`glme`: a C-program for parameter estimation using Gibbs-sampling in large linear mixed-effects models, with applications to DNA microarray data

Norwegian Computing Center
APPLIED RESEARCH AND DEVELOPMENT

NOTE

NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY
TRONDHEIM, NORWAY

SAMBA/10/04

PREPRINT STATISTICS
NO. 2/2004

Turid Follestad
Mette Langaas
Håvard Rue
Marit Holden
Anders Løland

March 2004

NR Norwegian Computing Center
APPLIED RESEARCH AND DEVELOPMENT

NR Note

Title: `glme`: a C-program for parameter estimation using Gibbs-sampling in large linear mixed-effects models, with applications to DNA microarray data

Date: March 2004
Year: 2004
Note no.: SAMBA/10/04

PREPRINT STATISTICS
NO. 2/2004

Author: Turid Follestad <turid.follestad@math.ntnu.no>
Mette Langaas <mette.langaas@math.ntnu.no>
Håvard Rue <havard.rue@math.ntnu.no>
Marit Holden <marit.holden@nr.no>
Anders Løland <anders.loland@nr.no>

Abstract: This note describes a C-routine for the estimation of parameters in linear mixed-effects models, developed for the analysis of large data sets. The `glme`-routine implements a fully Bayesian approach to parameter estimation using the Gibbs sampler. The program is applicable for data from experiments based on crossed as well as nested designs. The development of the program was motivated by the need for computationally efficient algorithms for analysing data from DNA microarray experiments, in situations where the effects of interest include both overall and gene-specific effects. We describe how the program can be applied in this context.

http://www.math.ntnu.no/preprint/statistics/2004/S2-2004.pdf

Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway

Keywords: Linear mixed-effects model, DNA microarray data, Gibbs-sampling

Target group:

Availability: Open

Project:

Project no.: 220091

Research field: Bioinformatics

No. of pages: 20

Norwegian Computing Center
Gaustadalléen 23, P.O. Box 114 Blindern, NO-0314 Oslo, Norway
Telephone: +47 2285 2500, telefax: +47 2269 7660, http://www.nr.no

`glme`: a `C`-program for parameter estimation using Gibbs-sampling in large linear mixed-effects models, with application to DNA microarray data

Turid Follestad, Mette Langaas and Håvard Rue

Department of Mathematical Sciences
Norwegian University of Science and Technology
N–7491 Trondheim, Norway.
E–MAIL: {turid.follestad,mette.langaas,havard.rue}@math.ntnu.no

Marit Holden and Anders Løland

Norwegian Computing Center
P.O.Box 114 Blindern
N–0314 Oslo, Norway.
E–MAIL: {marit.holden,anders.loland}@nr.no

March 2004

Abstract

This note describes a `C`-routine for the estimation of parameters in linear mixed-effects models, developed for the analysis of large data sets. The `glme`-routine implements a fully Bayesian approach to parameter estimation using the Gibbs sampler. The program is applicable for data from experiments based on crossed as well as nested designs. The development of the program was motivated by the need for computationally efficient algorithms for analysing data from DNA microarray experiments, in situations where the effects of interest include both overall and gene-specific effects. We describe how the program can be applied in this context.

Version

This documentation is valid for revision 1.70 and higher of the program.

Contents

1 Introduction

In this report we describe the C-routine `glme` implementing a Bayesian approach to estimation of parameters in linear mixed effects models for large data sets, using the Gibbs sampler.

The implementation of the program was motivated from the need for efficient algorithms for the analysis of data from DNA microarray experiments, in situations where the effects of interest include overall effects that, unlike differential expression, are not to be assessed on a gene by gene basis. The microarray technique has become a very useful tool in functional genomics, enabling the simultaneous comparison of a large number of genes. However, the high dimension of the resulting data sets also represents a computational challenge. By utilising efficient programming techniques, joint estimation of the parameters of a linear mixed effects model becomes computationally feasible even for large data sets, and the `glme` program can be applied to data from experiments that are based on crossed as well as nested experimental design.

Using the microarray technique, gene expression is quantified by hybridising mRNA samples representing biological material and conditions of interest to DNA probes representing known genes or genomic sequences, immobilised on the array. The mRNA samples are labelled by a fluorescent dye, and the fluorescent intensity corresponding to each probe, obtained by scanning and image analysis, is used as a measure of gene expression. There are two main types of microarrays in current use, spotted two-colour DNA arrays, where two mRNA samples, first reverse transcribed to cDNA and then labelled with different dyes (red and green) are hybridised to each array, and high-density single-colour oligonucleotide arrays. The two types differ mainly in the length of the DNA probe sequences and in how these are distributed on the array. For an overview of different microarray techniques and other topics regarding microarrays, see e.g. the *Nature Genetics* supplements *The Chipping Forecast* (vol. 21, 1999) and *Chipping Forecast II* (vol. 32, 2002).

A major aim of many microrarray experiments is to identify genes that are differentially expressed for different varieties. The term variety refers to the mRNA samples that are studied, e.g. different tissues (like healthy and cancer tissue) or samples representing a biological system at different points in time. In addition to, or in place of, the gene-by-gene differential expression, effects measured at an overall, *not gene-specific* level can be of interest. For example, if the individuals under study in a medical experiment can be allocated into different groups according to e.g. age or medical condition, questions of interest include whether persons in different groups respond differently to treatment and whether there is a gene-group interaction effect. Depending on the aim of the experiment, different inferential approaches are suitable.

The `glme`-program is designed for the general problem of estimating both gene-specific and not gene-specific effects. Linear mixed effects models (McCulloch and Searle, 2001; Pinheiro and Bates, 2001) provide a framework for the joint estimation of different types of effects, as well as for handling of multiply spotted genes. The application of linear mixed effects models to microarray data was first introduced by Kerr, Martin and Churchill (2000) and further elaborated by among others Kerr, Afshari, Bennett, Bushel, Martinez, Walker and Churchill (2002) and Wolfinger, Gibson, Wolfinger, Bennett, Hamadeh, Bushel, Ashfari and Paules (2001). They model the intensity in each spot, after a suitable transformation, as a sum of effects representing e.g. array, dye, variety and gene, as well as interaction effects between these factors. Aiming at assessing differential expression, the interaction between gene and variety is the effect of interest. Due to the computational burden, and to enable gene-specific error distributions, the LME models described are fitted using a two step procedure, first estimating overall effects, and then fitting the gene-specific effects based on the residuals from the first step. This approach is reasonable when the aim is to assess differential expression per gene, but in more general situations, a one-step procedure for estimating gene-specific as well as not gene-specific effects is preferred. The `glme` program is developed to provide a computationally efficient routine for estimation of the model parameters in one step.

In Section 2 we give a brief description of the LME in a microarray data analysis context. The parameter estimation algorithm is described in Section 3, and the `glme`-program is documented in Section 4. Some examples are given in Section 5.

2 A brief description of linear mixed-effects models applied to microarray data

In this section we first introduce the general linear mixed effects model, and then give a description of how such models can be applied to microarray data.

2.1 General model

A linear mixed effects model (LME) is of the form

$$\boldsymbol{y} = \mathbf{1}\mu + \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{b} + \boldsymbol{\epsilon}, \tag{1}$$

where

- $\boldsymbol{y}$ is a vector of n suitably transformed observed response values,
- μ is an overall mean,
- $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T, \ldots, \boldsymbol{\beta}_{m_F}^T)^T$ is a vector of *fixed* effects corresponding to the levels of m_F factors,
- $\boldsymbol{b} = (\boldsymbol{b}_1^T, \boldsymbol{b}_2^T, \ldots, \boldsymbol{b}_{m_R}^T)^T$ is a vector of *random* effects corresponding to levels of m_R factors, and where the elements of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{m_R}$ are assumed to be independent realisations of zero mean Gaussian variables with variances $\sigma^2_{R_1}, \ldots, \sigma^2_{R_{m_R}}$,
- $\boldsymbol{X}$ and $\boldsymbol{Z}$ are design matrices for the fixed and random effects respectively, and
- $\boldsymbol{\epsilon}$ is a vector of residuals, assumed to be zero mean independent Gaussian variables with common variance σ^2.

Effects should be specified as fixed if we are mainly interested in estimating the specific values associated with the different levels of the corresponding factor. If the main interest lies in assessing the variability associated with a factor, it should be treated as random. In general, random effects can be used as a tool to explicitly account for different sources of random variation other than the residual variance, as well as for correlations within groups of observations. For brevity, we will use the terms "fixed factors" and "random factors" to refer to factors for which the effects are fixed and random, respectively.

A component-wise representation of an LME is often used instead of the matrix representation (1). As an example, consider the model

$$y_{ijk} = \mu + \beta_i + b_j + c_{ij} + \epsilon_{ijk}, \tag{2}$$

where $\beta_i,\ i = 1, 2$ are the fixed effects corresponding to two levels of a factor, $b_j,\ j = 1, 2, 3$ are random effects corresponding to three realisations from the set of possible levels of a second factor, and $\{b_{ij}\}$ are random interaction effects. Each pair of fixed and random effects is observed twice. For this model, $\boldsymbol{\beta} = \boldsymbol{\beta}_1 = (\beta_1, \beta_2)^T$ and $\boldsymbol{b} = (\boldsymbol{b}_1^T, \boldsymbol{b}_2^T)^T$, where $\boldsymbol{b}_1 = (b_1, b_2, b_3)^T$ and $\boldsymbol{b}_2 = (c_{1,1}, c_{1,2}, \ldots, c_{2,3})^T$. The design matrices are $\boldsymbol{X}$ and $\boldsymbol{Z} = [\boldsymbol{Z}_1\ \boldsymbol{Z}_2]$, where

$$\boldsymbol{X} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}, \quad \boldsymbol{Z}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{Z}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

2.2 Linear mixed effects models for microarray data

As pointed out earlier, the implementation of the `glme`-program was motivated by problems where overall effects are of main interest. However, the program can also be applied to assess differential expression, which is the problem for which linear mixed effects models for microarray data were first introduced. In the following subsections we first present some recent applications of LME as a tool for identifying differentially expressed genes, and then we describe a problem of the type that motivated the development of the `glme`-program.

2.2.1 Identifying differentially expressed genes by LME

A model for the analysis of microarray data within the LME framework was first proposed by Kerr et al. (2000) and further elaborated in Kerr et al. (2002). They proposed a linear fixed effects model (i.e. an LME with no random effects) for the identification of differentially expressed genes taking into account array, dye and overall gene-effects as well as the gene-specific variety effect which is the effect of interest in this context. Their model, according to Kerr et al. (2000), which they denote an ANOVA model, is of the form

$$y_{ijkg} = \mu + A_i + D_j + V_k + G_g + (AG)_{ig} + (VG)_{kg} + \epsilon_{ijkg}, \tag{4}$$

where y_{ijkg} is the log-transformed intensity of gene g for variety k labelled with dye j on array i. The effect of interest is the variety-gene interaction $(VG)_{kg}$. The model is a special case of the general model (1) with no random effects term. The vector $\boldsymbol{\beta}$ of fixed effects is $\boldsymbol{\beta} = (\boldsymbol{A}^T, \boldsymbol{D}^T, \boldsymbol{V}^T, \boldsymbol{G}^T, (\boldsymbol{AG})^T, (\boldsymbol{VG})^T)^T$, where $\boldsymbol{A} = (A_1, \ldots, A_{n_A})^T$ is the vector of fixed effects corresponding to the n_A levels of the the array factor, $\boldsymbol{D} = (D_1, D_2)^T$ is the vector of dye-effects, and so on. In Kerr et al. (2002) a dye-gene interaction term $(DG)_{jg}$ is included in the above model to account for gene-specific dye-effects, and replication is allowed for.

Due to the large number of genes, estimation of the parameters in (4) is computationally expensive. Also, assuming i.i.d. noise implies that the residual variances for all genes are assumed to be equal, an assumption that can be questioned. Therefore, the model, including the dye-gene interaction and replication, is fit in two steps (Cui and Churchill, 2003):

$$\begin{aligned} \text{Step 1:} \quad y_{ijkgr} &= \mu + A_i + D_j + V_k + r_{ijkgr}, \\ \text{Step 2:} \quad r_{ijkgr} &= G_g + (AG)_{ig} + (VG)_{kg} + (DG)_{jg} + \epsilon_{ijkgr}, \end{aligned} \tag{5}$$

where index r refers to replications and $\text{Var}(\epsilon_{ijkgr}) = \sigma_g^2$. First, Step 1 is fitted for all genes simultaneously. After obtaining the residuals from Step 1, Step 2 is then fitted on a gene by gene basis. For a balanced design and for $\sigma_g^2 = \sigma^2, \ \forall g$, fitting the model in two steps is equivalent to fitting the full model (4) (Wu, Kerr, Cui and Churchill, 2003). Taking a frequentist approach to inference, the fixed effects are estimated by least squares, and the significance of the different effects are assessed by non-parametric tests and bootstrapping, avoiding Gaussian assumptions on the residuals. This LME approach is implemented in the R-package `MAANOVA` (Wu et al., 2003) which has recently been extended to handle random effects as well.

A similar two-step mixed effects model is given in Wolfinger et al. (2001). The variety-gene interaction effect $(VG)_{kg}$, measuring differential expression, is treated as fixed, as is the main effect G_g for genes. The dye effect D_j is also treated as fixed, since it has only two levels, red or green. The remaining main effects of array A_i and variety V_k, as well as the array-gene interaction $(AG)_{ig}$ are random, which means that they are considered as realisations from a distribution over a hypothetical set of levels of the factor. A similar model is applied by Jin, Riley, Wolfinger, White, Passador-Gurgel and Gibson (2001).

For two-colour spotted DNA arrays, an alternative approach to assessing differentially expressed genes is to specify tests on a gene-by-gene basis based on log-ratios of intensities at each spot. However, conclusions drawn on a gene-by-gene basis can be misleading, due to the fact that data from a microarray experiment typically consists of only a few observations per gene. One remedy to this problem is to borrow strength from observations for the remaining genes. Work along these lines include the empirical Bayes approaches of Lönnstedt and Speed (2002) and Smyth (2003b), the latter implemented in the `R`-package `limma` (Smyth, 2003a), as well as the related approach of Efron, Tibshirani, Storey and Tusher (2001). Using these or similar approaches, the data are usually adjusted for systematic effects that can be attributed to the microarray technology rather than true biological processes (normalisation). In the linear mixed effects models described above, array and dye effects are explicitly included in the model, such that data normalisation and estimation of the gene-dependent variety effects are integrated in a common framework, explicitly accounting for the degrees of freedom that are lost in the normalisation process. However, the linear normalisation model in Step 1 in (5) has proved to be inadequate in practice, as experience indicates spatial patterns in intensity readings across arrays, as well as non-linear dependence of log-ratios on intensities, see e.g. Yang, Dudoit, Luu, Lin, Peng, Ngai and Speed (2002) and Kerr et al. (2002). Therefore, additional pre-processing is most often needed, but the normalisation part of the linear model could still be kept to account for remaining systematic bias. It should also be pointed out that other transformations than the log-transformation might reduce systematic bias and the need for additional pre-processing, an example being the variance stabilising transform of Huber, von Heydebreck, Sültmann and Vingron (2002). For a discussion of different alternative transformations, see Cui, Kerr and Churchill (2002).

2.2.2 An application of LME to the estimation of overall effects

A published application of the `glme`-program is found in Nygaard, Løland, Holden, Langaas, Rue, Liu, Myklebost, Fodstad, Hovig and Smith-Sørensen (2003). They study the effects of different degrees of mRNA amplification on the variability of estimates of gene expression ratios in cDNA experiments. Their model includes main effects for array, cell line, batch, dye and amplification protocol, as well as two-factor and three-factor interaction effects. See the cited paper for a further description of the model. In this application, the major interest lies in the main effects of amplification, averaged over all genes. However, gene-specific effects enter the model through random interaction effects, included to explicitly account for the variability associated with these effects. The resulting model should be estimated in one step, and therefore a computationally efficient approach to parameter estimation, as provided by the `glme`-program, is needed.

3 The Gibbs sampling algorithm

The `glme` program implements a fully Bayesian approach to estimation of the parameters of a general linear mixed effects model of the form (1), using the Gibbs sampler. The unknown parameters of the model are the overall mean μ, the vector $\boldsymbol{\beta}$ of fixed effects the variances $\sigma^2_{R_1}, \ldots, \sigma^2_{R_{m_R}}$ of the random effects and the residual variance σ^2. At each iteration of the Gibbs sampler, predictions of the random effects $\boldsymbol{b}$ are also generated. We will denote by $\boldsymbol{\theta}$ the vector of unknown parameters, including the realisations of the random effects, and we write

$$\boldsymbol{\theta} = (\mu, \boldsymbol{\beta}^T, \boldsymbol{b}^T, \sigma^2_{R_1}, \ldots, \sigma^2_{R_{m_R}}, \sigma^2)^T.$$

To avoid over-parameterisation, constraints or re-parameterisation might have to be invoked, as discussed in Section 4.3 and Section 5.2, respectively.

3.1 The Bayesian model

Given the data vector $\boldsymbol{y}$ and assuming conditional independence between observations, the likelihood of the Bayesian model is

$$\pi(\boldsymbol{y}; \boldsymbol{\theta}) = \prod_{i=1}^{n} \pi(y_i \mid \boldsymbol{\theta}), \tag{6}$$

where $\pi(y_i \mid \boldsymbol{\theta}),\ i = 1, \ldots, n$ are given by the rows of (1). We assign a uniform prior to the overall mean, independent Gaussian priors to each of the fixed and random effects, and inverse Gamma priors on the variances. More specifically,

$$
\begin{aligned}
\pi(\mu) &\propto 1, && (7)\\
\pi(\boldsymbol{\beta}_j) &\sim \mathcal{N}(\mathbf{0}, \sigma_F^2 \boldsymbol{I}), \quad j = 1, \dots, m_F, && (8)\\
\pi(\boldsymbol{b}_k \mid \sigma_{R_k}^2) &\sim \mathcal{N}(\mathbf{0}, \sigma_{R_k}^2 \boldsymbol{I}), \quad k = 1, \dots, m_R, && (9)\\
\pi(1/\sigma_{R_k}^2) &\sim \text{Gamma}(\alpha_R, \beta_R), \quad k = 1, \dots, m_R, && (10)\\
\pi(1/\sigma^2) &\sim \text{Gamma}(\alpha_{\sigma^2}, \beta_{\sigma^2}). && (11)
\end{aligned}
$$

The hyper-parameters σ_F^2, α_R, β_R, α_{σ^2} and β_{σ^2} are treated as fixed, and can be specified by the user using input options, as described in Section 4.1. Observe that the value of σ_F^2 is to be specified in terms of the precision $\tau_F = 1/\sigma_F^2$. The joint posterior distribution is

$$
\pi(\boldsymbol{\theta} \mid \boldsymbol{y}) \propto \prod_{i=1}^{n} \{\pi(y_i \mid \boldsymbol{\theta})\} \prod_{j=1}^{m_F} \{\pi(\boldsymbol{\beta}_j)\} \prod_{k=1}^{m_R} \{\pi(\boldsymbol{b}_k \mid \sigma_{R_k}^2)\pi(1/\sigma_{R_k}^2)\}\pi(1/\sigma^2)\pi(\mu). \tag{12}
$$

Due to the high dimensionality of the problem, direct maximisation of the posterior distribution is computationally infeasible, but as we describe in the following subsection, the problem can be dealt with using simulation based inference. There, we describe the Gibbs sampling algorithm that is implemented in `glme`.

3.2 The Gibbs sampler

Due to the high dimensionality of microarray data, alternatives to standard approaches to parameter estimation for the linear mixed-effects model is needed. The Gibbs sampler is a simulation based approach that is suitable for complex problems. It is a special case of the class of Markov chain Monte Carlo methods, where inference is based on samples from a Markov chain with the distribution of interest as its stationary distribution. For an introduction to MCMC methods see e.g. Gilks, Richardson and Spiegelhalter (1996).

The general idea of the Gibbs sampler is to generate approximate samples from the joint posterior distribution by sampling one parameter (or one group of parameters) at the time, conditionally on the most recently sampled values of the remaining parameters. These conditional distributions are denoted the full conditional distributions. It can be shown that samples generated by this procedure converge to samples from the joint posterior distribution. To simplify the presentation we illustrate the approach by a linear mixed-effects model with one random and one fixed effect, but generalisation to several random and fixed effects is straightforward. We consider the model

$$
y_{jkr} = \mu + \beta_j + b_k + \epsilon_{jkr}, \tag{13}
$$

which is a special case of (1) with $m_F = m_R = 1$ and with replicates $r = 1, \dots, N$ for each combination of random and fixed effects. The unknown quantities are

- the overall mean μ,
- the fixed effects $\beta_j;\ j = 1, \dots, J$,
- the variance σ_R^2 of the random effects $b_k \sim \mathcal{N}(0, \sigma_R^2)$ as well as the effects $b_k,\ k = 1, \dots, K$ associated with the K realisations of the corresponding factor, and

- the residual variance σ^2.

The full conditional distributions can be derived from (12), and are given by

$$\begin{aligned}
\pi(\mu \mid \boldsymbol{y}, \{\beta_j\}, \{b_k\}, \sigma_R^2, \sigma^2) &\propto \prod_{j,k,r} \{\pi(y_{jkr} \mid \mu, \beta_j, b_k, \sigma^2)\}\pi(\mu), && (14)\\
\pi(\beta_j \mid \boldsymbol{y}, \mu, \{\beta_l\}_{l\neq j}, \{b_k\}, \sigma_R^2, \sigma^2) &\propto \prod_{k,r} \{\pi(y_{jkr} \mid \mu, \beta_j, b_k, \sigma^2)\}\pi(\beta_j), && (15)\\
\pi(\sigma_R^2 \mid \boldsymbol{y}, \mu, \{\beta_j\}, \{b_k\}, \sigma^2) &\propto \prod_{k} \{\pi(b_k \mid \sigma_R^2)\}\pi(1/\sigma_R^2), && (16)\\
\pi(b_k \mid \boldsymbol{y}, \mu, \{\beta_j\}, \{b_l\}_{l\neq k}, \sigma_R^2, \sigma^2) &\propto \prod_{j,r} \{\pi(y_{jkr} \mid \mu, \beta_j, b_k, \sigma^2)\}\pi(b_k \mid \sigma_R^2), && (17)\\
\pi(\sigma^2 \mid \boldsymbol{y}, \mu, \{\beta_j\}, \{b_k\}, \sigma_B^2) &\propto \prod_{j,k,r} \{\pi(y_{jkr} \mid \mu, \beta_j, b_k, \sigma^2)\}\pi(1/\sigma^2). && (18)
\end{aligned}$$

Since all priors are conjugate priors, the full conditional distributions are all standard distributions, that is, inverse Gamma distributions for the variances and Gaussian distributions for the β_j and b_k. Thus, all full conditional distributions can be sampled directly.

The Gibbs sampling algorithm can be summarised as follows:

```
Set μ = 0
Set σ² = 1
Set σ_R² = 1
for (j = 1,...,J) do
    Set β_j = 0
end for
for (k = 1,...,K) do
    Set b_k = 0
end for
for (i = 1,...,n_iter) do
    Sample μ from the Gaussian full conditional (14)
    Sample σ² from the inverse Gamma full conditional (18)
    Sample σ_R² from from the inverse Gamma full conditional (16)
    for (j = 1,...,J) do
        Sample β_j from from the Gaussian full conditional (15)
    end for
    for (k = 1,...,K) do
        Sample b_k from from the Gaussian full conditional (17)
    end for
end for
```

Based on the resulting set of n_{iter} samples, estimated posterior means or medians of the parameters are found by their empirical counterparts, discarding a set of *burn-in* samples to avoid any effect of arbitrary initial values. Since the number of burn-in iterations needed is usually not known on beforehand, we need to monitor the algorithm to decide whether the underlying Markov chain has converged (approximately) to the joint posterior distribution. A useful option of the `glme`-program is the ability to store the full state, including the current state of the random number generator, at exit. This means that the program can be interrupted to check for convergence and then continued at the current values.

Successive samples from the Gibbs sampler are correlated by design. This correlation does not have any profound effect on the estimated mean values, but variance estimates will be negatively biases. The problem can be reduced by thinning, computing the estimates based on every t'th iteration, where t depend on the degree of correlation.

4 Documentation of the `C`-program `glme`

The `glme`-program estimates the the posterior distributions of the unknown parameters of the Bayesian model in Section 3.1 based on information from an input data file and additional command line information. The program can be run until it is interrupted, which is the default, or alternatively for a specified number of iterations. In any case, the state of the program at exit can be stored, such that a new run can be started from this state. The calling sequence and the complete list of options are listed in Section 4.1. In the subsequent subsections, we give a more detailed description of how to specify the components of the model, and the output returned by the program.

4.1 Calling sequence

Synopsis

glme `[-R restore_state_fnm] -d datafile [-s seed]`
`[-S save_state_fnm][-t thinning][-r dump_residuals]`
`[-e dump_effects [col1 col2 ...]][-n maxiter][-a A][-b B]`
`[-f prec][-c][-v][-l][-m dump_mean][-w][-V][-T niter][-h]`

Options

`-R restore_state_fnm`
: Restore the state of a previous run the program, saved in the file `restore_state_fnm`, and start the new run at this state. If used, this should be the first option.

`-d datafile` **(required)**
: Input file containing the specification of the linear mixed-effects model. The file should be of the format described in Section 4.2.

`-s seed`
: Set the seed of the random number generator to `seed`. The default seed is fixed, such that repeated runs are identical.

`-S save_state_fnm`
: Save the current state of the program in file `save_state_fnm`. A new run of the program can be started from this state by using the option -R.

`-t thinning`
: Updated values for every `thinning` iteration is printed, the default is 1. The output is printed to standard output, and can be directed to a user specified output file.

-r dump_residuals

Dump the residuals every dump_residuals iteration, the default is never (0). The residuals are dumped on a file named FILE.residuals, where FILE equals the name of the input data file (the argument of the -d option) without suffix. If a file of that name already exists, the residuals are appended to that file. Each row of the file is on the format

$$q \quad \tau^{(q)} \quad r_1^{(q)} \quad r_2^{(q)} \quad \cdots \quad r_n^{(q)},$$

where q is the iteration index, $\tau^{(q)}$ is the residual precision and $r_i^{(q)},\ i = 1, \ldots, n$ are the corresponding residuals from the model for the n observations.

-e dump_effects [col1 col2 ...]

Dump the values of the random effects corresponding to (a subset of) the random factors every dump_effects iteration, the default is never (0). The values are dumped on a file named FILE.reffects, where FILE equals the name of the input data file (the argument of the -d option) without suffix. If a file of that name already exists, the values are appended to that file. Each row of the file is on the format

$$q \quad \text{col} \quad \tau_{R_{\text{col}}}^{(q)} \quad re_1^{(q)} \quad re_2^{(q)} \quad \cdots \quad re_{m_{\text{col}}}^{(q)},$$

where q is the iteration index, col is the column index for the random factor for which the effects are to be printed, $\tau_{R_{\text{col}}}^{(q)}$ is the precision and $re_i^{(q)};\ i = 1, \ldots, m_{\text{col}}$ are the effects corresponding to the m_{col} distinct realisations of that factor. The factors are numbered according to the column index in the input data file, including *all* columns on the file. Note that the columns are numbered beginning with 0. Invoking the option -e 10 2 3 means printing the random effects corresponding to the *third* and *fourth* column every 10 iterations.

-n maxiter

Run only maxiter iterations. The default is ∞, such that the program is run until it is interrupted. If this option is used with the -R option, maxiter should include the number of iterations from the run that is restored.

-a A The first parameter, α_R, in the gamma priors (10) for the precisions $\{\tau_{R_k} = 1/\sigma^2_{R_k}\}$ of the random effects and in the prior (11) for the residual precision τ, such that $E(\tau_{R_k}) = \alpha_R/\beta_R$. The default is 0.01.

-b B The second parameter, β, in the gamma prior (10) for the precisions of the random effects and in the prior (11) for the residual precision τ, such that $E(\tau_{R_k}) = \alpha_R/\beta_R$. The default is 0.01.

-f prec

The precision $\tau_F = 1/\sigma^2_F$ of the Gaussian priors (8) of the fixed effects. The default is 0, which corresponds to replacing (8) by a uniform prior $\pi(\boldsymbol{\beta}_j) \propto 1$.

`-c` Constrain the sum of the random effects corresponding to each random factor to sum to zero.

`-v` Verbose output, including a summary of the model specification.

`-l` Log-transform the data read into the program

`-m dump_mean`
Dump the mean of the random effects for each random factor at every `dump_mean` iteration, the default is never (0). The values are dumped on a file named `FILE`.mean, where `FILE` equals the name of the input data file (the argument of the -d option) without suffix. If a file of that name already exists, the mean values are appended to that file. Each row of the file is on the format

$$q \quad M_1^{(q)} \quad M_2^{(q)} \quad \cdots \quad M_m^{(q)}$$

where q is the iteration index, m is the total number of factors and

$$M_j^{(q)} = \begin{cases} \text{mean of the effects of factor } j\text{, at iteration } q, & \text{if factor } j \text{ is random} \\ 0, & \text{if factor } j \text{ is fixed} \end{cases}$$

The factors are numbered according to the column index in the input data file. See also the option `-e`.

`-w` Print warning messages.

`-V` Show the version of the program.

`-T niter`
Run a CPU test on `niter` iterations.

`-h` Show help on `glme`.

4.2 Input data file

The input data file contains the specification of the fixed and random effects and the measurements for each observation. The input data file should be on the format

$$\begin{array}{ccccc} n & m & & & \\ f_{11} & f_{12} & \cdots & f_{1m} & y_1 \\ f_{21} & f_{22} & \cdots & f_{2m} & y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ f_{i1} & f_{i2} & \cdots & f_{im} & y_i \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ f_{2n} & f_{2n} & \cdots & f_{2m} & y_n \end{array} \qquad (19)$$

where

n = the number of observations,

m = the number of factors (fixed and random),

f_{ij} = the level (fixed factor) or realisation index (random factor) of factor j for observation i (see more detailed explanation below)

y_i = response value (typically log-intensity or log-ratio), for observation i.

Here, we use a common notation for fixed and random effects, denoting by f_{ij} the level j of a fixed factor or the realisation from the population of levels of a random factor, for observation i. The response values $\{y_i\}$ can be original or pre-processed data. As pointed out in Section 2 some normalisation might be necessary even if array effects are included in the model.

Specification of fixed and random effects

The number of levels of a fixed factor or realisations of a random factor, is evaluated from the information on the input file. The effects of a factor F_j (using the common term F_j to refer to a fixed or random factor) is interpreted as fixed if the number of distinct levels of F_j, other than 0, is one *and* this level is coded by 1, and otherwise it is interpreted as random. For both types of effects, a value of 0 means "no effect". The specification of fixed and random effects can be summarised as follows.

- *Fixed effects:*
  - The effect of a factor F_j is interpreted as fixed if there is only one distinct level, other than 0, among the $f_{ij},\ i = 1, \ldots, n$. This level should be coded 1, otherwise the effect is interpreted as random.
  - A value of 0 means no effect.
  - To include a fixed factor with more than one level, other than 0, define one factor per level.
  - An intercept is specified by a column of 1's in the data file.

  In terms of the matrix formulation (1) of the LME model, the coding of the fixed effects corresponds to the columns of the design matrix $\boldsymbol{X}$. See also Example 2 in Section 5.2 for a discussion of identifiability of the fixed effects parameters of the model.

- *Random effects:*
  - The realisations of a random factor should be represented by integer indices different from 0. The indices should be coded successively by 1,2,3,. . . , such that if factor j is random, $\max_i(f_{ij})$ in the input file (19) equals the number of distinct realisations.
  - The effects of a factor are interpreted as random if the number of distinct realisations, with code other than 0, is greater than 1, or if there is only one realisation, but with code different from 1.
  - Preceding the index of the realisation with a minus sign indicates that the effect of the realisation is equal to the negative of the corresponding positive realisation. For example, $f_{ij} = -2$ means that for observation i, the effect of F_j is the negative of that of realisation number 2. The negative and positive of a realisation are counted as *one* distinct realisation.
  - A value of 0 means no effect.

Examples of input data files are given in Section 5.

4.3 Sum-to-zero constraints for random effects

Even though in principle all parameters should be identifiable taking a fully Bayesian approach, near non-identifiability might in practice lead to slow convergence of the Gibbs sampler. Therefore, the random effects can be constrained to sum to zero by invoking the command line option `-c` as described in Section 4.1. This will constrain the random effects for *each* random factor to sum to zero. The constraints are treated correctly in the program, that is, all random effects corresponding to a random factor are sampled from their joint density conditionally on the sum-to-zero constraint.

For fixed effects, non-identifiability can be handled by re-parameterisation, see Section 5.2 for an example.

4.4 Program output

At each iteration, the current values of the unknown parameters are printed to standard output, which can be directed to an output file. Optionally, thinning can be invoked, such that for example only every 10th or 100th iteration is printed. Each row of the output is of the format

$$\text{iter} \quad q \quad X_1^{(q)} \quad X_2^{(q)} \quad \cdots \quad X_m^{(q)} \quad \tau^{(q)}$$

where

$$\begin{aligned} q &= \text{iteration index}, \\ X_j^{(q)} &= \begin{cases} \beta_j^{(q)} \text{ (current update of effect } j), & \text{if effect } j \text{ is fixed} \\ \tau_j^{(q)} \text{ (current update of the precision of effect } j), & \text{if effect } j \text{ is random}, \end{cases} \\ \tau^{(q)} &= \text{current update of the residual precision } (1/(\sigma^2)^{(q)}). \end{aligned}$$

Invoking the options `-e` and `-m`, the traces of the random effects corresponding to (a subset of) the random factors and of the corresponding mean values over the random effects for each factor can be dumped to output files, as described in Section 4.1.

To study the performance of the Gibbs sampler and compute posterior estimates of the parameters, the program output has to be post-processed. A useful tool for convergence assessment and computation of summary statistics is the `R/S-Plus` package `CODA` (Best, Cowles and Vines, 1995), purpose built for processing output from the `BUGS` program (Spiegelhalter, Thomas and Gilks, 1996), but also suitable for processing Gibbs sampling output from other programs.

5 Examples

5.1 Example 1

In our first example, we consider a special case of model (13) with $J = 0$, $K = 4$ and $N = 2$, that is, there are no fixed effects other than an overall mean, one set of random effects associated with four realisations from the levels of a random factor and two replications for each realisation. Thus, the model can be written

$$y_{kr} = \mu + b_k + \epsilon_{kr}, \tag{20}$$

for $k = 1, 2, 3, 4$ and $r = 1, 2$. For this model, the input data file `input-ex1.dat` is

```
8 2
1 1 -1.853988
1 1 -1.129544
1 2  0.9037422
1 2 -1.124410
1 3 -0.7398462
1 3 -0.6665096
1 4 -0.3317344
1 4  2.607377
```

where the rightmost column is simulated data imitating log-ratios $y_{kr},\ k = 1, 2, 3, 4,\ r = 1, 2$ of intensities from a microarray experiment. The first column represents the overall mean and the second column represents the four random effects, coded 1,2,3 and 4. We run the the Gibbs sampler using default settings by the command

```
glme -d input-ex1.dat
```

and obtain the following first 10 lines of output

```
iter   1 -0.6894353 1.1094635 0.92486334
iter   2 0.12125956 0.19953019 0.5604369
iter   3 -0.16457997 0.060155064 0.9493721
iter   4 -0.12441083 0.2435661 0.4064641
iter   5 -1.0425286 0.50105595 0.84835486
iter   6 -1.6630148 0.63185392 0.20803519
iter   7 -1.5694546 2.8588174 0.4980522
iter   8 -0.29239713 0.77783937 0.06623933
iter   9 1.0584113 0.4469442 0.62540024
iter   10 0.61644837 3.1687189 0.81821507
```

Here, the second column is the iteration index, and then follow the sampled values of the overall mean μ, the precision $\tau_R = 1/\sigma_R^2$ of the random effects and the residual precision $\tau = 1/\sigma^2$.

To illustrate the save and restore options, we first run two iterations of the Gibbs sampler by

```
glme -d input-ex1.dat -s 123 -n 2 -S ex1-save.dat
```

using seed `123` and saving the final state of the program in `ex1-save.dat`. The output from this calling sequence is

```
iter   1 -0.37728822 5.7518639 0.64558984
iter   2 -0.41119993 5.3394122 0.63696234
```

We then run the program invoking the `-R` option to start the program in the final state of the first run. Two additional iterations of the Gibbs sampler are generated by

```
glme -R ex1-save.dat -n 4
```

Observe that the number of iterations, specified by the `-n` option, is four, and thus includes the two iterations from the run that is restored. The output from this second run is

```
iter   3 -1.3770948 0.95671158 0.51358401
iter   4  0.1281061 0.79487455 0.75978085
```

The results are identical to the ones obtained by running four iterations invoking the program once by

```
glme -d input-ex1.dat -s 123 -n 4
```

The output from this run is

```
iter   1 -0.37728822 5.7518639 0.64558984
iter   2 -0.41119993 5.3394122 0.63696234
iter   3 -1.3770948 0.95671158 0.51358401
iter   4  0.1281061 0.79487455 0.75978085
```

5.2 Example 2

Consider a model of the form (13) with $J = 2$, $K = 4$ and $N = 2$. As in example 1, we use simulated data imitating log-ratios of intensities. Using non-informative priors on the overall mean μ and the fixed effects parameters β_1 and β_2, the model is over-parameterised. We see that the sums $\mu + \beta_k,\ k = 1, 2$ do not change if the same amount is added to μ and subtracted from β_1 and β_2. The problem can be solved by re-parameterising the model, and we illustrate two alternative re-parameterisations:

1. Replace (μ, β_1, β_2) by (μ', α), where $\mu' = \mu + \beta_1$ and $\alpha = \beta_2 - \beta_1$.

2. Replace (μ, β_1, β_2) by (α_1, α_2), where $\alpha_1 = \mu + \beta_1$ and $\alpha_2 = \mu + \beta_2$.

The parameters of the two models are related by $\mu' = \alpha_1$ and $\mu' + \alpha = \alpha_2$. The input data files for Model 1 (left) and Model 2 (right) are

```
1 0 1 -0.758368            1 0 1 -0.758368
1 1 1  1.128266            0 1 1  1.128266
1 0 1  0.01157952          1 0 1  0.01157952
1 1 1  0.9121233           0 1 1  0.9121233
1 0 2 -1.026323            1 0 2 -1.026323
1 1 2 -1.699259            0 1 2 -1.699259
1 0 2 -1.197100            1 0 2 -1.197100
1 1 2 -1.322309            0 1 2 -1.322309
1 0 3 -0.1225574           1 0 3 -0.1225574
1 1 3  1.449209            0 1 3  1.449209
1 0 3 -0.4012761           1 0 3 -0.4012761
1 1 3  1.846157            0 1 3  1.846157
1 0 4 -1.825652            1 0 4 -1.825652
1 1 4 -0.6654628           0 1 4 -0.6654628
1 0 4 -0.794649            1 0 4 -0.794649
1 1 4 -0.558287            0 1 4 -0.558287
```

We run the program for 10000 iterations by the command

```
glme -d input-ex2.dat -n 10000 -a 0.1 -b 0.1 -c > output-ex2.txt
```

using a Gamma$(0.1, 0.1)$-prior for τ_R and constraining the random effects to sum to zero. In the two leftmost columns of Figure 1 we show trace plots of the Gibbs sampling updates for the parameters μ', $\mu' + \alpha$, $\tau_R = 1/\sigma_R^2$ and $\tau = 1/\sigma^2$ for Model 1 and α_1, α_2, τ_R and τ for Model 2. The plots indicate that the algorithm converges fast, but the precision estimates for this example with only 4 realisations of the random effect and only 2 replicates, are highly variable. From Figure 1 and Table 1 we observe that the two parameterisations lead to similar estimates, as expected.

Model 1	$\widehat{\mu'} = -0.771$	$\widehat{\mu' + \alpha} = 0.139$	$\widehat{\alpha} = 0.908$	$\widehat{\tau_R} = 1.138$	$\widehat{\tau} = 2.50$
Model 2	$\widehat{\alpha_1} = -0.766$	$\widehat{\alpha_2} = 0.139$		$\widehat{\tau_R} = 1.004$	$\widehat{\tau} = 2.45$
True values	$\alpha_1 = -0.5$	$\alpha_2 = 0.5$	$\alpha = 1.0$	$\tau_R = 1.0$	$\tau = 4.0$

Table 1: Estimated posterior medians for Example 2.

6 Acknowledgements

Turid Follestad was supported by the cross-disciplinary research project BIOEMIT at the Norwegian University of Science and Technology.

References

Best, N. G., Cowles, M. K. and Vines, S. K. (1995). CODA Convergence Diagnosis and Output Analysis software for Gibbs sampler output: Version 0.3, *Technical report*, MRC Biostatistics Unit, Cambridge, UK.

Cui, X. and Churchill, G. A. (2003). Statistical tests for differential expression in cDNA microarray experiments, *Genome Biology* **4**(4): 210.1–210.10.

Cui, X., Kerr, M. K. and Churchill, G. A. (2002). Data transformations for cDNA microarray data, Submitted.

Efron, B., Tibshirani, R., Storey, J. D. and Tusher, V. (2001). Empirical Bayes analysis of a microarray experiment, *Journal of the American Statistical Association* **96**: 1151–1160.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*, Chapman & Hall, London, UK.

Huber, W., von Heydebreck, A., Sültmann, H. Poustka, A. and Vingron, M. (2002). Variance stabilization applied to microarray data calibration and to the quantification of differential expression, **1**(1): 1–9.

Jin, W., Riley, R. M., Wolfinger, R. D., White, K. P., Passador-Gurgel, G. and Gibson, G. (2001). The contributions of sex, genotype and age to transcriptional variance in Drosophila melanogaster, *Nature Genetics* **29**: 389–395.

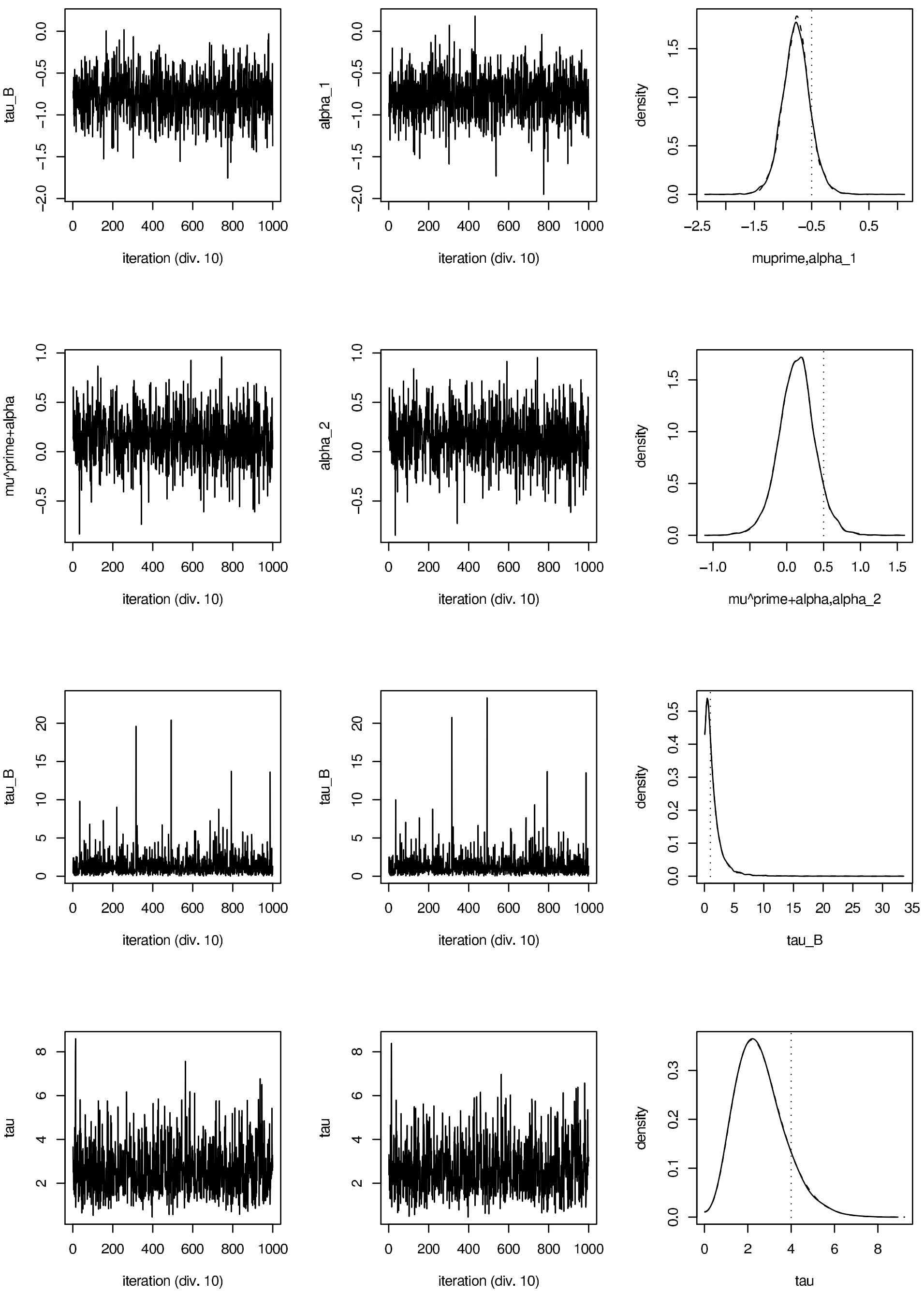

Figure 1: Trace plots for Model 1 (left) and Model 2 (middle), and density plots (right) for Model 1 (full lines) and Model 2 (dotted lines) for Example 2. The trace plots display every 10th of 10000 iterations.

Kerr, M. K., Afshari, C. A., Bennett, L., Bushel, P., Martinez, J., Walker, N. J. and Churchill, G. A. (2002). Statistical analysis of a gene expression microarray experiment with replication, *Statistica Sinica* **12**: 203–217.

Kerr, M. K., Martin, M. and Churchill, G. A. (2000). Analysis of variance for gene expression microarray data, *Journal of Computational Biology* **7**(6): 819–837.

Lönnstedt, I. and Speed, T. (2002). Replicated microarray data, *Statistica Sinica* **12**: 31–46.

McCulloch, C. E. and Searle, S. R. (2001). *Generalized, Linear and Mixed Models*, Wiley Series in Probability and Statistics, Wiley, New York.

Nygaard, V., Løland, A., Holden, M., Langaas, M., Rue, H., Liu, F., Myklebost, O., Fodstad, Ø., Hovig, E. and Smith-Sørensen, B. (2003). Effects of mRNA amplification on gene expression ratios in cDNA experiments estimated by analysis of variance, *BMC Genomics*. URL `http://www.biomedcentral.com/1471-2164/4/11`.

Pinheiro, J. and Bates, D. M. (2001). *Mixed Effects Models in S and S-PLUS*, Springer-Verlag, New York.

Smyth, G. K. (2003a). limma:linear models for microarray data. User's guide, *Technical report*, Walter and Eliza Hall Institute of Medical Research, Melbourne, Australia. `http://bioinf.wehi.edu.au/limma/`.

Smyth, G. K. (2003b). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments, *Technical report*, Walter and Eliza Hall Institute of Medical Research, Melbourne, Australia.

Spiegelhalter, D. J., Thomas, A. Best, N. and Gilks, W. R. (1996). BUGS: Bayesian inference using Gibbs sampling, Version 0.5, (version ii) ., *Technical report*, MRC Biostatistics Unit, Cambridge, UK.

Wolfinger, R. D., Gibson, G., Wolfinger, E. D., Bennett, L., Hamadeh, H., Bushel, P., Ashfari, C. and Paules, R. S. (2001). Assessing gene significance from cDNA microarray expression data via mixed models, *Journal of Computational Biology* **8**(6): 625–637.

Wu, H., Kerr, K. M., Cui, X. and Churchill, G. A. (2003). MAANOVA: A software package for the analysis of spotted cDNA microarray experiments, *in* G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger (eds), *The Analysis of Gene Expression Data. Methods and Software*, Statistics for Biology and Health, Springer-Verlag, New York, pp. 313–341.

Yang, Y. H., Dudoit, S., Luu, P., Lin, D. M., Peng, V., Ngai, J. and Speed, T. P. (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation, *Nucleic Acids Research*.