was diagnosed as having AIDS at baseline (\( a_i = 1 \)) or not (\( a_i = 0 \)). Each of these covariates is allowed to influence the intercept, slope and change, and hence

\[
X_i = (W_i | d_i W_i | a_i W_i),
\]

so that \( p = 3q = 9 \).

We complete our model specification with minimally informative priors, taking care to ensure that they do not lead to improper posterior distributions for the variance components \( \sigma^2 \) and \( \Delta \). Following previous work, we set \( \rho = 24 \) and \( R = 24 \times \text{Diag}(2^2, (.25)^2, (.25)^2) \), which should preserve identifiability while still allowing the random effects a reasonable amount of freedom. For the prior on \( \sigma^2 \) we take \( a = 1, b = 100 \) (a prior with both mean and standard deviation equal to \((1/10)^2\)), while for \( \alpha \) we set

\[
c = (10, 0, 0, 0, 0, 0, -3, 0, 0), \quad \text{and}
\]

\[
\Delta = \text{Diag}(2^2, 1^2, 1^2, (1)^2, 1^2, 1^2, 1^2, 1^2, 1^2),
\]

a prior biased strongly away from 0 only for the baseline intercept, \( \alpha_4 \), and the intercept adjustment for a positive AIDS diagnosis, \( \alpha_7 \). This prior also forces the drug group intercept (i.e., the effect at baseline) \( \alpha_4 \) to be very small, since patients were assigned to drug group at random.

Here is the BUGS code to fit this model, where \( \text{ind} \) indexes the individual in the study, and \( i \) and \( j \) index the rows and columns of the design matrices, respectively. By placing \( \text{NA}'s \) in the data file, the \( W \) matrix is common to all individuals, but the \( X \) matrix is still individual-specific.

```plaintext
model ddIddC;
const
   N = 467,  # number of patients
   s = 5,    # number of time points
   q = 3,    # number of random effects
   p = 9;    # number of fixed effects
var
   X[N,s,p],W[s,q],Y[N,s],alpha[p],beta[N,q],d[N],a[N],Omega[q,q],V[q,q],
   Sigma2[q,q],sigma,tau,R[q,q],rho,c[p],Omega.alpha[p,p],
   mu[N,s],mu.beta[q];
```
Figure 3: Graph of the ddIddc example.
data d,a in "drugAIDS.dat", Y in "Y.dat",
W in "W.dat", c in "priorMean.dat", Omega.alpha in "priorPrec.dat";

init in "ddidC.in";

{ for (ind in 1:N) {
  for (i in 1:s) {

    for (j in 1:q) {
      X[ind,i,j] <- W[i,j];
      X[ind,i,j+3] <- d[ind]*W[i,j];
      X[ind,i,j+6] <- a[ind]*W[i,j];
    }

    Y[ind,i] ~ dnorm(mu[ind,i],tau);
    mu[ind,i] <- inprod(X[ind,i,],alpha[])+inprod(W[i,],beta[ind,]);
  }

  beta[ind,] ~ dmnorm(mu.beta[],Omega[,]);
  # trivariate Normal
}

tau ~ dgamma(1, 100); sigma <- 1.0/sqrt(tau);
Omega[,] ~ dwish(R[,]|24); # Wishart prior on precision matrix
R[1,1] <- 96.0; R[1,2] <- 0.0; R[1,3] <- 0.0;
R[2,1] <- 0.0; R[2,2] <- 1.5; R[2,3] <- 0.0;
R[3,1] <- 0.0; R[3,2] <- 0.0; R[3,3] <- 1.5;
V[,] <- inverse(Omega[,]);

mu.beta[1] <- 0.0; mu.beta[2] <- 0.0; mu.beta[3] <- 0.0;
# for (j in 1:p){alpha[j] ~ dnorm(c[j],Omega.alpha[,j]);} # univ norms
alpha[] ~ dmnorm(c[],Omega.alpha[,]);
# mv normal -- better convergence!

Running this BUGS code for 5000 iterations produces the summaries in Table 1. The results are quite comparable to those given by Carlin and Louis (1996)[pp.280-281]. Interestingly, this original work took several hundred lines of code in Fortran 77, augmented with IMSL subroutine calls for matrix manipulation and random variate generation – a stark contrast with the fewer than 40 lines of BUGS code above.

The single line that is commented out in the above BUGS code can be used to specify the (independence) prior for α componentwise using dnorm, instead of all at once using dmnorm. While mathematically equivalent, Table 2 shows that the univariate specification to be inferior in terms of convergence speed, since BUGS then updates the α_i one at a time, instead of as a vector. Laboring against the cross-correlations within this vector, overall performance deteriorates.

We remark that the hierarchically centered version of this model recommended for this dataset by Gelfand et al. (1995), namely

$$\eta_i = \alpha^{(0)} + d_i\alpha^{(d)} + a_i\alpha^{(a)} + \beta_i,$$

where $$\alpha^{(0)} = (\alpha_1, \alpha_2, \alpha_3),$$ $$\alpha^{(d)} = (\alpha_4, \alpha_5, \alpha_6),$$ and $$\alpha^{(a)} = (\alpha_7, \alpha_8, \alpha_9),$$ is not possible within the current version of BUGS. This is because BUGS cannot calculate the proper multivariate normal mean and precision matrix when the “data” (in the centered version, the $$\eta_i$$) are not univariate, unless
the data mean is identical to the multivariate normal prior. **BUGS** can however accommodate some simpler, univariate centering forms, as in the revised **epil** example.

5.5 **PK: a nonlinear population pharmacokinetic model**

Wakefield et al. (1994) consider the data in Table 3, which record the plasma concentration \( Y_{ij} \) of the drug Cadralazine at various time lags \( x_{ij} \) following the administration of a single dose of 30 mg in 10 cardiac failure patients. Here, \( i = 1, \ldots, 10 \) indexes the patient, while \( j = 1, \ldots, n_i \) indexes the observations, \( 5 \leq n_i \leq 8 \). These authors suggest a “one-compartment” nonlinear pharmacokinetic model wherein the mean plasma concentration \( \eta_{ij}(x_{ij}) \) is given by

\[
\eta_{ij}(x_{ij}) = 30 \alpha_i^{-1} \exp(-\beta_i x_{ij}/\alpha_i).
\]

Subsequent unpublished work by these same authors suggests this model is best fit on the log scale. That is, we suppose

\[
Z_{ij} = \log Y_{ij} = \log \eta_{ij}(x_{ij}) + \epsilon_{ij},
\]

where \( \epsilon_{ij} \sim N(0, \tau) \). The mean structure for the \( Z_{ij} \)’s thus emerges as

\[
\log \eta_{ij}(x_{ij}) = \log \left[ 30 \alpha_i^{-1} \exp(-\beta_i x_{ij}/\alpha_i) \right] = \log 30 - \log \alpha_i - \beta_i x_{ij}/\alpha_i = \log 30 - a_i - \exp(b_i - a_i) x_{ij},
\]

where \( a_i = \log \alpha_i \) and \( b_i = \log \beta_i \).

Following the analysis by Wakefield et al. (1994), we assume the subject-specific random effects \( \boldsymbol{\theta}_i = (a_i, b_i)' \) are i.i.d. from a \( N_2(\boldsymbol{\mu}, \Omega) \) distribution, where \( \boldsymbol{\mu} = (\mu_a, \mu_b) \). These authors also

<table>
<thead>
<tr>
<th>parameter</th>
<th>mean</th>
<th>95% interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_1 )</td>
<td>9.938E+0</td>
<td>9.338E+0</td>
</tr>
<tr>
<td>( \alpha_2 )</td>
<td>-3.817E-2</td>
<td>-2.532E-1</td>
</tr>
<tr>
<td>( \alpha_3 )</td>
<td>-1.437E-1</td>
<td>-3.890E-1</td>
</tr>
<tr>
<td>( \alpha_4 )</td>
<td>9.326E-3</td>
<td>-1.767E-1</td>
</tr>
<tr>
<td>( \alpha_5 )</td>
<td>3.280E-1</td>
<td>9.792E-2</td>
</tr>
<tr>
<td>( \alpha_6 )</td>
<td>-3.596E-1</td>
<td>-6.270E-1</td>
</tr>
<tr>
<td>( \alpha_7 )</td>
<td>-4.279E+0</td>
<td>-4.996E+0</td>
</tr>
<tr>
<td>( \alpha_8 )</td>
<td>-3.297E-1</td>
<td>-5.607E-1</td>
</tr>
<tr>
<td>( \alpha_9 )</td>
<td>3.835E-1</td>
<td>1.139E-1</td>
</tr>
<tr>
<td>( \beta_{8,1} )</td>
<td>-7.618E+0</td>
<td>-9.738E+0</td>
</tr>
<tr>
<td>( \beta_{8,2} )</td>
<td>-3.349E-1</td>
<td>-8.263E-1</td>
</tr>
<tr>
<td>( \beta_{8,3} )</td>
<td>3.071E-1</td>
<td>-2.040E-1</td>
</tr>
<tr>
<td>( \sigma )</td>
<td>1.681E+0</td>
<td>1.587E+0</td>
</tr>
<tr>
<td>( \Omega_{1,1} )</td>
<td>1.048E-1</td>
<td>6.838E-2</td>
</tr>
<tr>
<td>( \Omega_{1,2} )</td>
<td>-1.591E-1</td>
<td>-6.515E-1</td>
</tr>
</tbody>
</table>

**Table 1:** Posterior summaries, ddf/ddC data model
<table>
<thead>
<tr>
<th>parameter</th>
<th>$\alpha$ updating method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>componentwise vector</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>0.979 0.798</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>0.995 0.194</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>0.994 0.207</td>
</tr>
<tr>
<td>$\alpha_4$</td>
<td>0.686 0.204</td>
</tr>
<tr>
<td>$\alpha_5$</td>
<td>0.993 0.436</td>
</tr>
<tr>
<td>$\alpha_6$</td>
<td>0.993 0.408</td>
</tr>
<tr>
<td>$\alpha_7$</td>
<td>0.972 0.811</td>
</tr>
<tr>
<td>$\alpha_8$</td>
<td>0.991 0.134</td>
</tr>
<tr>
<td>$\alpha_9$</td>
<td>0.988 0.154</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>0.529 0.530</td>
</tr>
<tr>
<td>$D_{11}$</td>
<td>0.385 0.388</td>
</tr>
<tr>
<td>$D_{21}$</td>
<td>0.942 0.942</td>
</tr>
<tr>
<td>$D_{22}$</td>
<td>0.891 0.891</td>
</tr>
<tr>
<td>$D_{31}$</td>
<td>0.934 0.934</td>
</tr>
<tr>
<td>$D_{32}$</td>
<td>0.967 0.967</td>
</tr>
<tr>
<td>$D_{33}$</td>
<td>0.917 0.918</td>
</tr>
</tbody>
</table>

Table 2: Lag 1 sample autocorrelations, algorithms for ddI/ddC data model

recommend the usual conjugate prior specification, namely $\mu \sim N_2(\lambda, C)$, $\tau_i^{iid} \sim G(\nu_0/2, \nu_0\tau_0/2)$, and $\Omega \sim $ Wishart $(R, \rho)$. Since the full conditional distributions of the random effects $\theta_i$ are neither simple conjugate forms nor guaranteed to be log-concave, the new Metropolis capability of **BUGS** 0.6 is required. This Metropolis routine requires bounds to be placed on variables using the $I(,)$ construction, so unfortunately, the model for $\theta_i$ cannot be specified bivariately as above, since **BUGS** currently cannot handle multivariate range restrictions. However, the model may still be specified in **BUGS** using the product formulation of the bivariate normal, namely

$$a_i \sim N(\mu_a, \tau_a) I(L_a, U_a)$$
$$b_i | a_i \sim N(k_0 + k_1(a_i - c), \tau_b) I(L_b, U_b),$$

where $(L_a, U_a)$ and $(L_b, U_b)$ are broad truncation regions to enable the grid-based Metropolis algorithm, and $c$ is a constant used to roughly center the $a_i$'s (hence reduce correlation between the intercept $k_0$ and slope $k_1$). Under this formulation, we replace the normal prior for $\mu$ and the Wishart prior for $\Omega$ with gamma priors for $\tau_a$ and $\tau_b$ and normal priors for $\mu_a$, $k_0$ and $k_1$.

The **BUGS** code to fit this model follows. As can be seen, we adopt the tuning constants $c = 3$, $L_a = L_b = -5$, and $U_a = U_b = 10$. The latter values comfortably contain all the posterior mass for the $a_i$ and $b_i$; significantly more widely dispersed values (say, $L_a = L_b = -50$ and $U_a = U_b = 100$) do in fact lead to sharp drops in the Metropolis acceptance rate, hence reductions in efficiency.

```r
model PK;
c
const
  N = 10, # number of patients
  T = 8; # number of time points
var
```
Figure 4: Graph of the PK example.
Table 3: Cadralazine concentration data

\[
X_T, Z_T, \theta_N, N_2, a_N, b_N, \lnu_N, \tau_N, \sigma_N, Y_N, T, \\
u.a, tau.a, mub.N, tau.b, int.b, slope.b, mu.b;
\]
data Z in "PK.Z.dat", X in "PKX.dat";

\{
    for (i in 1:N) {
        for (j in 1:T) {
            Z[i,j] ~ dnorm(\lnu[i,j], tau[i]);
            Y[i,j] <- exp(Z[i,j]);
            \lnu[i,j] <- log(30) - a[i] - exp(b[i]-a[i])*X[j];
        }  
    }  
    a[i] ~ dnorm(mu.a, tau.a) I(-5, 10);
    b[i] ~ dnorm(mub[i], tau.b) I(-5, 10);
    mub[i] <- int.b + slope.b * (a[i] - 3.0);  
    tau[i] ~ dgamma(.0001, .0001);  
    sigma[i] <- 1.0/sqrt(tau[i]);
\}

\}

mu.a ~ dnorm(0.0, 0.0001);
int.b ~ dnorm(0.0, 0.0001);  
slope.b ~ dnorm(0.0, 0.0001);
mu.b <- int.b + slope.b * (mu.a - 3.0);
tau.a ~ dgamma(1, 0.04); tau.b ~ dgamma(1, 0.04);  
\}

Using a sequence of univariate Metropolis (Gaussian proposals) and Gibbs steps, Sargent et al. (1997) fit the original Wishart formulation of this model using the priors recommended by Wakefield et al. (1994), namely \( \nu_0 = 0, \lambda = 0, C^{-1} = \text{Diag}(0.01, 0.01), \rho = 2, \) and \( R = \rho * \text{Diag}(0.04, 0.04). \) We attempt a comparable prior in our formulation by taking \( G(0.0001, 0.0001) \) priors for the \( \tau_i, N(0, 0.0001) \) priors for \( \mu_\alpha, k_0 \) and \( k_1, \) and \( G(1, 0.04) \) priors for \( \tau_\alpha \) and \( \tau_\beta. \)

Running this BUGS code for 5000 iterations (following a 250 iteration burn-in period) produces
<table>
<thead>
<tr>
<th>parameter</th>
<th>BUGS V0.6</th>
<th>Sargent et al. (1997)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>sd</td>
</tr>
<tr>
<td>$a_1$</td>
<td>2.956</td>
<td>0.0479</td>
</tr>
<tr>
<td>$a_2$</td>
<td>2.692</td>
<td>0.0772</td>
</tr>
<tr>
<td>$a_3$</td>
<td>2.970</td>
<td>0.1106</td>
</tr>
<tr>
<td>$a_8$</td>
<td>2.828</td>
<td>0.1417</td>
</tr>
<tr>
<td>$b_1$</td>
<td>1.259</td>
<td>0.0335</td>
</tr>
<tr>
<td>$b_2$</td>
<td>0.234</td>
<td>0.0648</td>
</tr>
<tr>
<td>$b_7$</td>
<td>1.157</td>
<td>0.0879</td>
</tr>
<tr>
<td>$b_8$</td>
<td>0.936</td>
<td>0.1458</td>
</tr>
<tr>
<td>$\tau_1$</td>
<td>362.4</td>
<td>260.4</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>84.04</td>
<td>57.60</td>
</tr>
<tr>
<td>$\tau_7$</td>
<td>18.87</td>
<td>12.07</td>
</tr>
<tr>
<td>$\tau_8$</td>
<td>2.119</td>
<td>1.139</td>
</tr>
<tr>
<td>$\mu_a$</td>
<td>2.838</td>
<td>0.0715</td>
</tr>
<tr>
<td>$\mu_b$</td>
<td>1.051</td>
<td>0.1472</td>
</tr>
<tr>
<td>$k_0$</td>
<td>1.324</td>
<td>0.2087</td>
</tr>
<tr>
<td>$k_1$</td>
<td>1.773</td>
<td>1.154</td>
</tr>
<tr>
<td>$\tau_6$</td>
<td>44.58</td>
<td>23.24</td>
</tr>
<tr>
<td>$\tau_9$</td>
<td>14.75</td>
<td>12.72</td>
</tr>
<tr>
<td>$Y_{2,8}$</td>
<td>0.1338</td>
<td>0.0339</td>
</tr>
<tr>
<td>$Y_{7,8}$</td>
<td>0.00891</td>
<td>0.00443</td>
</tr>
</tbody>
</table>

Table 4: Posterior summaries and lag 1 sample autocorrelations, PK data model

the posterior summaries and lag 1 sample autocorrelations given in Table 4. Also shown are the results produced by Sargent et al. (1997), to which the BUGS results are quite comparable, given the slight differences in model and prior formulation. (Results for $\Omega_{1,1}$ in the Wishart model are given in the $\tau_9$ row of the table; however, results for $\Omega_{2,2}$ are not shown in the $\tau_9$ row since $\tau_9$ is a conditional precision, given the $a_i$.) The relatively large posterior means for $\tau_7$ and $\tau_8$ (and correspondingly large posterior variances for $a_7, a_8, b_7,$ and $b_8$) at first seem counter-intuitive, since these two patients had the most data available for study. However, their final 2 to 3 observations fit the overall model poorly (with those for patient 8 not even being monotone), explaining this oddity. Finally (and relatedly), note the predicted values of the final observations for patients 2 (whose clearance rate is the slowest) and 7 (whose rate is amongst the fastest). The former has mean somewhat larger than that suggested by the posterior predictive distribution under the “power model” fit on the original (unlogged) scale by Wakefield et al. (1994).

Acknowledgements

We are grateful to all those people who have proposed ways of improving the BUGS software, and apologise for implementing so few of their suggestions. We are particularly grateful for Brad Carlin for providing the dd1ddc and the PK examples.
References


