MCMC Methods for Multi-response Generalized Linear Mixed Models: The MCMCglmm R Package

Jarrod Hadfield
University of Edinburgh

Abstract

Generalized linear mixed models provide a flexible framework for modeling a range of data, although with non-Gaussian response variables the likelihood cannot be obtained in closed form. Markov chain Monte Carlo methods solve this problem by sampling from a series of simpler conditional distributions that can be evaluated. The R package **MCMCglmm**, implements such an algorithm for a range of model fitting problems. More than one response variable can be analysed simultaneously, and these variables are allowed to follow Gaussian, Poisson, multi(bi)nominal, exponential, zero-inflated and censored distributions. A range of variance structures are permitted for the random effects, including interactions with categorical or continuous variables (i.e., random regression), and more complicated variance structures that arise through shared ancestry, either through a pedigree or through a phylogeny. Missing values are permitted in the response variable(s) and data can be known up to some level of measurement error as in meta-analysis. All simulation is done in C/ C++ using the **CSparse** library for sparse linear systems. If you use the software please cite this article, as published in the Journal of Statistic Software (?)

Keywords: MCMC, linear mixed model, pedigree, phylogeny, animal model, multivariate, sparse, R.

Due to their flexibility, linear mixed models are now widely used across the sciences (???). However, generalizing these models to non-Gaussian data has proved difficult because integrating over the random effects is intractable (?). Although techniques that approximate these integrals (?) are now popular, Markov chain Monte Carlo (MCMC) methods provide an alternative strategy for marginalizing the random effects that may be more robust (??). Developing MCMC methods for generalized linear mixed models (GLMM) is an active area of research (e.g., ????), and several software packages are now available that implement these techniques (e.g., **WinBUGS** (?), **MLwiN** (?), **glmmBUGS** (?), **JAGS** (?)). However, these methods often require a certain level of expertise on behalf of the user and may take a great deal of computing time. The **MCMCglmm** package for R (?) implements Markov chain Monte Carlo routines for fitting multi-response generalized linear mixed models. A range of distributions are supported and several types of variance structure for the random effects and the residuals can be fitted. The aim is to provide routines that require little expertise on behalf of the user while reducing the amount of computing time required to adequately sample the posterior distribution.

In this paper we explain the underlying structure of GLMM's and then briefly describe a general strategy for estimating the parameters. Few new results are presented, and we would like to acknowledge that many of the statistical results can be found in **?** and many of the algorithm details that allow the models to be fitted efficiently can be found in **?**. The main body of the paper introduces the software, using a worked example taken from a quantitative genetic experiment. We end by comparing the routines with WinBUGS (**?**), and find **MCMCglmm** to be nearly 40 times faster per iteration, and to have an effective sample size per iteration more than 3 times greater.

1. Model form

The model has three components: a) probability density functions that relate the data y to latent variables l, on the link scale b) a standard linear mixed model with fixed and random predictors applied to l and c) variance structures that describe the expected (co)variances between the location effects (fixed and random effects). Although we develop these models in a Bayesian context where the distinction between fixed and random effects does not technically exist, we make the distinction throughout the manuscript as the terminology is well entrenched and understood.

1.1. Probability of the data y given the latent variable l

The probability of the i^{th} data point is represented by:

$$f_i(y_i|l_i) \tag{1}$$

where f_i is the probability density function associated with y_i. For example, if y_i was assumed to be Poisson distributed and we used the canonical log link function, then Equation 1 would have the form:

$$f_P\left(y_i|\lambda = \exp(l_i)\right) \tag{2}$$

where λ is the canonical parameter of the Poisson density function f_P.

1.2. Linear model for the latent variables l

The vector of latent variables are predicted by the linear model

$$\mathbf{l} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \mathbf{e} \tag{3}$$

where $\mathbf{X}$ is a design matrix relating fixed predictors to the data, and $\mathbf{Z}$ is a design matrix relating random predictors to the data. These predictors have associated parameter vectors $\boldsymbol{\beta}$ and $\mathbf{u}$, and $\mathbf{e}$ is a vector of residuals. In the Poisson case these residuals deal with any over-dispersion in the data after accounting for fixed and random sources of variation.

1.3. Variance structures for the model parameters

The location effects ($\boldsymbol{\beta}$ and $\mathbf{u}$), and the residuals ($\mathbf{e}$) are assumed to come from a multivariate normal distribution:

$$\left[\begin{array}{c}\boldsymbol{\beta}\\ \mathbf{u}\\ \mathbf{e}\end{array}\right] \sim N\left(\left[\begin{array}{c}\boldsymbol{\beta}_0\\ \mathbf{0}\\ \mathbf{0}\end{array}\right], \left[\begin{array}{ccc}\mathbf{B} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{G} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{R}\end{array}\right]\right) \tag{4}$$

where $\boldsymbol{\beta}_0$ are the prior means for the fixed effects with prior covariance matrix $\mathbf{B}$, and $\mathbf{G}$ and $\mathbf{R}$ are the expected (co)variances of the random effects and residuals respectively. The zero off-diagonal matrices imply *a priori* independence between fixed effects, random effects, and residuals. Generally, $\mathbf{G}$ and $\mathbf{R}$ are large square matrices with dimensions equal to the number of random effects and residuals. Typically they are unknown, and must be estimated from the data, usually by assuming they are structured in a way that they can be parametrized by few parameters. Below we will focus on the structure of $\mathbf{G}$, but the same logic can be applied to $\mathbf{R}$.

At its most general, **MCMCglmm** allows variance structures of the form:

$$\mathbf{G} = (\mathbf{V}_1 \otimes \mathbf{A}_1) \oplus (\mathbf{V}_2 \otimes \mathbf{A}_2) \oplus \dots \tag{5}$$

where the parameter (co)variance matrices ($\mathbf{V}$) are usually low-dimensional and are to be estimated, and the structured matrices ($\mathbf{A}$) are usually high dimensional and treated as known. We will refer to terms separated by a direct sum ($\oplus$) as component terms, and the use of a direct sum explicitly assumes random effects associated with different component terms are independent. Each component term, however, is formed through the Kronecker product ($\otimes$) which allows for possible dependence between random effects *within* a component term. Equation 24 can be expanded to give:

$$\mathbf{G} = \left[\begin{array}{cc}\mathbf{V}_1 \otimes \mathbf{A}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_2 \otimes \mathbf{A}_2\end{array}\right] \tag{6}$$

where the zero off-diagonals represent the independence between component terms.

In the simplest models the structured matrices of each component term are often assumed to be identity matrices and the parameter (co)variance matrices scalar variances:

$$\mathbf{V}_1 \otimes \mathbf{A}_1 = \sigma^2_1\mathbf{I} \tag{7}$$

which assumes that random effects within a component term are independent but have a common variance. However, independence between different levels is often too strong an assumption. For example, if we had made two visits to a sample of schools and recorded test scores for the children, we may expect dependence between measurements made in the same school although they were sampled at different times. If the random effects are ordered schools within ages ($\mathbf{u}^{\top} = [\mathbf{u}_1\ \mathbf{u}_2]$) where $\mathbf{u}_1$ are the random effects for the schools at time period one, and $\mathbf{u}_2$ for the same set of schools at time period 2, then an appropriate $\mathbf{G}$ component may have the form:

$$\mathbf{V}_1 \otimes \mathbf{A}_1 = \left[\begin{array}{cc}\sigma^2_{u_1} & \sigma_{u_1,u_2}\\ \sigma_{u_2,u_1} & \sigma^2_{u_2}\end{array}\right] \otimes \mathbf{I} \tag{8}$$

Here the diagonal elements model different variances for the two sampling periods, and the covariance captures any persistent differences between schools. The identity matrix in the Kronecker product implies the schools are independent. Although the assumption of independence may be adequate in many applications, there are situations where it is not tenable. For example, when data have been collected on related individuals, or related species, then complicated patterns of dependence can arise if the characteristics are heritable. In these cases $\mathbf{A}$ is not an identity matrix but a matrix whose elements are equal to the proportion of genes the two individuals have in common.

2. Parameter estimation and DIC

For most types of model (non-Gaussian data) the distribution of l is not in a recognizable form and is updated using either Metropolis-Hastings updates or the slice sampling method of (**?**). Latent variables whose residuals are non-independent are sampled in blocks using Metropolis-Hastings updates and an efficient proposal distribution is determined during the burn-in phase using adaptive methods (**??**). The parameters of the mixed model ($\boldsymbol{\beta}$ and $\mathbf{u}$) follow a multivariate normal distribution and can be Gibbs sampled in a single block using the method of **?**. This method requires solving a large, but often sparse set of linear equations which can be done efficiently using methods provided in the **CSparse** library (**?**). With conjugate priors the variance structures ($\mathbf{R}$ and $\mathbf{G}$) follow an inverse-Wishart distribution which can also be Gibbs sampled in a single block in many instances. By fitting non-identified multiplicative working parameters for the random effects non-central F-distributed priors for the variance components can be fitted (**?**). This involves updating the working parameters each iteration which again can be achieved using the method of **?**.

The deviance and hence the deviance information criterion (DIC) can be calculated in different ways depending on what is in ‘focus’ (**?**). For non-Gaussian response variables (including censored Gaussian) **MCMCglmm** calculates the deviance using the probability of the data given the latent variables. For Gaussian data, however, the deviance is calculated using the probability of the data given the location parameters $\boldsymbol{\theta}^{\top} = [\boldsymbol{\beta}\ \mathbf{u}]$.

In the appendix the conditional distributions, and computational strategies for sampling from them, are described in more detail, together with a more in depth explanation on the computation of deviance and DIC.

3. Software

To illustrate the software we reanalyze experimental data collected on the Eurasian passerine bird, the Blue tit (*Cyanistes caeruleus*) (**?**). The data consist of measurements taken on 828 chicks distributed across 106 broods:

```
R> library("MCMCglmm")
R> data("BTdata")
R> BTdata[1,]
      tarsus      back  animal      dam fosternest  hatchdate sex
```

```
1 -1.892297 1.146421 R187142 R187557      F2102 -0.6874021 Fem
```

The day after the chicks hatch, approximately half of the brood are reciprocally swapped with chicks from another nest. This results in an unbalanced cross-classified data structure where chicks share a `fosternest` with both relatives and non-relatives. Using molecular methods (**?**) the `sex` of the chicks were determined in 94% of cases, and the response variables, `tarsus` length and `back` color, were measured in all birds. The response variables are approximately normal and were mean centered and scaled to unit variance. The date on which the chicks hatched was recorded for all nests. The parental generation is assumed to consist of unrelated individuals and all chicks from the same family are assumed to share the same mother and father. Although in this example, family structure can be modeled more efficiently by fitting genetic mother (`dam`) as a random effect, we will use the more general animal model **?** which is parametrized in terms of the relationship matrix, **A**. The relationship matrix is defined by the pedigree;

```
R> data("BTped")
R> BTped[1,]
   animal  dam sire
1 R187557 <NAR> <NAR>
```

a 3 column data frame with an individual's identifier (`animal`) in the first column and its parental identifiers in the second and third columns. The pedigree often contains more individuals than are present in the data frame (in this example the pedigree also includes the parental generation) but all `animal`'s in the data frame must have a row in the pedigree.

3.1. MCMCglmm arguments

The function `MCMCglmm` within the R library of the same name is used for model fitting. **?** were interested in estimating the covariance between `tarsus` and `back` for different sources of variation and to achieve this we fitted the model:

```
R> m1<-MCMCglmm(cbind(tarsus, back) ~ trait:sex + trait:hatchdate - 1,
R>   random = ~ us(trait):animal + us(trait):fosternest, rcov = ~ us(trait):units,
R>   prior = prior, family = rep("gaussian", 2), nitt = 60000, burnin = 10000,
R>   thin=25, data = BTdata, pedigree=BTped)
```

In the following sections we work through the four main arguments taken by `MCMCglmm`: those that specify the response variables and fixed effects (`fixed`), the distribution of the response variables (`family`), the random effects and associated **G**-structure (`random`), and the **R**-structure (`rcov`). The syntax used to specify the model closely follows that used by **asreml** (**?**), an R interface to **ASReml** (**?**) - a program for fitting GLMM using restricted maximum likelihood (REML).

3.2. fixed: Response variables and fixed effects

The `fixed` argument follows the standard R formula language, and although multiple responses can be passed as a single vector, it is perhaps easier in many cases to pass them as a matrix using `cbind`. For example,

```
fixed = cbind(tarsus, back) ~ trait:sex + trait:hatchdate - 1
```

defines a bivariate model with the responses `tarsus` and `back`. For multi-response models it is usual to make use of the reserved variables `trait` and `units` which index columns and rows of the response matrix, respectively. To understand the use of these variables it can be easier to think of the response as stacked column-wise:

$$\begin{array}{ccc} & \texttt{tarsus} & \texttt{back} \\ 1 & \texttt{-1.89229718} & \texttt{1.1464212} \\ 2 & \texttt{1.13610981} & \texttt{-0.7596521} \\ \vdots & \vdots & \vdots \\ 828 & \texttt{0.833269} & \texttt{-1.438743} \end{array} \Longrightarrow \begin{array}{ccc} \texttt{y} & \texttt{trait} & \texttt{units} \\ \texttt{-1.89229718} & \texttt{tarsus} & 1 \\ \texttt{1.13610981} & \texttt{tarsus} & 2 \\ \vdots & \vdots & \vdots \\ \texttt{0.833269} & \texttt{tarsus} & 828 \\ \texttt{1.1464212} & \texttt{back} & 1 \\ \texttt{-0.7596521} & \texttt{back} & 2 \\ \vdots & \vdots & \vdots \\ \texttt{-1.438743} & \texttt{back} & 828 \end{array}$$

By fitting `trait` as a fixed effect we allow the two responses to have different means, and by fitting interactions such as `trait:hatchdate` we allow different regression slopes of the traits on `hatchdate`. Multi-response models models are generally easier to interpret when an overall intercept is suppressed (`-1`) otherwise the parameter estimates associated with `back` are interpreted as contrasts with `tarsus`.

3.3. `family`: Response variable distributions

For the above model, two distributions must be specified in the `family` argument, and we assume Gaussian distributions with identity link functions for both:

```
family = c("gaussian", "gaussian")
```

Other distributions and link functions can be specified (See Table **??**). Some distributions require more data columns than linear predictors. For example, censored data are passed as two columns, the first specifying the lowest value the data could take, and the second column specifying the highest value the data could take. However, only a single linear predictor (associated with the uncensored but unobserved data) is fitted for that distribution and it should be remembered that in this case `trait` is really indexing linear predictors, not data. Another example of this is the binomial distribution (specified as `"multinomial2"` in the family argument) which is generally specified as a two column response of successes and failures, but is parametrized by a single linear predictor of the log odds ratio. In addition, some distributions actually have more linear predictors than data columns. For example, the zero-inflated Poisson has two linear predictors; one for predicting zero-inflation and one for predicting the Poisson counts. Similarly, categorical data although passed as a single response are treated as a multinomial response with $J-1$ linear predictors (where J are the number of categories). Again, it should be remembered that in this case several levels of `trait` may be associated with different aspects of the same data column.

3.4. `random`: Random effects and G

Simple variance structures, as represented in Equation 7, can also be specified as a standard R formula:

```
random = ~ fosternest + ...
```

although this is often inappropriate, especially for multi-response models where the implicit assumption has been made that `fosternest` effects are identical for both traits. Table 3 summarizes covariance matrix specifications for the general 3 case, but to illustrate, we will focus on a 2 × 2 (co)variance matrix ($\mathbf{V}_{\texttt{f}}$) associated with `fosternest` effects:

The diagonal elements are the `fosternest` variance components for tarsus length and back color, and the off-diagonal elements are the covariance between `fosternest` effects on the two traits. The specification above, without an interaction, forces the structure:

$$\mathbf{V}_{\texttt{f}} = \left[\begin{array}{cc} \sigma^2_{\texttt{f}} & \sigma^2_{\texttt{f}} \\ \sigma^2_{\texttt{f}} & \sigma^2_{\texttt{f}} \end{array}\right] \tag{9}$$

where all components are forced to be the same. It is natural to form interactions with `trait` as we did with the fixed effects, although there are three possible ways this could be done. The straight forward interaction `trait:fosternest` although still fitting a single variance component across both traits, assumes that individual effects are independent between traits:

$$\mathbf{V}_{\texttt{f}} = \left[\begin{array}{cc} \sigma^2_{\texttt{f}} & 0 \\ 0 & \sigma^2_{\texttt{f}} \end{array}\right] \tag{10}$$

More useful interactions can be formed using the `idh()` and `us()` functions. For example, `idh(trait):fosternest` fits heterogeneous variances across traits:

$$\mathbf{V}_{\texttt{f}} = \left[\begin{array}{cc} \sigma^2_{\texttt{f:tarsus}} & 0 \\ 0 & \sigma^2_{\texttt{f:back}} \end{array}\right] \tag{11}$$

although still assumes that the two traits are independent at the `fosternest` level. The specification `us(trait):fosternest` fits the completely parametrized matrix that allows for covariance across traits:

$$\mathbf{V}_{\texttt{f}} = \left[\begin{array}{cc} \sigma^2_{\texttt{f:tarsus}} & \sigma_{\texttt{f:tarsus,back}} \\ \sigma_{\texttt{f:back,tarsus}} & \sigma^2_{\texttt{f:back}} \end{array}\right] \tag{12}$$

Since the experiment was designed to measure the covariances between the two response variables, completely parametrized (co)variance matrices are specified:

```
random = ~ us(trait):fosternest + us(trait):animal
```

For models that have pedigree or phylogenetic effects the vector of random effects needs to be associated with the inverse relationship matrix $\mathbf{A}^{-1}$. This matrix is formed by passing a pedigree or phylogeny to the `pedigree` argument of `MCMCglmm`. The individuals (or taxa) need to be associated with a column in the data frame, and this column must be called `animal`.

It is also possible to fit random interactions between categorical and continuous variables as in random regression models. For example, a random intercept-slope model with a covariance term fitted could be specified:

```
random = ~ us(1+age):individual
```

or for higher order polynomials the `poly` function could be used:

```
random = ~ us(1+poly(age, 2)):individual
```

Another form of random effect structure that does not arise in the worked example is that arising in meta analysis. In meta-analysis each data point is measured with some error. If the sampling error around the true value is approximately normal, and the variance of the sampling errors known, then random effect meta-analyses can be fitted by passing the sampling variances to the `mev` argument of `MCMCglmm`. In the simplest case, without additional random effects and i.i.d **R**-structure, the latent variables are assumed to have the multivariate normal distribution:

$$\mathbf{l} \sim N\left(\mathbf{X}\boldsymbol{\beta}, \mathbf{D} + \sigma^2_e\mathbf{I}\right) \tag{13}$$

where **D** is a diagonal matrix with `mev` along the diagonal.

3.5. `rcov`: Residual variance structure R

The **R**-structure can be parametrized in the same way as the **G**-structure although currently direct sums are not possible. However, unlike the **G**-structure it is important that the residual model is specified in away that allows each linear predictor to have a unique residual. For multi-response models forming an interaction between `trait` and `units` satisfies this condition and as with the **G**-structure various types of interaction could be considered. Again, we will use a fully parametrized covariance matrix:

```
rcov = ~ us(trait):units
```

3.6. `prior`: Response variables and fixed effects

If not defined, default priors are used which are not proper and this can lead to both inferential and numerical problems. The prior specification is passed to `MCMCglmm` via the argument `prior` which takes a list of three elements specifying the priors for the fixed effects (`B`), the **G**-structure (`G`) and the **R**-structure (`R`).

For the fixed effects, a multivariate normal prior distribution can be specified through the mean vector `mu` ($\boldsymbol{\beta}_0$) and a (co)variance matrix `V` ($\boldsymbol{B}$) passed as list elements of `B`. The default has a zero mean vector and a diagonal variance matrix with large variances (1e+10).

For non-parameter expanded models, the parameter (co)variance matrices are assumed to have (conditional) inverse-Wishart prior distributions and individual elements for each component of the variance structure take the arguments `V`, `n` and `fix` which specify the expected (co)variance matrix at the limit, the degree of freedom parameter, and the partition to condition on. The variance structure prior specification for the above models was

```
R> prior = list(R = list(V = diag(2)/3, n = 2),
R>              G = list(G1 = list(V = diag(2)/3, n = 2),
R>                       G2 = list(V = diag(2)/3, n = 2)))
```

where the expected covariance matrices for all three components of the variance structure are diagonal matrices implying *a priori* independence between `tarsus` and `back`. The traits were scaled to have unit variance prior to analysis and so the specification implies the prior belief that the total variance is evenly split across all three terms. The term `fix` has been left unspecified and so all variance parameters are estimated. However, for certain types of model it is advantageous to be able to fix sub-matrices at certain values and not estimate them. The `fix` argument partitions `V` into (potentially) 4 sub-matrices where the partition occurs on the $\texttt{fix}^{th}$ diagonal element. For example, if `V` is an $n \times n$ matrix then `V` is partitioned:

$$\texttt{V} = \begin{bmatrix} \texttt{V}_{\texttt{1:(fix-1)},\texttt{1:(fix-1)}} & \texttt{V}_{\texttt{1:(fix-1)},\texttt{fix:}n} \\ \texttt{V}_{\texttt{fix:}n,\texttt{1:(fix-1)}} & \texttt{V}_{\texttt{fix:}n,\texttt{fix:}n} \end{bmatrix} \tag{14}$$

and the lower right sub-matrix ($\texttt{V}_{\texttt{fix:}n,\texttt{fix:}n}$) is fixed and not estimated. When `fix = 1` the whole matrix is fixed.

Two further arguments that can passed are `alpha.mu` and `alpha.V` which specify the prior distribution for the non-identified working parameters. When the matrix `alpha.V` is non-null parameter expanded models are fitted. When the variance-structure defines a single variance, the prior distribution is a scaled non-central F-distribution (?). Without loss of generality we can have `V = 1` in the prior to give:

$$Pr(\sigma^2) = f_F(\sigma^2/\texttt{alpha.V}|1, \texttt{nu}, (\texttt{alpha.mu}^2)/\texttt{alpha.V})$$

where f_F is the density function of the F-distribution defined by three parameters: the numerator and denominator degrees of freedom and the non-centrality parameter, respectively.

3.7. MCMC output

The model was ran for 60,000 iterations with a burn-in phase of 10,000 and a thinning interval of 25. `MCMCglmm` returns a list with elements:

- `Sol`: Posterior distribution of location effects (and cutpoints for ordinal models)
- `VCV`: Posterior distribution of (co)variance matrices

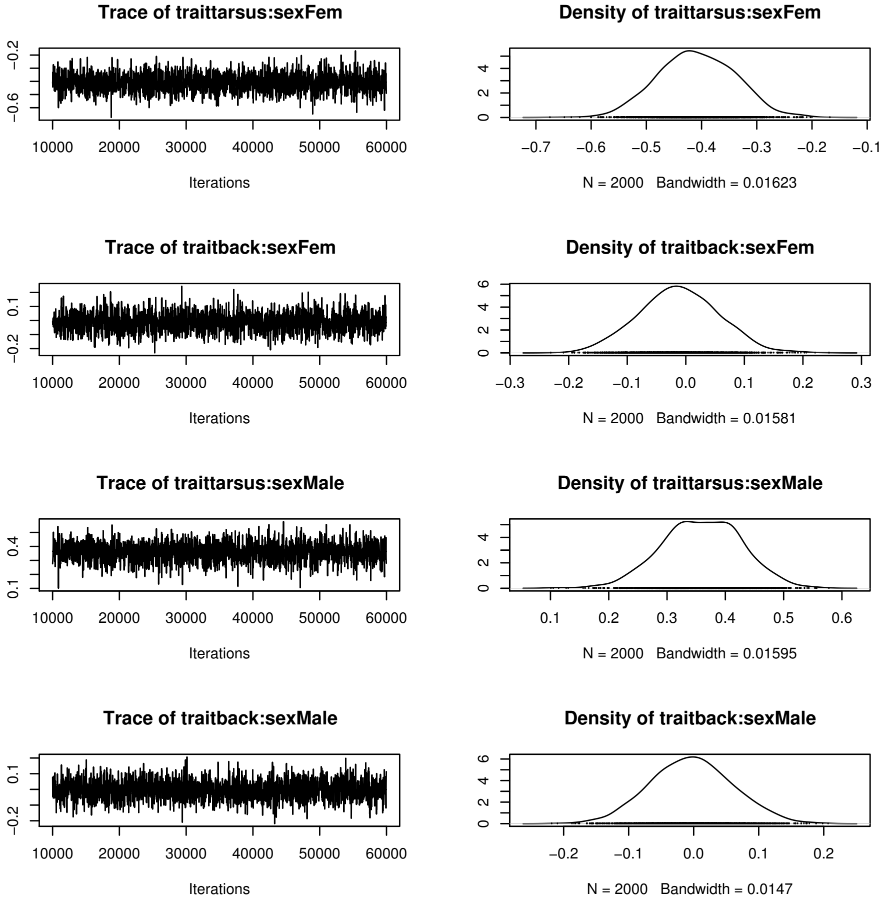

Figure 1: Trace of the sampled output and density estimates for male and female tarsus length and back color.

- `Liab`: Posterior Distribution of latent variables
- `Deviance`: Deviance
- `DIC`: Deviance Information Criterion

The samples from the posterior distribution are stored as `mcmc` objects, which can be summarized and visualized using the **coda** package (**?**). The element `Sol` contains the fixed effects ($\boldsymbol{\beta}$), and if `pr=TRUE` then also the random effects ($\mathbf{u}$). The element `VCV` contains the parameter (co)variance matrices stacked column-wise, and if `pl=TRUE` then `Liab` contains the posterior distribution of latent variables $\mathbf{l}$. The element `Deviance` contains the deviance at each stored iteration and `DIC` contains the deviance information criterion (**?**) calculated over all iterations after burn-in. Traces of the sampled output and density estimates are shown for the effects of gender on trait expression (Figure 1) and the genetic covariance matrix associated with `animal` (See Figure 2).

We also fitted alternative variance structures where some or all covariances were set to zero,

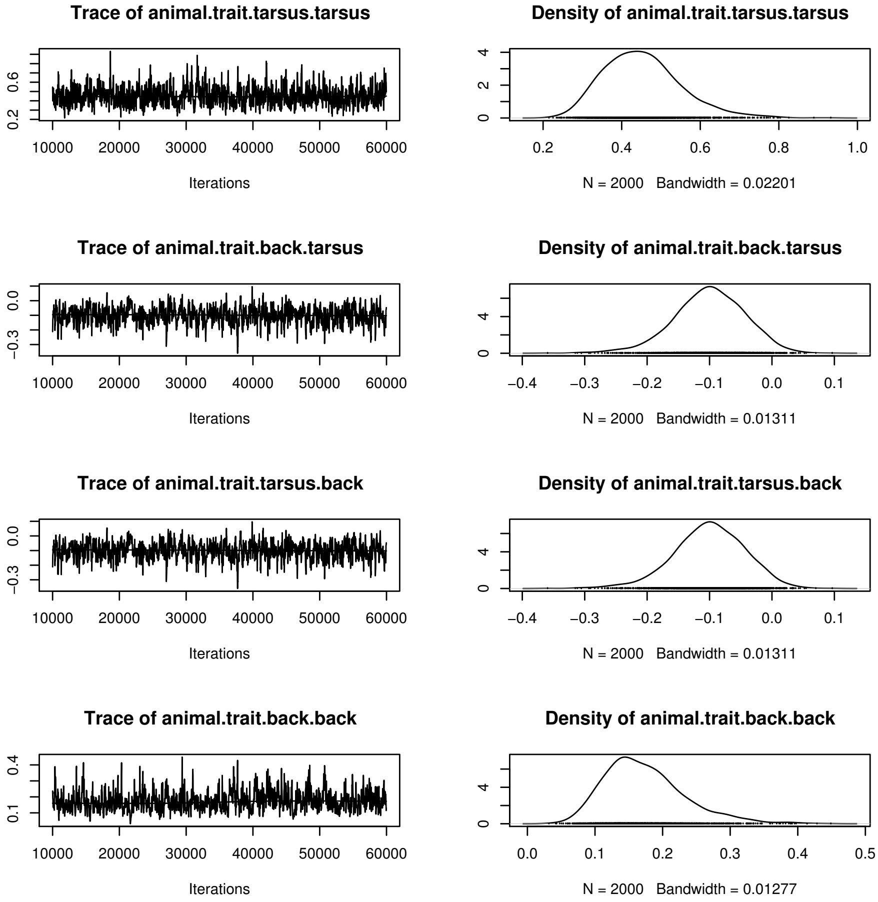

Figure 2: Trace of the sampled output and density estimates for the genetic covariance matrix of tarsus length and back color.

`animal` variance function	`fosternest` variance function	`units` variance function	DIC
us	us	us	4043.8/4041.9
idh	us	us	4050.5/4050.7
idh	idh	us	4063.0/4062.8
idh	idh	idh	4077.9/4076.7
us	idh	us	4056.2/4059.2
us	idh	idh	4091.1/4089.5
idh	us	idh	4069.8/4069.9
us	us	idh	4081.8/4082.4

Table 1: Deviance Information Criteria for several models where the covariance between the response variable for a designated source of variation was either estimated (us) or set to zero (idh). Each model was ran twice in order to asses the level of Monte Carlo error in calculating DIC.

and Table 1 shows the DIC for each model. The priors on the reduced models were set up so that the marginal prior for the variances was the same as that in the full model. The sampling error of DIC can be large and so we ran all models for an additional 500,000 iterations.

3.8. Comparison with WinBUGS

We also fitted an identical model in **WinBUGS** (code available from the author) using a multivariate extension to the method proposed by **?**. On a 2.5Ghz dual core MacBook Pro with 2GB RAM, **MCMCglmm** took 7.6 minutes and **WinBUGS** took 4.8 hours to fit the model. Moreover, the number of effective samples was 3.2 times higher in **MCMCglmm** (averaged over all parameters) indicating that the chain has better mixing properties. Because **MCMCglmm** samples all location parameters in a single block the gains in efficiency are expected to be even higher when the parameters show stronger posterior correlation.

4. Concluding remarks

This paper introduces an R package for fitting multi-response generalized linear mixed models using Markov chain Monte Carlo techniques developed in quantitative genetics (**?**). A key aspect of these techniques is that they update all location effects (fixed and random) as a single block which results in better mixing properties and shorter chain lengths than alternative strategies. This can involve repeatedly solving a very large but sparse set of mixed model equations, and the computational cost of doing this is minimized by using the **CSparse** C libraries for solving sparse linear systems (**?**). For the example data set analysed, **MCMCglmm** collected 120 times more effective samples per unit time than the same model fitted in **WinBUGS**. A range of distributions for the response variables are permitted, and flexible variance structures for the random effects and residuals included. It is hoped that this package makes the flexibility and simplicity of generalized linear mixed modeling available to a wider range of researchers.

Acknowledgements

This work would not have been possible without the **CSparse** written by Tim Davis and the comprehensive book on MCMC and mixed models by Sorensen & Gianola. This work was funded by NERC and a Leverhulme trust award to Loeske Kruuk, who together with Shinichi Nakagawa and two anonymous reviewers made helpful comments on this manuscript. I also thank Dylan Childs for help with fitting the example in **WinBUGS**.

References

A. Appendix

A.1. Updating the latent variables l

The conditional density of l is given by:

$$Pr(l_i|\mathbf{y},\boldsymbol{\theta},\mathbf{R},\mathbf{G}) \propto f_i(y_i|l_i)f_N(e_i|\mathbf{r}_i\mathbf{R}_{/i}^{-1}\mathbf{e}_{/i}, r_i-\mathbf{r}_i\mathbf{R}_{/i}^{-1}\mathbf{r}_i^{\top}) \tag{15}$$

where f_N indicates a Multivariate normal density with specified mean vector and covariance matrix. Equation 15 is the probability of the data point y_i with linear predictor l_i on the link scale for distribution f_i, multiplied by the probability of the linear predictor residual. The linear predictor residual follows a conditional normal distribution where the conditioning is on the residuals associated with data points other than i. Vectors and matrices with the row and/or column associated with i removed are denoted $/i$. In practice, this conditional distribution only involves other residuals which are expected to show some form of residual covariation, as defined by the $\mathbf{R}$ structure. Because of this we actually update latent variables in blocks, where the block is defined as groups of residuals which are expected to be correlated:

$$Pr(\mathbf{l}_j|\mathbf{y},\boldsymbol{\theta},\mathbf{R},\mathbf{G}) \propto \prod_{i\in j} p_i(y_i|l_i)f_N(\mathbf{e}_j|\mathbf{0},\mathbf{R}_j) \tag{16}$$

where j indexes blocks of latent variables that have non-zero residual covariances. A special case arises for multi-parameter distributions in which each parameter is associated with a linear predictor. For example, in the zero-inflated Poisson two linear predictors are used to model the same data point, one to predict zero-inflation, and one to predict the Poisson variable. In this case the two linear predictors are updated in a single block even when the residual covariance between them is set to zero, because the first probability in Equation 16 cannot be factored:

$$Pr(\mathbf{l}_j|\mathbf{y}, \boldsymbol{\theta}, \mathbf{R}, \mathbf{G}) \propto p_i(y_i|\mathbf{l}_j) f_N(\mathbf{e}_j|\mathbf{0}, \mathbf{R}_j) \tag{17}$$

We use adaptive methods during the burn-in phase to determine an efficient multivariate normal proposal distribution entered at the previous value of $\mathbf{l}_j$ with covariance matrix $m\mathbf{M}$. For computational efficiency we use the same $\mathbf{M}$ for each block j, where $\mathbf{M}$ is the average posterior (co)variance of $\mathbf{l}_j$ within blocks and is updated each iteration of the burn-in period **?**. The scalar m is chosen using the method of **?** so that the proportion of successful jumps is optimal, with a rate of 0.44 when $\mathbf{l}_j$ is a scalar declining to 0.23 when $\mathbf{l}_j$ is high dimensional (**?**).

For the standard linear mixed model with a Gaussian response and identity link, $Pr(l_i = y_i|\mathbf{y}, \boldsymbol{\theta}, \mathbf{R}, \mathbf{G})$ is always unity and so the Metropolis-Hastings steps are always omitted. When the latent variables within a block j are associated with missing data then their conditional distribution is multivariate normal and can be Gibbs sampled directly:

$$Pr(\mathbf{l}_j|\mathbf{y}, \boldsymbol{\theta}, \mathbf{R}, \mathbf{G}) \sim N(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}, \mathbf{R}_j) \tag{18}$$

where design matrices subscripted by j are the rows of the original design matrices associated with the latent variables in block j.

A.2. Updating the location vector $\boldsymbol{\theta} = \left[\boldsymbol{\beta}^\top \; \mathbf{u}^\top\right]^\top$

? provide a method for sampling $\boldsymbol{\theta}$ as a complete block that involves solving the sparse linear system:

$$\tilde{\boldsymbol{\theta}} = \mathbf{C}^{-1}\mathbf{W}^\top\mathbf{R}^{-1}(\mathbf{l} - \mathbf{W}\boldsymbol{\theta}_\star - \mathbf{e}_\star) \tag{19}$$

where $\mathbf{C}$ is the mixed model coefficient matrix:

$$\mathbf{C} = \mathbf{W}^\top\mathbf{R}^{-1}\mathbf{W} + \left[\begin{array}{cc} \mathbf{B}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^{-1} \end{array}\right] \tag{20}$$

and $\mathbf{W} = [\mathbf{X} \; \mathbf{Z}]$, and $\mathbf{B}$ is the prior (co)variance matrix for the fixed effects.

$\boldsymbol{\theta}_\star$ and $\mathbf{e}_\star$ are random draws from the multivariate normal distributions:

$$\boldsymbol{\theta}_\star \sim N\left(\left[\begin{array}{c} \boldsymbol{\beta}_0 \\ \mathbf{0} \end{array}\right], \left[\begin{array}{cc} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{array}\right]\right) \tag{21}$$

and

$$\mathbf{e}_{\star} \sim N\left(\mathbf{W}\boldsymbol{\theta}_{\star}, \mathbf{R}\right) \tag{22}$$

$\tilde{\boldsymbol{\theta}} + \boldsymbol{\theta}_{\star}$ gives a realization from the required probability distribution:

$$Pr(\boldsymbol{\theta}|\mathbf{l}, \mathbf{W}, \mathbf{R}, \mathbf{G}) \tag{23}$$

Equation 19 is solved using Cholesky factorization. Because $\mathbf{C}$ is sparse and the pattern of non-zero elements fixed, an initial symbolic Cholesky factorization of $\mathbf{PCP}^{\top}$ is preformed where $\mathbf{P}$ is a fill-reducing permutation matrix (**?**). Numerical factorization must be performed each iteration but the fill-reducing permutation (found via a minimum degree ordering of $\mathbf{C} + \mathbf{C}^{\top}$) reduces the computational burden dramatically compared to a direct factorization of $\mathbf{C}$ (**?**).

Forming the inverse of the variance structures is usually simpler because they can be expressed as a series of direct sums and Kronecker products:

$$\mathbf{G} = (\mathbf{V}_1 \otimes \mathbf{A}_1) \oplus (\mathbf{V}_2 \otimes \mathbf{A}_2) \oplus \dots \tag{24}$$

and the inverse of such a structure has the form

$$\mathbf{G}^{-1} = \left(\mathbf{V}_1^{-1} \otimes \mathbf{A}_1^{-1}\right) \oplus \left(\mathbf{V}_2^{-1} \otimes \mathbf{A}_2^{-1}\right) \oplus \dots \tag{25}$$

which involves inverting the parameter (co)variance matrices ($\mathbf{V}$), which are usually of low dimension, and inverting $\mathbf{A}$. For many problems $\mathbf{A}$ is actually an identity matrix and so inversion is not required. When $\mathbf{A}$ is a relationship matrix associated with a pedigree, **??** give efficient recursive algorithms for obtaining the inverse, and **?** derive a similar procedure for phylogenies.

A.3. Updating the variance structures G and R

Components of the direct sum used to construct the desired variance structures are conditionally independent. The sum of squares matrix associated with each component term has the form:

$$\mathbf{S} = \mathbf{U}^{\top}\mathbf{A}^{-1}\mathbf{U} \tag{26}$$

where $\mathbf{U}$ is a matrix of random effects where each column is associated with the relevant row/column of $\mathbf{V}$ and each row associated with the relevant row/column of $\mathbf{A}$. The parameter (co)variance matrix can then be sampled from the inverse Wishart distribution:

$$\mathbf{V} \sim IW((\mathbf{S}_p + \mathbf{S})^{-1},\ n_p + n_u) \tag{27}$$

where n_u is the number of rows in $\mathbf{U}$, and $\mathbf{S}_p$ and n_p are the prior sum of squares and prior degrees of freedom, respectively.

In some models, some elements of a parameter (co)variance matrix cannot be estimated from the data and all the information comes from the prior. In these cases it can be advantageous to fix these elements at some value and **?** provide a strategy for sampling from a conditional inverse-Wishart distribution which is appropriate when the rows/columns of the parameter matrix can be permuted so that the conditioning occurs on some diagonal sub-matrix. When this is not possible Metropolis-Hastings updates can be made.

A.4. Ordinal models

For ordinal models it is necessary to update the cutpoints which define the bin boundaries for latent variables associated with each category of the outcome. To achieve good mixing we used the method developed by (**?**) that allows the latent variables and cutpoints to be updated simultaneously using a Hastings-with-Gibbs update.

A.5. Parameter expansion

As the covariance matrix approaches a singularity the mixing of the chain becomes notoriously slow. This problem is often encountered in single-response models when a variance component is small and the chain becomes stuck at values close to zero. Similar problems occur for the EM algorithm and (**?**) introduced parameter expansion to speed up the rate of convergence. The idea was quickly applied to Gibbs sampling problems **?** and has now been extensively used to develop more efficient mixed-model samplers (e.g., **???**).

The columns of the design matrix ($\mathbf{W}$) can be multiplied by the non-identified working parameters $\boldsymbol{\alpha} = [1,\ \alpha_1,\ \alpha_2,\ \ldots \alpha_k]^\top$:

$$\mathbf{W}_\alpha = [\mathbf{X}\ \mathbf{Z}_1\alpha_1\ \mathbf{Z}_2\alpha_2\ \ldots\ \mathbf{Z}_k\alpha_k] \tag{28}$$

where the indices denote sub-matrices of $\mathbf{Z}$ which pertain to effects associated with the same variance component. Replacing $\mathbf{W}$ with $\mathbf{W}_\alpha$ we can sample the new location effects $\boldsymbol{\theta}_\alpha$ as described above, and rescale them to obtain $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = (\mathbf{I}_{n_\beta} \oplus_{i=1}^{k} \mathbf{I}_{n_{\mathbf{u}_i}}\ \alpha_i)\boldsymbol{\theta}_\alpha \tag{29}$$

where the identity matrices are equal in dimension to n_x the number of elements in the subscripted parameter vector x.

Likewise, the (co)variance matrices can be rescaled by the set of α's associated with the variances of a particular variance structure component ($\boldsymbol{\alpha}_\mathcal{V}$):

$$\mathbf{V} = Diag(\boldsymbol{\alpha}_\mathcal{V})\mathbf{V}_\alpha Diag(\boldsymbol{\alpha}_\mathcal{V}) \tag{30}$$

The working parameters are not identifiable in the likelihood, but do have a proper conditional distribution. Defining $\mathbf{X}_\alpha$ as an $n \times (k+1)$ design matrix, with each column equal to the sub-matrices in Equation 28 post-multiplied by the relevant sub-vectors of $\boldsymbol{\theta}_\alpha$, we can see that $\boldsymbol{\alpha}$ is a vector of regression coefficients:

$$\mathbf{l} = \quad \mathbf{X}_{\alpha}\boldsymbol{\alpha} + \mathbf{e} \tag{31}$$

and so the methods described above can be used to update them.

A.6. Deviance and DIC

The deviance D is defined as:

$$D = -2\text{log}(\text{Pr}(\mathbf{y}|\boldsymbol{\Omega})) \tag{32}$$

where $\boldsymbol{\Omega}$ is some parameter set of the model. The deviance can be calculated in different ways depending on what is in 'focus', and `MCMCglmm` calculates this probability for the lowest level of the hierarchy (**?**). For Gaussian response variables the likelihood is the density:

$$f_N(\mathbf{y}|\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u},\ \mathbf{R}) \tag{33}$$

where $\boldsymbol{\Omega} = \{\boldsymbol{\theta},\ \mathbf{R}\}$ but for other response variables variables it is the product:

$$\prod_i f_i(y_i|l_i) \tag{34}$$

with $\boldsymbol{\Omega} = \mathbf{l}$.

For multivariate models with mixtures of Gaussian and non-Gaussian data (including missing values) the likelihood of the Gaussian data is the density of $\mathbf{y}_g$ in the conditional density:

$$f_N\left(\mathbf{y}_g|\mathbf{X}_g\boldsymbol{\beta} + \mathbf{Z}_g\mathbf{u} + \mathbf{R}_{g,l}\mathbf{R}^{-1}_{l,l}(\mathbf{l} - \mathbf{X}_l\boldsymbol{\beta} - \mathbf{Z}_l\mathbf{u}),\ \mathbf{R}_{g,g} - \mathbf{R}_{g,l}\mathbf{R}^{-1}_{l,l}\mathbf{R}_{l,g}\right) \tag{35}$$

where the subscripts g and l denote rows of the data vector/design matrices that pertain to Gaussian data, and non-Gaussian data respectively. Subscripts on the **R**-structure index both rows and columns. The likelihood of the non-Gaussian data are identical to Equation 34 giving the complete parameter set $\boldsymbol{\Omega} = \{\boldsymbol{\theta}_g, \mathbf{R}, \mathbf{l}\}$.

The deviance is calculated at each iteration if `DIC=TRUE` and stored each `thin`th iteration after burn-in. The mean deviance ($\bar{D}$) is calculated over all iterations, as is the mean of the latent variables ($\mathbf{l}$) the **R**-structure and the vector of predictors ($\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}$). The deviance is calculated at the mean estimate of the parameters ($D(\bar{\boldsymbol{\Omega}})$) and the deviance information criterion calculated as:

$$\text{DIC} = 2\bar{D} - D(\bar{\boldsymbol{\Omega}}) \tag{36}$$

Distribution type	No. Data columns	No. latent columns	Density	function
`"gaussian"`	1	1	$Pr(y) =$	$f_N(\mathbf{w}\boldsymbol{\theta}, \sigma^2_e)$
`"poisson"`	1	1	$Pr(y) =$	$f_P(\exp(l))$
`"categorical"`	1	J-1	$Pr(y = k \vert k \neq 1) =$ $Pr(y = 1) =$	$\frac{\exp(l_k)}{1+\sum_{j=1}^{J-1}\exp(l_j)}$ $\frac{1}{1+\sum_{j=1}^{J-1}\exp(l_j)}$
`"multinomial`J`"`	J	J-1	$Pr(y_k = n_k \vert k \neq J) =$ $Pr(y_k = n_k \vert k = J) =$	$\left(\frac{\exp(l_k)}{1+\sum_{j=1}^{J-1}\exp(l_j)}\right)^{n_k}$ $\left(\frac{1}{1+\sum_{j=1}^{J-1}\exp(l_j)}\right)^{n_k}$
`"ordinal"`	1	1	$Pr(y = k) =$	$F_N(\gamma_k \vert l, 1) - F_N(\gamma_{k-1} \vert l, 1)$
`"exponential"`	1	1	$Pr(y) =$	$f_E(\exp(-l))$
`"geometric"`	1	1	$Pr(y) =$	$f_G(\frac{\exp(l)}{1+\exp(l)})$
`"cengaussian"`	2	1	$Pr(y_1 > y > y_2) =$	$F_N(y_2 \vert \mathbf{w}\boldsymbol{\theta}, \sigma^2_e) - F_N(y_1 \vert \mathbf{w}\boldsymbol{\theta}, \sigma^2_e)$
`"cenpoisson"`	2	1	$Pr(y_1 > y > y_2) =$	$F_P(y_2 \vert l) - F_P(y_1 \vert l)$
`"cenexponential"`	2	1	$Pr(y_1 > y > y_2) =$	$F_E(y_2 \vert l) - F_E(y_1 \vert l)$
`"zipoisson"`	1	2	$Pr(y = 0) =$ $Pr(y \vert y > 0) =$	$\frac{\exp(l_2)}{1+\exp(l_2)} + \left(1 - \frac{\exp(l_2)}{1+\exp(l_2)}\right) f_P(y \vert \exp(l_1))$ $\left(1 - \frac{\exp(l_2)}{1+\exp(l_2)}\right) f_P(y \vert \exp(l_1))$

`"ztpoisson"`	1	1	$Pr(y) = \frac{f_P(y\vert \exp(l))}{1-f_P(0\vert \exp(l))}$
`"hupoisson"`	1	2	$Pr(y=0) = \frac{\exp(l_2)}{1+\exp(l_2)}$ $Pr(y\vert y>0) = \left(1-\frac{\exp(l_2)}{1+\exp(l_2)}\right)\frac{f_P(y\vert \exp(l_1))}{1-f_P(0\vert \exp(l_1))}$
`"zapoisson"`	1	2	$Pr(y=0) = 1-\exp(\exp(l_2))$ $Pr(y\vert y>0) = \exp(\exp(l_2))\frac{f_P(y\vert \exp(l_1))}{1-f_P(0\vert \exp(l_1))}$
`"zibinomial"`	2	2	$Pr(y_1=0) = \frac{\exp(l_2)}{1+\exp(l_2)}+\left(1-\frac{\exp(l_2)}{1+\exp(l_2)}\right)f_B(0,n=y_1+y_2\vert \frac{\exp(l_1)}{1+\exp(l_1)})$ $Pr(y_1\vert y_1>0) = \left(1-\frac{\exp(l_2)}{1+\exp(l_2)}\right)f_B(y_1,n=y_1+y_2\vert \frac{\exp(l_1)}{1+\exp(l_1)})$

Table 2: Distribution types that can fitted using `MCMCglmm`. The prefixes `"zi"`, `"zt"`, `"hu"` and `"za"` stand for zero-inflated, zero-truncated, hurdle and zero-altered respectively. The prefix `"cen"` standards for censored where y_1 and y_2 are the upper and lower bounds for the unobserved datum y. J stands for the number of categories in the multinomial/categorical distributions and this must be specified in the family argument for the multinomial distribution. The density function is for a single datum in a univariate model with $\mathbf{w}$ being a row vector of $\mathbf{W}$. f and F are the density and distribution functions for the subscripted distribution (N=Normal, P=Poisson, E=Exponential, G=Geometric, B=Binomial). The $J-1$ γ's in the ordinal models are the cutpoints, with γ_1 set to zero.

Affiliation:

Jarrod Hadfield
The EGI
Department of Zoology
University of Oxford
Oxford, OX1 3PS, UK
E-mail: jarrod.hadfield@zoo.ox.ac.uk
URL: http://www.zoo.ox.ac.uk/egi/people/researchfellows/jarrod_harrod.htm

Syntax	n	Covariance	Correlation
`rfactor`	1	$\begin{bmatrix} V & V & V \\ V & V & V \\ V & V & V \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
`us(ffactor):rfactor`	6	$\begin{bmatrix} V_{1,1} & V_{1,2} & V_{1,3} \\ V_{1,2} & V_{2,2} & V_{2,3} \\ V_{1,3} & C_{2,3} & V_{3,3} \end{bmatrix}$	$\begin{bmatrix} 1 & r_{1,2} & r_{1,3} \\ r_{1,2} & 1 & r_{2,3} \\ r_{1,3} & r_{2,3} & 1 \end{bmatrix}$
`ffactor:rfactor`	1	$\begin{bmatrix} V & 0 & 0 \\ 0 & V & 0 \\ 0 & 0 & V \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
`rfactor+ffactor:rfactor`	2	$\begin{bmatrix} V_1+V_2 & V_1 & V_1 \\ V_1 & V_1+V_2 & V_1 \\ V_1 & V_1 & V_1+V_2 \end{bmatrix}$	$\begin{bmatrix} 1 & r & r \\ r & 1 & r \\ r & r & 1 \end{bmatrix}^{\dagger}$
`idh(ffactor):rfactor`	3	$\begin{bmatrix} V_{1,1} & 0 & 0 \\ 0 & V_{2,2} & 0 \\ 0 & 0 & V_{3,3} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table 3: Different random effect specifications where `ffactor` is a factor with three levels and `rfactor` is a factor with (usually) many levels. The resulting 3×3 covariance and correlation matrices of `rfactor` effects within and across `ffactor` factor levels are given, together with the number of parameters to be estimated (n). Continuous variables can also go within the variance structure functions (e.g., `us`,`idh`). In this case the associated parameters are regression coefficients for which (co)variances are estimated. $^{\dagger}: rR > 0$