hglm: A Package for Fitting Hierarchical Generalized Linear Models

by Lars Rönnegård, Xia Shen and Moudud Alam

Abstract We present the **hglm** package for fitting hierarchical generalized linear models. It can be used for linear mixed models and generalized linear mixed models with random effects for a variety of links and a variety of distributions for both the outcomes and the random effects. Fixed effects can also be fitted in the dispersion part of the model.

Introduction

The **hglm** package (Alam et al., 2010) implements the estimation algorithm for hierarchical generalized linear models (HGLM; Lee and Nelder, 1996). The package fits generalized linear models (GLM; McCullagh and Nelder, 1989) with random effects, where the random effect may come from a distribution conjugate to one of the exponential-family distributions (normal, gamma, beta or inverse-gamma). The user may explicitly specify the design matrices both for the fixed and random effects. In consequence, correlated random effects, as well as random regression models can be fitted. The dispersion parameter can also be modeled with fixed effects.

The main function is `hglm()` and the input is specified in a similar manner as for `glm()`. For instance,

```
R> hglm(fixed = y ~ week, random = ~ 1|ID,
        family = binomial(link = logit))
```

fits a logit model for `y` with `week` as fixed effect and `ID` representing the clusters for a normally distributed random intercept. Given an `hglm` object, the standard generic functions are `print()`, `summary()` and `plot()`.

Generalized linear mixed models (GLMM) have previously been implemented in several R functions, such as the `lmer()` function in the **lme4** package (Bates and Maechler, 2010) and the `glmmPQL()` function in the **MASS** package (Venables and Ripley, 2002). In GLMM, the random effects are assumed to be Gaussian whereas the `hglm()` function allows other distributions to be specified for the random effect. The `hglm()` function also extends the fitting algorithm of the **dglm** package (Dunn and Smyth, 2009) by including random effects in the linear predictor for the mean, i.e. it extends the algorithm so that it can cope with mixed models. Moreover, the model specification in `hglm()` can be given as a formula or alternatively in terms of `y`, `X`, `Z` and `X.disp`. Here `y` is the vector of observed responses, `X` and `Z` are the design matrices for the fixed and random effects, respectively, in the linear predictor for the means and `X.disp` is the design matrix for the fixed effects in the dispersion parameter. This enables a more flexible modeling of the random effects than specifying the model by an R formula. Consequently, this option is not as user friendly but gives the user the possibility to fit random regression models and random effects with known correlation structure.

The **hglm** package produces estimates of fixed effects, random effects and variance components as well as their standard errors. In the output it also produces diagnostics such as deviance components and leverages.

Three illustrating models

The **hglm** package makes it possible to

1. include fixed effects in a model for the residual variance,
2. fit models where the random effect distribution is not necessarily Gaussian,
3. estimate variance components when we have correlated random effects.

Below we describe three models that can be fitted using `hglm()`, which illustrate these three points. Later, in the Examples section, five examples are presented that include the R syntax and output for the `hglm()` function.

Linear mixed model with fixed effects in the residual variance

We start by considering a normal-normal model with heteroscedastic residual variance. In biology, for instance, this is important if we wish to model a random genetic effect (e.g., Rönnegård and Carlborg, 2007) for a trait y, where the residual variance differs between the sexes.

For the response y and observation number i we have:

$$y_i \mid \beta, u, \beta_d \sim N\left(X_i\beta + Z_i u, \exp\left(X_{d,i}\beta_d\right)\right)$$
$$u \sim MVN\left(0, \mathbf{I}\sigma_u^2\right)$$

where β are the fixed effects in the mean part of the model, the random effect u represents random variation among clusters of observations and β_d is the fixed effect in the residual variance part of the model. The variance of the random effect u is given by σ_u^2.

The subscript i for the matrices $\mathbf{X}$, $\mathbf{Z}$, and $\mathbf{X}_d$ indicates the i'th row. Here, a log link function is used for the residual variance and the model for the residual variance is therefore given by $\exp(X_{d,i}\beta_d)$. In the more general GLM notation, the residual variance here is described by the dispersion term ϕ, so we have $\log(\phi_i) = X_{d,i}\beta_d$.

This model cannot be fitted with the **dglm** package, for instance, because we have random effects in the mean part of the model. It is also beyond the scope of the `lmer()` function since we allow a model for the residual variance.

The implementation in `hglm()` for this model is demonstrated in Example 2 in the Examples section below.

A Poisson model with gamma distributed random effects

For dependent count data it is common to model a Poisson distributed response with a gamma distributed random effect (Lee et al., 2006). If we assume no overdispersion conditional on u and thereby have a fixed dispersion term, this model may be specified as:

$$E(y_i \mid \beta, u) = \exp(X_i\beta + Z_i v)$$

where a level j in the random effect v is given by $v_j = \log(u_j)$ and u_j are iid with gamma distribution having mean and variance: $E(u_j) = 1$, $var(u_j) = \lambda$.

This model can also be fitted with the **hglm** package, since it extends existing GLMM functions (e.g. `lmer()`) to allow a non-normal distribution for the random effect. Later on, in Example 3, we show the `hglm()` code used for fitting a gamma-Poisson model with fixed effects included in the dispersion parameter.

A linear mixed model with a correlated random effect

In animal breeding it is important to estimate variance components prior to ranking of animal performances (Lynch and Walsh, 1998). In such models the genetic effect of each animal is modeled as a level in a random effect and the correlation structure $\mathbf{A}$ is a matrix with known elements calculated from the pedigree information. The model is given by

$$y_i \mid \beta, u \sim N\left(X_i\beta + Z_i u, \sigma_e^2\right)$$
$$u \sim MVN\left(0, \mathbf{A}\sigma_u^2\right)$$

This may be reformulated as (see Lee et al., 2006; Rönnegård and Carlborg, 2007)

$$y_i \mid \beta, u \sim N\left(X_i\beta + Z_i^* u^*, \sigma_e^2\right)$$
$$u^* \sim MVN(0, \mathbf{I}\sigma_u^2)$$

where $\mathbf{Z}^* = \mathbf{ZL}$ and $\mathbf{L}$ is the Cholesky factorization of $\mathbf{A}$.

Thus the model can be fitted using the `hglm()` function with a user-specified input matrix `Z` (see R code in Example 4 below).

Overview of the fitting algorithm

The fitting algorithm is described in detail in Lee et al. (2006) and is summarized as follows. Let n be the number of observations and k be the number of levels in the random effect. The algorithm is then:

1. Initialize starting values.
2. Construct an augmented model with response $y_{aug} = \begin{pmatrix} y \\ E(u) \end{pmatrix}$.
3. Use a GLM to estimate β and v given the vector ϕ and the dispersion parameter for the random effect λ. Save the deviance components and leverages from the fitted model.
4. Use a gamma GLM to estimate β_d from the first n deviance components d and leverages h obtained from the previous model. The response variable and weights for this model are $d/(1-h)$ and $(1-h)/2$, respectively. Update the dispersion parameter by putting ϕ equal to the predicted response values for this model.
5. Use a similar GLM as in Step 4 to estimate λ from the last k deviance components and leverages obtained from the GLM in Step 3.
6. Iterate between steps 3-5 until convergence.

For a more detailed description of the algorithm in a particular context, see below.

H-likelihood theory

Let y be the response and u an unobserved random effect. The **hglm** package fits a hierarchical model $y \mid u \sim f_m(\mu, \phi)$ and $u \sim f_d(\psi, \lambda)$ where f_m and f_d are specified distributions for the mean and dispersion parts of the model.

We follow the notation of Lee and Nelder (1996), which is based on the GLM terminology by McCullagh and Nelder (1989). We also follow the likelihood approach where the model is described in terms of likelihoods. The conditional (log-)likelihood for y given u has the form of a GLM

$$\ell(\theta', \phi; y \mid u) = \frac{y\theta' - b(\theta')}{a(\phi)} + c(y, \phi) \qquad (1)$$

where θ' is the canonical parameter, ϕ is the dispersion term, μ' is the conditional mean of y given u

where $\eta' = g(\mu')$, i.e. $g()$ is a link function for the GLM. The linear predictor is given by $\eta' = \eta + v$ where $\eta = X\beta$ and $v = v(u)$ for some strict monotonic function of u. The link function $v(u)$ should be specified so that the random effects occur linearly in the linear predictor to ensure meaningful inference from the h-likelihood (Lee et al., 2007). The *h-likelihood* or hierarchical likelihood is defined by

$$h = \ell(\theta', \phi; y \mid u) + \ell(\alpha; v) \tag{2}$$

where $\ell(\alpha; v)$ is the log density for v with parameter α. The estimates of β and v are given by $\frac{\partial h}{\partial \beta} = 0$ and $\frac{\partial h}{\partial v} = 0$. The dispersion components are estimated by maximizing the *adjusted profile h-likelihood*

$$h_p = \left(h - \frac{1}{2}\log|-\frac{1}{2\pi}H| \right)_{\beta=\hat{\beta}, v=\hat{v}} \tag{3}$$

where H is the Hessian matrix of the h-likelihood. The dispersion term ϕ can be connected to a linear predictor $X_d\beta_d$ given a link function $g_d(\cdot)$ with $g_d(\phi) = X_d\beta_d$. The adjusted profile likelihoods of ℓ and h may be used for inference of β, v and the dispersion parameters ϕ and λ (pp. 186 in Lee et al., 2006). More detail and discussion of h-likelihood theory is presented in the **hglm** vignette.

Detailed description of the hglm fitting algorithm for a linear mixed model with heteroscedastic residual variance

In this section we describe the fitting algorithm in detail for a linear mixed model where fixed effects are included in the model for the residual variance. The extension to distributions other than Gaussian is described at the end of the section.

Lee and Nelder (1996) showed that linear mixed models can be fitted using a hierarchy of GLM by using an augmented linear model. The linear mixed model

$$y = \mathbf{X}b + \mathbf{Z}u + e$$
$$v = \mathbf{Z}\mathbf{Z}^T\sigma_u^2 + \mathbf{R}\sigma_e^2$$

where $\mathbf{R}$ is a diagonal matrix with elements given by the estimated dispersion model (i.e. ϕ defined below). In the first iteration of the HGLM algorithm, $\mathbf{R}$ is an identity matrix. The model may be written as an augmented weighted linear model:

$$y_a = \mathbf{T}_a\delta + e_a \tag{4}$$

where

$$y_a = \begin{pmatrix} y \\ 0_q \end{pmatrix} \qquad \mathbf{T}_a = \begin{pmatrix} \mathbf{X} & \mathbf{Z} \\ \mathbf{0} & \mathbf{I}_q \end{pmatrix}$$
$$\delta = \begin{pmatrix} b \\ u \end{pmatrix} \qquad e_a = \begin{pmatrix} e \\ -u \end{pmatrix}$$

Here, q is the number of columns in $\mathbf{Z}$, 0_q is a vector of zeros of length q, and $\mathbf{I}_q$ is the identity matrix of size $q \times q$. The variance-covariance matrix of the augmented residual vector is given by

$$V(e_a) = \begin{pmatrix} \mathbf{R}\sigma_e^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_q\sigma_u^2 \end{pmatrix}$$

Given σ_e^2 and σ_u^2, this weighted linear model gives the same estimates of the fixed and random effects (b and u respectively) as Henderson's mixed model equations (Henderson, 1976).

The estimates from weighted least squares are given by:

$$\mathbf{T}_a^t\mathbf{W}^{-1}\mathbf{T}_a\hat{\delta} = \mathbf{T}_a^t\mathbf{W}^{-1}y_a$$

where $\mathbf{W} \equiv V(e_a)$.

The two variance components are estimated iteratively by applying a gamma GLM to the residuals e_i^2 and u_i^2 with intercept terms included in the linear predictors. The leverages h_i for these models are calculated from the diagonal elements of the hat matrix:

$$\mathbf{H}_a = \mathbf{T}_a(\mathbf{T}_a^t\mathbf{W}^{-1}\mathbf{T}_a)^{-1}\mathbf{T}_a^t\mathbf{W}^{-1} \tag{5}$$

A gamma GLM is used to fit the dispersion part of the model with response

$$y_{d,i} = e_i^2/(1-h_i) \tag{6}$$

where $E(y_d) = \mu_d$ and $\mu_d \equiv \phi$ (i.e. σ_e^2 for a Gaussian reponse). The GLM model for the dispersion parameter is then specified by the link function $g_d(.)$ and the linear predictor $X_d\beta_d$, with prior weights $(1-h_i)/2$, for

$$g_d(\mu_d) = X_d\beta_d \tag{7}$$

Similarly, a gamma GLM is fitted to the dispersion term α (i.e. σ_u^2 for a GLMM) for the random effect v, with

$$y_{\alpha,j} = u_j^2/(1-h_{n+j}), j = 1,2,...,q \tag{8}$$

and

$$g_\alpha(\mu_\alpha) = \lambda \tag{9}$$

where the prior weights are $(1-h_{n+j})/2$ and the estimated dispersion term for the random effect is given by $\hat{\alpha} = g_\alpha^{-1}(\hat{\lambda})$.

The algorithm iterates by updating both $\mathbf{R} = diag(\hat{\phi})$ and $\sigma_u^2 = \hat{\alpha}$, and subsequently going back to Eq. (4).

For a non-Gaussian response variable y, the estimates are obtained simply by fitting a GLM instead of Eq. (4) and by replacing e_i^2 and u_j^2 with the deviance components from the augmented model (see Lee et al., 2006).

Implementation details

Distributions and link functions

There are two important classes of models that can be fitted in **hglm**: GLMM and conjugate HGLM. GLMMs have Gaussian random effects. Conjugate HGLMs have been commonly used partly due to the fact that explicit formulas for the marginal likelihood exist. HGLMs may be used to fit models in survival analysis (frailty models), where for instance the complementary-log-log link function can be used on binary responses (see e.g., Carling et al., 2004). The gamma distribution plays an important role in modeling responses with a constant coefficient of variation (see Chapter 8 in McCullagh and Nelder, 1989). For such responses with a gamma distributed random effect we have a gamma-gamma model. A summary of the most important models is given in Tables 1 and 2. Note that the random-effect distribution can be an arbitrary conjugate exponential-family distribution. For the specific case where the random-effect distribution is a conjugate to the distribution of y, this is called a *conjugate HGLM*. Further implementation details can be found in the **hglm** vignette.

Possible future developments

In the current version of `hglm()` it is possible to include a single random effect in the mean part of the model. An important development would be to include several random effects in the mean part of the model and also to include random effects in the dispersion parts of the model. The latter class of models is called Double HGLM and has been shown to be a useful tool for modeling heavy tailed distributions (Lee and Nelder, 2006).

The algorithm of `hglm()` gives true marginal likelihood estimates for the fixed effects in conjugate HGLM (Lee and Nelder, 1996, pp. 629), whereas for other models the estimates are approximated. Lee and co-workers (see Lee et al., 2006, and references therein) have developed higher-order approximations, which are not implemented in the current version of the **hglm** package. For such extensions, we refer to the commercially available GenStat software (Payne et al., 2007), the recently available R package **HGLMMM** (Molas, 2010) and also to coming updates of **hglm**.

Examples

Example 1: A linear mixed model

Data description The output from the `hglm()` function for a linear mixed model is compared to the results from the `lme()` function in the **nlme** (Pinheiro et al., 2009) package using simulated data. In the simulated data there are five clusters with 20 observations in each cluster. For the mean part of the model, the simulated intercept value is $\mu = 0$, the variance for the random effect is $\sigma_u^2 = 0.2$, and the residual variance is $\sigma_e^2 = 1.0$.

Both functions produce the same estimate of the fixed intercept effect of 0.1473 (s.e. 0.16) and also the same variance component estimates. The `summary.hglm()` function gives the estimate of the variance component for the random intercept (0.082) as well as the residual variance (0.84). It also gives the logarithm of the variance component estimates together with standard errors below the lines `Model estimates for the dispersion term` and `Dispersion model for the random effects`. The `lme()` function gives the square root of the variance component estimates.

The model diagnostics produced by the `plot.hglm` function are shown in Figures 1 and 2. The data are completely balanced and therefore produce equal leverages (hatvalues) for all observations and also for all random effects (Figure 1). Moreover, the assumption of the deviance components being gamma distributed is acceptable (Figure 2).

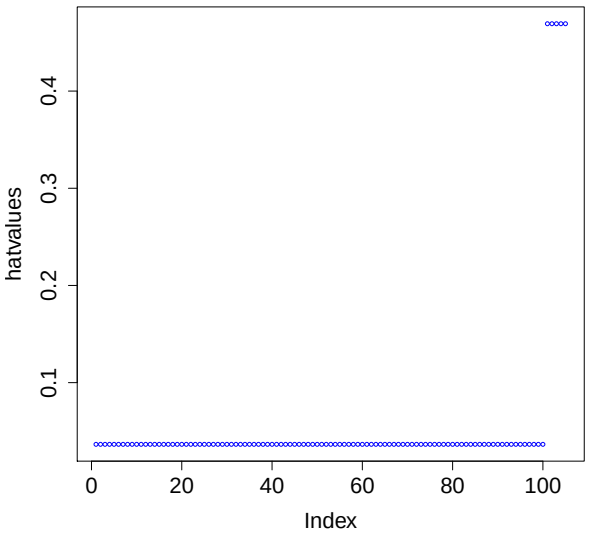

Figure 1: Hatvalues (i.e. diagonal elements of the augmented hat-matrix) for each observation 1 to 100, and for each level in the random effect (index 101-105).

Table 1: Commonly used distributions and link functions possible to fit with `hglm()`

Model name	$y \mid u$ distribution	Link $g(\mu)$	u distribution	Link $v(u)$
Linear mixed model	Gaussian	identity	Gaussian	identity
Binomial conjugate	Binomial	logit	Beta	logit
Binomial GLMM	Binomial	logit	Gaussian	identity
Binomial frailty	Binomial	comp-log-log	Gamma	log
Poisson GLMM	Poisson	log	Gaussian	identity
Poisson conjugate	Poisson	log	Gamma	log
Gamma GLMM	Gamma	log	Gaussian	identity
Gamma conjugate	Gamma	inverse	Inverse-Gamma	inverse
Gamma-Gamma	Gamma	log	Gamma	log

Table 2: `hglm` code for commonly used models

Model name	Setting for `family` argument	Setting for `rand.family` argument
Linear mixed model[a]	`gaussian(link = identity)`	`gaussian(link = identity)`
Beta-Binomial	`binomial(link = logit)`	`Beta(link = logit)`
Binomial GLMM	`binomial(link = logit)`	`gaussian(link = identity)`
Binomial frailty	`binomial(link = cloglog)`	`Gamma(link = log)`
Poisson GLMM	`poisson(link = log)`	`gaussian(link = identity)`
Poisson frailty	`poisson(link = log)`	`Gamma(link = log)`
Gamma GLMM	`Gamma(link = log)`	`gaussian(link = identity)`
Gamma conjugate	`Gamma(link = inverse)`	`inverse.gamma(link = inverse)`
Gamma-Gamma	`Gamma(link = log)`	`Gamma(link = log)`

[a] For example, the `hglm()` code for a linear mixed model is

```
hglm(family = gaussian(link = identity), rand.family = gaussian(link = identity), ...)
```

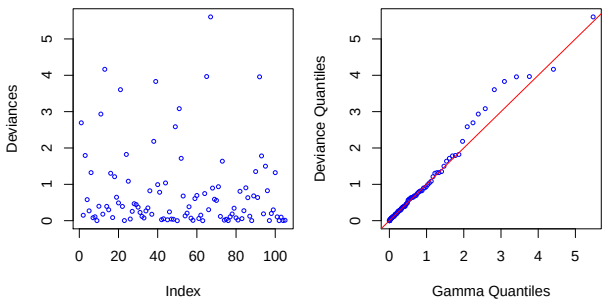

Figure 2: Deviance diagnostics for each observation and each level in the random effect.

The R code and output for this example is as follows:

```
R> set.seed(123)
R> n.clus <- 5 #No. of clusters
R> n.per.clus <- 20  #No. of obs. per cluster
R> sigma2_u <- 0.2 #Variance of random effect
R> sigma2_e <- 1 #Residual variance
R> n <- n.clus*n.per.clus
R> X <- matrix(1, n, 1)
R> Z <- diag(n.clus)%x%rep(1, n.per.clus)
R> a <- rnorm(n.clus, 0, sqrt(sigma2_u))
R> e <- rnorm(n, 0, sqrt(sigma2_e))
R> mu <- 0
R> y <- mu + Z%*%a + e
R> lmm <- hglm(y = y, X = X, Z = Z)
R> summary(lmm)
```

```
R> plot(lmm)

Call:
hglm.default(X = X, y = y, Z = Z)

DISPERSION MODEL
WARNING: h-likelihood estimates through EQL can be biased.
Model estimates for the dispersion term:[1] 0.8400608

Model estimates for the dispersion term:
Link = log
Effects:
  Estimate Std. Error
   -0.1743     0.1441

Dispersion = 1 is used in Gamma model on deviances
to calculate the standard error(s).
Dispersion parameter for the random effects
[1] 0.08211

Dispersion model for the random effects:
Link = log
Effects:
  Estimate Std. Error
   -2.4997     0.8682

Dispersion = 1 is used in Gamma model on deviances
to calculate the standard error(s).

MEAN MODEL
Summary of the fixed effects estimates
    Estimate Std. Error t value Pr(>|t|)
X.1   0.1473     0.1580   0.933    0.353
Note: P-values are based on 96 degrees of freedom
Summary of the random effects estimate
     Estimate Std. Error
[1,]  -0.3237     0.1971
[2,]  -0.0383     0.1971
```

```
[3,]   0.3108     0.1971
[4,]  -0.0572     0.1971
[5,]   0.1084     0.1971

EQL estimation converged in 5 iterations.

R> #Same analysis with the lme function
R> library(nlme)
R> clus <- rep(1:n.clus,
+              rep(n.per.clus, n.clus))
R> summary(lme(y ~ 0 + X,
+              random = ~ 1 | clus))

Linear mixed-effects model fit by REML
Data: NULL
      AIC      BIC    logLik
  278.635 286.4203 -136.3175

Random effects:
Formula: ~1 | clus
        (Intercept) Residual
StdDev:   0.2859608   0.9166

Fixed effects: y ~ 0 + X
      Value Std.Error DF   t-value p-value
X 0.1473009 0.1573412 95 0.9361873  0.3516

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-2.5834807 -0.6570612  0.0270673  0.6677986  2.1724148

Number of Observations: 100
Number of Groups: 5
```

Example 2: Analysis of simulated data for a linear mixed model with heteroscedastic residual variance

Data description Here, a heteroscedastic residual variance is added to the simulated data from the previous example. Given the explanatory variable x_d, the simulated residual variance is 1.0 for $x_d = 0$ and 2.72 for $x_d = 1$. The output shows that the variance of the random effect is 0.109, and that $\hat{\beta}_d = (-0.32, 1.47)$, i.e. the two residual variances are estimated as 0.72 and 3.16. (Code continued from Example 1)

```
R> beta.disp <- 1
R> X_d <- matrix(1, n, 2)
R> X_d[,2] <- rbinom(n, 1, .5)
R> colnames(X_d) <- c("Intercept", "x_d")
R> e <- rnorm(n, 0,
+       sqrt(sigma2_e*exp(beta.disp*X_d[,2])))
R> y <- mu + Z%*%a + e
R> summary(hglm(y = y, X = X, Z = Z,
+               X.disp = X_d))

Call:
hglm.default(X = X, y = y, Z = Z, X.disp = X_d)

DISPERSION MODEL
WARNING: h-likelihood estimates through EQL can be biased.
Model estimates for the dispersion term:
Link = log
Effects:
         Estimate Std. Error
Intercept  -0.3225     0.2040
x_d         1.4744     0.2881

Dispersion = 1 is used in Gamma model on deviances
to calculate the standard error(s).
Dispersion parameter for the random effects
[1] 0.1093

Dispersion model for the random effects:
Link = log
Effects:
  Estimate Std. Error
   -2.2135     0.8747

Dispersion = 1 is used in Gamma model on deviances
to calculate the standard error(s).
MEAN MODEL
Summary of the fixed effects estimates
    Estimate Std. Error t value Pr(>|t|)
X.1  -0.0535     0.1836  -0.291    0.771
Note: P-values are based on 96 degrees of freedom
Summary of the random effects estimate
     Estimate Std. Error
[1,]   0.0498     0.2341
[2,]  -0.2223     0.2276
[3,]   0.4404     0.2276
[4,]  -0.1786     0.2276
[5,]  -0.0893     0.2296

EQL estimation converged in 5 iterations.
```

Example 3: Fitting a Poisson model with gamma random effects, and fixed effects in the dispersion term

Data description We simulate a Poisson model with random effects and estimate the parameter in the dispersion term for an explanatory variable x_d. The estimated dispersion parameter for the random effects is 0.6556. (Code continued from Example 2)

```
R> u <- rgamma(n.clus,1)
R> eta <- exp(mu + Z%*%u)
R> y <- rpois(length(eta), eta)
R> gamma.pois <- hglm(y = y, X = X, Z = Z,
+                     X.disp = X_d,
+                     family = poisson(
+                     link = log),
+                     rand.family =
+                     Gamma(link = log))
R> summary(gamma.pois)

Call:
hglm.default(X = X, y = y, Z = Z,
    family = poisson(link = log),
    rand.family = Gamma(link = log), X.disp = X_d)

DISPERSION MODEL
WARNING: h-likelihood estimates through EQL can be biased.
Model estimates for the dispersion term:
Link = log
Effects:
         Estimate Std. Error
Intercept  -0.0186     0.2042
x_d         0.4087     0.2902

Dispersion = 1 is used in Gamma model on deviances
to calculate the standard error(s).
Dispersion parameter for the random effects
```

```
[1] 1.926

Dispersion model for the random effects:
Link = log
Effects:
  Estimate Std. Error
    0.6556     0.7081

Dispersion = 1 is used in Gamma model on deviances
to calculate the standard error(s).
MEAN MODEL
Summary of the fixed effects estimates
    Estimate Std. Error t value Pr(>|t|)
X.1   2.3363     0.6213    3.76 0.000293
---

Note: P-values are based on 95 degrees of freedom
Summary of the random effects estimate
     Estimate Std. Error
[1,]   1.1443     0.6209
[2,]  -1.6482     0.6425
[3,]  -2.5183     0.6713
[4,]  -1.0243     0.6319
[5,]   0.2052     0.6232

EQL estimation converged in 3 iterations.
```

Example 4: Incorporating correlated random effects in a linear mixed model - a genetics example

Data description The data consists of 2025 individuals from two generations where 1000 individuals have observed trait values y that are approximately normal (Figure 3). The data we analyze was simulated for the QTLMAS 2009 Workshop (Coster et al., 2010)[1]. A longitudinal growth trait was simulated. For simplicity we analyze only the values given on the third occasion at age 265 days.

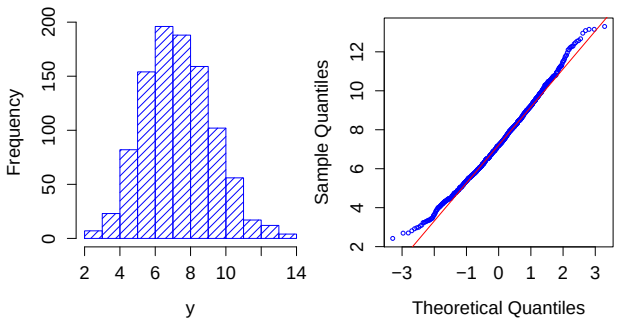

Figure 3: Histogram and qqplot for the analyzed trait.

We fitted a model with a fixed intercept and a random animal effect, a, where the correlation structure of a is given by the additive relationhip matrix **A** (which is obtained from the available pedigree information). An incidence matrix $\mathbf{Z}_0$ was constructed and relates observation number with id-number in the pedigree. For observation y_i coming from individual j in the ordered pedigree file $\mathbf{Z}_0[i,j] = 1$, and all other elements are 0. Let **L** be the Cholesky factorization of **A**, and $\mathbf{Z} = \mathbf{Z}_0\mathbf{L}$. The design matrix for the fixed effects, **X**, is a column of ones. The estimated variance components are $\hat{\sigma}^2_e = 2.21$ and $\hat{\sigma}^2_u = 1.50$.

The R code for this example is given below.

```
R> data(QTLMAS)
R> y <- QTLMAS[,1]
R> Z <- QTLMAS[,2:2026]
R> X <- matrix(1, 1000, 1)
R> animal.model <- hglm(y = y, X = X, Z = Z)
R> print(animal.model)

Call:
hglm.default(X = X, y = y, Z = Z)

Fixed effects:
     X.1
7.279766
Random effects:
   [1] -1.191733707  1.648604776  1.319427376 -0.928258503
   [5] -0.471083317 -1.058333534  1.011451565  1.879641994
   [9]  0.611705900 -0.259125073 -1.426788944 -0.005165978
...

Dispersion parameter for the mean model:[1] 2.211169
Dispersion parameter for the random effects:[1] 1.502516

EQL estimation converged in 2 iterations
```

Example 5: Binomial-beta model applied to seed germination data

Data description The seed germination data presented by Crowder (1978) has previously been analyzed using a binomial GLMM (Breslow and Clayton, 1993) and a binomial-beta HGLM (Lee and Nelder, 1996). The data consists of 831 observations from 21 germination plates. The effect of seed variety and type of root extract was studied in a 2×2 factorial lay-out. We fit the binomial-beta HGLM used by Lee and Nelder (1996) and setting `fix.disp = 1` in `hglm()` produces comparable estimates to the ones obtained by Lee and Nelder (with differences $< 2 \times 10^{-3}$). The beta distribution parameter α in Lee and Nelder (1996) was defined as $1/(2a)$ where a is the dispersion term obtained from `hglm()`. The output from the R code given below gives $\hat{a} = 0.0248$ and the corresponding estimate given in Lee and Nelder (1996) is $\hat{a} = 1/(2\hat{\alpha}) = 0.023$. We conclude that the **hglm** package produces similar results as the ones presented in Lee and Nelder (1996) and the dispersion parameters estimated using the EQL method in GenStat differ by less than 1%. Additional examples, together with comparisons to estimates produced by GenStat, are given in the **hglm** vignette included in the package on CRAN.

```
R> data(seeds)
R> germ <- hglm(
+    fixed = r/n ~ extract*I(seed=="073"),
```

[1] http://www.qtlmas2009.wur.nl/UK/Dataset

```
+     weights = n, data = seeds,
+     random = ~1|plate, family = binomial(),
+     rand.family = Beta(), fix.disp = 1)
R> summary(germ)

Call:
hglm.formula(family = binomial(), rand.family = Beta(),
    fixed = r/n ~ extract * I(seed == "O73"),
    random = ~1 | plate, data = seeds,
    weights = n, fix.disp = 1)

DISPERSION MODEL
WARNING: h-likelihood estimates through EQL can be biased.
Model estimates for the dispersion term:[1] 1

Model estimates for the dispersion term:
Link = log
Effects:
[1] 1

Dispersion = 1 is used in Gamma model on deviances to
calculate the standard error(s).
Dispersion parameter for the random effects
[1] 0.02483

Dispersion model for the random effects:
Link = log

Effects:
  Estimate Std. Error
   -3.6956     0.5304

Dispersion = 1 is used in Gamma model on deviances to
calculate the standard error(s).
MEAN MODEL
Summary of the fixed effects estimates
                              Estimate Std. Error t value
(Intercept)                    -0.5421    0.1928  -2.811
extractCucumber                 1.3386    0.2733   4.898
I(seed == "O73")TRUE            0.0751    0.3114   0.241
extractCucumber:I(seed=="O73")  -0.8257    0.4341  -1.902
                              Pr(>|t|)
(Intercept)                   0.018429
extractCucumber               0.000625
I(seed == "O73")TRUE          0.814264
extractCucumber:I(seed=="O73") 0.086343
---

Note: P-values are based on 10 degrees of freedom
Summary of the random effects estimate
      Estimate Std. Error
 [1,]  -0.2333     0.2510
 [2,]   0.0085     0.2328
 ...
[21,]  -0.0499     0.2953

EQL estimation converged in 7 iterations.
```

Summary

The hierarchical generalized linear model approach offers new possibilities to fit generalized linear models with random effects. The **hglm** package extends existing GLMM fitting algorithms to include fixed effects in a model for the residual variance, fits models where the random effect distribution is not necessarily Gaussian and estimates variance components for correlated random effects. For such models there are important applications in, for instance: genetics (Noh et al., 2006), survival analysis (Ha and Lee, 2005), credit risk modeling (Alam and Carling, 2008), count data (Lee et al., 2006) and dichotomous responses (Noh and Lee, 2007). We therefore expect that this new package will be of use for applied statisticians in several different fields.

Bibliography

M. Alam and K. Carling. Computationally feasible estimation of the covariance structure in generalized linear mixed models GLMM. *Journal of Statistical Computation and Simulation*, 78:1227–1237, 2008.

M. Alam, L. Ronnegard, and X. Shen. *hglm: Hierarchical Generalized Linear Models*, 2010. URL http://CRAN.R-project.org/package=hglm. R package version 1.1.1.

D. Bates and M. Maechler. *lme4: Linear mixed-effects models using S4 classes*, 2010. URL http://CRAN.R-project.org/package=lme4. R package version 0.999375-37.

N. E. Breslow and D. G. Clayton. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, 88:9–25, 1993.

K. Carling, L. Rönnegård, and K. Roszbach. An analysis of portfolio credit risk when counterparties are interdependent within industries. *Sveriges Riksbank Working Paper*, 168, 2004.

A. Coster, J. Bastiaansen, M. Calus, C. Maliepaard, and M. Bink. QTLMAS 2010: Simulated dataset. *BMC Proceedings*, 4(Suppl 1):S3, 2010.

M. J. Crowder. Beta-binomial ANOVA for proportions. *Applied Statistics*, 27:34–37, 1978.

P. K. Dunn and G. K. Smyth. *dglm: Double generalized linear models*, 2009. URL http://CRAN.R-project.org/package=dglm. R package version 1.6.1.

I. D. Ha and Y. Lee. Comparison of hierarchical likelihood versus orthodox best linear unbiased predictor approaches for frailty models. *Biometrika*, 92: 717–723, 2005.

C. R. Henderson. A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. *Biometrics*, 32(1):69–83, 1976.

Y. Lee and J. A. Nelder. Double hierarchical generalized linear models with discussion. *Applied Statistics*, 55:139–185, 2006.

Y. Lee and J. A. Nelder. Hierarchical generalized linear models with discussion. *J. R. Statist. Soc. B*, 58: 619–678, 1996.

Y. Lee, J. A. Nelder, and Y. Pawitan. *Generalized linear models with random effects*. Chapman & Hall/CRC, 2006.

Y. Lee, J. A. Nelder, and M. Noh. H-likelihood: problems and solutions. *Statistics and Computing*, 17: 49–55, 2007.

M. Lynch and B. Walsh. *Genetics and analysis of Quantitative Traits*. Sinauer Associates, Inc., 1998. ISBN 087893481.

P. McCullagh and J. A. Nelder. *Generalized linear models*. Chapman & Hall/CRC, 1989.

M. Molas. *HGLMMM: Hierarchical Generalized Linear Models*, 2010. URL http://CRAN.R-project.org/package=HGLMMM. R package version 0.1.1.

M. Noh and Y. Lee. REML estimation for binary data in GLMMs. *Journal of Multivariate Analysis*, 98:896–915, 2007.

M. Noh, B. Yip, Y. Lee, and Y. Pawitan. Multicomponent variance estimation for binary traits in family-based studies. *Genetic Epidemiology*, 30:37–47, 2006.

R. W. Payne, D. A. Murray, S. A. Harding, D. B. Baird, and D. M. Soutar. GenStat for Windows (10th edition) introduction, 2007. URL http://www.vsni.co.uk/software/genstat.

J. Pinheiro, D. Bates, S. DebRoy, D. Sarkar, and the R Core team. *nlme: Linear and Nonlinear Mixed Effects Models*, 2009. URL http://CRAN.R-project.org/package=nlme. R package version 3.1-96.

L. Rönnegård and Ö. Carlborg. Separation of base allele and sampling term effects gives new insights in variance component QTL analysis. *BMC Genetics*, 8(1), 2007.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL http://www.stats.ox.ac.uk/pub/MASS4. ISBN 0-387-95457-0.

Lars Rönnegård
Statistics Unit
Dalarna University, Sweden
and
Department of Animal Breeding and Genetics
Swedish University of Agricultural Sciences, Sweden
lrn@du.se

Xia Shen
Department of Cell and Molecular Biology
Uppsala University, Sweden
and
Statistics Unit
Dalarna University, Sweden
xia.shen@lcb.uu.se

Moudud Alam
Statistics Unit
Dalarna University, Sweden
maa@du.se