

The **hglm** Package (Version 2.0)

by Lars Rönnegård, Moudud Alam, Xia Shen.

Package Maintainer: Xia Shen <www.shen.se>

Abstract This vignette describes the R **hglm** package via a series of applications that may be of interest to applied scientists. The **hglm** package implements the estimation algorithm for hierarchical generalized linear models. The package fits generalized linear models with random effects, where the random effect may come from a conjugate exponential-family distribution (Gaussian, Gamma, Beta or inverse-Gamma). The design matrices both for the fixed and random effects can be explicitly specified, which allows fitting correlated random effects as well as random regression models. Fixed effects may also be modeled in the dispersion parameter. The **hglm** package produces estimates of fixed effects, random effects, variance components as well as their standard errors. Model diagnostics such as deviances and leverages can be visualized. At the end of this vignette, estimates from the **hglm** package are compared to the ones from other software packages including GenStat for several examples previously published by Lee and Nelder.

Contents

The hglm Package (Version 2.0)	1
2.1 Introduction	2
2.2 Important implementation details	3
2.2.1 Brief overview of the fitting algorithm	3
2.2.2 The h-likelihood theory	3
2.2.3 Detailed description of the fitting algorithm for a linear mixed model with heteroscedastic residual variance	4
2.2.4 Distributions and link functions	5
2.2.5 Interacting with the hglm function	6
2.2.6 HL(1,1) estimator and EQL1 correction for GLMM	8
2.2.7 Possible future developments	10
2.3 Linear mixed model with fixed effects in the residual variance	10
2.4 Poisson model with Gamma distributed random effects	13
2.5 Poisson-Gamma model with an offset for the mean model	15
2.6 Linear mixed model with a correlated random effect	15
2.7 Random regression with a Gamma distributed random effect	17
2.8 Binomial model with a Beta distributed random effect	18
2.9 Gamma HGLM with a structured dispersion	20
2.10 More than one random effect in the hglm2 function	21
2.11 EQL1 correction for a Poisson GLMM	26
2.12 Fitting a spatial Markov Random Field model using the CAR family	27
2.13 Fitting several random effects from different families	30
2.14 Likelihood-ratio test for dispersion parameters	31
2.15 Discussions on h-likelihood theory	32

2.16 Comparison with estimates using GenStat	34
2.17 Comparison with estimates using HGLMMM	36
2.18 Comparison with estimates using spaMM	37

Introduction

The **hglm** package implements the estimation algorithm for hierarchical generalized linear model (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 conjugate exponential-family distribution (normal, gamma, beta or inverse-gamma). The user may explicitly specify the design matrices both for the fixed and random effects, which means that correlated random effects as well as random regression models can be fitted. Fixed effects may also be modeled in the dispersion parameter.

Generalized linear mixed models (GLMM) have previously been implemented in several R (R Development Core Team, 2009) function, such as the `glmer()` function in the **lme4** library and in the `glmmPQL()` function in the **MASS** library. In GLMM, the random effects are assumed to be Gaussian whereas the `hglm()` function allow for other distributions for the random effect. The `hglm()` function also extends the fitting algorithm of Gordon Smyth's **dglm** package by including random effects in the linear predictor for the mean. Moreover, the model specification in `hglm()` can be given as a formula or alternatively in terms of y , X , Z and $X.\text{disp}$, where 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 mean, and $X.\text{disp}$ 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 a possibility to fit random regression models and random effects with known correlation structure.

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

New in version 1.2

From version 1.2 of the package it is possible to fit several random effects (see Salamnder example below). The `bigRR = TRUE` option was added for fitting $p \gg n$ problems (see also the **bigRR** package; Shen et al., 2013). Likelihoods are computed with the `calc.like = TRUE` option.

New in version 2.0

From version 2.0 of the package (Alam et al., 2014) it is possible to

- Fit several random effects from different distributions (e.g. `rand.family = list(Gamma(), gaussian())`).
- Fit a linear predictor for the dispersion of the random effects (e.g. `X.rand.disp = X`).
- Fit a spatial CAR model for the random effects (e.g. `rand.family = CAR(D = nbr)`).
- Fit a spatial SAR model for the random effects (e.g. `rand.family = SAR(D = nbr)`).
- Use “HL(1,1)” correction on EQL, see Lee and Lee (2012); Noh and Lee (2007) (e.g. `method = "EQL1"`).
- Perform a likelihood-ratio test for the the dispersion parameter of the random effects (function `lrt()`).

See Section 2.11, 2.12, 2.13 and 2.14 for examples.

Table 1: `hglm` functions.

Function	Description	Reference
Beta	Extended usage of the Beta family	Lee et al. (2006) Lee and Nelder (1996)
CAR	Conditional autoregressive family	
GLM.MME	Internal IWLS estimation for hglm()	
hglm	Fitting hierarchical generalized linear models	
hglm2	hglm model specified in lmer single equation style	
inverse.gamma	Extended usage of the inverse-Gamma family	
SAR	Simultaneous autoregressive family	
- Utilities -		
logLik	Print log-likelihoods while calc.like = TRUE	
lrt	Print likelihood-ratio test while calc.like = TRUE	
plot	Plot individual deviances and hatvalues for the fitted hglm objects	
print	Produce basic statistics from hglm estimation in a simplified way	
summary	Produce standard summary statistics for the fitted hglm objects	

Important implementation details

Brief overview of the fitting algorithm

The fitting algorithm is described in detail in Lee et al. (2006) and we summarize it here. Let n be the number of observations and k be the number of levels in the random effect. The algorithm is then given by:

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 deviances and leverages from the fitted model;
4. Use a gamma GLM to estimate β_d from the first n deviance residuals 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 fitted response values for this model;
5. Use a similar GLM as in Step 4 to estimate λ from the last k deviance residuals and leverages obtained from the GLM in Step 3;
6. Iterate between steps 3-5 until convergence.

The h-likelihood theory

Let y be the response and u an unobserved random effects. The `hglm` package fits a hierarchical model $y|u \sim f_m(\mu, \phi)$ 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:

$$l(\theta', \phi; y|u) = \frac{y\theta' - b(\theta')}{a(\phi)} + c(y, \phi) \quad (1)$$

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. $g(\cdot)$ is a link function for the GLM. The linear predictor μ' is given by $\eta' = \eta + v$ where $\eta = X\beta$ and $v = v(u)$ for some strict monotonic function of u . The hierarchical likelihood (h -likelihood) is defined by:

$$h = l(\theta', \phi; y|u) + l(\alpha; v) \quad (2)$$

where $l(\alpha; v)$ 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:

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

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](#)).

Detailed description of the 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 other distributions than Gaussian are 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

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{Z}\mathbf{u} + \mathbf{e}$$

$$\mathbf{V} = \mathbf{Z}\mathbf{Z}'\sigma_u^2 + \mathbf{R}\sigma_e^2$$

where \mathbf{R} is a diagonal matrix, and in the first iteration of the HGLM algorithm \mathbf{R} is equal to the identity matrix. The model may be written as an augmented weighted linear model:

$$\mathbf{y}_a = \mathbf{T}_a\delta + \mathbf{e}_a \quad (4)$$

where

$$\begin{aligned} \mathbf{y}_a &= \begin{pmatrix} \mathbf{y} \\ \mathbf{0}_q \end{pmatrix} \\ \mathbf{T}_a &= \begin{pmatrix} \mathbf{X} & \mathbf{Z} \\ \mathbf{0} & \mathbf{I}_q \end{pmatrix} \\ \delta &= \begin{pmatrix} \beta \\ \mathbf{u} \end{pmatrix} \\ \mathbf{e}_a &= \begin{pmatrix} \mathbf{e} \\ -\mathbf{u} \end{pmatrix} \end{aligned}$$

Here, q is the number of columns in \mathbf{Z} , $\mathbf{0}_q$ 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

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

Given σ_e^2 and σ_u^2 , this weighted linear model gives the same estimates of the fixed and random effects (β and \mathbf{u} respectively) as [Henderson \(1976\)](#)'s mixed model equations.

The estimates from weighted least squares are given by:

$$\mathbf{T}_a' \mathbf{W}^{-1} \mathbf{T}_a \hat{\delta} = \mathbf{T}_a' \mathbf{W}^{-1} \mathbf{y}_a$$

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

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:

$$\mathbf{H}_a = \mathbf{T}_a (\mathbf{T}_a' \mathbf{W}^{-1} \mathbf{T}_a)^{-1} \mathbf{T}_a' \mathbf{W}^{-1} \quad (5)$$

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

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

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

$$g_d(\mu_d) = X_d \beta_d \quad (7)$$

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

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

and

$$g_\alpha(\mu_\alpha) = \lambda \quad (9)$$

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 $\mathbf{R} = \text{diag}(\hat{\phi})$ and 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 residuals from the augmented model ([Lee et al., 2006](#)).

Based on $\log f_\theta(y|v)$, [Lee and Nelder \(1996\)](#) proposed using the scaled deviance for the goodness-of-fit test, having the estimated degrees of freedom, $d.f. = n - p_D$, where

$$p_D = \text{trace}\{(\mathbf{T}_m' \mathbf{\Sigma}_m^{-1} \mathbf{T}_m)^{-1}\} \mathbf{T}_m' \mathbf{\Sigma}_0^{-1} \mathbf{T}_m$$

with $\mathbf{\Sigma}_0^{-1} = \mathbf{W}_{ma}\{\text{diag}(\mathbf{\Phi}^{-1}, 0)\}$, and m represents the mean model. $E(D)$ is the expected deviance. [Lee and Nelder \(1996\)](#) showed that, under the assumed model, degrees of freedom can be estimated as $E(D) \approx n - p_D$. This extends the scaled deviance test for GLMs to HGLMs.

Distributions and link functions

There are two important classes of models that can be fitted in `hglm`: GLMM and conjugate HGLM. In GLMM we have a Gaussian random effect, whereas the conjugate HGLM has also been commonly used since explicit formulas for the marginal likelihood exist. HGLMs can also 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 (e.g. [Carling et al., 2004](#); [Alam and Carling, 2008](#)). The gamma distribution plays an important role in modeling responses with a constant coefficient of variation (see Chapter 8 in [McCullagh and Nelder, 1989](#)), and for such responses with a gamma distributed random effect we have a gamma-gamma model. A summary of the most important models are given in Tables 2 and 3.

Table 2: Commonly used distributions and link functions possible to fit with `hglm()`

Model name	$y u$ 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 3: `hglm()` code for commonly used models

Model name	<code>hglm()</code> code: <code>family =</code>	<code>hglm()</code> code: <code>rand.family =</code>
Linear mixed model	<code>gaussian(link = identity)</code>	<code>gaussian(link = identity)</code>
Beta-Binomial	<code>binomial(link = logit)</code>	<code>Beta(link = logit)</code>
Binomial GLMM	<code>binomial(link = logit)</code>	<code>gaussian(link = identity)</code>
Binomial frailty	<code>binomial(link = cloglog)</code>	<code>Gamma(link = log)</code>
Poisson GLMM	<code>poisson(link = log)</code>	<code>gaussian(link = identity)</code>
Poisson frailty	<code>poisson(link = log)</code>	<code>Gamma(link = log)</code>
Gamma GLMM	<code>Gamma(link = log)</code>	<code>gaussian(link = identity)</code>
Gamma conjugate	<code>Gamma(link = inverse)</code>	<code>inverse.gamma(link = inverse)</code>
Gamma-Gamma	<code>Gamma(link = log)</code>	<code>Gamma(link = log)</code>

Interacting with the `hglm` function

The main function is `hglm()` and the input is specified in a similar manner as for `glm()`. For instance, to fit a logit model for y with `week` as fixed effect and `ID` represents the clusters for a normally distributed random intercept, we run

```
require(hglm)
data <- data.frame(y = y, week = week, ID = factor(ID))
hglm(fixed = y ~ week, random = ~1 | ID, family = binomial(link = logit), data = data)
```

Given an `hglm` object, the standard generic functions are `print()`, `summary()` and `plot()`. For this example, `hglm` allows an alternative command if the user would like to define the design matrices directly. If the design matrices of `week` and `ID` have been defined as

```
fixed.design <- model.matrix(~week)
random.design <- model.matrix(~factor(ID) - 1)
```

respectively, we may run the following command instead.

```
hg1m(X = fixed.design, y = y, Z = random.design, family = binomial(link = logit))
```

Now we analyze the bacteria data available in the [MASS](#) library using a binomial GLMM. The data consists of observations on the presence of *H.influenzae* at five occasions (at weeks 0, 2, 4, 6 and 11) on 50 individuals. Thirty observations were not reported and there are in total 220 observations.

```
library(MASS)
data(bacteria)
g1 <- hg1m(fixed = y ~ week, random = ~1 | ID, data = bacteria, family = binomial(link = logit))
summary(g1)

## Call:
## hg1m.formula(family = binomial(link = logit), fixed = y ~ week,
##      random = ~1 | ID, data = bacteria)
##
## -----
## MEAN MODEL
## -----
##
## Summary of the fixed effects estimates:
##
##      Estimate Std. Error t-value Pr(>|t|)
## (Intercept)  2.30216     0.33627   6.846 9.84e-11 ***
## week        -0.13510     0.04127  -3.273  0.00126 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Note: P-values are based on 193 degrees of freedom
##
## Summary of the random effects estimates:
##
##      Estimate Std. Error
## IDX01    0.7472     0.9897
## IDX02   -0.2844     0.8385
## IDX03    0.8602     0.9591
## ...
## NOTE: to show all the random effects, use print(summary(hg1m.object), print.ranef = TRUE).
##
## -----
## DISPERSION MODEL
## -----
##
## NOTE: h-likelihood estimates through EQL can be biased.
##
## Dispersion parameter for the mean model:
## [1] 0.7581
##
## Model estimates for the dispersion term:
##
## Link = log
##
## Effects:
##      Estimate Std. Error
##      -0.2769     0.1019
##
## Dispersion = 1 is used in Gamma model on deviances to calculate the standard error(s).
##
## Dispersion parameter for the random effects:
```



```
## [1] 1.613
##
## Dispersion model for the random effects:
##
## Link = log
##
## Effects:
## .|Random1
## Estimate Std. Error
## 0.4779 0.2816
##
## Dispersion = 1 is used in Gamma model on deviances to calculate the standard error(s).
##
## EQL estimation converged in 14 iterations.
```

The model diagnostics produced by the `hg1m()` function are shown in Figures 1 and 2. In Figure 1 there are two random effects with leverages > 0.7 , which correspond to two individuals that only have two observations each. We also see that the assumption of the deviance residuals being gamma distributed is acceptable (Figure 2). The variance of the random individual effect was estimated to 1.6126 and the algorithm converged in 14 iterations.

```
plot(g1, device = "pdf", name = "fig_bacteria")
```

HL(1,1) estimator and EQL1 correction for GLMM

The theory below summarizes the first order fixed effects correction for generalized linear mixed models (GLMM) derived in Lee & Lee (2012). Consider a GLMM with response \mathbf{y} , fixed effect $\boldsymbol{\beta}$ and random effect \mathbf{v}

$$\boldsymbol{\mu} = E[\mathbf{y}|\mathbf{v}] \quad (10)$$

$$g(\boldsymbol{\mu}) = \boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{v} \quad (11)$$

$$\mathbf{v} \sim N(0, \mathbf{W}_2^{-1}) \quad (12)$$

Here, \mathbf{X} and \mathbf{Z} are design matrices for the fixed and random effect, respectively. The linearized response is $\mathbf{z}_1 = \boldsymbol{\eta} + (\mathbf{y} - \boldsymbol{\mu})\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\mu}}$ having a diagonal variance matrix \mathbf{W}_1 . The hierarchical likelihood is $h(\boldsymbol{\beta}, \sigma_v^2, \phi) = \log(f(\mathbf{y}|\mathbf{v})) + \log(f(\mathbf{v}))$, where we have $\mathbf{W}_2^{-1} = \sigma_v^2 \mathbf{I}$ and ϕ is the dispersion parameter. $\frac{\partial h}{\partial \boldsymbol{\beta}} = 0$ and $\frac{\partial h}{\partial \sigma_v^2} = 0$ give the following mixed model equations for GLMM

$$\mathbf{X}'\mathbf{W}_1\mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{X}'\mathbf{W}_1\mathbf{Z}\hat{\mathbf{v}} = \mathbf{X}'\mathbf{W}_1\mathbf{z}_1 \quad (13)$$

$$\mathbf{X}'\mathbf{W}_1\mathbf{Z}\hat{\boldsymbol{\beta}} + (\mathbf{Z}'\mathbf{W}_1\mathbf{Z} + \mathbf{W}_2)\hat{\mathbf{v}} = \mathbf{Z}'\mathbf{W}_1\mathbf{z}_1 \quad (14)$$

The estimating equations for the HL(1,1) estimator are derived from $\frac{\partial p_v(h)}{\partial \boldsymbol{\beta}} = 0$ and $\frac{\partial h}{\partial \sigma_v^2} = 0$, where $p_v(h)$ is the adjusted profile h-likelihood (Lee and Nelder, 1996). Lee and Lee (2012) showed that $\frac{\partial p_v(h)}{\partial \boldsymbol{\beta}} = 0$ and $\frac{\partial h}{\partial \sigma_v^2} = 0$ give the following estimating equations for GLMM

$$\mathbf{X}'\mathbf{W}_1\mathbf{X}\hat{\boldsymbol{\beta}} + \mathbf{X}'\mathbf{W}_1\mathbf{Z}\hat{\mathbf{v}} = \mathbf{X}'\mathbf{W}_1(\mathbf{z}_1 - \mathbf{s}\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\mu}}) \quad (15)$$

$$\mathbf{X}'\mathbf{W}_1\mathbf{Z}\hat{\boldsymbol{\beta}} + (\mathbf{Z}\mathbf{W}_1\mathbf{Z} + \mathbf{W}_2)\hat{\mathbf{v}} = \mathbf{Z}'\mathbf{W}_1\mathbf{z}_1 \quad (16)$$

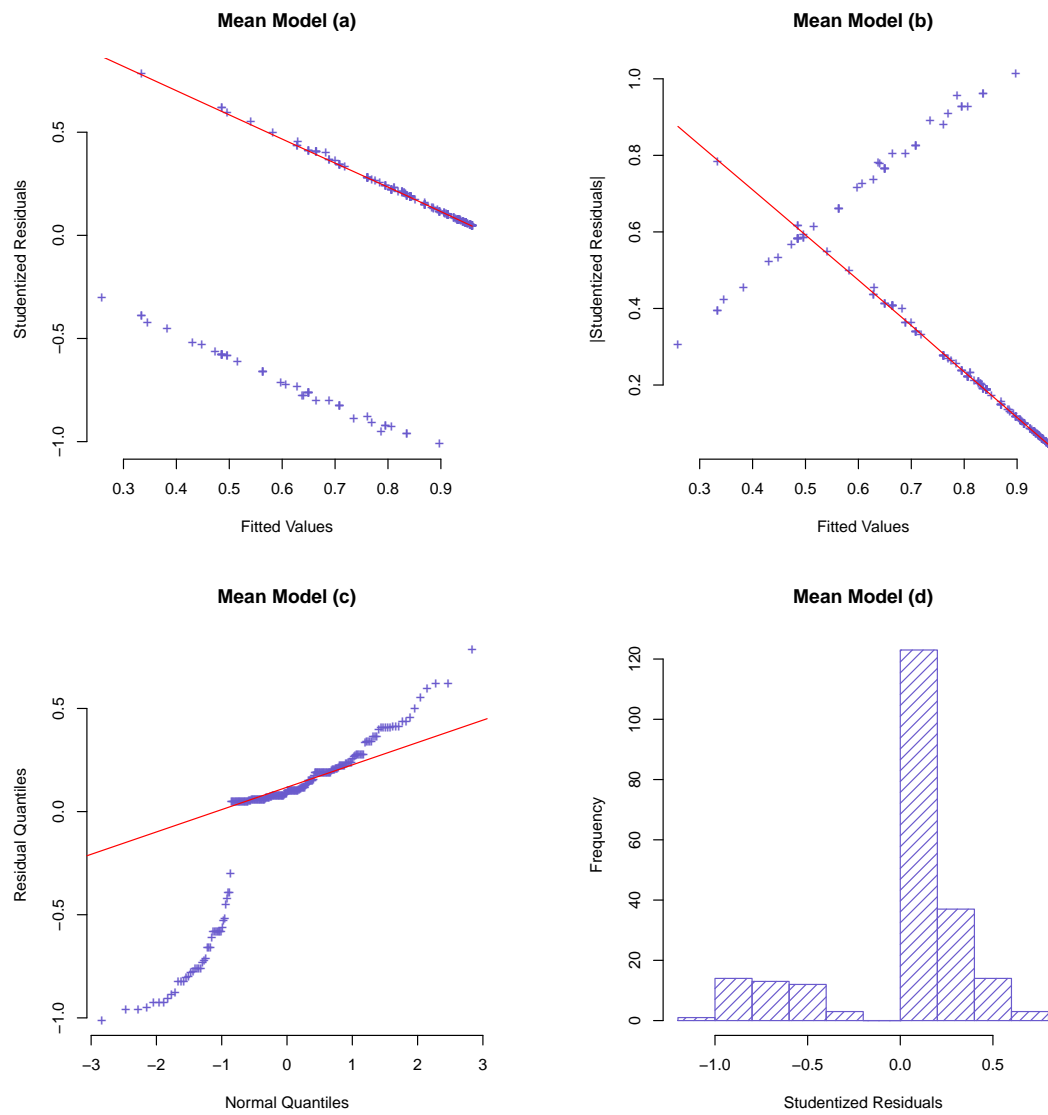


Figure 1: Residual diagnostics for the observations in the mean model.

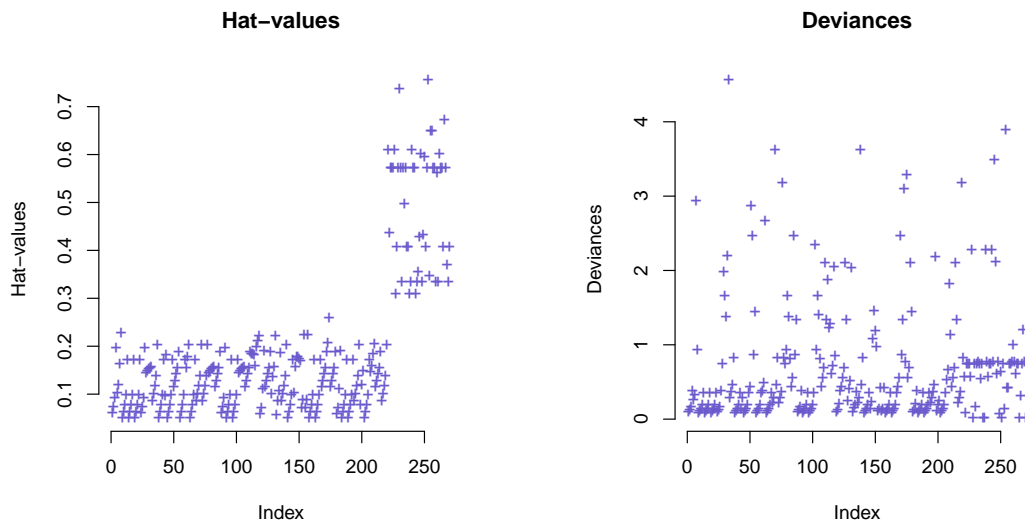


Figure 2: Leverages (i.e. diagonal elements of the augmented hat-matrix) for each observation 1 to 220, and for each level in the random effect (index 221-282).

Hence, the HL(1,1) estimator can be computed merely by adjusting \mathbf{z}_1 in the mixed model equations by the vector $\mathbf{s}_{\frac{\partial \eta}{\partial \mu}}$, given in the Appendix of [Lee and Lee \(2012\)](#). See also the comparison with the implementation in the **HGLMMM** package in section 2.17. In **hglm**, such correction is implemented as `method = "EQL1"`. The slight difference between EQL1 and [Lee and Lee \(2012\)](#)'s "HL(1,1)" correction is that [Lee and Lee \(2012\)](#) applied the correction to their HL(0,1) estimates whereas we apply such correction directly to EQL. The EQL algorithm does not account for the fact that the estimator for the random effects is a function of the dispersion parameters for non-normal mixed models. HL(0,1) accounts for this dependency, but otherwise the estimators are identical.

Possible future developments

In the current version of `hglm()` it is possible to random effects in the mean part of the model. An important development would be to include random effects in the dispersion parts of the model too. The latter class of models are called Double HGLM and have 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 conjugate HGLM, 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 give very good estimates. Only the EQL1 correction is currently implemented (from version 2.0), whereas higher-order approximations are not implemented. For these possible future extensions, we refer to the commercially available GenStat software and also to coming updates of the **hglm** package.

Linear mixed model with fixed effects in the residual variance

We consider 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 and where the residual variance is different between sexes.

For the response y and observation number i we have:

$$y_i | \beta, u, \beta_d \sim N(X_i \beta + Z_i u, \exp(X_{d,i} \beta_d))$$

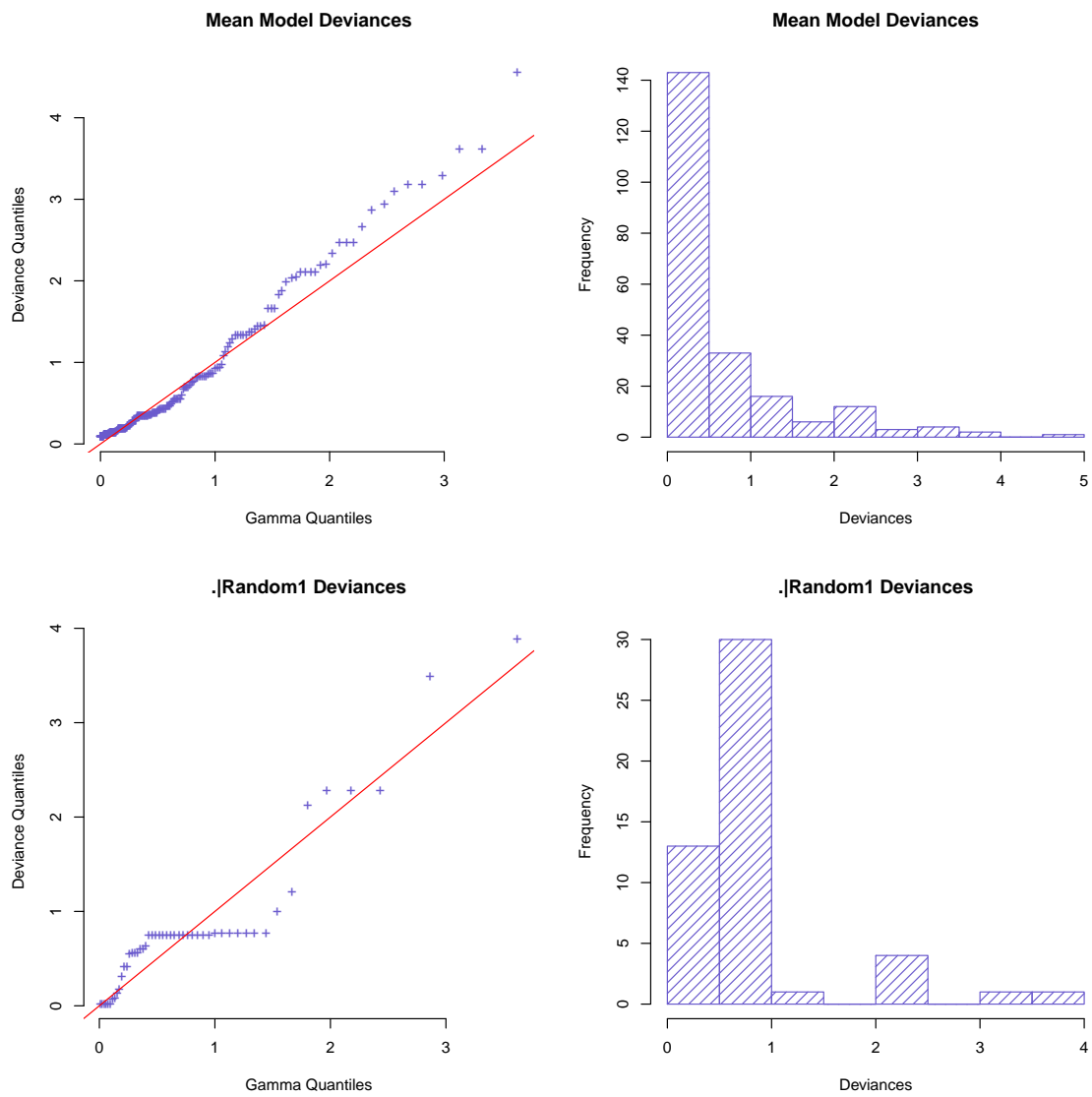


Figure 3: Deviance diagnostics for each observation and level in the random effect.

$$u \sim MVN(0, \mathbf{I}\sigma_u^2)$$

where β and u are the fixed and random effects in the mean part of the model, 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 indicate the i :th row. Here, a log link function is used for the dispersion term (i.e. 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 dispersion term ϕ is given by the residual variance here and $\log(\phi_i) = X_{d,i}\beta_d$.

This model is not possible to fit in the `dglm` package, for instance, because we have random effects in the mean part of the model and it is also an improvement compared to the `glmer()` function since we allow a model for the residual variance.

We simulate data where there are five clusters with 20 observations in each cluster. For the mean part of the model, The simulated intercept value is $\mu = 0$ and the variance for the random effect is $\sigma_u^2 = 0.2$. Given the explanatory variable x_d , the simulated residual variance is 1.0 for $x_d = 0$ and 2.72 for $x_d = 1$. In this example, and the following ones, we show how the input code can be given in terms of the model matrices y , X , Z and $X.d$ instead of using R formula.

```
n.cluster <- 5
n.per.cluster <- 20
sigma2_u <- 0.2
sigma2_e <- 1
beta.disp <- 1
mu <- 0
n <- n.cluster * n.per.cluster
set.seed(1234)
X <- matrix(1, n, 1)
Z <- diag(n.cluster) %x% rep(1, n.per.cluster)
a <- rnorm(5, 0, sqrt(sigma2_u))
X_d <- matrix(1, n, 2)
X_d[, 2] <- rbinom(n, 1, 0.5)
e <- rnorm(n, 0, sqrt(sigma2_e * exp(beta.disp * X_d[, 2])))
y <- mu + Z %*% a + e
simul1 <- hglm(y = y, X = X, Z = Z, X.d = X_d)
summary(simul1)

## Call:
## hglm.default(X = X, y = y, Z = Z, X.d = X_d)
##
## -----
## MEAN MODEL
## -----
##
## Summary of the fixed effects estimates:
##
##      Estimate Std. Error t-value Pr(>|t|)
## X.1 -0.004186  0.267928  -0.016   0.988
## Note: P-values are based on 96 degrees of freedom
##
## Summary of the random effects estimates:
##
##      Estimate Std. Error
## Z.1  0.0454    0.3163
## Z.2  0.0284    0.3183
## Z.3  0.4311    0.3173
## Z.4 -0.8330    0.3163
## Z.5  0.3282    0.3129
```

```
##
## -----
## DISPERSION MODEL
## -----
##
## NOTE: h-likelihood estimates through EQL can be biased.
##
## Model estimates for the dispersion term:
##
## Link = log
##
## Effects:
##   Estimate Std. Error
## 1    0.0247    0.1859
## 2    0.5048    0.2958
##
## Dispersion = 1 is used in Gamma model on deviances to calculate the standard error(s).
##
## Dispersion parameter for the random effects:
## [1] 0.298
##
## Dispersion model for the random effects:
##
## Link = log
##
## Effects:
## .|Random1
##   Estimate Std. Error
##    -1.2107    0.7758
##
## Dispersion = 1 is used in Gamma model on deviances to calculate the standard error(s).
##
## EQL estimation converged in 5 iterations.
```

The output shows that the variance of the random effect is 0.298, and that $\hat{\beta}_d = (0.0247, 0.5048)$.

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:

$$E(y_i|\beta, u) = \exp(X_i\beta + Z_iv)$$

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: $E(u_j) = 1$, $\text{var}(u_j) = \lambda$.

This model is also possible to fit with the **hglm** package and extends other GLMM functions (e.g. `glmer()`) to allow for non-normal distributions for the random effect.

We simulate a Poisson model with random effects and test if there are differences in the dispersion term for an explanatory variable x_d . This example uses the data from the previous example. Hence, the simulated parameters and matrices `mu`, `a` and `Z` are the same. (Continued from the previous example)

```
eta <- exp(mu + Z %*% a)
y <- rpois(length(eta), eta)
```

```

simul.pois <- hglm(y = y, X = X, Z = Z, X.disp = X_d, family = poisson(link = log))
summary(simul.pois)

## Call:
## hglm.default(X = X, y = y, Z = Z, family = poisson(link = log),
##             X.disp = X_d)
##
## -----
## MEAN MODEL
## -----
##
## Summary of the fixed effects estimates:
##
##      Estimate Std. Error t-value Pr(>|t|)
## X.1 -0.07242    0.34406   -0.21   0.834
## Note: P-values are based on 95 degrees of freedom
##
## Summary of the random effects estimates:
##
##      Estimate Std. Error
## Z.1  -0.7040    0.4189
## Z.2   0.3625    0.3748
## Z.3   0.8082    0.3640
## Z.4  -0.7171    0.4196
## Z.5   0.2504    0.3752
##
## -----
## DISPERSION MODEL
## -----
##
## NOTE: h-likelihood estimates through EQL can be biased.
##
## Model estimates for the dispersion term:
##
## Link = log
##
## Effects:
##      Estimate Std. Error
## 1  -0.0367    0.1859
## 2   0.3427    0.2963
##
## Dispersion = 1 is used in Gamma model on deviances to calculate the standard error(s).
##
## Dispersion parameter for the random effects:
## [1] 0.5244
##
## Dispersion model for the random effects:
##
## Link = log
##
## Effects:
## .|Random1
##      Estimate Std. Error
##      -0.6454    0.7515
##
## Dispersion = 1 is used in Gamma model on deviances to calculate the standard error(s).
##
## EQL estimation converged in 5 iterations.

```

The estimated variance of the random effect is 0.5244. The output also gives the estimate and standard error (0.7515) of $\log(\sigma_u^2)$.

Poisson-Gamma model with an offset for the mean model

The pump failure data of [Gaver and O’Muircheartaigh \(1987\)](#) contains the number of failures y_i and the period of operation t_i recorded for each of 10 pumps, so that the empirical failure rate is y_i/t_i . [Gaver and O’Muircheartaigh \(1987\)](#) mentioned that there are two groups of the pumps, where 4 pumps were operated continuously, and the rest intermittently. We fit the conjugate Poisson-Gamma HGLM with the group effects as fixed effects, the effects of the pumps as random effects, and an offset of $\log t_i$.

```
data(pump)
offset.model <- hglm(fixed = S ~ factor(Gr), random = ~1 | System, offset = log(t),
  fix.disp = 1, family = poisson(), rand.family = Gamma(link = log), data = pump)
print(offset.model)

## Call:
## hglm.formula(family = poisson(), rand.family = Gamma(link = log),
##   fixed = S ~ factor(Gr), random = ~1 | System, data = pump,
##   fix.disp = 1, offset = log(t))
##
## -----
## Estimates of the mean model
## -----
##
## Fixed effects:
## (Intercept) factor(Gr)1
##      0.07479      -1.66527
##
## Random effects:
## as.factor(System)1 as.factor(System)2 as.factor(System)3
##      0.2951      0.1092      0.4324
## ...
## as.factor(System)9 as.factor(System)10
##      1.542      1.874
## NOTE: to show all the random effects estimates, use print(hglm.object, print.ranef = TRUE).
##
## Dispersion parameter for the mean model: 1
##
## Dispersion parameter for the random effects: 1.047
##
## Estimation converged in 4 iterations
```

On a log scale, the dispersion estimate of the random effects is $\log(1.0467) = 0.5523$.

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 modelled 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

$$y_i | \beta, u \sim N(X_i \beta + Z_i u, \sigma_e^2)$$

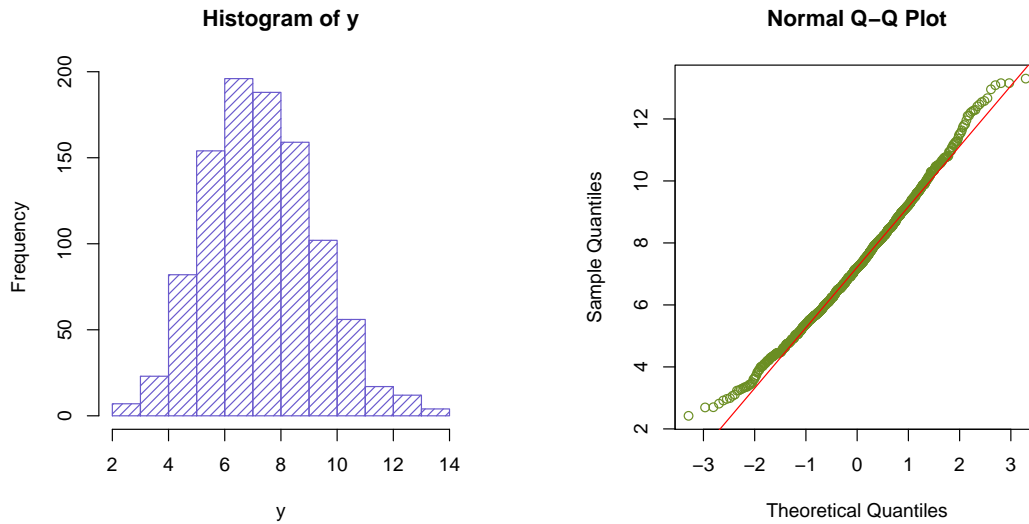


Figure 4: Histogram and qqplot for the analyzed trait in the QTLMAS data.

$$u \sim \text{MVN}(0, \mathbf{A}\sigma_u^2)$$

The model may be reformulated as (e.g. Lee et al., 2006; Rönnegård and Carlborg, 2007)

$$y_i | \beta, u \sim N(X_i \beta + Z_i^* u^*, \sigma_e^2)$$

$$u^* \sim \text{MVN}(0, \mathbf{I}\sigma_u^2)$$

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

The model can be fitted with the `hglm()` function since the input matrix \mathbf{Z} is user-specified. Now we analyze the data set QTLMAS included in the `hglm` package as an example. The data consists of 2025 individuals from two generations where 1000 individuals have observed trait values y that are approximately normal (Figure 4). The data we analyze was simulated for the QTLMAS 2009 Workshop (Coster et al., 2009)¹.

```
data(QTLMAS)
y <- QTLMAS[, 1]
pdf("fig_qtlmas_y.pdf", height = 5, width = 10)
par(mfrow = c(1, 2), pty = "s")
hist(y, density = 18, col = "slateblue")
qqnorm(y, col = "olivedrab")
qqline(y, col = 2)
dev.off()

## pdf
## 2
```

A longitudinal growth trait was simulated but for simplicity we analyze only the values given on the third occasion at age 265 days. 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 relationship matrix \mathbf{A} (which is obtained from the available pedigree information). A design matrix \mathbf{Z}_0 was constructed giving relating observation 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 \mathbf{L} be the Cholesky factorization of \mathbf{A} , and $\mathbf{Z} = \mathbf{Z}_0\mathbf{L}$. The design matrix for the fixed effects, \mathbf{X} , is a column of ones.

¹<http://www.qtlmas2009.wur.nl/UK/Dataset>

```

rm(list = ls())
data(QTLMAS)
y <- QTLMAS[, 1]
Z <- as.matrix(QTLMAS[, 2:2026])
X <- matrix(1, 1000, 1)
system.time(test1 <- hglm(y = y, X = X, Z = Z))

## NOTE: You are fitting a model with one Gaussian random effect term,
## and the number of effects (p) is greater than the number of
## observations (n). Consider turning on the argument 'bigRR' that may
## speed up a lot if p >> n.
## user system elapsed
## 9.701 0.246 10.149

print(test1)

## Call:
## hglm.default(X = X, y = y, Z = Z)
##
## -----
## Estimates of the mean model
## -----
##
## Fixed effects:
## X.1
## 7.28
##
## Random effects:
## Z1 Z2 Z3
## -1.192 1.649 1.319
## ...
## Z2024 Z2025
## 0.4102 -0.2024
## NOTE: to show all the random effects estimates, use print(hglm.object, print.ranef = TRUE).
##
## Dispersion parameter for the mean model: 2.211
##
## Dispersion parameter for the random effects: 1.503
##
## Estimation converged in 2 iterations

```

The estimated variance components are $\hat{\sigma}_e^2 = 2.2112$ and $\hat{\sigma}_u^2 = 1.5025$.

Random regression with a Gamma distributed random effect

The observed trait values y are the same as the previous example. Here we model a Gamma distributed random marker effect on chromosome 1. The incidence matrix \mathbf{Z}_m is 1000×90 since there are 1000 observed phenotypes and 90 genetic markers on chromosome 1 in this data set. An element $\mathbf{Z}_m[i, j]$ is given by the number of copies (0, 1 or 2) for marker j in individual i . The design matrix for the fixed effects, \mathbf{X} , is a column of ones. The marker with the highest estimated random effect (Figure 5) was close to the main genetic effect simulated on chromosome 1. (Continued from the previous example)

```

Z.markers <- as.matrix(QTLMAS[, 2027:2116])
test2 <- hglm(y = y, X = X, Z = Z.markers, rand.family = Gamma(link = log))

```

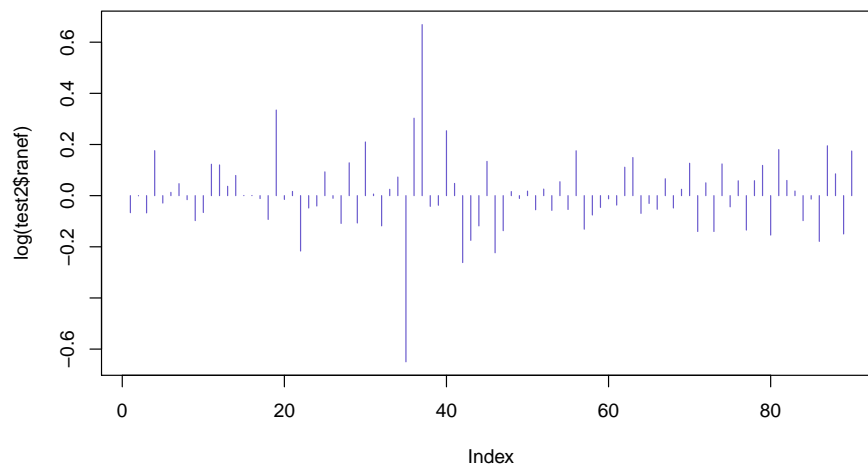


Figure 5: Estimated random effects for the 90 markers on chromosome 1 for trait values at 265 days. The main genetic effect was simulated close to marker number 40.

```
pdf("fig_qtlmas_effects.pdf", width = 8, height = 5)
plot(log(test2$ranef), type = "h", col = "slateblue")
dev.off()
```

```
## pdf
## 2
```

Binomial model with a Beta distributed random effect

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 the convergence criteria in `hglm()` to 0.005 produces the same estimates for the fixed effects as the ones obtained by Lee and Nelder (with differences $< 10^{-2}$). The beta distribution parameter α in Lee and Nelder (1996) was defined as $1/(2a)$ where a is the dispersion term obtained from `hglm()`.

```
data(seeds)
germ <- hglm(fixed = r/n ~ extract * I(seed == "073"), weights = n, data = seeds,
  random = ~1 | plate, family = binomial(), rand.family = Beta(), fix.disp = 1)
summary(germ)
```

```
## Call:
## hglm.formula(family = binomial(), rand.family = Beta(), fixed = r/n ~
##   extract * I(seed == "073"), random = ~1 | plate, data = seeds,
##   weights = n, fix.disp = 1)
##
## -----
## MEAN MODEL
## -----
```

```
##
## Summary of the fixed effects estimates:
##
##               Estimate Std. Error t-value Pr(>|t|)
## (Intercept)      -0.54240    0.19108  -2.839 0.017591
## extractCucumber    1.33916    0.27085   4.944 0.000583
## I(seed == "073")TRUE    0.07651    0.30897   0.248 0.809442
## extractCucumber:I(seed == "073")TRUE -0.82567    0.43077  -1.917 0.084261
##
## (Intercept)          *
## extractCucumber      ***
## I(seed == "073")TRUE
## extractCucumber:I(seed == "073")TRUE .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Note: P-values are based on 10 degrees of freedom
##
## Summary of the random effects estimates:
##
##               Estimate Std. Error
## as.factor(plate)1    0.4430    0.2482
## as.factor(plate)2    0.5021    0.2304
## as.factor(plate)3    0.4405    0.2259
## ...
## NOTE: to show all the random effects, use print(summary(hglm.object), print.ranef = TRUE).
##
## -----
## DISPERSION MODEL
## -----
##
## NOTE: h-likelihood estimates through EQL can be biased.
##
## Dispersion parameter for the mean model:
## [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] 0.02442
##
## Dispersion model for the random effects:
##
## Link = log
##
## Effects:
## .|Random1
##   Estimate Std. Error
##   -3.7124    0.5348
##
## Dispersion = 1 is used in Gamma model on deviances to calculate the standard error(s).
##
```

```
## EQL estimation converged in 10 iterations.
```

The output from the R code gives $\hat{a} = 0.0244$ and the corresponding estimate given in [Lee and Nelder \(1996\)](#) is $\hat{a} = 1/(2\hat{a}) = 0.023$.

Gamma HGLM with a structured dispersion

In this example, we analyze the semiconductor data taken from [Myers et al. \(2002\)](#), which involves a designed experiment in a semiconductor plant. Six factors, lamination temperature, lamination time, lamination pressure, firing temperature, firing cycle time and firing dew point, are employed, and we are interested in the curvature of the substrate devices produced in the plant. The curvature measurement is made four times on each device produced. Each design variable is taken at two levels. The measurement is known to be non-normally distributed, and the measurements taken on the same device are correlated. [Myers et al. \(2002\)](#) considered a gamma response model with a log link and used a GEE method assuming an AR(1) working correlation.

We consider a gamma HGLM by adding a random effect for the device in the mean model

$$\log \mu = \beta_0 + x_1\beta_1 + x_3\beta_3 + x_5\beta_5 + x_6\beta_6.$$

And the dispersion model is

$$\log \phi = \gamma_0 + x_2\gamma_2 + x_3\gamma_3$$

The variance λ of random effects represents the between-group variance, while ϕ represents the within-group variance. Results are shown as follows, and residual plots for the mean and dispersion models are in Figures (6, 7, 8, 9). The estimates are the same as those obtained using EQL in GenStat.

```
data(semiconductor)
gamma.model <- hglm(fixed = y ~ x1 + x3 + x5 + x6, random = ~1 | Device, family = Gamma(link = log),
  disp = ~x2 + x3, data = semiconductor)
summary(gamma.model)

## Call:
## hglm.formula(family = Gamma(link = log), fixed = y ~ x1 + x3 +
##   x5 + x6, random = ~1 | Device, disp = ~x2 + x3, data = semiconductor)
##
## -----
## MEAN MODEL
## -----
##
## Summary of the fixed effects estimates:
##
##           Estimate Std. Error t-value Pr(>|t|)
## (Intercept) -4.71168    0.06696  -70.368  < 2e-16 ***
## x1           0.20979    0.06638   3.160  0.00263 **
## x3           0.32893    0.06696   4.913  9.34e-06 ***
## x5          -0.17314    0.06638  -2.608  0.01185 *
## x6          -0.35690    0.06633  -5.380  1.80e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Note: P-values are based on 52 degrees of freedom
##
## Summary of the random effects estimates:
##
##           Estimate Std. Error
```

```
## as.factor(Device)1    0.2724    0.1787
## as.factor(Device)2    0.0097    0.1787
## as.factor(Device)3   -0.2697    0.1584
## ...
## NOTE: to show all the random effects, use print(summary(hglm.object), print.ranef = TRUE).
##
## -----
## DISPERSION MODEL
## -----
##
## NOTE: h-likelihood estimates through EQL can be biased.
##
## Model estimates for the dispersion term:
##
## Link = log
##
## Effects:
##      Estimate Std. Error
## (Intercept)  -2.5887    0.1972
## x2            -0.6861    0.1971
## x3            -0.5024    0.1971
##
## Dispersion = 1 is used in Gamma model on deviances to calculate the standard error(s).
##
## Dispersion parameter for the random effects:
## [1] 0.0486
##
## Dispersion model for the random effects:
##
## Link = log
##
## Effects:
## .|Random1
##      Estimate Std. Error
##      -3.0242    0.5172
##
## Dispersion = 1 is used in Gamma model on deviances to calculate the standard error(s).
##
## EQL estimation converged in 4 iterations.
```

```
plot(gamma.model, cex = 0.6, pch = 1, cex.axis = 1/0.6, cex.lab = 1/0.6, cex.main = 1/0.6,
     mar = c(3, 4.5, 0, 1.5), device = "pdf", name = "fig_semi")
```

More than one random effect in the hglm2 function

From version 1.2 of the [hglm](#) package, it is possible to fit models having several random effects. The model is specified using the [lme4](#) formula conversion in the `hglm2` function, or using the matrix-type input X, y, Z in the `hglm` function as described earlier. Here the `hglm2` function is illustrated using the classical data set from a Salamander mating experiment [McCullagh and Nelder \(1989\)](#). The binary response variable (Mate) is mating success or not, and two variance components σ_f^2 and σ_m^2 are estimated for females and males, respectively.

```
data(salamander)
hglm.salam <- hglm2(meanmodel = Mate ~ TypeF + TypeM + TypeF * TypeM + (1 |
```

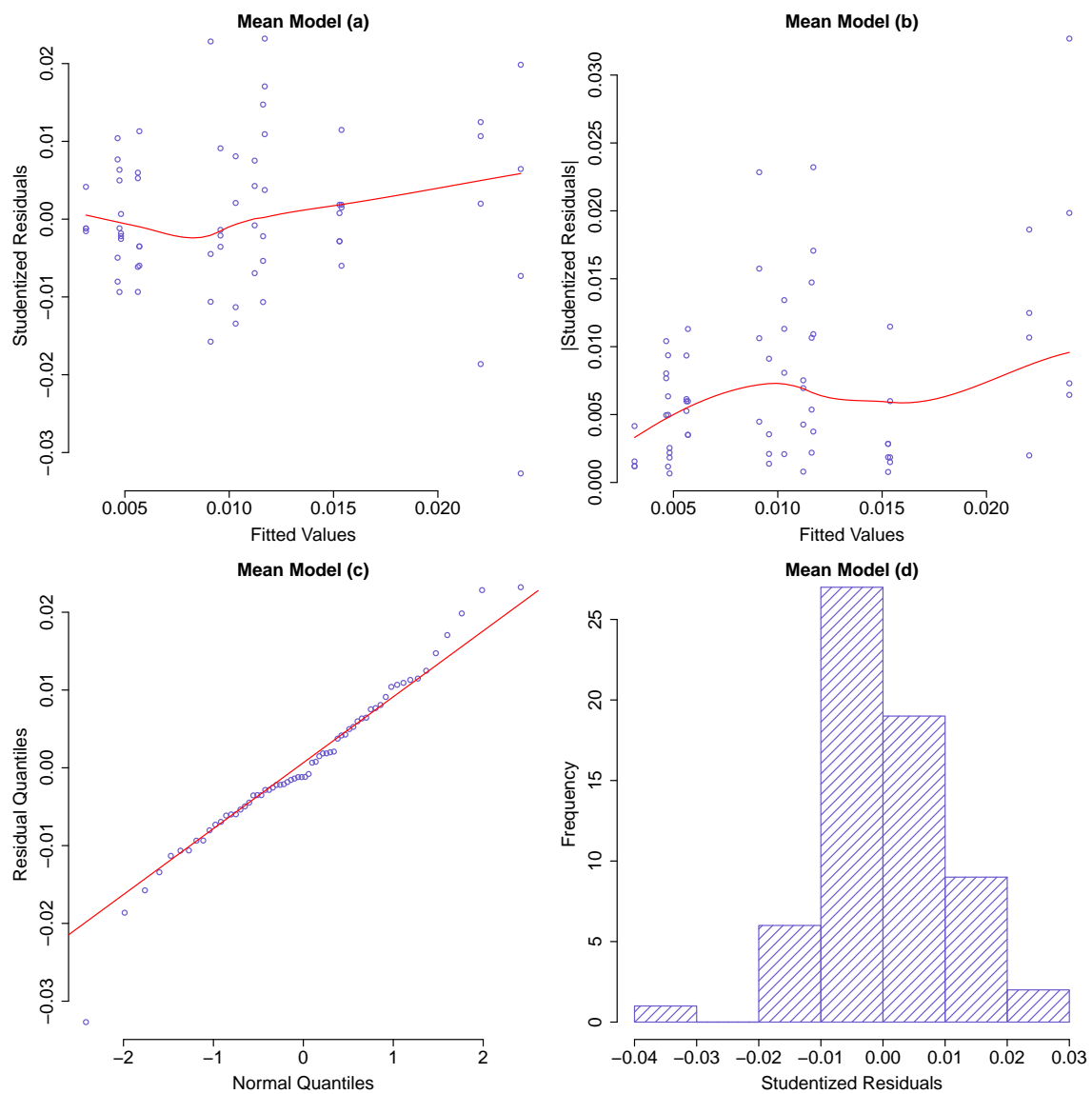


Figure 6: Residual plots of the mean model for the semiconductor data.

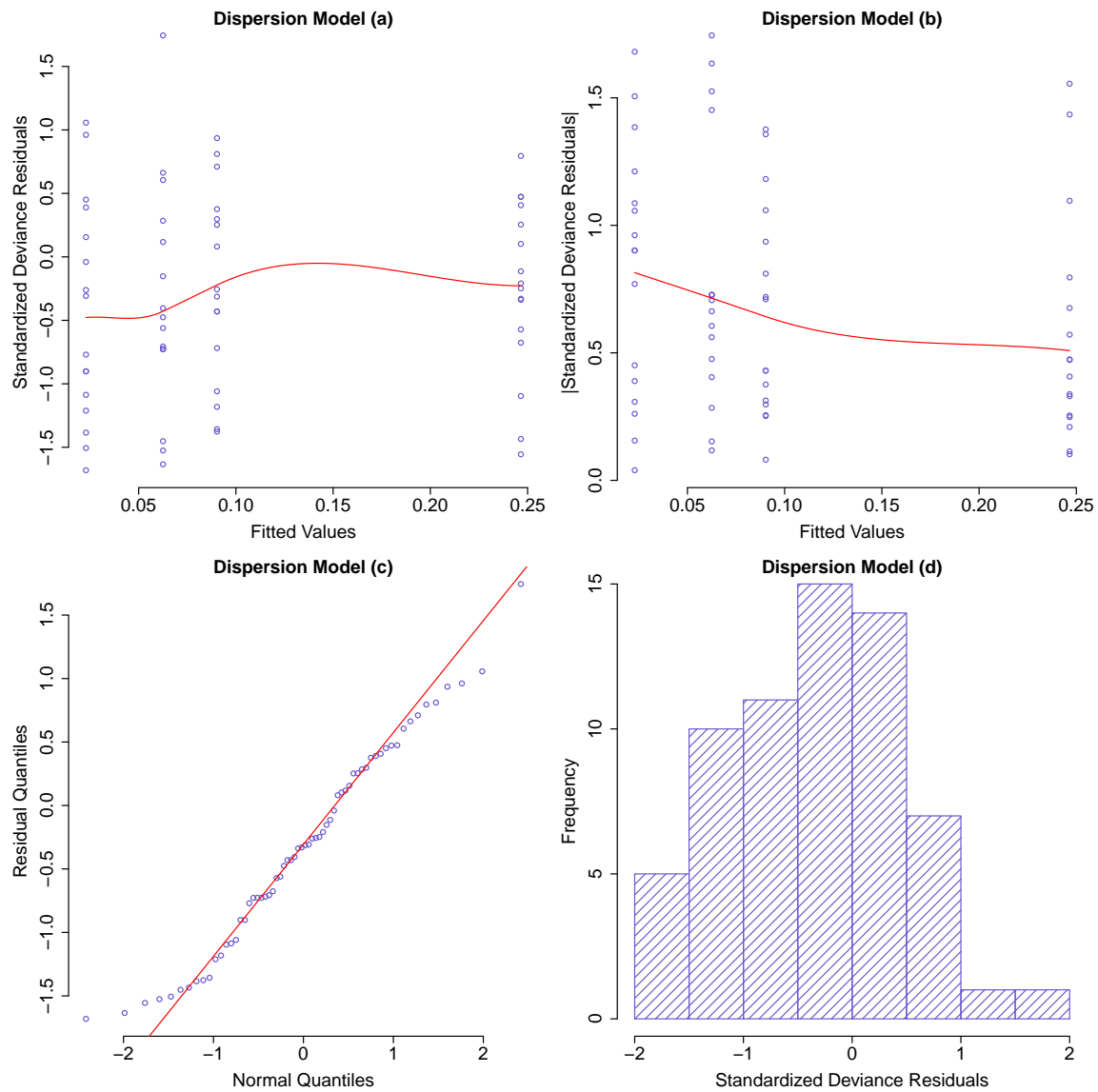


Figure 7: Residual plots of the dispersion model for the semiconductor data.

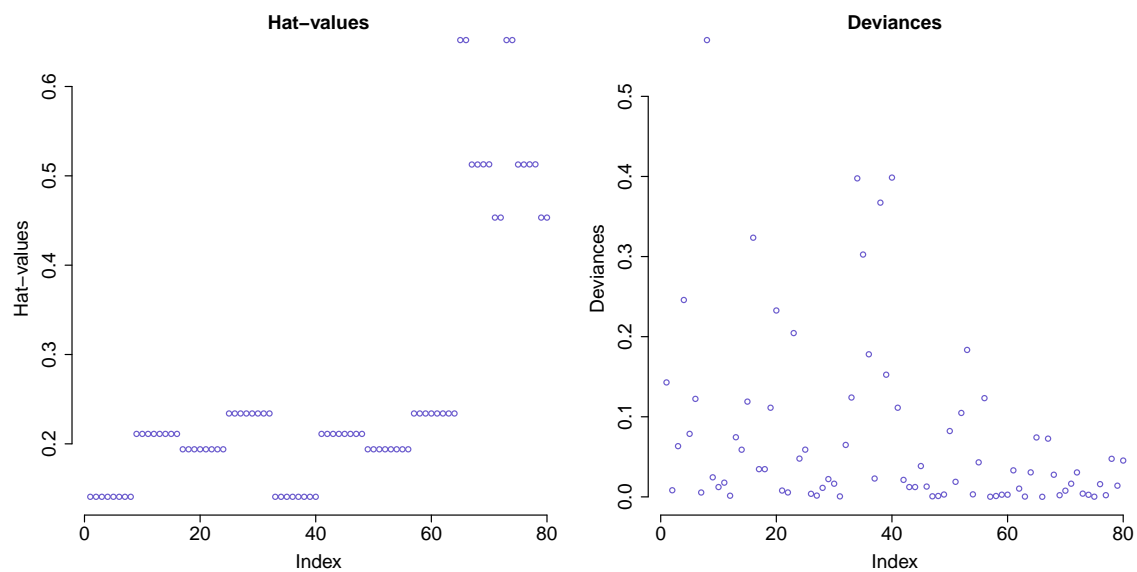


Figure 8: Diagnostics plots of the hat-values and deviances for the semiconductor data.

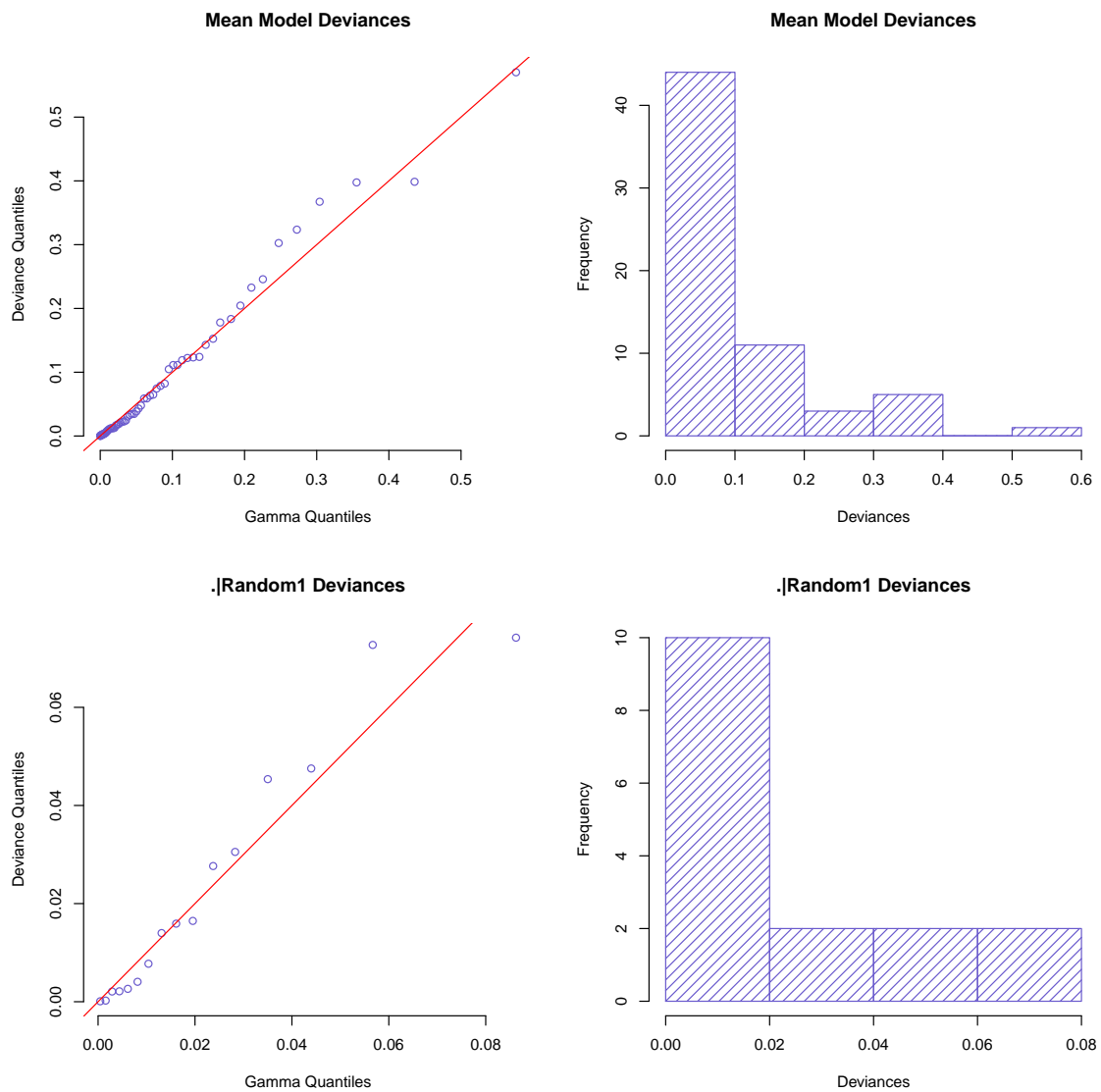


Figure 9: Deviance diagnostics for each observation and level in the random effects (the semiconductor data).

```

    Female) + (1 | Male), family = binomial(), data = salamander, conv = 1e-08,
    maxit = 40)
summary(hglm.salam)

## Call:
## hglm2.formula(meanmodel = Mate ~ TypeF + TypeM + TypeF * TypeM +
##   (1 | Female) + (1 | Male), data = salamander, family = binomial(),
##   conv = 1e-08, maxit = 40)
##
## -----
## MEAN MODEL
## -----
##
## Summary of the fixed effects estimates:
##
##              Estimate Std. Error t-value Pr(>|t|)
## (Intercept)    0.7881    0.3210    2.455  0.0146 *
## TypeFW         -2.2903    0.4315   -5.308 2.12e-07 ***
## TypeMW         -0.5410    0.3890   -1.391  0.1654
## TypeFW:TypeMW   2.8208    0.4974    5.671 3.26e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Note: P-values are based on 308 degrees of freedom
##
## Summary of the random effects estimates:
##
##              Estimate Std. Error
## (Intercept)| Female:1    0.3982    0.6527
## (Intercept)| Female:2    0.7099    0.6902
## (Intercept)| Female:3    0.3160    0.6595
## ...
## NOTE: to show all the random effects, use print(summary(hglm.object), print.ranef = TRUE).
##
## Summary of the random effects estimates:
##
##              Estimate Std. Error
## (Intercept)| Male:1     0.7426    0.6451
## (Intercept)| Male:2     0.0169    0.6435
## (Intercept)| Male:3     0.7487    0.6439
## ...
## NOTE: to show all the random effects, use print(summary(hglm.object), print.ranef = TRUE).
##
## -----
## DISPERSION MODEL
## -----
##
## NOTE: h-likelihood estimates through EQL can be biased.
##
## Dispersion parameter for the mean model:
## [1] 0.9954
##
## Model estimates for the dispersion term:
##
## Link = log
##
## Effects:
##      Estimate Std. Error
##      -0.0046    0.0806

```

```
##
## Dispersion = 1 is used in Gamma model on deviances to calculate the standard error(s).
##
## Dispersion parameter for the random effects:
## [1] 0.7245 0.6402
##
## Dispersion model for the random effects:
##
## Link = log
##
## Effects:
## .|Random1
##   Estimate Std. Error
##   -0.3222    0.2833
##
## .|Random2
##   Estimate Std. Error
##   -0.4460    0.2941
##
## Dispersion = 1 is used in Gamma model on deviances to calculate the standard error(s).
##
## EQL estimation converged in 28 iterations.
```

The results show that $\hat{\sigma}_f^2 = 0.7245$ and $\hat{\sigma}_m^2 = 0.6402$ (the EQL method in GenStat gives $\hat{\sigma}_f^2 = 0.72$ and $\hat{\sigma}_m^2 = 0.62$).

EQL1 correction for a Poisson GLMM

The EQL1 method gives improved estimates compared to EQL1 for a Poisson GLMM when the number of levels in the random effect are large and i.i.d. The implementation follows the Appendix in [Lee and Lee \(2012\)](#). In the following example we show how the bias using EQL1, for the extreme case of having the number of levels equal to the number of observations, can be resolved by using EQL1.

```
# A simulation study to compare EQL and EQL1 estimates in Poisson GLMM where
# No. of levels in the random effect = No. of observations
n <- 100 # No. of observations
p <- 100 # No. of levels in the random effect
Z <- diag(p)
sigma2u <- 1 # Variance of the random effects
mu <- 0 # Simulated intercept term
n.rep <- 30 # Number of simulation replicates
set.seed(123)
results.EQL <- results.EQL1 <- matrix(NA, n.rep, 3)
colnames(results.EQL) <- colnames(results.EQL1) <- c("Intercept", "Variance component",
  "Converged")
# This takes a while to run!
t0 <- proc.time()[3]
for (i.rep in 1:n.rep) {
  u <- rnorm(p, 0, sqrt(sigma2u))
  eta <- mu + Z %*% u
  y <- rpois(n, exp(eta))
  hglm <- hglm(y = y, X = matrix(1, n, 1), Z = Z, family = poisson(link = log),
    fix.disp = 1, maxit = 100)
  results.EQL[i.rep, 1] = hglm$fixef
  results.EQL[i.rep, 2] = hglm$varRanef
```

```

results.EQL[i.rep, 3] = as.numeric(hglm$Converge == "converged")
hglm.EQL1 <- hglm(y = y, X = matrix(1, n, 1), Z = Z, family = poisson(link = log),
  fix.disp = 1, maxit = 100, method = "EQL1")
results.EQL1[i.rep, 1] = hglm.EQL1$fixef
results.EQL1[i.rep, 2] = hglm.EQL1$varRanef
results.EQL1[i.rep, 3] = hglm.EQL1$Converge == "converged"
if (i.rep%10 == 0)
  cat(i.rep, " ")
}

## 10  20  30

proc.time()[3] - t0

## elapsed
##   18.94

summary(results.EQL)

##      Intercept      Variance component      Converged
## Min.      :-0.151    Min.      :0.361      Min.      :1
## 1st Qu.: 0.141    1st Qu.:0.594      1st Qu.:1
## Median : 0.219    Median :0.749      Median :1
## Mean      : 0.212    Mean      :0.741      Mean      :1
## 3rd Qu.: 0.285    3rd Qu.:0.898      3rd Qu.:1
## Max.      : 0.462    Max.      :1.131      Max.      :1

summary(results.EQL1)

##      Intercept      Variance component      Converged
## Min.      :-0.5545   Min.      :0.508      Min.      :0.0
## 1st Qu.: -0.1721   1st Qu.:0.830      1st Qu.:1.0
## Median : -0.0540   Median :1.011      Median :1.0
## Mean      : -0.0687   Mean      :1.023      Mean      :0.9
## 3rd Qu.: 0.0371    3rd Qu.:1.221      3rd Qu.:1.0
## Max.      : 0.2586   Max.      :1.614      Max.      :1.0

```

Note that both the intercept term and the variance component are highly biased for the EQL method (and that one replicate did not converge). The EQL1 method corrects estimate of the intercept term (from a median of 0.2188 down to -0.054 , with a true simulated value of 0) and as consequence the estimate of the variance component is also improved (from a median of 0.7489 to 1.0109, with a true simulated value of 1).

Fitting a spatial Markov Random Field model using the CAR family

In spatial data analysis, Generalized Linear Mixed Models (GLMM) with spatially correlated random effects are commonly used (Cressie, 1993). A spatial GLMM with random effects being distributed as the Gaussian Markov Random Field (GMRF) is presented as follows

$$\begin{aligned}
 E[z_s | u_s] &= \mu_s \\
 g(\mu_s) = \eta_s &= \mathbf{X}_s^T \boldsymbol{\beta} + \mathbf{Z}_s u_s \\
 s &= 1, 2, \dots, n
 \end{aligned} \tag{17}$$

$$z_s|u_s \sim \text{Exponential Family} \quad (18)$$

with $z_s|u_s \perp z_t|u_t, \forall s \neq t$ and

$$\mathbf{u} = (u_1, u_2, \dots, u_n)^T \sim N(\mathbf{0}, \Sigma = \tau(\mathbf{I} - \rho\mathbf{D})^{-1}) \quad (19)$$

where s represents a location identified by the coordinates $(x(s), y(s))$, $\boldsymbol{\beta}$ is a vector of fixed effects, \mathbf{X}_s is the vector observed covariates at location s and u_s being the location specific random effects. The D matrix in Equation 19 is generally a function of the location coordinates (or a neighbourhood matrix for areas) and are known (see e.g. Clayton and Kaldor, 1987). The two parameters τ and ρ are estimated.

A well-known data set is the Scottish Lip Cancer data (Clayton and Kaldor, 1987), which is available in the **hglm** package. An example is given as follows.

```
data(cancer)
logE <- log(E)
X11 <- model.matrix(~Paff)
m41 <- hglm(X = X11, y = 0, Z = diag(length(0)), family = poisson(), rand.family = CAR(D = nbr),
  offset = logE, conv = 1e-09, maxit = 200, fix.disp = 1, method = "EQL1")
summary(m41)

## Call:
## hglm.default(X = X11, y = 0, Z = diag(length(0)), family = poisson(),
##   rand.family = CAR(D = nbr), method = "EQL1", conv = 1e-09,
##   maxit = 200, fix.disp = 1, offset = logE)
##
## -----
## MEAN MODEL
## -----
##
## Summary of the fixed effects estimates:
##
##           Estimate Std. Error t-value Pr(>|t|)
## (Intercept)  0.08118    0.21145   0.384  0.70441
## Paff         0.04094    0.01247   3.283  0.00314 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Note: P-values are based on 24 degrees of freedom
##
## Summary of the random effects estimates:
##
##           Estimate Std. Error
## [1,]    0.7779    1.0587
## [2,]    0.6808    0.3853
## [3,]    0.5281    0.5302
## ...
## NOTE: to show all the random effects, use print(summary(hglm.object), print.ranef = TRUE).
##
## -----
## DISPERSION MODEL
## -----
##
## NOTE: h-likelihood estimates through EQL can be biased.
##
## Dispersion parameter for the mean model:
## [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] 391.7
##
## Dispersion model for the random effects:
##
## Link = log
##
## Effects:
## .|Random1
##
##           Estimate Std. Error
## 1/CAR.tau      5.971      1.5718
## -CAR.rho/CAR.tau -1.034      0.2763
## CAR.tau (estimated spatial variance component): 0.1675
## CAR.rho (estimated spatial correlation): 0.1733
##
## Dispersion = 1 is used in Gamma model on deviances to calculate the standard error(s).
##
## EQL1 estimation converged in 12 iterations.
```

hglm provides efficient estimation procedure of the GMRF models, which enables fitting moderately sized CAR model on an ordinary computer (see also the comparison with **spaMM** in section 2.18).

It should be noted that there are concerns or drawbacks of CAR/SAR models, listed as follows, which require further investigation in spatial statistics. However, these would not preclude the use of such models in practice.

1. How to construct **D** matrix? Should it be a neighborhood matrix or some other type of weight matrix (LeSage and Pace, 2009; Wall, 2004; Ord, 1975). The choices are often made subjectively.
2. What to do with the locations on the edges so that the global dependence structure is uniquely represented by the model defined for the sample locations (Cressie, 1993)? So far there is no clear answer to this question.
3. Should one let $(\mathbf{I} - \rho\mathbf{D})$ to be non-singular in CAR models? If one wants it to be non-singular then restrictions on ρ are needed to make the covariance matrix positive definite. This in turn might put a certain restriction on the correlation structure (unwanted) between locations (Wall, 2004).
4. How to compare ρ and τ parameters between CAR and SAR models (Cressie, 1993; Wall, 2004)? Again, there is no straight answer to this question, either.

Though CAR and SAR structures have their own drawbacks (see Cressie, 1993; Wall, 2004), they are very frequently used and widely suggested in standard text books on spatial data analysis (see e.g. Cressie, 1993; LeSage and Pace, 2009; Hodges, 2013). For CAR models, $\rho d_{i,j}$ represents partial correlation, i.e. $\text{Cor}(u_i, u_j | \mathbf{u}_{-(i,j)}) = \rho d_{i,j}$ where $\mathbf{u}_{-(i,j)}$ represents a vector all the u 's but the i 'th and j 'th.

Fitting several random effects from different families

Suppose we wish to fit a spatial random effect and an i.i.d. Gaussian random effect. The following example shows how this is possible (from version 2.0).

```
set.seed(123)
V <- matrix(rnorm(20000), 100, 200)
Sigma <- tcrossprod(V)
rho <- 0.5
tau <- 1.5
D <- (diag(100) - solve(Sigma/tau))/rho
require(mvtnorm)
u1 <- as.numeric(rmvnorm(1, sigma = Sigma))
Z1 <- diag(100)
z2 <- factor(rep(LETTERS[1:20], rep(5, 20)))
Z2 <- model.matrix(~0 + z2)
u2 <- rnorm(20, 0, sqrt(2))
x1 <- rnorm(100)
x2 <- rnorm(100)
y <- 1 + 2 * x1 + 3 * x2 + Z1 %*% u1 + Z2 %*% u2 + rnorm(100, 0, 1)
carnorm <- hglm(X = cbind(rep(1, 100), x1, x2), y = y, Z = cbind(Z1, Z2), rand.family = list(CAR(D = D),
  gaussian()), RandC = c(100, 20))
print(carnorm)

## Call:
## hglm.default(X = cbind(rep(1, 100), x1, x2), y = y, Z = cbind(Z1,
##      Z2), rand.family = list(CAR(D = D), gaussian()), RandC = c(100,
##      20))
##
## -----
## Estimates of the mean model
## -----
##
## Fixed effects:
##           x1      x2
## 0.751 2.292 4.361
##
## Random effects:
##
## -4.880 -1.546 -29.415
## ...
##
## 33.32 -15.67
## NOTE: to show all the random effects estimates, use print(hglm.object, print.ranef = TRUE).
##
## Random effects:
##      z2A      z2B      z2C
## -2.095  0.837  1.776
## ...
##      z2S      z2T
## 0.002018 0.455228
## NOTE: to show all the random effects estimates, use print(hglm.object, print.ranef = TRUE).
##
## Dispersion parameter for the mean model: 8.462
##
## Dispersion parameter for the random effects: 2.674 1.222
##
## Estimation converged in 45 iterations
```

Likelihood-ratio test for dispersion parameters

From version 2.0, **hglm** package provides the `lrt` function for performing likelihood-ratio test (LRT) for the estimated random effects variance components or dispersion parameters. The test statistic follows [Self and Liang \(1987\)](#)'s mixture χ^2 distribution since the tested parameters are bounded above zero. The function is able to test a single random effects model against the null model without random effects, or compare two estimated HGLMs to test the parameters that they differ. See the examples as follows.

```
set.seed(911)
x1 <- rnorm(100)
x2 <- rnorm(100)
x3 <- rnorm(100)
z1 <- factor(rep(LETTERS[1:10], rep(10, 10)))
z2 <- factor(rep(letters[1:5], rep(20, 5)))
Z1 <- model.matrix(~0 + z1)
Z2 <- model.matrix(~0 + z2)
u1 <- rnorm(10, 0, sqrt(2))
u2 <- rnorm(5, 0, sqrt(3))
y <- 1 + 2 * x1 + 3 * x2 + Z1 %*% u1 + Z2 %*% u2 + rnorm(100, 0, sqrt(exp(x3)))
dd <- data.frame(x1 = x1, x2 = x2, x3 = x3, z1 = z1, z2 = z2, y = y)
m20 <- hglm(X = cbind(rep(1, 100), x1, x2), y = y, Z = Z1, calc.like = TRUE)
lrt(m20)

##
## Likelihood-ratio test
##
## data: m20
## LRT statistic = 74.38, p-value < 2.2e-16

m21 <- hglm(X = cbind(rep(1, 100), x1, x2), y = y, Z = cbind(Z1, Z2), RandC = c(10,
5), calc.like = TRUE)
lrt(m20, m21)

##
## Likelihood-ratio test
##
## data: m20 v.s. m21
## LRT statistic = 0.8245, p-value = 0.1819
```

Summary

The hierarchical generalized linear model approach gives 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, fit models where the random effect distribution is not necessarily Gaussian and estimate 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](#)), 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.

Discussions on h-likelihood theory

For the users not previously acquainted with h-likelihood theory, we summarize here the discussions in [Lee and Nelder \(1996\)](#), [Lee et al. \(2007\)](#), [Lee and Nelder \(2006\)](#) and the collection of discussion papers in Statistical Science vol. 24 no. 3 ([Lee and Nelder, 2009a](#); [Meng, 2009](#); [Louis, 2009](#); [Molenberghs et al., 2009](#); [Lee and Nelder, 2009b](#)). We try to keep this summary objective to reflect the most important parts of the discussions. Thereafter, we give our view of which parts of the discussion that has not been settled yet and may have consequences on the [hglm](#) package.

Summary of discussions by Clayton, Kuha and Firth following [Lee and Nelder \(1996\)](#) and the Discussion section in [Lee et al. \(2007\)](#)

In the Discussions following [Lee and Nelder \(1996\)](#), it is claimed that: i) the h-likelihood is not a true likelihood, ii) if there is little information on each of the random effects we can expect the estimates to be biased, and iii) the h-likelihood could be interpreted in a Bayesian viewpoint as a posterior distribution with uniform priors so that we can only expect the estimates to be satisfactory if there is plenty of information on each of the random effects.

In [Lee et al. \(2007\)](#) the authors address these criticisms. Specifically they address the following three points:

- i) The h-likelihood is not a valid likelihood.
- ii) The h-likelihood is not invariant to non-linear transformation of the random effects.
- iii) There are problems in the analysis of binary matched pairs data.

The authors claim that the h-likelihood is a valid likelihood. Unlike the classical likelihood, however, which contain the two objects: data (observed random variables) and unknown fixed parameters, the h-likelihood also contains a third object, unobserved random variables. Based on [Bjornstad \(1996\)](#) extended likelihood principle we should expect a joint likelihood, including both fixed and random parameters, to carry all the information about these parameters. (For an excellent review on the extended likelihood principle, see [Pawitan \(2001\)](#)). In [Lee and Nelder \(2005\)](#) the authors explained why they consider the h-likelihood to be a joint likelihood appropriate for statistical inference of HGLMs.

Concerning the h-likelihood and invariance, here it is important to note that the h-likelihood is not a general joint likelihood. In the h-likelihood it is imposed that a proper scale for the random effects is chosen, and consequently the problem of invariance may be an issue for a general joint likelihood but not for the h-likelihood.

One might expect that the h-likelihood has similar problems as the Penalized Quasi-Likelihood (PQL) method ([Breslow and Clayton, 1993](#)) for analysis of binary matched pairs (i.e. the extreme case of having few observations for each of the random effects). The authors show however that the estimation method they call HL(2) gives estimates very close to the marginal likelihood estimates obtained by numerical integration. The method HL(2) is a higher order approximation to the h-likelihood. Hence, there does not seem to be a major problem with the h-likelihood itself but rather there is an issue of choosing a computational method that produces correct h-likelihood results.

This is not the only paper where binary outcomes with little information on each of the random effects have been investigated for the h-likelihood. [Noh et al. \(2006\)](#) showed that the bias in variance component estimates were very small, compared to PQL, for binary outcomes in related individuals of small families. To obtain good estimates for the h-likelihood an appropriate computational method was developed (similar to the HL(2) method in [Noh and Lee \(2007\)](#)).

Summary of Firth's discussion in [Lee and Nelder \(2006\)](#)

- i) The choice of the function for the random effects cannot be made on grounds of additivity, i.e. the transformation $v(u)$ that makes fixed and random effects additive. Two simple examples are given.
- ii) The h-likelihood may lead to inconsistent estimators and it is not reasonable for Lee and Nelder to claim that the h-likelihood gives a general method for generalized linear models with random effects.
- iii) The problem of bias should not be specific for binary outcomes but rather be a general problem for situations where there is little information on each of the random effects.

The authors reply that it should be noted that although the choice of link function for the random effects is difficult in the examples given by Firth, the adjusted profile h-likelihood still gives satisfactory estimates for the fixed parameters.

Discussion in papers of *Statistical Science* vol. 24 no. 3

[Lee and Nelder \(2009a\)](#) base their argumentation on the extended likelihood principle ([Bjornstad, 1996](#)) to explain why the h-likelihood should be used for inference of models with unobservable random effects. The idea of the h-likelihood is that the marginal likelihood should be used for estimating fixed effects, the h-likelihood should be maximized to estimate random effects and the adjusted profile likelihood for the estimation of dispersion parameters. When the marginal likelihood is difficult to compute, the adjusted profile h-likelihood (with profiling over the random effects) is proposed as an alternative.

They criticize the use of estimation methods that are not included in a probabilistic framework, such as GEE and empirical Bayes methods. It is emphasized that the h-likelihood inference can be made without the necessity of inventing priors, as opposed to Bayesian methods.

By using the example of [Bayarri et al. \(1988\)](#), they illustrate the importance of choosing the scale for the random effects so that the maximization of the h-likelihood gives the MLE for the fixed effects. In this example the outcome y follows an exponential distribution conditional on a random effect u , where u is also exponentially distributed. The point of this example is that useless estimators are derived if the joint likelihood is maximized. However, [Lee and Nelder \(2009a\)](#) claim that the h-likelihood is defined for the specific scale $v = \log u$, which is shown to produce sensible estimates of the fixed parameter and the random effects. They admit that “the choice of the scale in defining the h-likelihood is important to guarantee the meaningfulness of the mode estimation”. Furthermore: “The (weak) canonical scale in HGLMs leads to an invariance of a certain extended likelihood. However, in general the validity of such a scale has not been established.”

[Louis \(2009\)](#) considers the paper by Lee and Nelder as a piece of “over promotion” and “more of an opinion-piece than a scientific comparison of approaches”. He agrees that h-likelihood estimation can be valid and efficient under some settings but it is not globally valid.

[Molenberghs et al. \(2009\)](#) review different estimation methods for models with random effects and to them it is clear that no method can “claim uniform superiority over all others”. They also emphasize the importance of assessing a suite of models to judge, through sensitivity analysis, how key inferences are vulnerable to model choice, which was neglected in [Lee and Nelder \(2009a\)](#).

[Meng \(2009\)](#) derives the necessary constraints for the scale of the random effects to be used in the h-likelihood. The similarity (and differences) to Bayesian analysis with uniform priors is pointed out, and concludes that the h-likelihood is just an approximation to the Bayesian posterior for the special case of having uniform priors.

In response, [Lee and Nelder \(2009b\)](#) note that both Louis and Meng say that the extended likelihood such as the h-likelihood does carry information about the unobservables but that the Bayesian

approach is best suited for the inference. In contrast to the many possible priors in the Bayesian framework, the authors say that “in our system there is only one corresponding prior likelihood”, i.e. a uniform prior. Furthermore, the adjusted profile h-likelihood is not merely a Laplace approximation to the marginal posterior distribution in a Bayesian framework, because “it can also eliminate fixed unknowns”, and the authors argue that the adjusted profile h-likelihood is quite different from the Bayesian marginal posterior.

Summary of points not settled yet and possible consequences

The conclusion that can be drawn from the studies performed by Lee, Nelder and co-workers over the past 15 years is that we can expect the h-likelihood to give good estimates, but the computational procedure to get good approximations for the maximum h-likelihood estimates might be computationally demanding for data such as binary matched pairs. Uncertainty measures, such as standard errors, have been shown to be possible to derive from the curvature of the h-likelihood for many applications of HGLMs, but the general validity of using the h-likelihood for inference is still being argued by several authors.

In the **hglm** package, we have implemented the Extended Quasi-Likelihood (EQL) procedure, which is the one described in the book by Lee et al. (2006), where the estimates are obtained by fitting an interconnected set of GLMs in a very elegant way. The dispersion components are estimated by correcting for the leverages in the mean part of the model, which is a result of applying an adjusted profile likelihood function to the EQL.

The EQL method can give biased results, which we clearly state in the output of the package, but this is not a problem of the h-likelihood itself but rather that the EQL method may sometimes give estimates that are substantially different from the maximum h-likelihood estimates.

The most important point from the above discussions is that a proper scale for the random effects is required by the h-likelihood to avoid problems of invariance. For many of the commonly used HGLM this is not an issue, but the user of the **hglm** package should be aware of this potential problem.

Comparison with estimates using GenStat

In the **hglm** package, interconnected GLMs are used to fit HGLMs. The algorithm was implemented using the `glm` function in the R **stats** package and by following Table 7.1 and Table 7.3 in Lee et al. (2006). An important part of the implementation was to define a separate GLM function for HGLMs with gamma distributed random effects (as explained in Table 7.1 and Example 6.3 in Lee et al. (2006)), and also to compute the deviance components for beta distributed random effects according to Table 7.1 in Lee et al. (2006). For a linear mixed model, this algorithm gives REML estimates, and for general HGLMs it gives EQL estimates. The standard errors are the ones produced by the interconnected GLM functions at convergence.

The estimates produced by the `hglm` function are very similar to the ones using the EQL method implemented in GenStat. For a binomial-beta model using the seed germination data presented in Lee and Nelder (1996) with fixed binomial dispersion parameter, the estimates of fixed effects (Table 4) differed by less than 2×10^{-3} to the EQL estimates in GenStat, and the dispersion parameter differed by less than 1%.

For a Poisson-gamma model using the pump failure data presented in Lee and Nelder (1996), the estimates (Table 5) differed by less than 1×10^{-3} . By comparing to the HL(0,0) estimates, we can see that here the EQL method seems to give a poor approximation for the dispersion parameter.

For a gamma-normal model using the semiconductor data presented on page 218 of Lee et al. (2006), the estimates of fixed effects (Table 6) differed by less than 4×10^{-3} to the EQL estimates in GenStat, and the dispersion parameter differed by less than 1%.

Table 4: Comparison of estimates for a binomial-beta model using the seed germination data presented in Lee and Nelder (1996).

	GenStat: HL(0,0) ^a	GenStat: EQL	hglm EQL
Intercept	-0.543	-0.542	-0.542
Seed	0.080	0.077	0.075
Extract	1.337	1.339	1.339
Interaction	-0.822	-0.825	-0.826
Dispersion parameter for the random effects:			
log α	3.096	3.022	3.003

^a Same estimates as in Lee and Nelder (1996)**Table 5:** Comparison of estimates for a Poisson-gamma model using the pump failure data presented in Lee and Nelder (1996).

	GenStat: HL(0,0) ^a	GenStat: EQL	hglm EQL
Intercept	-1.599	-1.590	-1.590
Group(2)	1.668	1.665	1.665
Dispersion parameter for the random effects:			
log α	-0.165	0.046	0.046

^a Same estimates as in Lee and Nelder (1996)**Table 6:** Comparison of estimates for a gamma-normal model using the semiconductor data presented in Lee et al. (2006).

	GenStat: HL(0,0) ^a	GenStat: EQL	hglm EQL
Mean model			
Intercept	-4.711	-4.712	-4.711
x_1	0.209	0.210	0.209
x_3	0.328	0.329	0.329
x_5	-1.174	-0.173	-0.173
x_6	-0.357	-0.357	-0.357
Dispersion model			
Intercept	-2.610	-2.588	-2.591
x_2	-0.673	-0.686	-0.682
x_3	-0.492	-0.503	-0.499
Dispersion parameter for the random effects:			
log λ	-3.014	-3.028	-3.004

^a Same estimates as on page 218 of Lee et al. (2006)

Comparison with estimates using HGLMMM

We compare our EQL1 implementation to that in the CRAN-archived version of **HGLMMM** where the HL(1,1) estimator by [Noh and Lee \(2007\)](#) was correctly implemented. We conducted the following simulation procedure to compare the fixed effect (intercept) estimate and the residual variance component estimate. The results are also visualized in Figures 10 and 11.

It can be seen that the method option EQL1 in **hglm**, based on [Lee and Lee \(2012\)](#), produces variance component estimates nearly the same as the Laplace approximation implemented in **HGLMMM**. Nevertheless, our implementation does generate more bias compared to the HL(1,1) estimates from **HGLMMM** for this particular example.

```
n <- 100 # No. of observations
p <- 100 # No. of levels in the random effect
Z <- diag(p)
sigma2u <- 1 # Variance of the random effects
mu <- 0 # Simulated intercept term
n.rep <- 100 # Number of simulation replicates
set.seed(123)
results <- matrix(NA, n.rep, 3)
for (i.rep in 1:n.rep) {
  u <- rnorm(p, 0, sqrt(sigma2u))
  eta <- mu + Z %*% u
  y <- rpois(n, exp(eta))
  hglm1 <- try(hglm(y = y, X = matrix(1, n, 1), Z = Z, family = poisson(link = log),
    fix.disp = 1, maxit = 100, method = "EQL1"), silent = TRUE)
  if (class(hglm1) == "hglm") {
    if (hglm1$Converge == "converged") {
      results[i.rep, 3] <- as.numeric(hglm1$Converge == "converged")
      results[i.rep, 1] <- hglm1$fixef
      results[i.rep, 2] <- hglm1$varRanef
    }
  }
}
colnames(results) <- c("Intercept", "Variance component", "Converged")
summary(results)
```

##	Intercept	Variance component	Converged
## Min.	:-0.557	Min. :0.521	Min. :1
## 1st Qu.:	-0.123	1st Qu.:0.822	1st Qu.:1
## Median	:-0.029	Median :1.039	Median :1
## Mean	:-0.039	Mean :1.015	Mean :1
## 3rd Qu.:	0.054	3rd Qu.:1.169	3rd Qu.:1
## Max.	: 0.268	Max. :1.710	Max. :1
## NA's	:7	NA's :7	NA's :7

```
require(HGLMMM)
n <- 100 # No. of observations
p <- 100 # No. of levels in the random effect
Z <- diag(p)
sigma2u <- 1 # Variance of the random effects
mu <- 0 # Simulated intercept term
n.rep <- 100 # Number of simulation replicates
set.seed(123)
results2 <- matrix(NA, n.rep, 3)
```



```

for (i.rep in 1:n.rep) {
  u <- rnorm(p, 0, sqrt(sigma2u))
  eta <- mu + Z %*% u
  y <- rpois(n, exp(eta))
  RSC <- data.frame(int = rep(1, p))
  Subject <- factor(1:n)
  simPoisson <- as.data.frame(cbind(y, Subject))
  modG <- HGLMfit(DistResp = "Poisson", DistRand = c("Normal"), Link = "Log",
    LapFix = TRUE, ODEst = FALSE, ODEstVal = c(0), formulaMain = y ~ 1 +
      (1 | Subject), formulaRand = list(one = ~1), formulaOD = ~1, DataMain = simPoisson,
      DataRand = list(RSC), Offset = NULL, INFO = TRUE, DEBUG = FALSE)
  results2[i.rep, 1] <- modG$Results$Beta
  results2[i.rep, 2] <- exp(modG$Results$Dispersion)
  results2[i.rep, 3] <- as.numeric(abs(modG$Results$GradientDisp) < 1e-04)
}
colnames(results2) <- c("Intercept", "Variance component", "Converged")
summary(results2)

```

```

## Intercept      Variance component    Converged
## Min.   :-0.5230   Min.   :0.435      Min.   :1
## 1st Qu.: -0.0949   1st Qu.:0.759      1st Qu.:1
## Median :-0.0151   Median :0.950      Median :1
## Mean   :-0.0266   Mean   :0.968      Mean   :1
## 3rd Qu.: 0.0636   3rd Qu.:1.159      3rd Qu.:1
## Max.    : 0.2785   Max.    :1.753      Max.    :1

```

```

NAs <- is.na(results[, 1])
summary(results2[!NAs, ])
plot(results[!NAs, 1], results2[!NAs, 1], main = "Estimated Intercept", xlab = "hglm",
  ylab = "HGLMMM")
abline(0, 1)
plot(results[!NAs, 2], results2[!NAs, 2], main = "Estimated Variances", xlab = "hglm",
  ylab = "HGLMMM")
abline(0, 1)

```

Comparison with estimates using **spaMM**

For fitting the CAR() family in **hglm**, *i.e.* Gaussian Markov Random Field (GMRF) models, an alternative package **spaMM** can also be used. The following codes perform a simulation study to compare the estimates and execution time using the two different packages. The simulation results are displayed in Figures 12 and 13 (codes for plotting not shown).

It can be seen that the inter-connected GLM algorithm of **hglm** fits GMRF models much faster than **spaMM** which uses a numerical optimization algorithm. The estimated parameters were very similar from the EQL method of **hglm** and the "EQL-" method of **spaMM**, with correlation coefficients between the two methods 1.0000, 0.9998, 0.9997 and 0.9999 for the residual variance, ρ , τ and the intercept, respectively.

```

require(hglm)
require(spaMM)

data(ohio)

```

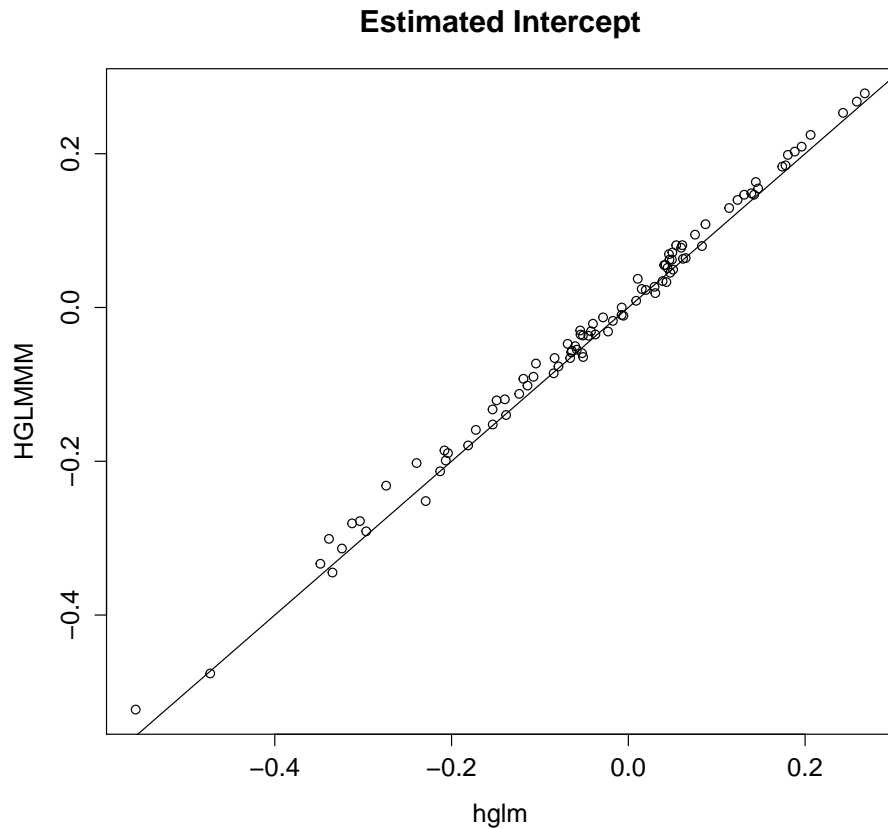


Figure 10: Comparison of the intercept estimates using `hglm` and `HGLMMM`.

```

thglm0 <- tspaMM0 <- rhothglm0 <- phihtglm0 <- tauhtglm0 <- betathglm0 <- rhospaMM0 <- phispMM0 <- tauspaMM0 <- b
  10), n100 = rep(0, 10), n200 = rep(0, 10), n300 = rep(0, 10), n400 = rep(0,
  10), n500 = rep(0, 10))
n <- c(50, seq(100, 500, 100))

run1 <- function(subMedian, subD) {
  t1 <- system.time(hg0 <- hglm(fixed = MedianScore ~ 1, random = ~1 | district,
    rand.family = CAR(D = subD), data = subMedian))[3]
  eigD <- eigen(subD)
  t3 <- system.time(sp0 <- corrHLfit(MedianScore ~ 1 + adjacency(1 | district),
    data = subMedian, HLmethod = "EQL-", adjMatrix = subD, lower = list(rho = 1/min(eigD$values)),
    upper = list(rho = 1/max(eigD$values))))[3]
  return(list(t1 = t1, t3 = t3, hg0 = hg0, sp0 = sp0))
}

for (k in 1:length(n)) {
  set.seed(911)
  for (i in 1:10) {
    lv <- sample(levels(ohioMedian$district[1:500]), n[k])
    idx <- which(ohioMedian$district %in% lv)
    subdistrict <- factor(as.character(ohioMedian$district)[idx])
    subMedian <- data.frame(MedianScore = ohioMedian$MedianScore[idx], district = subdistrict)
    subD <- ohioDistrictDistMat[levels(subdistrict), levels(subdistrict)]
    res <- try(run1(subMedian, subD), silent = TRUE)
    while (inherits(res, "try-error")) {
      lv <- sample(levels(ohioMedian$district[1:500]), n[k])
      idx <- which(ohioMedian$district %in% lv)
    }
  }
}

```

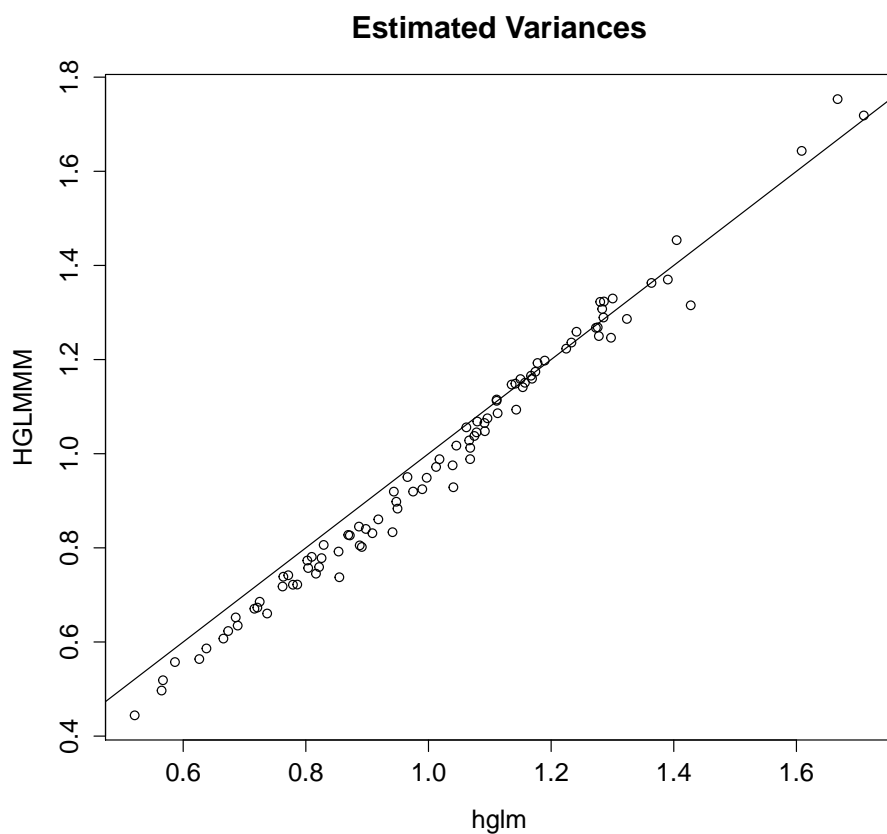


Figure 11: Comparison of the residual variance estimates from `hglm` and `HGLMMM`.

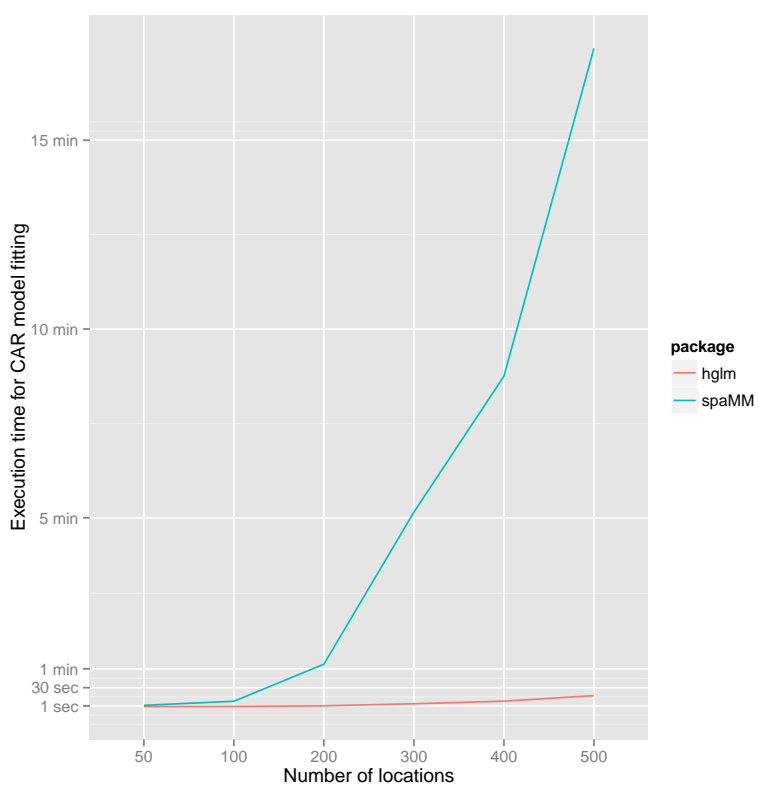


Figure 12: Comparison of the execution time for fitting CAR models using `hglm` and `spaMM`.

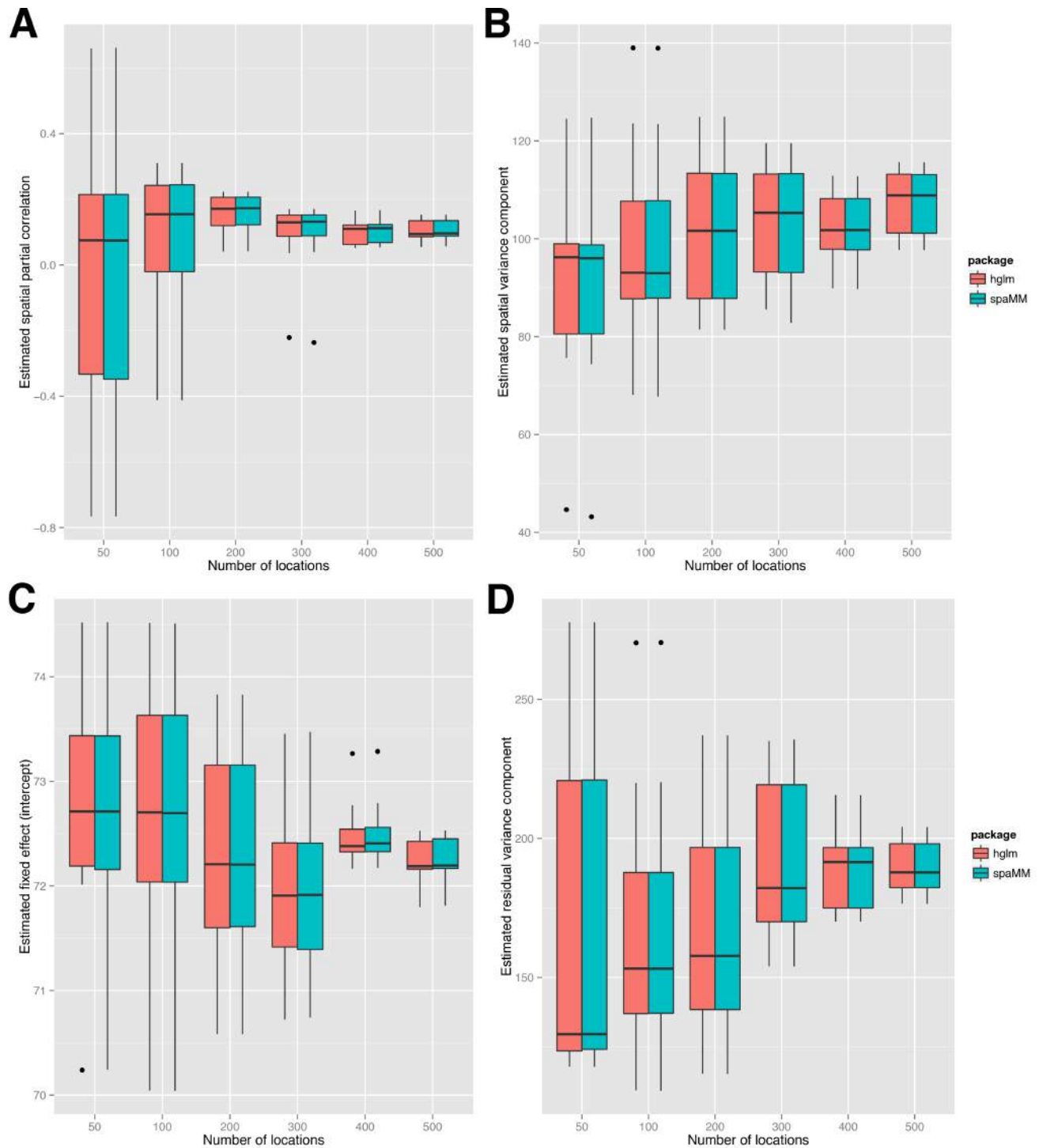


Figure 13: Comparison of the CAR models estimated (A) spatial partial correlation parameters, (B) spatial variance components, (C) fixed effects, and (D) residual variance estimates from **hglm** (EQL1) and **spaMM** (HL(1,1)).

```

subdistrict <- factor(as.character(ohioMedian$district)[idx])
subMedian <- data.frame(MedianScore = ohioMedian$MedianScore[idx],
  district = subdistrict)
subD <- ohioDistrictDistMat[levels(subdistrict), levels(subdistrict)]
res <- try(run1(subMedian, subD))
}
thglm0[i, k] <- res$t1
tspaMM0[i, k] <- res$t3
rhothglm0[i, k] <- res$hg0$CAR.rho
phithglm0[i, k] <- res$hg0$varFix
tauthglm0[i, k] <- res$hg0$CAR.tau
betathglm0[i, k] <- res$hg0$fixef
rhospaMM0[i, k] <- res$sp0$corrPars$rho
phispaMM0[i, k] <- res$sp0$phi
tauspaMM0[i, k] <- res$sp0$lambda
betaspaMM0[i, k] <- res$sp0$fixef
}
}

```

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. [p6]
- M. Alam, L. Rönnegård, and X. Shen. Fitting conditional and simultaneous autoregressive spatial models in hglm. *Submitted*, 0(0):0–0, 2014. [p2]
- M. Bayarri, M. DeGroot, and J. Kadane. What is the likelihood function? (with discussion). In *Statistical Decision Theory and Related Topics IV*. (eds. S. S. Gupta and J. O. Berger) Springer, New York., 1, 1988. [p33]
- J. Bjornstad. On the generalization of the likelihood function and likelihood principle. *Journal of the American Statistical Association*, 91:791–806, 1996. [p32, 33]
- N. E. Breslow and D. G. Clayton. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, 88:9–25, 1993. [p18, 32]
- K. Carling, L. Rönnegård, and K. Roszbach. An analysis of portfolio credit risk when counterparties are interdependent within industries. *Sveriges Riksbank Working Paper*, 169, 2004. [p6]
- D. Clayton and J. Kaldor. Empirical bayes estimates of age-standardized relative risks for use in disease mapping. *Biometrics*, 43:671–681, 1987. [p28]
- A. Coster, J. Bastiaansen, M. Calus, C. Maliepaard, and M. Bink. Qtlmas 2009: Simulated dataset. *Submitted*, 2009. [p16]
- N. A. Cressie. *Statistics for Spatial Data (Revised Edition)*. Wiley, New York, 1993. [p27, 29]
- M. J. Crowder. Beta-binomial anova for proportions. *Applied Statistics*, 27:34–37, 1978. [p18]
- D. P. Gaver and I. G. O’Muircheartaigh. Robust empirical bayes analyses of event rates. *Technometrics*, 29(1):1–15, 1987. [p15]
- 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. [p31]

- 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. [p5]
- J. S. Hodges. *Richly Parametarized Linear Models: Additive Linear and Time Series and Spatial Models Using Random Effects*. Chapman and Hall/CRC, Boca Raton, 2013. [p29]
- W. Lee and Y. Lee. Modifications of REML algorithm for HGLMs. *Statistical Computing*, 22:959–966, 2012. [p2, 8, 10, 26, 36]
- Y. Lee and J. A. Nelder. Likelihood for random-effect models (with discussion). *Statistical and Operational Research Transactions*, 29:141–182, 2005. [p32]
- Y. Lee and J. A. Nelder. Likelihood inference for models with unobservables: another view. *Statistical Science*, 24:255–269, 2009a. [p32, 33]
- Y. Lee and J. A. Nelder. Rejoinder: Likelihood inference for models with unobservables: another view. *Statistical Science*, 24:294–302, 2009b. [p32, 33]
- Y. Lee and J. A. Nelder. Double hierarchical generalized linear models with discussion. *Applied Statistics*, 55:139–185, 2006. [p10, 32, 33]
- Y. Lee and J. A. Nelder. Hierarchical generalized linear models with discussion. *J. R. Statist. Soc. B*, 58: 619–678, 1996. [p2, 3, 4, 5, 8, 18, 20, 32, 34, 35]
- Y. Lee, J. A. Nelder, and Y. Pawitan. *Generalized linear models with random effects: a unified analysis via h-likelihood*. Chapman & Hall/CRC, 2006. [p3, 4, 5, 10, 13, 16, 31, 34, 35]
- Y. Lee, J. A. Nelder, and M. Noh. H-likelihood: problems and solutions. *Statistics and Computing*, 17: 49–55, 2007. [p32]
- J. LeSage and R. Pace. *Introduction to Spatial Econometrics*. Chapman & Hall/CRC, Boca Raton, 2009. [p29]
- T. A. Louis. Discussion of likelihood inference for models with unobservables: another view. *Statistical Science*, 24:270–272, 2009. [p32, 33]
- M. Lynch and B. Walsh. *Genetics and analysis of Quantitative Traits*. Sinauer Associates, Inc., 1998. ISBN 087893481. [p15]
- P. McCullagh and J. A. Nelder. *Generalized linear models*. Chapman & Hall/CRC, 1989. [p2, 4, 6, 21]
- X.-L. Meng. Decoding the h-likelihood. *Statistical Science*, 24:280–293, 2009. [p32, 33]
- G. Molenberghs, M. G. Kenward, and G. Verbeke. Discussion of likelihood inference for models with unobservables: another view. *Statistical Science*, 24:273–279, 2009. [p32, 33]
- P. H. Myers, D. C. Montgomery, and G. G. Vining. *Generalized linear models with applications in engineering and the sciences*. New York: John Wiley and Sons, 2002. [p20]
- M. Noh and Y. Lee. Reml estimation for binary data in glmms. *Journal of Multivariate Analysis*, 98: 896–915, 2007. [p2, 31, 32, 36]
- 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. [p31, 32]
- K. Ord. Estimation methods for models of spatial interaction. *J. Am. Stat. Assoc.*, 70:120–126, 1975. [p29]

- Y. Pawitan. *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford Science Publications, 2001. [p32]
- R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. URL <http://www.R-project.org>. ISBN 3-900051-07-0. [p2]
- 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. [p10, 16]
- S. G. Self and K.-Y. Liang. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association*, 82(398): 605–610, 1987. [p31]
- X. Shen, M. Alam, F. Fikse, and L. Rönnegård. A novel generalized ridge regression method for quantitative genetics. *Genetics*, 193(4):1255–1268, Apr. 2013. doi: 10.1534/genetics.112.146720. URL <http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?dbfrom=pubmed&id=23335338&retmode=ref&cmd=prlinks>. [p2]
- M. M. Wall. A close look at the spatial structure implied by the CAR and SAR models. *Journal of Statistical Planning and Inference*, 121:311–324, 2004. [p29]