June 26, 2008

Package glmmAK: Example Epileptic

Arnošt Komárek

Department of Probability and Mathematical Statistics, Charles University in Prague
Sokolovská 83, CZ–186 75, Praha 8, Czech Republic

E-mail: arnost.komarek@mff.cuni.cz

This document shows how to perform the analysis of the Epileptic data presented in Komárek and Lesaffre (2008) using the functions of the package glmmAK. To process the MCMC output, we also extensively use the coda package (Plummer et al., 2006). It is assumed that the user reads Komárek and Lesaffre's paper first. In this manual, the same notation is used, often without redefining it.

This manual especially supplements the help pages of the following functions of the package glmmAK:

- logpoissonRE,
- summaryGspline2.

The user is encouraged to take a look on the manual pages of these functions first! You can try

```
> help(logpoissonRE, package = glmmAK, htmlhelp = TRUE)
> help(summaryGspline2, package = glmmAK, htmlhelp = TRUE)
```

1 Getting started

We start by loading the package, specifying the working directory and loading the data. Note that data epileptic are the original data as reported by Thall and Vail (1990) and data epilepticBC are data where the variables are transformed to fit the models presented by Breslow and Clayton (1993) and Komárek and Lesaffre (2008).

```
> library(glmmAK)
> root <- "/home/komarek/Rlib/glmmAK/Doc/"
> setwd(root)
> data(epileptic)
> data(epilepticBC)
```

Brief summary of the data:

```
> summary(epileptic)
```

```
       id           seizure           visit       trt              age       
 Min.   :101.0   Min.   :  0.00   Min.   :0   Min.   :0.0000   Min.   :18.00  
 1st Qu.:118.0   1st Qu.:  3.00   1st Qu.:1   1st Qu.:0.0000   1st Qu.:23.00  
 Median :147.0   Median :  6.00   Median :2   Median :1.0000   Median :28.00  
 Mean   :168.4   Mean   : 12.85   Mean   :2   Mean   :0.5254   Mean   :28.34  
 3rd Qu.:217.0   3rd Qu.: 14.50   3rd Qu.:3   3rd Qu.:1.0000   3rd Qu.:32.00  
 Max.   :238.0   Max.   :151.00   Max.   :4   Max.   :1.0000   Max.   :42.00  
```

```
> summary(epilepticBC)
```

```
       id            visit          seizure0           age           Seizure       
 Min.   :101.0   Min.   :1.00   Min.   :  6.00   Min.   :18.00   Min.   :  0.000  
 1st Qu.:118.0   1st Qu.:1.75   1st Qu.: 12.00   1st Qu.:23.00   1st Qu.:  2.750  
 Median :147.0   Median :2.50   Median : 22.00   Median :28.00   Median :  4.000  
 Mean   :168.4   Mean   :2.50   Mean   : 31.22   Mean   :28.34   Mean   :  8.263  
 3rd Qu.:217.0   3rd Qu.:3.25   3rd Qu.: 41.00   3rd Qu.:32.00   3rd Qu.:  9.000  
 Max.   :238.0   Max.   :4.00   Max.   :151.00   Max.   :42.00   Max.   :102.000  
      Base              Trt             Base.Trt            Age            Visit       
 Min.   :0.4055   Min.   :0.0000   Min.   :0.0000   Min.   :2.890   Min.   :-0.30  
 1st Qu.:1.0986   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:3.135   1st Qu.:-0.15  
 Median :1.7047   Median :1.0000   Median :0.5596   Median :3.332   Median : 0.00  
 Mean   :1.7680   Mean   :0.5254   Mean   :0.9484   Mean   :3.320   Mean   : 0.00  
 3rd Qu.:2.3273   3rd Qu.:1.0000   3rd Qu.:1.7918   3rd Qu.:3.466   3rd Qu.: 0.15  
 Max.   :3.6310   Max.   :1.0000   Max.   :3.6310   Max.   :3.738   Max.   : 0.30  
```

2 Data and models

Thall and Vail (1990) report the data from a longitudinal study of seizures in epileptic patients. In total, $N = 59$ patients were randomized to receive either the antiepileptic drug progabide (Trt=1) or placebo (Trt=0), as an adjuvant to standard chemotherapy. Patients underwent four successive postrandomization clinic visits. For the ith patient, the response variable $Y_{i,l}$ denotes the number of seizures during the 2-weeks period before the lth visit. GLMM's to this data were fitted using an approximate method of penalized quasilikelihood (PQL) under the assumption of normality of random effects by Breslow and Clayton (1993). We will specify the linear predictor of the GLMM in the same way as a way equivalent to Breslow and Clayton's Model IV and will consider two PGM GLMM and two Normal GLMM's. In the following, let Visit be the centered visit time in weeks divided by 10 (-0.3, -0.1, 0.1, 0.3), Base be the logarithm of $\frac{1}{4}$ the 8-week prerandomization seizure count and Age be the logaritm of age in years.

2.1 PGM GLMM, not hierarchically centered

PGM GLMM, not hierarchically centered model is the following:

$$\begin{aligned}&\log\bigl\{\mathrm{E}(Y_{i,l}\,|\,\boldsymbol{\beta},\,\boldsymbol{b}_i)\bigr\} =\\ &\qquad \beta_1 + \beta_2\mathsf{Visit}_{i,l} + \beta_3\mathsf{Base}_i + \beta_4\mathsf{Trt}_i + \beta_5\mathsf{Base}_i\cdot\mathsf{Trt}_i + \beta_6\mathsf{Age}_i + b_{i,1} + b_{i,2}\mathsf{Visit}_{i,l}, \quad (1)\end{aligned}$$

where

$$\boldsymbol{b}_i = \begin{pmatrix} b_{i,1}\\ b_{i,2}\end{pmatrix} \overset{\text{i.i.d.}}{\sim} \sum_{j_1=-K_1}^{K_1}\sum_{j_2=-K_2}^{K_2} w_{j_1,j_2}(\boldsymbol{a})\mathcal{N}_2\left(\begin{pmatrix}\tau_1\mu_{1,j_1}\\ \tau_2\mu_{2,j_2}\end{pmatrix},\,\begin{pmatrix}(\tau_1\sigma_1)^2 & 0\\ 0 & (\tau_2\sigma_2)^2\end{pmatrix}\right)$$

$$(i = 1,\dots,N).$$

In a sequel, we will denote this model as **PGM GLMM(nhc)**.

The results of this model are shown in Komárek and Lesaffre (2008).

2.2 PGM GLMM, hierarchically centered

PGM GLMM, hierarchically centered model is the following:

$$\log\bigl\{\mathrm{E}(Y_{i,l}\,|\,\boldsymbol{\beta},\,\boldsymbol{b}_i)\bigr\} = \beta_3\mathsf{Base}_i + \beta_4\mathsf{Trt}_i + \beta_5\mathsf{Base}_i\cdot\mathsf{Trt}_i + \beta_6\mathsf{Age}_i + b_{i,1} + b_{i,2}\mathsf{Visit}_{i,l}, \qquad (2)$$

where

$$\boldsymbol{b}_i = \begin{pmatrix} b_{i,1}\\ b_{i,2}\end{pmatrix} \overset{\text{i.i.d.}}{\sim} \begin{pmatrix}\alpha_1\\ \alpha_2\end{pmatrix} + \sum_{j_1=-K_1}^{K_1}\sum_{j_2=-K_2}^{K_2} w_{j_1,j_2}(\boldsymbol{a})\mathcal{N}_2\left(\begin{pmatrix}\tau_1\mu_{1,j_1}\\ \tau_2\mu_{2,j_2}\end{pmatrix},\,\begin{pmatrix}(\tau_1\sigma_1)^2 & 0\\ 0 & (\tau_2\sigma_2)^2\end{pmatrix}\right)$$

$$(i = 1,\dots,N).$$

In a sequel, we will denote this model as **PGM GLMM(hc)**.

2.3 Normal GLMM, not hierarchically centered

Normal GLMM, not hierarchically centered model is the following:

$$\begin{aligned}\log\bigl\{\mathrm{E}(Y_{i,l}\,|\,\boldsymbol{\beta},\,\boldsymbol{b}_i)\bigr\} = \\ \beta_1 + \beta_2\mathsf{Visit}_{i,l} + \beta_3\mathsf{Base}_i + \beta_4\mathsf{Trt}_i + \beta_5\mathsf{Base}_i\cdot\mathsf{Trt}_i + \beta_6\mathsf{Age}_i + b_{i,1} + b_{i,2}\mathsf{Visit}_{i,l},\end{aligned} \tag{3}$$

where

$$\boldsymbol{b}_i = \begin{pmatrix} b_{i,1} \\ b_{i,2}\end{pmatrix} \overset{\text{i.i.d.}}{\sim} \mathcal{N}_2\left(\begin{pmatrix}0\\0\end{pmatrix},\, \underbrace{\begin{pmatrix} d_{1,1} & d_{2,1} \\ d_{2,1} & d_{2,2}\end{pmatrix}}_{\mathbb{D}}\right) \qquad (i = 1,\dots,N).$$

In a sequel, we will denote this model as **Normal GLMM(nhc)**.

The results of this model are shown in Komárek and Lesaffre (2008).

2.4 Normal GLMM, hierarchically centered

Normal GLMM, hierarchically centered model is the following:

$$\log\bigl\{\mathrm{E}(Y_{i,l}\,|\,\boldsymbol{\beta},\,\boldsymbol{b}_i)\bigr\} = \beta_3\mathsf{Base}_i + \beta_4\mathsf{Trt}_i + \beta_5\mathsf{Base}_i\cdot\mathsf{Trt}_i + \beta_6\mathsf{Age}_i + b_{i,1} + b_{i,2}\mathsf{Visit}_{i,l}, \tag{4}$$

where

$$\boldsymbol{b}_i = \begin{pmatrix} b_{i,1} \\ b_{i,2}\end{pmatrix} \overset{\text{i.i.d.}}{\sim} \mathcal{N}_2\left(\begin{pmatrix}\alpha_1\\ \alpha_2\end{pmatrix},\, \underbrace{\begin{pmatrix} d_{1,1} & d_{2,1} \\ d_{2,1} & d_{2,2}\end{pmatrix}}_{\mathbb{D}}\right) \qquad (i = 1,\dots,N).$$

In a sequel, we will denote this model as **Normal GLMM(hc)**.

2.5 Remarks

From the probabilistic point of view, PGM GLMM(nhc) is indeed equivalent to PGM GLMM(hc) and Normal GLMM(nhc) is equivalent to Normal GLMM(hc).

3 Specification of the prior distributions

Choices for the prior distributions are passed as `list` objects to the function `logpoissonRE`. In this Section, we create objects holding the prior information for considered models.

3.1 Prior for the fixed effects $\boldsymbol{\beta}$

In all models, we will assume that the prior distribution for the components of the vector of fixed effects $\boldsymbol{\beta}$ is a product of independent normal distributions $\mathcal{N}(0,\,10\,000)$:

```
> prior.fixed <- list(mean = 0, var = 10000)
```

3.2 Prior for the parameters of the penalized Gaussian mixture in the PGM GLMM's

For the PGM GLMM's (1) and (2), the following choices of the parameters defining the PGM will be used: $K_1 = K_2 = 15$, that is, $2 \cdot 15 + 1 = 31$ knots in each margin. Further, the distance between the two consecutive knots in each margin will be $\delta_1 = \delta_2 = 0.3$, that is, the knots are

$$\boldsymbol{\mu}_1 = \{\mu_{1,-15}, \dots, \mu_{1,15}\} = \{j_1\delta_1 :\; j_1 = -15, \dots, 15\} = \{-4.5,\, -4.2, \dots, 4.2,\, 4.5\},$$
$$\boldsymbol{\mu}_2 = \{\mu_{2,-15}, \dots, \mu_{2,15}\} = \{j_2\delta_2 :\; j_2 = -15, \dots, 15\} = \{-4.5,\, -4.2, \dots, 4.2,\, 4.5\}.$$

The basis standard deviation will be the same in both margins and equal to 0.2, i.e., $\sigma_1 = \sigma_2 = 0.2$.

The prior distribution for the transformed PGM weights $\boldsymbol{a}$ will be the intrinsic Gaussian Markov random field (IGMRF) based on the 3rd order (CARorder=3) differences between the consecutive weights in each margin, i.e.,

$$p(\boldsymbol{a}\,|\,\boldsymbol{\lambda}) \propto \exp\Bigl\{-\frac{\lambda_1}{2}\sum_{j_2=-K_2}^{K_2}\;\sum_{j_1=-K_1+3}^{K_1}\bigl(a_{j_1,j_2} - 3a_{j_1-1,j_2} + 3a_{j_1-2,j_2} - a_{j_1-3,j_2}\bigr)^2 \\ -\frac{\lambda_2}{2}\sum_{j_1=-K_1}^{K_1}\;\sum_{j_2=-K_2+3}^{K_2}\bigl(a_{j_1,j_2} - 3a_{j_1,j_2-1} + 3a_{j_1,j_2-2} - a_{j_1,j_2-3}\bigr)^2\Bigr\},$$

where $\boldsymbol{\lambda} = (\lambda_1,\, \lambda_2)'$ are the smoothing hyperparameters.

For the smoothing hyperparameters λ_1 and λ_2 independent gamma priors Gamma(1, 0.005) will be used. The transformed weights $\boldsymbol{a}$ will be updated using the slice sampling of Neal (2003). All above information is stored in a `list`:

```
> prior.gspline <- list(K = 15, delta = 0.3, sigma = 0.2, CARorder = 3,
+     Ldistrib = "gamma", Lequal = FALSE, Lshape = 1, LinvScale = 0.005,
+     AtypeUpdate = "slice")
```

It is also possible to use different grids of knots and/or different basis standard deviations in each margin and/or different priors for the smoothing hyperparameters λ_1 and λ_2. For example, the choices $K_1 = 15$, $K_2 = 10$, $\delta_1 = 0.3$, $\delta_2 = 0.6$, $\sigma_1 = 0.2$, $\sigma_2 = 0.4$, $\lambda_1 \sim \text{Gamma}(1,\, 0.005)$, $\lambda_2 \sim \text{Gamma}(0.001,\, 0.001)$ would be specified in the following alternative:

```
> prior.gspline.Alternative <- list(K = c(15, 10), delta = c(0.3,
+     0.6), sigma = c(0.2, 0.4), CARorder = 3, Ldistrib = "gamma",
+     Lequal = FALSE, Lshape = c(1, 0.001), LinvScale = c(0.005, 0.001),
+     AtypeUpdate = "slice")
```

3.3 Prior for the remaining parameters of the random effects distribution in the PGM GLMM's

In both PGM GLMM's (1) and (2) we still have to specify prior choices for the PGM scale parameter vector $\boldsymbol{\tau} = (\tau_1,\, \tau_2)'$, in the PGM GLMM(hc) (2) we also have to specify the prior distribution for the PGM location $\boldsymbol{\alpha} = (\alpha_1,\, \alpha_2)'$. We will use the following priors:

$$\begin{aligned}\tau_1^{-2} &\sim \text{Gamma}(1,\, 0.005), &\quad \tau_2^{-2} &\sim \text{Gamma}(1,\, 0.005),\\ \alpha_1 &\sim \mathcal{N}(0,\, 10\,000), &\quad \alpha_2 &\sim \mathcal{N}(0,\, 10\,000),\end{aligned}$$

which in the case of the PGM GLMM(nhc) is in R specified as

```
> prior.random.gspl.nhc <- list(Ddistrib = "gamma", Dshape = 1, DinvScale = 0.005)
```

and in the case of the PGM GLMM(hc) as

```
> prior.random.gspl.hc <- list(Mdistrib = "normal", Mmean = 0, Mvar = 10000,
+     Ddistrib = "gamma", Dshape = 1, DinvScale = 0.005)
```

Alternatively, one can assume the uniform prior for the PGM scale parameters τ_1 and τ_2 which is often prefered to the gamma prior, see Gelman (2006) for the discussion of this point. For example, the prior distribution

$$\tau_1 \sim \text{Unif}(0,\, 100), \qquad \tau_2 \sim \text{Unif}(0,\, 200),$$

is specified in the following way:

```
> prior.random.gspl.nhc.Unif <- list(Ddistrib = "sduniform", Dupper = c(100,
+     200))
```

3.4 Prior for the parameters of the random effects distribution in the PGM GLMM's

In both Normal GLMM's (3) and (4) we have to specify prior distribution for the covariance matrix $\mathbb{D}$ of the random effects and in the Normal GLMM(hc) (4) also the prior for the mean $\boldsymbol{\alpha}$ of the random intercept. We will use the following priors:

$$\begin{aligned}\mathbb{D}^{-1} &\sim \text{Wishart}\left(2,\, \begin{pmatrix}0.005 & 0\\ 0 & 0.005\end{pmatrix}^{-1}\right),\\ \alpha_1 \sim \mathcal{N}(0,\, 10\,000), &\quad \alpha_2 \sim \mathcal{N}(0,\, 10\,000),\end{aligned}$$

where the Wishart distribution is parametrized in the same way as in Gelman et al. (2004), that is, a priori

$$\mathrm{E}\big(\mathbb{D}^{-1}\big) = 2\begin{pmatrix}0.005 & 0\\ 0 & 0.005\end{pmatrix}^{-1}.$$

These prior distributions are specified in R, in the case of the **Normal GLMM(nhc)** as

```
> prior.random.norm.nhc <- list(Ddistrib = "wishart", Ddf = 2, DinvScale = 0.005)
```

and in the case of the **Normal GLMM(hc)** as

```
> prior.random.norm.hc <- list(Mdistrib = "normal", Mmean = 0, Mvar = 10000,
+     Ddistrib = "wishart", Ddf = 2, DinvScale = 0.005)
```

4 MCMC simulation

Having specified the prior distribution we are almost ready to start the MCMC simulation to sample from the posterior distribution of the model parameters.

4.1 Directories to store the chains

For each considered model, we create one directory as a subdirectory of root/chEpileptic which will afterwards be used to store the sampled chains. Creation of directories can of course be performed outside R as well.

```
> if (!("chEpileptic" %in% dir(root))) dir.create(paste(root, "chEpileptic",
+     sep = ""))
> dirNames <- c("PGM_nhc", "PGM_hc", "Normal_nhc", "Normal_hc")
> dirPaths <- paste(root, "chEpileptic/", dirNames, "/", sep = "")
> for (i in 1:length(dirPaths)) {
+     if (!(dirNames[i] %in% dir(paste(root, "chEpileptic", sep = ""))))
+         dir.create(dirPaths[i])
+ }
> names(dirPaths) <- c("PGM_nhc", "PGM_hc", "Normal_nhc", "Normal_hc")
```

That is, the chains for considered models will be stored in the following directories:

```
> print(dirPaths)
```

```
                                                PGM_nhc
   "/home/komarek/Rlib/glmmAK/Doc/chEpileptic/PGM_nhc/"
                                                 PGM_hc
    "/home/komarek/Rlib/glmmAK/Doc/chEpileptic/PGM_hc/"
                                             Normal_nhc
"/home/komarek/Rlib/glmmAK/Doc/chEpileptic/Normal_nhc/"
                                              Normal_hc
 "/home/komarek/Rlib/glmmAK/Doc/chEpileptic/Normal_hc/"
```

4.2 Matrices of covariates

To pass the covariates to the function logpoissonRE, we have to create two matrices or data frames which will contain (i) the covariates that appear in the fixed effect part of the model and are not involved in the random effect part (Base, Trt, Base:Trt interaction, Age) and (ii) the covariates that appear in the random effect part of the model (Visit). Remember, that inclusion of the random intercept is treated separately by the argument intcpt.random of the function logpoissonRE. Needed matrices will be stored as X2mat and Xb2mat:

```
> X2mat <- epilepticBC[, c("Base", "Trt", "Base.Trt", "Age")]
> Xb2mat <- data.frame(Visit = epilepticBC[, "Visit"])
```

Let us take a look at first few rows of these matrices:

```
> print(X2mat[1:6, ])
```

```
      Base Trt Base.Trt      Age
2 2.944439   1 2.944439 2.890372
3 2.944439   1 2.944439 2.890372
4 2.944439   1 2.944439 2.890372
5 2.944439   1 2.944439 2.890372
7 2.251292   1 2.251292 3.465736
8 2.251292   1 2.251292 3.465736
```

```
> print(Xb2mat[1:6, ])
```

```
[1] -0.3 -0.1  0.1  0.3 -0.3 -0.1
```

4.3 Length of the MCMC

The length of the MCMC simulation will be passed to the function cumlogitRE as a list:

```
> nsimul <- list(niter = 2000, nthin = 10, nburn = 1000, nwrite = 100)
```

With this specification, we will perform in total 2000 iterations out of which 1000 iterations will be a burn-in period. Further, we will thin the sample and store only every 10th value. Finally, the iteration count will increase every 100 iterations. That is, for inference, we will have chains of length 1000.

Remark: In the paper Komárek and Lesaffre (2008), much longer MCMC simulation was used to derive the results presented there.

4.4 Running MCMC

At this stage, we have specified all the information to start the MCMC simulation by calling the function logpoissonRE for each considered model. Be aware that this can take some time, according to the length of the MCMC specified.

PGM GLMM(nhc)

```
> fit.PGM.nhc <- logpoissonRE(y = epilepticBC$Seizure, x = X2mat, xb = Xb2mat,
+     cluster = epilepticBC$id, intcpt.random = TRUE, hierar.center = FALSE,
+     drandom = "gspline", prior.fixed = prior.fixed,
+     prior.random = prior.random.gspl.nhc, prior.gspline = prior.gspline,
+     nsimul = nsimul, store = list(ecount = FALSE, b = TRUE),
+     dir = dirPaths["PGM_nhc"])
```

```
Simulation started on           Fri Jun  1 13:53:09 2007
Iteration 1000
Burn-up finished on             Fri Jun  1 13:54:02 2007   (iteration 1000)
Iteration 2000
Simulation finished on          Fri Jun  1 13:55:01 2007   (iteration 2000)
```

PGM GLMM(hc)

```
> fit.PGM.hc <- logpoissonRE(y = epilepticBC$Seizure, x = X2mat, xb = Xb2mat,
+     cluster = epilepticBC$id, intcpt.random = TRUE, hierar.center = TRUE,
+     drandom = "gspline", prior.fixed = prior.fixed,
+     prior.random = prior.random.gspl.hc, prior.gspline = prior.gspline,
+     nsimul = nsimul, store = list(ecount = FALSE, b = TRUE),
+     dir = dirPaths["PGM_hc"])
```

```
Simulation started on           Fri Jun  1 13:55:01 2007
Iteration 1000
Burn-up finished on             Fri Jun  1 13:55:54 2007   (iteration 1000)
Iteration 2000
Simulation finished on          Fri Jun  1 13:56:48 2007   (iteration 2000)
```

Normal GLMM(nhc)

```
> fit.Normal.nhc <- logpoissonRE(y = epilepticBC$Seizure, x = X2mat, xb = Xb2mat,
+     cluster = epilepticBC$id, intcpt.random = TRUE, hierar.center = FALSE,
+     drandom = "normal", prior.fixed = prior.fixed,
+     prior.random = prior.random.norm.nhc,
+     nsimul = nsimul, store = list(ecount = FALSE, b = TRUE),
+     dir = dirPaths["Normal_nhc"])
```

```
Simulation started on           Fri Jun  1 13:56:48 2007
Iteration 1000
Burn-up finished on             Fri Jun  1 13:56:53 2007   (iteration 1000)
Iteration 2000
Simulation finished on          Fri Jun  1 13:56:58 2007   (iteration 2000)
```

Normal GLMM(hc)

```
> fit.Normal.hc <- logpoissonRE(y = epilepticBC$Seizure, x = X2mat, xb = Xb2mat,
+     cluster = epilepticBC$id, intcpt.random = TRUE, hierar.center = TRUE,
+     drandom = "normal", prior.fixed = prior.fixed,
+     prior.random = prior.random.norm.hc,
+     nsimul = nsimul, store = list(ecount = FALSE, b = TRUE),
+     dir = dirPaths["Normal_hc"])
```

```
Simulation started on            Fri Jun  1 13:56:58 2007
Iteration 1000
Burn-up finished on              Fri Jun  1 13:57:03 2007   (iteration 1000)
Iteration 2000
Simulation finished on           Fri Jun  1 13:57:08 2007   (iteration 2000)
```

5 Basic posterior computation

In this Section, we compute posterior summary statistics for regression coefficients $\boldsymbol{\beta}$ and moments of the distribution of random effects. To get reasonable results, we will use the chains sampled for the analysis in Komárek and Lesaffre (2008) which were obtained under the following value of the argument nsimul:

```
> nsimul <- list(niter = 50000, nthin = 130, nburn = 25000, nwrite = 1000)
```

That is, in the paper, we performed in total 50000 iterations out of which 25000 iterations were considered as a burn-in period. Further, we thined the sample and stored only every 130th value. For inference, we have the chains of length 25000.

5.1 Reading the chains into **coda** objects

Using the commands below, it is possible to read all sampled chains and store them as coda mcmc objects. It is possible to skip some values at the beginning of the chains by setting the argument skip to a positive value.

```
> chPGM.nhc <- glmmAK.files2coda(dir = dirPaths["PGM_nhc"], drandom = "gspline",
+     skip = 0)
> chPGM.hc <- glmmAK.files2coda(dir = dirPaths["PGM_hc"], drandom = "gspline",
+     skip = 0)
> chNormal.nhc <- glmmAK.files2coda(dir = dirPaths["Normal_nhc"],
+     drandom = "normal", skip = 0)
> chNormal.hc <- glmmAK.files2coda(dir = dirPaths["Normal_hc"], drandom = "normal",
+     skip = 0)
```

5.2 Reading only needed chains

On this place, we will read only the chains that will be worked out now, that is the chains for regression coefficients $\boldsymbol{\beta}$ and the chains for the moments of the random effect distribution. We will use the function scanFH which is a customized version of the R base function scan. All chains will be stored as coda mcmc objects.

PGM GLMM(nhc)

The chains we need now will be stored in the object chPGM.nhc. Let us first explicitly mention which (derived) parameters, stored in the files betaF.sim, betaRadj.sim and varRadj.sim will be summarized.

betaF.sim, columns "Base", "Trt", "Base.Trt", "Age" are the chains for β_3, β_4, β_5, β_6, i.e., regression coefficients for the fixed effects covariates. We will store them as components Base, Trt, Base.Trt, Age of the object chPGM.nhc.

betaRadj.sim, column "`(Intercept)`" is the chain for

$$\gamma_1 = \beta_1 + \mathrm{E}(b_1) = \beta_1 + \tau_1\beta_1^*,$$

$$\text{where } \beta_1^* = \sum_{j_1=-K_1}^{K_1} w_{j_1,+}(\boldsymbol{a})\mu_{1,j_1}, \qquad w_{j_1,+}(\boldsymbol{a}) = \sum_{j_2=-K_2}^{K_2} w_{j_1,j_2}(\boldsymbol{a}) \quad (j_1 = -K_1,\ldots,K_1).$$

That is, γ_1 is the mean intercept value and its chain will be stored as a component Intcpt of the object chPGM.nhc.

betaRadj.sim, column "`Visit`" is the chain for

$$\gamma_2 = \beta_2 + \mathrm{E}(b_2) = \beta_2 + \tau_2\beta_2^*,$$

$$\text{where } \beta_2^* = \sum_{j_2=-K_2}^{K_2} w_{+,j_2}(\boldsymbol{a})\mu_{2,j_2}, \qquad w_{+,j_2}(\boldsymbol{a}) = \sum_{j_1=-K_1}^{K_1} w_{j_1,j_2}(\boldsymbol{a}) \quad (j_2 = -K_2,\ldots,K_2).$$

That is, γ_2 is the mean effect of the covariate Visit and its chain will be stored as a component Visit of the object chPGM.nhc.

varRadj.sim, column "`varR.1.1`" is the chain for

$$d_{1,1} = \mathrm{var}(b_1) = \tau_1^2\, d_{1,1}^*, \qquad \text{where } d_{1,1}^* = \sum_{j_1=-K_1}^{K_1} w_{j_1,+}(\boldsymbol{a})\bigl(\mu_{1,j_1} - \beta_1^*\bigr)^2 + \sigma_1^2.$$

That is, $d_{1,1}$ is the variance of the random intercept. In the following, we will store a standard deviation of the random intercept, i.e., $\sqrt{d_{1,1}}$ as a component SDIntcpt of the object chPGM.nhc.

varRadj.sim, column "`varR.2.2`" is the chain for

$$d_{2,2} = \mathrm{var}(b_2) = \tau_2^2\, d_{2,2}^*, \qquad \text{where } d_{2,2}^* = \sum_{j_2=-K_2}^{K_2} w_{+,j_2}(\boldsymbol{a})\bigl(\mu_{2,j_2} - \beta_2^*\bigr)^2 + \sigma_2^2.$$

That is, $d_{2,2}$ is the variance of the random Visit effect. In the following, we will store a standard deviation of the random Visit effect, i.e., $\sqrt{d_{2,2}}$ as a component SDVisit of the object chPGM.nhc.

varRadj.sim, column "`varR.2.1`" is the chain for

$$d_{2,1} = \mathrm{cov}(b_1,\, b_2) = \tau_1\tau_2\, d_{2,1}^*, \qquad \text{where } d_{2,1}^* = \sum_{j_1=-K_1}^{K_1}\sum_{j_2=-K_2}^{K_2} w_{j_1,j_2}(\boldsymbol{a})\bigl(\mu_{1,j_1}-\beta_1^*\bigr)\bigl(\mu_{2,j_2}-\beta_2^*\bigr).$$

That is, $d_{2,1}$ is the covariance between the random intercept and the random Visit effect. In the following, we will store a correlation between the random intercept and the random Visit effect, i.e., $d_{2,1}/\sqrt{d_{1,1}\, d_{2,2}}$ as a component Corr of the object chPGM.nhc.

```
> iters     <- scanFH(paste(dirPaths["PGM_nhc"], "iteration.sim", sep = ""))
> betaF     <- scanFH(paste(dirPaths["PGM_nhc"], "betaF.sim", sep = ""))
> betaRadj  <- scanFH(paste(dirPaths["PGM_nhc"], "betaRadj.sim", sep = ""))
> varRadj   <- scanFH(paste(dirPaths["PGM_nhc"], "varRadj.sim", sep = ""))
> chPGM.nhc <- mcmc(data.frame(Base=betaF[,"Base"],
+             Trt=betaF[,"Trt"],
+             Base.Trt=betaF[,"Base.Trt"],
+             Age=betaF[,"Age"],
+             Intcpt=betaRadj[,"(Intercept)"],
+             Visit=betaRadj[,"Visit"],
+             SDIntcpt=sqrt(varRadj[,"varR.1.1"]),
+             SDVisit=sqrt(varRadj[,"varR.2.2"]),
+             Corr=varRadj[,"varR.2.1"]/sqrt(varRadj[,"varR.1.1"]*varRadj[,"varR.2.2"])),
+                   start=iters[1,1])
> rm(list = c("iters", "betaF", "betaRadj", "varRadj"))
```

PGM GLMM(hc)

The chains we need now will be stored in the object chPGM.hc. Again, let us first explicitly mention which (derived) parameters, stored in the files betaF.sim, betaRadj.sim and varRadj.sim will be summarized.

betaF.sim, columns "Base", "Trt", "Base.Trt", "Age" are the chains for β_3, β_4, β_5, β_6, i.e., regression coefficients for the fixed effects covariates. We will store them as components Base, Trt, Base.Trt, Age of the object chPGM.hc.

betaRadj.sim, column "(Intercept)" is the chain for

$$\gamma_1 = \mathrm{E}(b_1) = \alpha_1 + \tau_1\beta_1^*,$$

$$\text{where } \beta_1^* = \sum_{j_1=-K_1}^{K_1} w_{j_1,+}(\boldsymbol{a})\mu_{1,j_1}, \qquad w_{j_1,+}(\boldsymbol{a}) = \sum_{j_2=-K_2}^{K_2} w_{j_1,j_2}(\boldsymbol{a}) \quad (j_1 = -K_1,\dots,K_1).$$

That is, γ_1 is the mean intercept value and its chain will be stored as a component Intcpt of the object chPGM.hc.

betaRadj.sim, column "Visit" is the chain for

$$\gamma_2 = \mathrm{E}(b_2) = \alpha_2 + \tau_2\beta_2^*,$$

$$\text{where } \beta_2^* = \sum_{j_2=-K_2}^{K_2} w_{+,j_2}(\boldsymbol{a})\mu_{2,j_2}, \qquad w_{+,j_2}(\boldsymbol{a}) = \sum_{j_1=-K_1}^{K_1} w_{j_1,j_2}(\boldsymbol{a}) \quad (j_2 = -K_2,\dots,K_2).$$

That is, γ_2 is the mean effect of the covariate Visit and its chain will be stored as a component Visit of the object chPGM.hc.

varRadj.sim, column "varR.1.1" is the chain for

$$d_{1,1} = \mathrm{var}(b_1) = \tau_1^2\, d_{1,1}^*, \qquad \text{where } d_{1,1}^* = \sum_{j_1=-K_1}^{K_1} w_{j_1,+}(\boldsymbol{a})\bigl(\mu_{1,j_1} - \beta_1^*\bigr)^2 + \sigma_1^2.$$

That is, $d_{1,1}$ is the variance of the random intercept. In the following, we will store a standard deviation of the random intercept, i.e., $\sqrt{d_{1,1}}$ as a component SDIntcpt of the object chPGM.hc.

varRadj.sim, column “`varR.2.2`” is the chain for

$$d_{2,2} = \mathrm{var}(b_2) = \tau_2^2\, d^*_{2,2}, \qquad \text{where } d^*_{2,2} = \sum_{j_2=-K_2}^{K_2} w_{+,j_2}(\boldsymbol{a})\bigl(\mu_{2,j_2} - \beta_2^*\bigr)^2 + \sigma_2^2.$$

That is, $d_{2,2}$ is the variance of the random Visit effect. In the following, we will store a standard deviation of the random Visit effect, i.e., $\sqrt{d_{2,2}}$ as a component SDVisit of the object chPGM.hc.

varRadj.sim, column “`varR.2.1`” is the chain for

$$d_{2,1} = \mathrm{cov}(b_1,\, b_2) = \tau_1\tau_2\, d^*_{2,1}, \qquad \text{where } d^*_{2,1} = \sum_{j_1=-K_1}^{K_1}\sum_{j_2=-K_2}^{K_2} w_{j_1,j_2}(\boldsymbol{a})\bigl(\mu_{1,j_1} - \beta_1^*\bigr)\bigl(\mu_{2,j_2} - \beta_2^*\bigr).$$

That is, $d_{2,1}$ is the covariance between the random intercept and the random Visit effect. In the following, we will store a correlation between the random intercept and the random Visit effect, i.e., $d_{2,1}/\sqrt{d_{1,1}\, d_{2,2}}$ as a component Corr of the object chPGM.hc.

```
> iters     <- scanFH(paste(dirPaths["PGM_hc"], "iteration.sim", sep = ""))
> betaF     <- scanFH(paste(dirPaths["PGM_hc"], "betaF.sim", sep = ""))
> betaRadj  <- scanFH(paste(dirPaths["PGM_hc"], "betaRadj.sim", sep = ""))
> varRadj   <- scanFH(paste(dirPaths["PGM_hc"], "varRadj.sim", sep = ""))
> chPGM.hc <- mcmc(data.frame(Base=betaF[,"Base"],
+           Trt=betaF[,"Trt"],
+           Base.Trt=betaF[,"Base.Trt"],
+           Age=betaF[,"Age"],
+           Intcpt=betaRadj[,"(Intercept)"],
+           Visit=betaRadj[,"Visit"],
+           SDIntcpt=sqrt(varRadj[,"varR.1.1"]),
+           SDVisit=sqrt(varRadj[,"varR.2.2"]),
+           Corr=varRadj[,"varR.2.1"]/sqrt(varRadj[,"varR.1.1"]*varRadj[,"varR.2.2"])),
+                  start=iters[1,1])
> rm(list = c("iters", "betaF", "betaRadj", "varRadj"))
```

Normal GLMM(nhc)

The chains we need now will be stored in the object chNormal.nhc. Let us first explicitly mention which (derived) parameters, stored in the files betaF.sim and varR.sim will be summarized.

betaF.sim, columns “`Base`”, “`Trt`”, “`Base.Trt`”, “`Age`” are the chains for β_3, β_4, β_5, β_6, i.e., regression coefficients for the fixed effects covariates. We will store them as components Base, Trt, Base.Trt, Age of the object chNormal.nhc.

betaF.sim, columns “`(Intercept)`”, “`Visit`” are the chains for β_1 and β_2, which are (due to the fact that $\mathrm{E}(\boldsymbol{b}) = (0,\,0)'$) the mean intercept value and the mean value of the Visit effect. We will store them as components Intcpt and Visit of the object chNormal.nhc.

varR.sim, columns “`varR.1.1`”, “`varR.2.1`”, “`varR.2.2`” are the chains for $d_{1,1}$, $d_{2,1}$, $d_{2,2}$, which is the lower traingle of the random effects covariance matrix $\mathbb{D}$. In the object chNormal.nhc, random effect standard deviations $\sqrt{d_{1,1}}$ and $\sqrt{d_{2,2}}$ will be stored as components SDIntcpt and SDVisit, respectively and the correlation between the random effects, $d_{2,1}/\sqrt{d_{1,1}\, d_{2,2}}$, will be stored as a component Corr.

```
> iters <- scanFH(paste(dirPaths["Normal_nhc"], "iteration.sim", sep = ""))
> betaF <- scanFH(paste(dirPaths["Normal_nhc"], "betaF.sim", sep = ""))
> varR  <- scanFH(paste(dirPaths["Normal_nhc"], "varR.sim", sep = ""))
> chPGM.nhc <- mcmc(data.frame(Base=betaF[,"Base"],
+            Trt=betaF[,"Trt"],
+            Base.Trt=betaF[,"Base.Trt"],
+            Age=betaF[,"Age"],
+            Intcpt=betaF[,"(Intercept)"],
+            Visit=betaF[,"Visit"],
+            SDIntcpt=sqrt(varR[,"varR.1.1"]),
+            SDVisit=sqrt(varR[,"varR.2.2"]),
+            Corr=varR[,"varR.2.1"]/sqrt(varR[,"varR.1.1"]*varR[,"varR.2.2"])),
+                   start=iters[1,1])
> rm(list = c("iters", "betaF", "varR"))
```

Normal GLMM(hc)

The chains we need now will be stored in the object chNormal.hc. Again, let us first explicitly mention which (derived) parameters, stored in the files betaF.sim, betaR.sim and varR.sim will be summarized.

betaF.sim, columns "Base", "Trt", "Base.Trt", "Age" are the chains for β_3, β_4, β_5, β_6, i.e., regression coefficients for the fixed effects covariates. We will store them as components Base, Trt, Base.Trt, Age of the object chNormal.hc.

betaR.sim, columns "(Intercept)", "Visit" are the chains for α_1 and α_2, which are the mean intercept value and the mean value of the Visit effect. We will store them as components Intcpt and Visit of the object chNormal.hc.

varR.sim, columns "varR.1.1", "varR.2.1", "varR.2.2" are the chains for $d_{1,1}$, $d_{2,1}$, $d_{2,2}$, which is the lower traingle of the random effects covariance matrix $\mathbb{D}$. In the object chNormal.hc, random effect standard deviations $\sqrt{d_{1,1}}$ and $\sqrt{d_{2,2}}$ will be stored as components SDIntcpt and SDVisit, respectively and the correlation between the random effects, $d_{2,1}/\sqrt{d_{1,1}\,d_{2,2}}$, will be stored as a component Corr.

```
> iters <- scanFH(paste(dirPaths["Normal_hc"], "iteration.sim", sep = ""))
> betaF <- scanFH(paste(dirPaths["Normal_hc"], "betaF.sim", sep = ""))
> betaR <- scanFH(paste(dirPaths["Normal_hc"], "betaR.sim", sep = ""))
> varR  <- scanFH(paste(dirPaths["Normal_hc"], "varR.sim", sep = ""))
> chPGM.nhc <- mcmc(data.frame(Base=betaF[,"Base"],
+            Trt=betaF[,"Trt"],
+            Base.Trt=betaF[,"Base.Trt"],
+            Age=betaF[,"Age"],
+            Intcpt=betaR[,"(Intercept)"],
+            Visit=betaR[,"Visit"],
+            SDIntcpt=sqrt(varR[,"varR.1.1"]),
+            SDVisit=sqrt(varR[,"varR.2.2"]),
+            Corr=varR[,"varR.2.1"]/sqrt(varR[,"varR.1.1"]*varR[,"varR.2.2"])),
+                   start=iters[1,1])
> rm(list = c("iters", "betaF", "betaR", "varR"))
```

5.3 Basic posterior summary statistics

Basic posterior summary statistics can be obtained using the coda `summary` function for objects of class `mcmc`:

PGM GLMM(nhc)

```
> summary(chPGM.nhc)


Iterations = 25001:50000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 25000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

            Mean     SD  Naive SE Time-series SE
Base      0.8658 0.1374 0.0008691       0.002598
Trt      -0.9952 0.4228 0.0026743       0.009159
Base.Trt  0.3706 0.2131 0.0013479       0.003993
Age       0.4875 0.3614 0.0022857       0.006344
Intcpt   -1.3766 1.2385 0.0078327       0.024679
Visit    -0.2747 0.1665 0.0010533       0.007553
SDIntcpt  0.5429 0.0754 0.0004771       0.003565
SDVisit   0.7114 0.1855 0.0011733       0.006676
Corr      0.0608 0.2196 0.0013886       0.026354

2. Quantiles for each variable:

             2.5%     25%     50%     75%    97.5%
Base       0.5908  0.7754  0.8667  0.9577  1.13206
Trt       -1.8346 -1.2744 -0.9947 -0.7133 -0.16488
Base.Trt  -0.0479  0.2293  0.3690  0.5128  0.79355
Age       -0.2299  0.2483  0.4880  0.7251  1.20148
Intcpt    -3.8308 -2.1842 -1.3764 -0.5692  1.06379
Visit     -0.6028 -0.3860 -0.2770 -0.1655  0.05612
SDIntcpt   0.4181  0.4900  0.5354  0.5869  0.71322
SDVisit    0.3747  0.5870  0.7013  0.8264  1.10404
Corr      -0.4367 -0.0777  0.0458  0.2045  0.48560
```

PGM GLMM(hc)

```
> summary(chPGM.hc)
```

```
Iterations = 25001:50000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 25000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

            Mean     SD  Naive SE Time-series SE
Base      0.9080 0.1386 0.0008768       0.002091
Trt      -0.8642 0.4246 0.0026853       0.005009
Base.Trt  0.3084 0.2155 0.0013632       0.002951
Age       0.4711 0.3649 0.0023076       0.024643
Intcpt   -1.3986 1.2400 0.0078426       0.086138
Visit    -0.2490 0.1598 0.0010109       0.002611
SDIntcpt  0.5320 0.0668 0.0004226       0.000812
SDVisit   0.7289 0.2018 0.0012763       0.005511
Corr     -0.0539 0.1077 0.0006808       0.013067

2. Quantiles for each variable:

            2.5%     25%     50%     75%    97.5%
Base      0.6344  0.8152  0.9084  1.0003  1.18117
Trt      -1.7111 -1.1488 -0.8611 -0.5816 -0.04256
Base.Trt -0.1133  0.1643  0.3089  0.4517  0.73546
Age      -0.2457  0.2207  0.4762  0.7142  1.20073
Intcpt   -3.8800 -2.2165 -1.4163 -0.5345  1.00589
Visit    -0.5725 -0.3530 -0.2465 -0.1404  0.05717
SDIntcpt  0.4164  0.4852  0.5272  0.5727  0.67790
SDVisit   0.3665  0.5921  0.7169  0.8510  1.16253
Corr     -0.2631 -0.1213 -0.0574  0.0067  0.17057
```

Normal GLMM(nhc)

```
> summary(chNormal.nhc)


Iterations = 25001:50000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 25000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

            Mean     SD  Naive SE Time-series SE
Base      0.8853 0.1366 0.0008642      0.0009286
Trt      -0.9435 0.4179 0.0026432      0.0028505
```

```
Base.Trt  0.3456 0.2127 0.0013453      0.0013762
Age       0.4917 0.3699 0.0023396      0.0021058
Intcpt   -1.4185 1.2564 0.0079463      0.0072118
Visit    -0.2733 0.1557 0.0009848      0.0011859
SDIntcpt  0.5305 0.0645 0.0004081      0.0003842
SDVisit   0.6131 0.2055 0.0012997      0.0017880
Corr      0.0422 0.3143 0.0019878      0.0017636

2. Quantiles for each variable:

            2.5%     25%     50%     75%    97.5%
Base      0.6165  0.7950  0.8852  0.9747  1.15444
Trt      -1.7741 -1.2242 -0.9442 -0.6621 -0.12047
Base.Trt -0.0718  0.2039  0.3454  0.4865  0.76296
Age      -0.2402  0.2443  0.4930  0.7388  1.21305
Intcpt   -3.8839 -2.2702 -1.4215 -0.5779  1.06445
Visit    -0.5767 -0.3769 -0.2742 -0.1706  0.03298
SDIntcpt  0.4173  0.4856  0.5259  0.5703  0.67129
SDVisit   0.1117  0.5018  0.6248  0.7457  0.98885
Corr     -0.5288 -0.1620  0.0328  0.2305  0.78734
```

Normal GLMM(hc)

```
> summary(chNormal.hc)


Iterations = 25001:50000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 25000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

            Mean     SD  Naive SE Time-series SE
Base      0.8844 0.1375 0.0008693       0.001492
Trt      -0.9422 0.4207 0.0026608       0.003125
Base.Trt  0.3433 0.2139 0.0013527       0.002231
Age       0.4655 0.3697 0.0023380       0.019473
Intcpt   -1.3282 1.2553 0.0079390       0.067613
Visit    -0.2724 0.1568 0.0009914       0.000983
SDIntcpt  0.5306 0.0649 0.0004106       0.000498
SDVisit   0.6112 0.2084 0.0013178       0.002086
Corr      0.0357 0.3201 0.0020242       0.004038

2. Quantiles for each variable:

            2.5%     25%     50%     75%    97.5%
```

```
Base      0.6158  0.7921  0.8841  0.9756  1.15473
Trt      -1.7624 -1.2243 -0.9402 -0.6609 -0.12179
Base.Trt -0.0778  0.2012  0.3438  0.4863  0.76057
Age      -0.2746  0.2175  0.4690  0.7173  1.17956
Intcpt   -3.7458 -2.1847 -1.3402 -0.4797  1.16613
Visit    -0.5819 -0.3758 -0.2731 -0.1687  0.03888
SDIntcpt  0.4193  0.4849  0.5258  0.5702  0.67361
SDVisit   0.1014  0.5002  0.6275  0.7470  0.98166
Corr     -0.5570 -0.1679  0.0253  0.2236  0.80353
```

5.4 Bayesian P-values

Bayesian P-values as defined in Komárek and Lesaffre (2008) can be computed as follows:

PGM GLMM(nhc)

```
> BPvalue(chPGM.nhc[,params])
```

```
    Base      Trt Base.Trt      Age    Visit
 0.00000  0.02072  0.08168  0.17368  0.10104
```

PGM GLMM(hc)

```
> BPvalue(chPGM.hc[,params])
```

```
    Base      Trt Base.Trt      Age    Visit
 0.00000  0.03944  0.14952  0.19096  0.11064
```

Normal GLMM(nhc)

```
> BPvalue(chNormal.nhc[,params])
```

```
    Base      Trt Base.Trt      Age    Visit
 0.00000  0.02392  0.10728  0.18248  0.08336
```

Normal GLMM(hc)

```
> BPvalue(chNormal.hc[,params])
```

```
    Base      Trt Base.Trt      Age    Visit
 0.00000  0.02472  0.10928  0.20824  0.08552
```

5.5 Highest posterior density intervals

Highest posterior density intervals can be computed using the coda function HPDinterval:

PGM GLMM(nhc)

```
> HPDinterval(chPGM.nhc, prob = 0.95)
```

```
              lower       upper
Base      0.6035998  1.14340284
Trt      -1.8485035 -0.18414713
Base.Trt -0.0448175  0.79529207
Age      -0.2018371  1.22327080
Intcpt   -3.7995117  1.08654797
Visit    -0.6016992  0.05661944
SDIntcpt  0.4081708  0.69554563
SDVisit   0.3668022  1.09183617
Corr     -0.4367999  0.48559052
attr(,"Probability")
[1] 0.95
```

PGM GLMM(hc)

```
> HPDinterval(chPGM.hc, prob = 0.95)
```

```
              lower       upper
Base      0.6331944  1.17915433
Trt      -1.7143816 -0.04690501
Base.Trt -0.1162877  0.73081797
Age      -0.2248783  1.21851045
Intcpt   -3.9051979  0.96909231
Visit    -0.5645966  0.06295498
SDIntcpt  0.4075452  0.66454562
SDVisit   0.3308356  1.12106139
Corr     -0.2591699  0.17375843
attr(,"Probability")
[1] 0.95
```

Normal GLMM(nhc)

```
> HPDinterval(chNormal.nhc, prob = 0.95)
```

```
              lower       upper
Base      0.6155912  1.15193728
Trt      -1.7645179 -0.11450856
```

```
Base.Trt -0.0690086  0.76408915
Age      -0.2449218  1.20725069
Intcpt   -3.9043743  1.03435788
Visit    -0.5739697  0.03562814
SDIntcpt  0.4110878  0.66122106
SDVisit   0.0959356  0.97249436
Corr     -0.5706093  0.71429813
attr(,"Probability")
[1] 0.95
```

Normal GLMM(hc)

```
> HPDinterval(chNormal.hc, prob = 0.95)
```

```
              lower       upper
Base      0.6207683  1.15915086
Trt      -1.7655127 -0.12578612
Base.Trt -0.0698370  0.76658142
Age      -0.2575308  1.19336622
Intcpt   -3.8250303  1.06536196
Visit    -0.5809721  0.03939429
SDIntcpt  0.4130962  0.66233863
SDVisit   0.0997530  0.97877056
Corr     -0.6142601  0.72709863
attr(,"Probability")
[1] 0.95
```

The chains can be further processed using the coda package to check for convergence, draw plots, etc. We will skip this in this manual to concentrate more on the issues specific for the glmmAK package.

6 Estimation of the random effect density in the PGM models

The estimate of the random effect density in the PGM models can be summarized using the pointwise posterior summary statistics (mean, median, quantiles). To compute these from the sampled chains, we use the function summaryGspline2.

6.1 Standardized version

Firstly, we summarize the standardized version of the random effect density. That is, when computing the posterior statistics, the random effect density at each iteration is standardized first to have zero means and unit variances and summarized afterwards. The pointwise posterior summary statistics will be computed in a grid of points stored in the variables grid1 (random intercept margin) and grid2 (random Visit effect margin). Besides computing pointwise posterior mean, we will also compute pointwise posterior 2.5%, 25%, 50%, 75% and 97.5% quantiles. Note that variables knots1 and sigma1 determine the PGM knots $\mu_{1,-K_1},\ldots,\mu_{1,K_1}$ and basis standard deviation σ_1, respectively. Similarly, variables knots2 and sigma2 determine the PGM knots $\mu_{2,-K_2},\ldots,\mu_{2,K_2}$ and basis standard deviation σ_2, respectively. Computed posterior summary statistics for the random effect density will be stored in objects stPGM.nhc and stPGM.hc for PGM GLMM(nhc) and PGM GLMM(hc) model, respectively. The following commands compute summaries for both joint (bivariate) random effect density and also the marginal (univariate) random intercept and random Visit effect densities.

```
> knots1 <- seq(-4.5, 4.5, by=0.3)
> knots2 <- seq(-4.5, 4.5, by=0.3)
> sigma1 <- 0.2
> sigma2 <- 0.2
> grid1 <- seq(-3, 3, length=20)
> grid2 <- seq(-3, 3, length=20)
>
> ### PGM GLMM(nhc)
> stPGM.nhc <- summaryGspline2(x1=grid1, x2=grid2,
+                               mu1=knots1, mu2=knots2,
+                               sigma1=sigma1, sigma2=sigma2,
+                               standard=TRUE,
+                               probs=c(0.025, 0.25, 0.5, 0.75, 0.975), values=FALSE,
+                               dir=dirPaths["PGM_nhc"])
>
> ### PGM GLMM(hc)
> stPGM.hc <- summaryGspline2(x1=grid1, x2=grid2,
+                              mu1=knots1, mu2=knots2,
+                              sigma1=sigma1, sigma2=sigma2,
+                              standard=TRUE,
+                              probs=c(0.025, 0.25, 0.5, 0.75, 0.975), values=FALSE,
+                              dir=dirPaths["PGM_hc"])
```

For example, for the PGM GLMM(nhc) model, the pointwise posterior summary statistics of the joint random effect density are stored in the subobject stPGM.nhc$summary, which has components labeled "x1" and "x2" (vectors of grid points), "Mean" (matrix with the pointwise posterior mean), "2.5%", "25%", "50%", "75%", "97.5%" (matrices with the pointwise posterior quantiles). The pointwise posterior summary statistics of the marginal random intercept density are stored in the subobject stPGM.nhc$summary1, which is a data frame with columns "x", "Mean", "2.5%", "25%", "50%", "75%", "97.5%" having an obvious meaning. Similarly, the pointwise posterior summary statistics of the marginal random Visit effect density are stored in the data frame stPGM.nhc$summary1.

Computed posterior summary statistics of the densities can be plotted as follows, see Figure 1 for the results. The example code below applies for the PGM GLMM(hc) model.

```
> obj  <- stPGM.nhc$summary
> obj1 <- stPGM.nhc$summary1
> obj2 <- stPGM.nhc$summary2
>
> par(mfrow=c(2, 2), bty="n", mar=c(4, 4, 1, 0)+0.1)
>
> ### Joint density (posterior mean only)
> contour(obj$x1, obj$x2, obj$Mean, col="red", xlab="b1[st]", ylab="b2[st]")
> persp(obj$x1, obj$x2, obj$Mean, col="seagreen3", theta=30, phi=60,
+       xlab="b1[st]", ylab="b2[st]", zlab="g(b1[st],b2[st])")
>
> ### Marginal random intercept density (posterior mean, 2.5% and 97.5% quantiles)
> plot(obj1$x, obj1[,"97.5%"], type="l", lty=1, col="red",
+      xlab="b1[st]", ylab="g(b1[st])", main="Random intercept")
> lines(obj1$x, obj1[,"2.5%"], lty=2, col="red")
> lines(obj1$x, obj1$Mean, lty=1, col="blue")
>
> ### Marginal random Visit effect density (posterior mean, 2.5% and 97.5% quantiles)
> plot(obj2$x, obj2[,"97.5%"], type="l", lty=1, col="red",
+      xlab="b2[st]", ylab="g(b2[st])", main="Random Visit effect")
> lines(obj2$x, obj2[,"2.5%"], lty=2, col="red")
> lines(obj2$x, obj2$Mean, lty=1, col="blue")
```

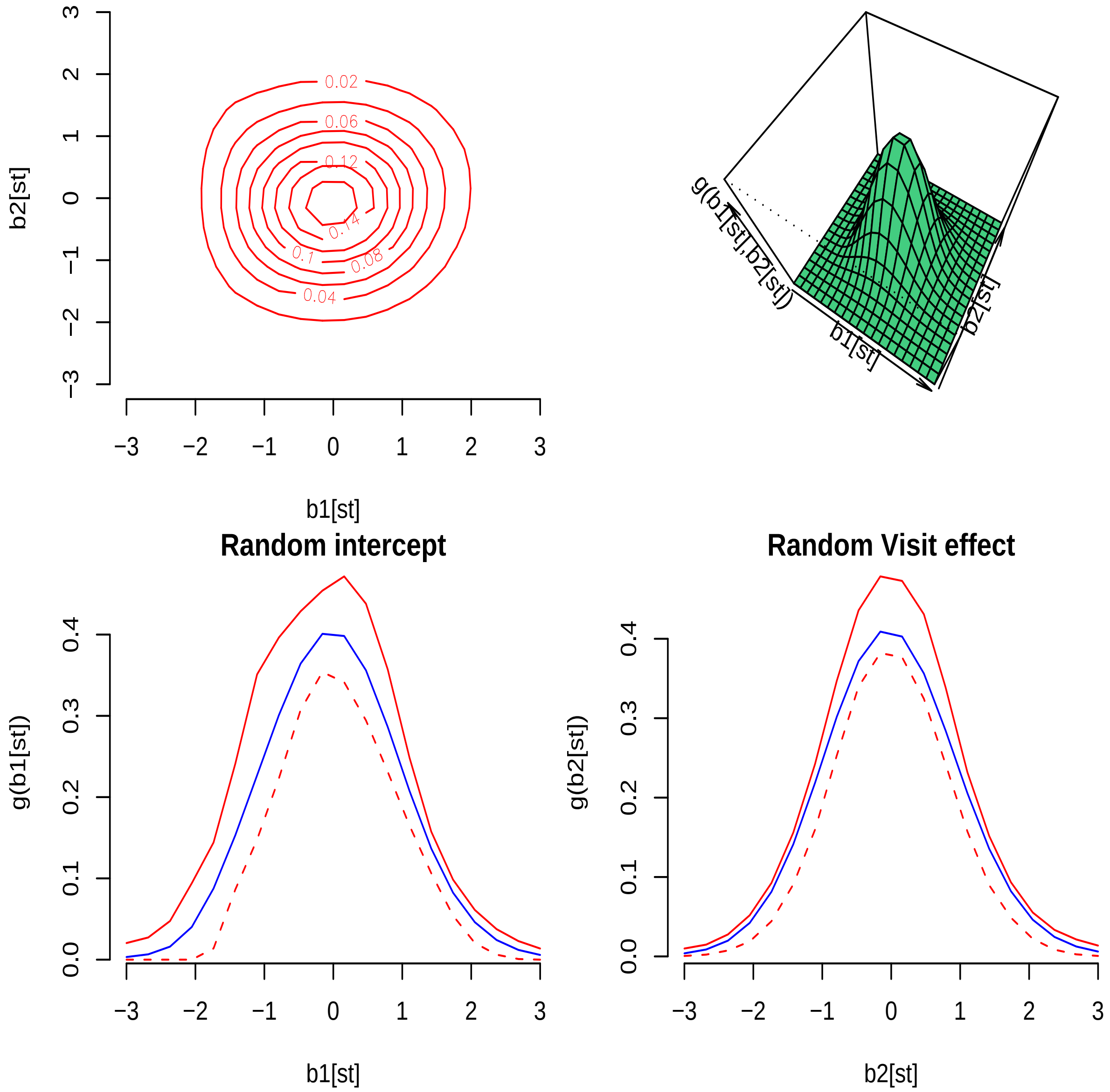


Figure 1: **PGM GLMM(nhc):** Pointwise posterior mean of the joint random effect density (upper panels), pointwise posterior mean, 2.5% and 97% quantiles of the marginal random intercept and random Visit effect densities (lower panels).

7 Summary for the values of individual random effects

Sampled values of the individual random effects are stored in the files b.sim. Posterior mean and quantiles can be used to infer on the individual random effects.

In this manual, we will show the results for the **PGM GLMM(nhc)** only. The results for the remaining models would have been obtained analogically. Note that we will compute posterior summary for $\beta_1 + b_{i,1}$ and $\beta_2 + b_{i,2}$ $(i = 1, \ldots, N)$, that is for random effects shifted by the corresponding location parameter.

Firstly, we extract from the original data identification numbers of the patients and divide also these id numbers into two groups according to the treatment.

```
> IDNR <- unique(epilepticBC$id)
> IDNR0 <- unique(subset(epilepticBC, Trt == 0)$id)
> IDNR1 <- unique(subset(epilepticBC, Trt == 1)$id)
> index.tr0 <- (1:length(IDNR))[IDNR %in% IDNR0]
> index.tr1 <- (1:length(IDNR))[IDNR %in% IDNR1]
```

Now, we read the sampled values of random effects and shift them by the sampled location parameters β_1 and β_2. Note that the sampled location parameters are stored in the columns “(Intercept)” and “Visit” of the file betaF.sim.

```
> betab.PGMnhc <- scanFH(paste(dirPaths["PGM_nhc"], "/betaF.sim", sep=""))
+          [,c("(Intercept)", "Visit")]
> b.PGMnhc <- scanFH(paste(dirPaths["PGM_nhc"], "b.sim", sep = "")) +
+          as.matrix(betab.PGMnhc)
> colnames(b.PGMnhc) <- paste(c("Intcpt", "Visit"), rep(IDNR, each=2), sep="")
```

We continue by computing posterior mean and median for the individual values of random intercepts. Note that the chains for individual random intercepts are stored in odd columns of the object b.PGMnhc.

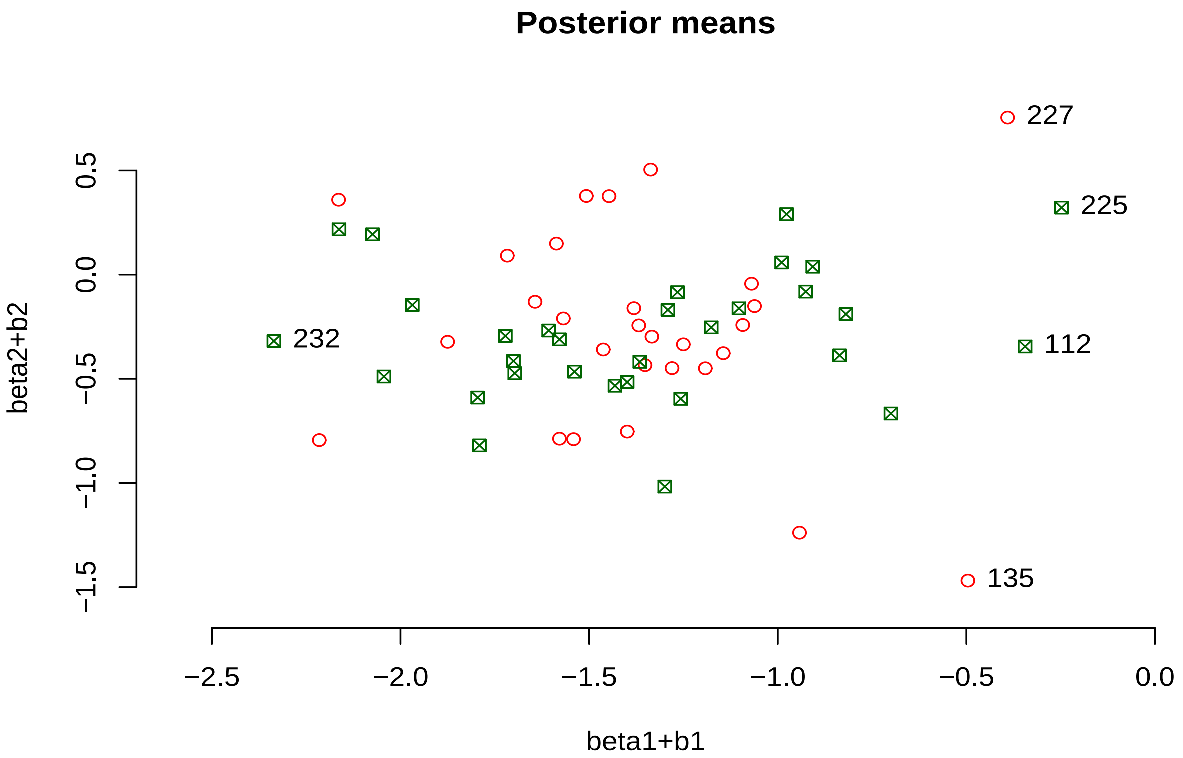
```
> indIntcpt <- seq(1, ncol(b.PGMnhc) - 1, by = 2)
> bIntcptMean.PGMnhc <- apply(b.PGMnhc[, indIntcpt], 2, mean)
> bIntcptMedian.PGMnhc <- apply(b.PGMnhc[, indIntcpt], 2, median)
```

Similarly, we compute posterior means and medians for the individual values of random Visit effects. Note that the chains for individual random Visit effects are stored in even columns of the object b.PGMnhc.

```
> indVisit <- seq(2, ncol(b.PGMnhc), by = 2)
> bVisitMean.PGMnhc <- apply(b.PGMnhc[, indVisit], 2, mean)
> bVisitMedian.PGMnhc <- apply(b.PGMnhc[, indVisit], 2, median)
```

Finally, we produce scatterplots of posterior means and medians of individual values of random effects. We will use different symbols and colors for the control and treatment group and identify some patients by their identification numbers. See Figure 2 for the result.

```
> showid <- c(112, 135, 225, 227, 232)
> index.show <- IDNR %in% showid
>
> par(mfrow=c(2, 1), bty="n", mar=c(4, 4, 4, 1)+0.1)
>
> ### Posterior means
> plot(bIntcptMean.PGMnhc[index.tr0], bVisitMean.PGMnhc[index.tr0], pch=1, col="red",
+      xlab="beta1+b1", ylab="beta2+b2",
+      xlim=range(bIntcptMean.PGMnhc), ylim=range(bVisitMean.PGMnhc),
+      main="Posterior means")
> points(bIntcptMean.PGMnhc[index.tr1], bVisitMean.PGMnhc[index.tr1], pch=7,
+        col="darkgreen")
> text(bIntcptMean.PGMnhc[index.show]+0.005, bVisitMean.PGMnhc[index.show],
+      labels=IDNR[index.show], pos=4)
>
> ### Posterior medians
> plot(bIntcptMedian.PGMnhc[index.tr0], bVisitMedian.PGMnhc[index.tr0], pch=1, col="red",
+      xlab="beta1+b1", ylab="beta2+b2",
+      xlim=range(bIntcptMedian.PGMnhc), ylim=range(bVisitMedian.PGMnhc),
+      main="Posterior medians")
> points(bIntcptMedian.PGMnhc[index.tr1], bVisitMedian.PGMnhc[index.tr1], pch=7,
+        col="darkgreen")
> text(bIntcptMedian.PGMnhc[index.show]+0.005, bVisitMedian.PGMnhc[index.show],
+      labels=IDNR[index.show], pos=4)
```



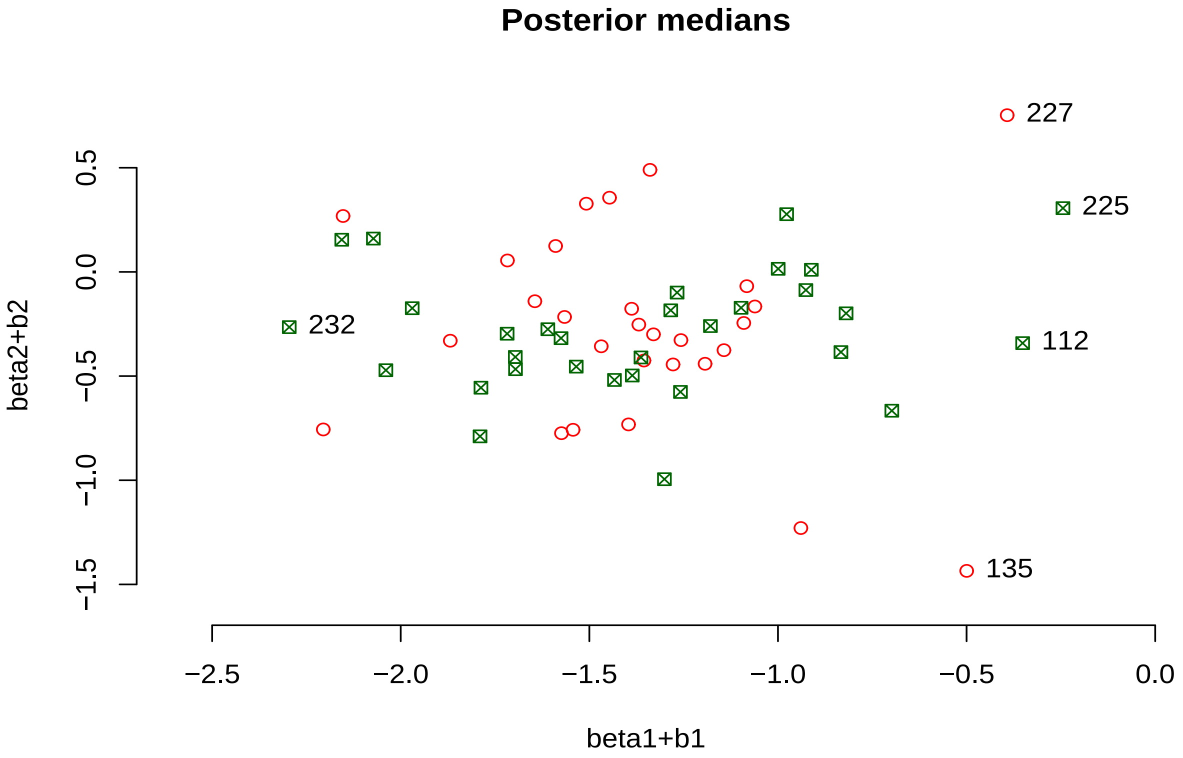


Figure 2: **PGM GLMM(nhc):** Scatterplot of the posterior means and posterior medians of individual random effects shifted by the locations β_1 and β_2.

References

BRESLOW, N. E. and CLAYTON, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, **88**, 9–25.

GELMAN, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, **1**, 515–533.

GELMAN, A., CARLIN, J. B., STERN, H. S., and RUBIN, D. B. (2004). *Bayesian Data Analysis*. Chapman & Hall/CRC, Boca Raton, Second edition. ISBN 1-58488-388-X.

KOMÁREK, A. and LESAFFRE, E. (2008). Generalized linear mixed model with a penalized Gaussian mixture as a random effects distribution. *Computational Statistics and Data Analysis*, **52**, 3441–3458.

NEAL, R. M. (2003). Slice sampling (with Discussion). *The Annals of Statistics*, **31**, 705–767.

PLUMMER, M., BEST, N., COWLES, K., and VINES, K. (2006). *coda: Output analysis and diagnostics for MCMC*. R package version 0.10-7.

THALL, P. F. and VAIL, S. C. (1990). Some covariance models for longitudinal count data with overdispersion. *Biometrics*, **46**, 657–671.