Journal of Statistical Software

August 2021, Volume 99, Issue 3. doi: 10.18637/jss.v099.i03

cold: An R Package for the Analysis of Count Longitudinal Data

M. Helena Gonçalves
Universidade do Algarve

M. Salomé Cabral
Universidade de Lisboa

Abstract

This paper describes the R package **cold** for the analysis of count longitudinal data. In this package marginal and random effects models are considered. In both cases estimation is via maximization of the exact likelihood and serial dependence among observations is assumed to be of Markovian type and referred as the integer-valued autoregressive of order one process. For random effects models adaptive Gaussian quadrature and Monte Carlo methods are used to compute integrals whose dimension depends on the structure of random effects. **cold** is written partly in R language, partly in Fortran 77, interfaced through R and is built following the S4 formulation of R methods.

Keywords: count longitudinal data, exact likelihood, Markov chain, marginal models, random effects models.

1. Introduction

In longitudinal studies, the natural experimental unit is the sequence of repeated responses in an individual subject or "cluster". Several approaches have been proposed for analyzing count longitudinal data. Zeger (1988) proposed an estimation equation approach for regression analysis with a time series of counts analogous to the generalized estimation equations (GEE) developed by Liang and Zeger (1986). Thall and Vail (1990) considered a mixed effects approach for modelling longitudinal count data with overdispersion, which may be viewed as an extension of the method proposed by Liang and Zeger (1986). Jowaheer and Sutradhar (2002) used generalized estimating equations to model longitudinal count data with overdispersion.

Breslow and Clayton (1993) extended the generalized linear models to longitudinal discrete data by incorporating random effects in the linear predictor. They used random effects to introduce correlation among observations within a cluster. These models, designated by

generalized linear mixed models (GLMMs), can model overdispersion and correlation usually present in these kind of data. A review of GLMMs with normal random effects is given by Molenberghs, Verbeke, and Demetrio (2007).

Azzalini (1994) proposed an approach where he made use of the idea of a discrete self-decomposable probability distribution following Steutel and Harn (1979) and used a non-negative sequence of integer variables modelled by the so-called integer-valued autoregressive (INAR) process introduced by McKenzie (1985) and Al-Osh and Alzaid (1987) where inference is based on likelihood approach.

Some of these approaches are available in a variety of softwares. In R (R Core Team 2021) the packages **gee** (Carey 2019), **geeM** (McDaniel, Henderson, and Rathouz 2013) and **geepack** (Højsgaard, Halekoh, and Yan 2006), implement the basic approach of GEE and some extensions. For fitting GLMMs, R has available, at least, the packages **lme4** (Bates, Mächler, Bolker, and Walker 2015) and **glmmML** (Broström and Holmberg 2018). The method proposed by Azzalini (1994) is implemented in S-PLUS described by Azzalini and Chiogna (1997).

Gonçalves (2002) and Gonçalves, Cabral, de Villa, Escrich, and Solanas (2007) developed the model proposed by Azzalini (1994) for Poisson response variables, where the serial dependence is assumed to be of Markovian type, through the inclusion of a random term in the linear predictor. Maximum likelihood estimation is used in both marginal and random effects models.

The present work extends the previous allowing the introduction of a second random effect in the linear predictor. For random effects model, likelihood estimation involves the integration over the random effects distributions that can be computational challenging (Fitzmaurice, Davidian, Verbeke, and Molenberghs 2009). Adaptive Gaussian quadrature and Monte Carlo methods are used to compute integrals whose dimension depends on the structure of random effects. Missing values are allowed on the response, provided they are missing at random (MAR) in the terminology of Little and Rubin (1987).

In this paper we describe the R package **cold** (Gonçalves and Cabral 2021) for the analysis of count longitudinal data where the aforementioned approach is considered. This package is available from the Comprehensive R Archive Network at `https://CRAN.R-project.org/package=cold`. Two features of **cold** distinguish it from other implementations: (i) maximum likelihood estimation is used when the marginal model is considered; (ii) serial dependence AR(1) can be incorporated in the random effects model allowing dependence between repeated measures in terms of numerical analysis which is ignored in the traditional approach (GLMM) implemented in the **lme4** and **glmmML**.

The paper is organized as follows. Section 2 provides the minimal theoretical background, Section 3 presents the main function and discuss its most important arguments. Section 4 introduces to the practical use of the software by analyzing two real datasets. Section 5 gives some guidelines of the `integration` approach in **cold** package. Section 6 compares **cold** package with **lme4** and **glmmML**. The paper is closed by some final remarks in Section 7.

The functions of **cold** have been written in R language, with some Fortran 77 routines which are interfaced through R. The package is built following the S4 formulation of R methods. The computational infrastructure of **cold** is similar to the one followed for **bild** package (Gonçalves, Cabral, and Azzalini 2012, 2020) but the technical development of the methodology is somewhat different in what concerns the serial dependence construction.

2. Parametric models for count longitudinal data

2.1. Introduction

In this section we consider marginal and random effects generalized linear models for longitudinal data. These two classes of models have different targets of inference; with marginal models the main focus is on inferences about the population while with random effects models the main focus is on the inferences about each individual. The GLMM accounts the correlation among the repeated observations of the same subject by the inclusion of random effects in the linear predictor. However, in GLMM it is assumed that the observations of the same subject are independent conditional to the random effects and covariates which may be not true. To overcome this problem the serial dependence is assumed to be of Markovian type and considered as the basic stochastic mechanism. For the marginal model the same stochastic mechanism is used to accommodate serial dependence.

2.2. Marginal model

Let y_{it} $(t = 1, \ldots, T_i)$, be the response value at time t from subject i $(i = 1, \ldots, n)$, and Y_{it}, its generating random variable which has a Poisson distribution whose mean value is $\mathsf{E}(Y_{it}) = \theta_{it}$. Associated with each observation time and each subject, a set of p covariates, x_{it}, is available.

The Poisson regression which links the covariates and the probability distribution of the response, is given by

$$\ln\{\mathsf{E}(Y_{it})\} = \ln(\theta_{it}) = x_{it}^{\top}\beta, \tag{1}$$

where β is the p dimensional vector of unknown parameters. The regression parameters β in a marginal model describe changes in the transformed population mean response vector. That is, the regression coefficients β describe the effects of covariates on changes in the population mean over time.

We drop temporarily the subscript i to simplify notation. To allow some form of dependence among observations of the same individual in Equation 1, Azzalini (1994) proposed a model where the serial dependence is assumed to be of Markovian type. In his approach Azzalini (1994) made use of the binomial thinning operation introduced by Steutel and Harn (1979) that can be summarized as follows. Assuming that $\rho \in (0, 1)$ and that W is a random variable taking values of the non-negative integers, $\rho \circ W$ is defined as

$$\rho \circ W = \sum_{h=1}^{W} Z_h, \tag{2}$$

where $Z_1, Z_2, \ldots$ is a sequence of independent Bernoulli variables with common probability of success ρ, $\mathsf{P}(Z_h = 1) = 1 - \mathsf{P}(Z_h = 0) = \rho$. From the definition of the thinning operator it is clear that $\rho \circ \omega \in \mathbb{N}_0$ with $1 \circ \omega = \omega, 0 \circ \omega = 0$ and $\mathsf{E}(\rho \circ \omega) = \rho(\mathsf{E}\omega)$ as in scalar multiplication. Consider the probability model introduced by McKenzie (1985) and Al-Osh and Alzaid (1987) and referred as the integer-valued autoregressive of order one (INAR(1)) process. The INAR(1) process is given by

$$Y_t = \rho \circ Y_{t-1} + \varepsilon_t, \qquad t = 2, 3, \ldots, \tag{3}$$

where $\rho \in (0,1)$ and ε_t is a sequence of independent non-negative and identically distributed integer-valued random variables with mean μ and variance σ^2. An INAR(1) process is a homogeneous Markov chain with transition probabilities given by

$$\mathsf{P}(Y_t = j|Y_{t-1} = i) = \sum_{k=0}^{\min(i,j)} \binom{i}{k} \rho^k (1-\rho)^{i-k} \times \mathsf{P}(\varepsilon_t = j-k),$$

which allows for several types of marginal distributions including the Poisson distribution.

In our work we consider the probability model given by Equation 3, for any given t, $\mathsf{E}(Y_t) = \theta_t > 0$ assuming that $\mathsf{E}(Y_1) = \theta_1$, where $\rho \circ Y_{t-1}$ is defined by Equation 2 and ε_t is a Poisson random disturbance. From Equation 3, the Poisson random variable ε_t has mean

$$\mathsf{E}(\varepsilon_t) = v_t = \theta_t - \rho\theta_{t-1}, \qquad t = 2, 3, \ldots, \tag{4}$$

where the mean values θ_t is related to the covariates by a logarithmic link function as in Equation 1, leading $\theta_t = \exp(x_t^\top \beta)$, we need that $\mathsf{E}(\varepsilon_t) = v_t > 0$, which is one peculiar feature of this model, see Gonçalves *et al.* (2007) for details. In addition, negative autocorrelations are not allowed.

The process for discrete random variables $Y_1, Y_2, \ldots, Y_T$ satisfying an equation of the form as the one in Equation 3 is an INAR(1) and so a Markov chain, and the matrix of transition probabilities can be constructed. For every instant in time t, the response Y_t is the sum of two independent random variables; one of which has the Poisson distribution with the expected value equal to v_t, and the other has binomial distribution with probability of success ρ. Following McKenzie (1985), Azzalini (1986) and McKenzie (1988) we have

$$\mathsf{P}(Y_t = j|Y_{t-1} = i) = \sum_{k=0}^{\min(i,j)} \binom{i}{k} \rho^k (1-\rho)^{i-k} \frac{\exp(-v_t) v_t^{j-k}}{(j-k)!}. \tag{5}$$

It is possible to generalize the above procedure in order to obtain the m-steps transition matrix of Y_t regarded as a Markov chain. Azzalini (1994) wrote

$$Y_t = \rho^m \circ Y_{t-m} + \varepsilon_t^{(m)}, \qquad t = m+1, m+2, \ldots, \tag{6}$$

where

$$\varepsilon_t^{(m)} = \varepsilon_t + \rho \circ \varepsilon_{t-1} + \rho^2 \circ \varepsilon_{t-2} + \ldots + \rho^{m-1} \circ \varepsilon_{t-(m-1)},$$

whose distribution is Poisson with mean

$$v_{t,m} = \theta_t - \rho^m \theta_{t-m}.$$

The m-step transition probabilities are given by

$$\mathsf{P}(Y_t = j|Y_{t-m} = i) = \sum_{k=0}^{\min(i,j)} \binom{i}{k} \rho^{mk} (1-\rho^m)^{i-k} \frac{\exp(-v_{t,m}) v_{t,m}^{j-k}}{(j-k)!}. \tag{7}$$

For the stationary case we have $corr(Y_{t-m}, Y_t) = \rho^m$. Full details to obtain Equation 7 are available in Gonçalves (2002).

The contribution from the subject i to the likelihood for the parameters (β, ρ), based on a sequence of observed data $y_{i1}, \ldots, y_{iT}$ of non-negative integer variables, possibly with some missing data, generated by model in Equation 3 with the condition Equation 4 greater than zero satisfied ($\mathsf{E}(\varepsilon_t) = v_t > 0$), is given by

$$L_i(\beta, \rho) = \frac{\exp(-\theta_1)\theta_1^{y_1}}{y_1!} \prod_{t=m+1}^{T} \mathsf{P}(Y_t = y_t | Y_{t-m} = y_{t-m}), \tag{8}$$

where the terms of the product are computed using Equation 7 with the indices t and $t - m$ referring to the non-missing observations.

The overall log-likelihood function is obtained as the sum of the n logarithmic individual contributions of the type given in Equation 8.

2.3. Random effects model

When the goal is to make inferences on individuals, the GLMMs are used. In these models, random effects are added to the linear predictor to account correlation among repeated observations of the same subject and carry information about how individuals vary in their longitudinal profiles. These models often called *subject-specific* models are also known as "conditional models" because they represent the transformed conditional mean of response Y, given the subject-specific random effect b_i, as a function of the covariates.

This can be formulated by adding a $q \times 1$ vector b_i of random effects in Equation 1 associated to a $q \times 1$ vector of covariates, z_{it}, (in general a subset of x_{it}),

$$\ln\{\mathsf{E}(Y_{it}|b_i)\} = \ln(\theta_{it}^b) = x_{it}^\top \beta + z_{it}^\top b_i, \qquad i = 1, \ldots, n, \tag{9}$$

where the b_i's are assumed to be sampled independently from each other and to have a multivariate normal distribution, with zero mean and a $q \times q$ covariance matrix, G (Fitzmaurice, Laird, and Ware 2004). The regression coefficients β, corresponding to the fixed effects regression parameters, have *subject-specific* interpretations in terms of changes in the transformed mean response for any individual.

When the vector b_i is reduced to a single ($q = 1$) random effect $b_i \sim N(0, \sigma^2)$ and $z_{it} = 1$ for all $i = 1, \ldots, n; t = 1, \ldots, T_i$. The model is referred as random intercept model and has been developed by Gonçalves (2002) and Gonçalves *et al.* (2007). Thus, for the random intercept model, if we consider $\beta_0^{b_i} = \beta_0 + b_i$, the model in Equation 9 can be expressed by

$$\ln(\theta_{it}^b) = x_{it}^\top \beta^{b_i}, \tag{10}$$

where β^{b_i} is a p dimensional vector of parameters like β, but where the first component is now $\beta_0 + b_i$, instead of β_0.

We consider the reparametrization $\omega = \ln(\sigma_b^2)$ introduced for numerical convenience, i.e., the individual random effects $b_i \sim N(0, \exp(\omega))$. For the random intercept model, the contribution of the ith subject to the likelihood function for the parameters (β, ρ, ω) based on a sequence of observed data $y_{i1}, \ldots, y_{iT}$ of non-negative integer variables, possibly with some missing data, generated by the model in Equation 3 with the condition Equation 4 greater than zero satisfied, is given by

$$L_i^R(\beta, \rho, \omega) = \frac{1}{\sqrt{2\pi e^{\omega/2}}} \int_{\mathbb{R}} L_i^F(\beta^{b_i}, \rho | b_i) \exp\left(-\frac{b_i^2}{2e^\omega}\right) db_i, \tag{11}$$

where L^F indicates the likelihood for the fixed effects model that can be computed by Equation 8.

In this paper, the random effects model has been extended allowing the inclusion of a second random effect in slope, b_{1i}, independent of b_{0i}. Thus, for each subject i, in the model given by Equation 9 a 2×1 ($q=2$) vector of random effects $b_i=(b_{0i},b_{1i})^\top$ is considered. Having $z_{it}^\top = (1,t)$ and G a diagonal matrix, where $\mathsf{VAR}(b_{0i})=\sigma^2_{b_0}$, $\mathsf{VAR}(b_{1i})=\sigma^2_{b_1}$ and $\mathsf{COV}(b_{0i},b_{1i})=0$ consequently b_{0i} and b_{1i} are independent normal random variables.

The corresponding density function is

$$f(b_0,b_1)=\frac{1}{2\pi\sigma_{b_0}\sigma_{b_1}}\exp\left[-\frac{1}{2}\left(\frac{b_0^2}{\sigma^2_{b_0}}+\frac{b_1^2}{\sigma^2_{b_1}}\right)\right].$$

Each subject i, follows a model analogous to the one in Equation 10, where β^{b_i} is a $p-$vector of parameters like β, but where the two first components are now β_0+b_{0i} and β_1+b_{1i} instead of β_0 and β_1, respectively, $\beta^{b_i}=(\beta_0+b_{0i},\beta_1+b_{1i},\beta_2,\ldots,\beta_{p-1})^\top$.

Similarly to the reparametrization for the random intercept model, let us consider $\omega_0=\ln(\sigma^2_{b_0})$ and $\omega_1=\ln(\sigma^2_{b_1})$ with matrix Ω given by

$$\Omega=\begin{pmatrix}\ln(\sigma^2_{b_0}) & 0\\ 0 & \ln(\sigma^2_{b_1})\end{pmatrix}.$$

For the two-dimensional random effects model, the contribution of the ith subject to the likelihood function for the parameters (β,ρ,ω) based on a sequence of observed data $y_{i1},\ldots,y_{iT}$ of non-negative integer variables and verifying the same conditions considered for the random intercept model, is given by

$$L_i^R(\beta,\rho,\Omega)=\int_{\mathbb{R}}\int_{\mathbb{R}}L_i^F(\beta^{b_i},\lambda|b_i)\frac{1}{2\pi e^{\omega_0/2}e^{\omega_1/2}}\exp\left[-\frac{1}{2}\left(\frac{b_0^2}{e^{\omega_0}}+\frac{b_1^2}{e^{\omega_1}}\right)\right]db_{0i}db_{1i}$$

$$=\frac{1}{2\pi e^{\omega_0/2}e^{\omega_1/2}}\int_{\mathbb{R}}\int_{\mathbb{R}}\exp\left[\ell_i^F(\beta^{b_i},\lambda|b_i)-\frac{1}{2}\left(\frac{b_0^2}{e^{\omega_0}}+\frac{b_1^2}{e^{\omega_1}}\right)\right]db_{0i}db_{1i}, \qquad (12)$$

where L^F indicates the likelihood for the fixed effects model that can be computed by Equation 8, similarly to the random intercept model.

The log-likelihood based on the a sample of n individual profiles, assumed to be independent from each other is

$$\ell^R(\beta,\rho,\omega)=\sum_i\log L_i^R(\beta,\rho,\omega).$$

The likelihood integrals given in Equation 11 and Equation 12 can not be evaluated analytically. Numerical methods are required to compute the integrals whose dimension depends on the structure of the random effects. In practice the integrals in Equation 11 and in Equation 12 are computed using adaptive Gaussian quadrature that simply approximate the integral appearing in the likelihood function as a weighted sum where the known quadrature points are chosen to provide an accurate numerical approximation. To improve efficiency of the numerical optimization of the log-likelihood, it is convenient to make use of its derivatives. See Appendix A for details or Gonçalves (2002) for a full description.

In addition to estimation of β, ρ and Ω, it is of interest to obtain the predicted random effects for the ith subject that are simply "estimated" as the conditional mean of b_i given y_i and the maximum likelihood estimates $\hat{\beta}, \hat{\rho}$ and $\hat{\Omega}$, $\hat{b}_i = \mathsf{E}(b_i|y_i; \hat{\beta}, \hat{\rho}, \hat{\Omega})$.

The computation of $\mathsf{E}(b_i|y_i; \hat{\beta}, \hat{\rho}, \hat{\Omega})$ requires integration over the distribution of the random effects, b_i. To obtain estimates $\hat{b}_i$ of the individual random effects, b_i's, we followed the idea used in package **bild** and described in Gonçalves *et al.* (2012) to avoid the difficult computation of the conditional expectation of b_i. The main idea is: If the parameters β of the systematic component $\eta_i = x_{it}^{\top}\beta$ in Equation 9 were available, one could estimate b_i by fitting a simple Poisson model, separately for each individual, regarding η_i as a fixed constant. In practice one replaces β by its estimate to compute η_i, and then fits a Poisson regression model to y_i, with η_i treated as an "offset".

3. Using package cold

The package is built around its main function `cold()` which performs the fit of parametric models as described in Section 2 via likelihood method. Maximization of the log-likelihood function is performed via `optim()` function. Serial dependence and two random effects are allowed according to the stochastic model chosen. This section provides some details on the implementation of the function `cold()` and explain its technical arguments. In the following we begin by presenting the arguments used in a call to the function `cold()`:

```
cold(formula, random, data,  id = "id", time = "time", subSET, start = NULL,
  dependence = "ind", method = "BFGS", integration = "QUADPACK",
  M = 6000, control = coldControl(), integrate = coldIntegrate(),
  cublim = coldcublim(), trace = FALSE)}
```

The function has standard arguments such as `formula`, `random` to specify the random components, `data` a `data.frame` related with the arguments `time` and `id`, the possibility to restrict the analysis to a `subSET` of the `data`. The `dependence` and `integration` arguments which specify, respectively, the serial dependence of the model and the approach to evaluate the integrals of the log-likelihood function when random effects are included in the model. The `control` argument is a list of algorithmic constants for the optimizer, `integrate` and `cublim` arguments are lists of algorithmic constants for the computation of a definite integral, the default settings of these arguments were chosen based on the study presented in Section 5. The rest of this section describes the most important arguments of `cold()`.

3.1. Two-type models

For marginal models, the basic `formula` in `cold` is a description of the model to be fitted of the type `response ~ predictors`. For random effects models, the random components must be specified through the `random` argument of type `random = ~ 1` if only the random intercept is considered or type `random = ~ 1 + time` if both intercept and slope are random.

3.2. Data structure

The structure of the `data` is a `data.frame`. Each element of the `data` argument must be identifiable by a name. The response variable represents the individual profiles of each subject,

it is expected a variable in the `data` that identifies the correspondence of each component of the response variable to the subject that it belongs, by default is named `id` variable. If an `id` variable is not part of the `data.frame` the name of the variable playing the role of `id` must be declared in the arguments of `cold()` function. When it is expected that all subjects in one experiment to be observed at the same time points, but in practice some of the subjects were not observed in some of the scheduled occasions, `NA` values can then be inserted in the response variable. By default, the program expects a variable named `time` to be present in the `data.frame`, otherwise the name of the variable playing the role of `time` must be declared in the arguments of `cold()` function. The `time` variable should identify the time points that each individual profile has been observed. Its possible to restrict the analysis to a `subSET` of the `data`.

To illustrate the data structure for function `cold()` the `seizure` dataset is presented. This dataset is a longitudinal study from a clinical trial where the antiepileptic drug prograbide was compared with a placebo and will be analyzed in Section 4.

```
R> data("seizure", package = "cold")
R> head(seizure)
```

```
   id y v4 time trt base age    lbase     lage
1 104 5  0    1   0   11  31 1.011601 3.433987
2 104 3  0    2   0   11  31 1.011601 3.433987
3 104 3  0    3   0   11  31 1.011601 3.433987
4 104 3  1    4   0   11  31 1.011601 3.433987
5 106 3  0    1   0   11  30 1.011601 3.401197
6 106 5  0    2   0   11  30 1.011601 3.401197
```

For each individual (`id`), we have the outcomes that consist of their number of epileptic seizures (`y`) during the two-weeks before and each of the four clinic visits (`time`), the number of seizures in a baseline eight-week interval preceding the entry into the trial (`base`), for prograbide (`trt = 1`) and placebo (`trt = 0`) the `age` in years, a dummy variable `v4` to account for a drop in seizures counts during the fourth interval as well as the log of a quarter of the number of baseline seizures (`lbase`) and the logarithm of the age (`lage`).

3.3. Dependence structure

According to the stochastic model chosen serial dependence of first order autoregressive model and random effects are allowed. The stochastic model that assumes `independence` is also available (the default). The six options are specified via the argument `dependence` as follows: `dependence = "ind"` (independence), `dependence = "indR"` (independence with random intercept), `dependence = "indR2"` (independence with random effects in both intercept and slope), `dependence = "AR1"` (first order autoregressive), `dependence = "AR1R"` (first order autoregressive with random intercept), `dependence = "AR1R2"` (first order autoregressive with random effects in both intercept and slope).

dependence	start components	start default values
ind	—	—
indR	(ω)	(0)
indR2	$(\omega_0,\ \omega_1)$	(0, 0)
AR1	(ρ)	(0.5)
AR1R	$(\rho,\ \omega)$	(0.5, 0)
AR1R2	$(\rho,\ \omega_0,\ \omega_1)$	(0.5, 0, 0)

Table 1: Components of `start` vector and correspondent default values.

3.4. Optimization

The methodology in the optimization process for the log-likelihood is identified through `method` argument to be passed to `optim()`. Numerical optimization requires starting values for the parameters in the model. For the regression parameters, `cold()` fits the `data` using the R function `glm()` to obtain the starting values. The starting values obtained for the regression parameters are then coerced with a `start` vector whose dimension depends on the structure of stochastic model chosen, `c(glm(formula, ...)$coefficients, start)`. Table 1 presents the structure of the vector `start` and corresponding default values.

3.5. Numerical integration

Numerical integration methods have been implemented based on Fortran 77 subroutine package `QUADPACK` only for random intercept models. For both, one and two-dimensional random effects models the **cubature** (Narasimhan, Johnson, Hahn, Bouvier, and Kiêu 2020) package is an alternative to compute the integrals for the likelihood and derivatives. The `cubature` approach is expect to obtain a close approximation to the likelihood. However, this approach can be computationally intensive as the model includes two random effects and an autoregressive structure. Monte Carlo methods were also implemented as an alternative to avoid numerical integration for both one and two-dimensional random effects models, some implementation details are presented in Appendix B. For models with random intercept (`dependence = "indR"` or `dependence = "AR1R"`), by default, the `integration` argument is set to `QUADPACK`. `cold()` function also allows the use of **cubature** package or the use of Monte Carlo methods to compute the integrals by setting `integration = "cubature"` or `integration = "MC"`, respectively.

For models with two random effects (`dependence = "indR2"` or `dependence = "AR1R2"`), with a random intercept and a random slope, the user has to define the `integration` argument by setting `integration = "cubature"` or `integration = "MC"`, see Table 2.

3.6. Fitting options

The fitting options in `cold()` are set trough arguments `control`, `integrate` and `cublim`.

The argument `control` is set by a call to function `coldControl()` and returns a list of algorithmic constants for the optimizer `optim()`, via a call of the form:

```
coldControl(maxit = 100, abstol = -Inf, reltol = sqrt(.Machine$double.eps))
```

	dependence			
	One random effect		*Two random effects*	
	indR	AR1R	indR2	AR1R2
integration				
QUADPACK	default	default	—	—
cubature	✓	✓	✓	✓
MC	✓	✓	✓	✓

Table 2: Options for `integration` according the dimension of random effects.

The argument `integrate` is set by a call to function `coldIntegrate()` that is an auxiliary function for controlling `cold()` fitting when `integration = "QUADPACK"` with the following defaults:

```
coldIntegrate(li = -4, ls = 4, epsabs = .Machine$double.eps^.25,
  epsrel = .Machine$double.eps^.25, limit = 100, key = 6, lig = -4, lsg = 4)
```

For given values of `li` and `ls`, the numerical integration is performed over the interval (`li`$\times\sigma$, `ls`$\times\sigma$) to compute the integral given by Equation 11 since it is sufficient to integrate over a small region around the mean to get a reasonable approximation for the integral over the infinite range, where $\sigma = \exp(\omega/2)$ is associated to the current parameter value ω examined by the `optim()` function. Integration limits for the gradient of Equation 11 are regulated similarly by `lig` and `lsg` and are used to compute the integrals presented in Appendix A.2. For some data sets, the user could have the need to do a specification of the `integrate` argument list changing the integration limits in the `coldIntegrate()` function.

By default, **cold** implements uni-dimensional adaptive Gauss-Kronrod quadrature based on the Fortran function `DQAGE` from `QUADPACK` with local integration rule a Gauss-Kronrod pair with 30–61 points, `key = 6` (default). The argument `key` is an integer from 1 to 6 for choice of local integration rule for the number of Gauss-Kronrod quadrature points (see **cold** documentation on `coldIntegrate()` function). Some results with different number of quadrature points are presented in Section 5.1.

The argument `cublim` is set by a call to function `coldcublim()` that is an auxiliary function for controlling `cold()` fitting when `integration = "cubature"` with the defaults:

```
coldcublim(l1i = -4, l2i = -4, l1s = 4, l2s = 4, tol = 1e-4, maxEval = 100)
```

For random intercept models, `coldcublim()` allows the user control the integration limits similarly as described above for `coldIntegrate()`.

To compute two-dimensional integrals as given by Equation 12, numerical integration is performed over the intervals (`l1i`$\times\sigma_0$, `l1s`$\times\sigma_0$) and (`l2i`$\times\sigma_1$, `l2s`$\times\sigma_1$). Integration limits for the gradient given by Equation 12 are regulated also by `(l1i, l2i, l1s, l2s)` and are used to compute the integrals presented in Appendix A.3.

When the `integration` argument is set to `MC`, `cold()` performs the numerical optimization of the log-likelihood using Monte Carlo methods. The default number of iterations considered to evaluate the integral is set to `M = 6000`. In Section 5.2 are presented some results considering different values for `M`.

Generic	Description of the returned value
`anova`	Analysis deviance table for two nested fitted model objects.
`coeftest`	Prints partial Wald tests of coefficients.
`fitted`	Extract fitted values.
`fixeff`	Extract the estimates of the fixed effects coefficients.
`getAIC`	Extract the Akaike information criterion.
`getcoef`	Extract estimated coefficient parameters of the fitted model.
`getLogLik`	Extract log-likelihood.
`getvcov`	Extract the variance-covariance matrix of the fixed effect estimates.
`model.mat`	Extract the fixed effects model matrix.
`plot`	Plots for fitted models.
`randeff`	Extract individual random effects.
`resid`	Extract residuals.
`show`	Print simple summary of a '`cold`' object.
`summary`	Summary of a fitted model.
`vareff`	Extract estimated random-effects variance.

Table 3: List of available methods for objects of the class '`cold`'.

3.7. Methods for class '`cold`'

The returned fitted-model object of class '`cold`' is a list that allows the user to extract several items produced by the maximum likelihood procedure. A set of standard methods available to extract information from the fitted model is described in Table 3. Some of the functions and methods have standard syntax as in other R packages.

The function `summary()` returns a list of statistics of the fitted linear model given a `cold()` object. The `anova()` method allows to compare two nested models via a likelihood ratio test for testing the difference between model M1 and a submodel M2. If the difference between M1 and M2 is the number of random effects the null hypothesis is on the boundary and standard asymptotic results on the null distribution of the likelihood ratio test do not hold (Self and Liang 1987; Stram and Lee 1994). To calculate the p values of `anova()` we followed Pinheiro and Bates (2000) and implemented in **cold** package the naive approach of using χ^2 distribution with the number of degrees of freedom determined by the difference of the number of non-redundant parameters in the models. The `AIC` and `BIC` are also displayed.

The `plot()` method allows some graphical facilities, namely, three plots selectable by `which`. The options are: the plot for the parametric fit (`which = 1`), the plot for the individual mean profile (`which = 2`) and the plot for the observed data and the corresponding predicted values (`which = 3`). The default is `which = 1`, the options `which = 2` and `which = 3` are used only if random effects are present. For `which = 1` the user has to define the argument `factor` if it is present in the data.

For a full description of the available quantities, see the list of slots of the class '`cold`' provided with the package documentation.

4. Applications

The usage of **cold** is illustrated in this section with two real datasets. Both marginal and random effects models are used. For random effects models the `integration` approach used are in agreement with the guidelines presented in Section 5.

To start analysing the data the package must be loaded first.

```
R> library("cold")
```

4.1. Seizure data

The `seizure` dataset were described in Section 3.2 and analysed by Thall and Vail (1990), Breslow and Clayton (1993), and Diggle, Heagerty, Liang, and Zeger (2002).

Marginal model

The function `cold()` was called to fit the marginal model

$$\ln(\theta_{it}) = \beta_0 + \beta_1 \texttt{lbase}_i + \beta_2 \texttt{trt}_i + \beta_3 \texttt{lage}_i + \beta_4 \texttt{v4}_{it} + \beta_5 \texttt{trt}_i \times \texttt{lbase}_i, \qquad (13)$$

assuming an independence correlation structure (`dependence = "ind"`):

```
R> seiz_ind <- cold(y ~ lbase + trt + lage + v4 + trt:lbase, data = seizure,
+    dependence = "ind")
R> summary(seiz_ind)

Call:
cold(formula = y ~ lbase + trt  + lage + v4 + trt:lbase, data = seizure,
    dependence = "ind")

Number of profiles in the dataset:  59
Number of profiles used in the fit:  59
Dependence structure: ind
Log likelihood:  -817.6593
AIC:  1647.319

Fixed effects:
              Estimate Std. Error z value  p-value
(Intercept) -2.7575825 0.40740799  -6.769 0.000000
lbase        0.9495236 0.04355775  21.799 0.000000
trt         -1.3411185 0.15672916  -8.557 0.000000
lage         0.8970507 0.11642503   7.705 0.000000
v4          -0.1610871 0.05457554  -2.952 0.003161
lbase:trt    0.5622252 0.06349172   8.855 0.000000

Message:  0
```

The above results of `summary()` show that all the parameters are significant at 5% level. In order to investigate the presence of serial dependence between successive observations from the same subject, a first order autoregressive model (`dependence = "AR1"`) was also fitted to data and the results can be visualized with function `coeftest()`:

```
R> seiz_AR1 <- cold(y ~ lbase + trt + lage + v4 + trt:lbase, data = seizure,
+    dependence = "AR1")
R> coeftest(seiz_AR1)

              Estimate Std. Error z value  p-value
(Intercept) -2.8367109 0.48276510  -5.876 0.000000
lbase        0.9519221 0.05140583  18.518 0.000000
trt         -1.4209512 0.18812621  -7.553 0.000000
lage         0.9219487 0.13749196   6.705 0.000000
v4          -0.1541182 0.05018799  -3.071 0.002135
lbase:trt    0.5874116 0.07597932   7.731 0.000000
rho          0.2433954 0.02968839   8.198 0.000000
```

The strong significance of the estimated AR(1) coefficient, `rho`, point out to the presence of substantial dependence between successive observations from the same subject. The likelihood ratio test to compare the two nested marginal models can be performed using `anova()` via the statement:

```
R> anova(seiz_ind, seiz_AR1)

Data:  seizure

Model1:  y ~  lbase + trt + lage + v4 + trt:lbase
dependence = ind
Model2:  y ~  lbase + trt + lage + v4 + trt:lbase
dependence = AR1

               AIC      BIC    logLik  Deviance df p-value
 Model1   1647.319 1668.102 -817.6593
 Model2   1574.109 1598.356 -780.0545    75.209  1       0
```

The results obtained by `anova()` method show that the change of deviance produces a p value of 0, confirming that the AR(1) structure is significantly preferable to the independence one.

The fitted values and the Pearson residuals of the two first subjects are given, respectively, by

```
R> fitted(seiz_AR1)[1:8]

       1        2        3        4        5        6        7        8
3.640840 3.640840 3.640840 3.120821 3.532422 3.532422 3.532422 3.027889

R> resid(seiz_AR1)[1:8]
```

```
      1         2         3         4         5         6         7         8
0.712312 -0.335852 -0.335852 -0.068392 -0.283282  0.780845 -0.283282 -0.016027
```

Model `seiz_ind` (`dependence = "ind"`) corresponds to Model I of Breslow and Clayton (1993) and the results of both models are in close agreement.

Random effects models

As well as in Breslow and Clayton (1993) models with random effects were also fitted to seizure data.

Model `seiz_indR` is a random intercept model with independence structure (`dependence = "indR"`) having the same predictors used in the model in Equation 13 and is given by

$$\ln(\theta_{it}) = \beta_0 + \beta_1 \texttt{lbase}_i + \beta_2 \texttt{trt}_i + \beta_3 \texttt{lage}_i + \beta_4 \texttt{v4}_{it} + \beta_5 \texttt{trt}_i \times \texttt{lbase}_i + b_{0i}. \qquad (14)$$

This model was fitted, considering the default settings, via the statement:

```
R> seiz_indR <- cold(y ~ lbase + trt + lage + v4 + trt:lbase, random = ~ 1,
+    data = seizure,  dependence = "indR")
R> summary(seiz_indR)

Call:
cold(formula = y ~ lbase + trt + lage + v4 + trt:lbase, random = ~1,
    data = seizure, dependence = "indR")

Number of profiles in the dataset:  59
Number of profiles used in the fit:  59
Dependence structure: indR
Log likelihood:  -665.2942
AIC:  1344.588

Fixed effects:
              Estimate Std. Error z value  p-value
(Intercept) -1.3397687 1.18225779  -1.133 0.257118
lbase        0.8843186 0.13122384   6.739 0.000000
trt         -0.9341057 0.40078426  -2.331 0.019769
lage         0.4849501 0.34723238   1.397 0.162529
v4          -0.1610698 0.05457551  -2.951 0.003164
lbase:trt    0.3389061 0.20332035   1.667 0.095543

Random effects:
             Variance
(Intercept) 0.2527949

Message:  0
```

The model `seiz_indR` corresponds to the Model II of Breslow and Clayton (1993) and the estimated values of the parameters of both models, as well as, their standard errors are in close agreement.

A model with two random effects with the form:

$$\ln(\theta_{it}^b) = \beta_0 + \beta_1 \texttt{lbase}_i + \beta_2 \texttt{trt}_i + \beta_3 \texttt{lage}_i + \beta_4 \texttt{visit}_{it} + \beta_5 \texttt{trt}_i \times \texttt{lbase}_i + b_{0i} + b_{1i} \texttt{visit}_{it} \quad (15)$$

was also fitted to seizure data where the fixed part is the same as that of model `seiz_indR` except that the variable `visit`, coded `(-0.3, -0.1, 0.1, 0.3)` corresponding to the four clinic visits, was used instead of `v4` as presented in Breslow and Clayton (1993).

The variable `visit` was added to the seizure dataset:

```
R> seizure$visit <- c(-0.3, -0.1, 0.1, 0.3)
```

An independence structure for serial correlation `dependence = "indR2"` was considered. As the model involves two random effects `integration = "MC"` was setted to compute integrals using Monte Carlo methods, the default value for number of iterations (`M = 6000`) was used:

```
R> seiz_indR2 <- cold(y ~ lbase + trt + lage + visit + trt:lbase, random = ~
+    1 + visit, data = seizure, dependence = "indR2", integration = "MC")
R> summary(seiz_indR2)

Call:
cold(formula = y ~ lbase + trt + lage + visit + trt:lbase, random = ~1 +
    visit, data = seizure, dependence = "indR2", integration = "MC")

Number of profiles in the dataset:  59
Number of profiles used in the fit:  59
Dependence structure: indR2
Log likelihood:  -655.7172
AIC:  1327.434

Fixed effects:
             Estimate Std. Error z value  p-value
(Intercept) -1.2280125  1.2378957  -0.992 0.321190
lbase        0.8868239  0.1114963   7.954 0.000000
trt         -0.9370026  0.3648511  -2.568 0.010223
lage         0.4346350  0.3662833   1.187 0.235382
visit       -0.2827952  0.1563205  -1.809 0.070440
lbase:trt    0.3460256  0.1690964   2.046 0.040725

Random effects:
             Variance
(Intercept) 0.2496109
visit       0.5094467

Message:  0
```

The model `seiz_indR2`, despite having independent random effects, is comparable to Model IV of Breslow and Clayton (1993) since they estimate the correlation between the two random effects to be effectively zero, once more, the results qualitatively agree.

The individual random effects of the first four individuals can be obtained using the `randeff()` extractor function,

```
R> randeff(seiz_indR2)[1:4,]
```

```
  (Intercept)        visit
1  0.07253222 -0.581584340
2  0.10334049 -0.003183831
3  0.46004462  1.200535006
4  0.22910040 -0.179861973
```

Their predicted values can also be obtained using the `fitted()` extractor function,

```
R> fitted(seiz_indR2)[1:16]
```

```
       1        2        3        4        5        6        7        8
4.452548 3.745670 3.151015 2.650766 3.805750 3.594185 3.394381 3.205684
       9       10       11       12       13       14       15       16
2.044846 2.456826 2.951810 3.546518 3.713992 3.385752 3.086521 2.813736
```

The plot of the individual mean profile for all subjects (by default) or for a subset (`subSET = ...`) can be obtained using the `plot` methods choosing `which = 2`. The identification of the subjects is also allowed by setting `ident = TRUE`. These options are only available for random effects models. The individual mean profiles for both treatments groups given in Figure 1 can be produced by:

```
R> op <- par(mfrow = c(1, 2))
R> plot(seiz_indR2, which = 2, ident = TRUE, subSET = trt == "1",
+    xlab = "visit", ylab = "Seizure count", main = "Prograbide")
R> plot(seiz_indR2, which = 2, ident = TRUE, subSET = trt == "0",
+    xlab = "visit", ylab = "Seizure count", main = "Control")
R> par(op)
```

Using `plot` methods with `which = 3` it is possible to display, in the same plot, observed and individual mean profiles as illustrated in Figure 2 and obtained by:

```
R> op <- par(mfrow = c(2, 2))
R> plot(seiz_indR2, which = 3, subSET = (id == c(116)), xlab = "visit",
+    ylab = "Seizure count", main = "Control_Subject116")
R> plot(seiz_indR2, which = 3, subSET = (id == c(126)), xlab = "visit",
+    ylab = "Seizure count", main = "Control_Subject126")
R> plot(seiz_indR2, which = 3, subSET = (id == c(112)), xlab = "visit",
+    ylab = "Seizure count", main = "Prograbide_Subject112")
R> plot(seiz_indR2, which = 3, subSET = (id == c(225)) ,xlab = "visit",
+    ylab = "Seizure count", main = "Prograbide_Subject225")
R> par(op)
```

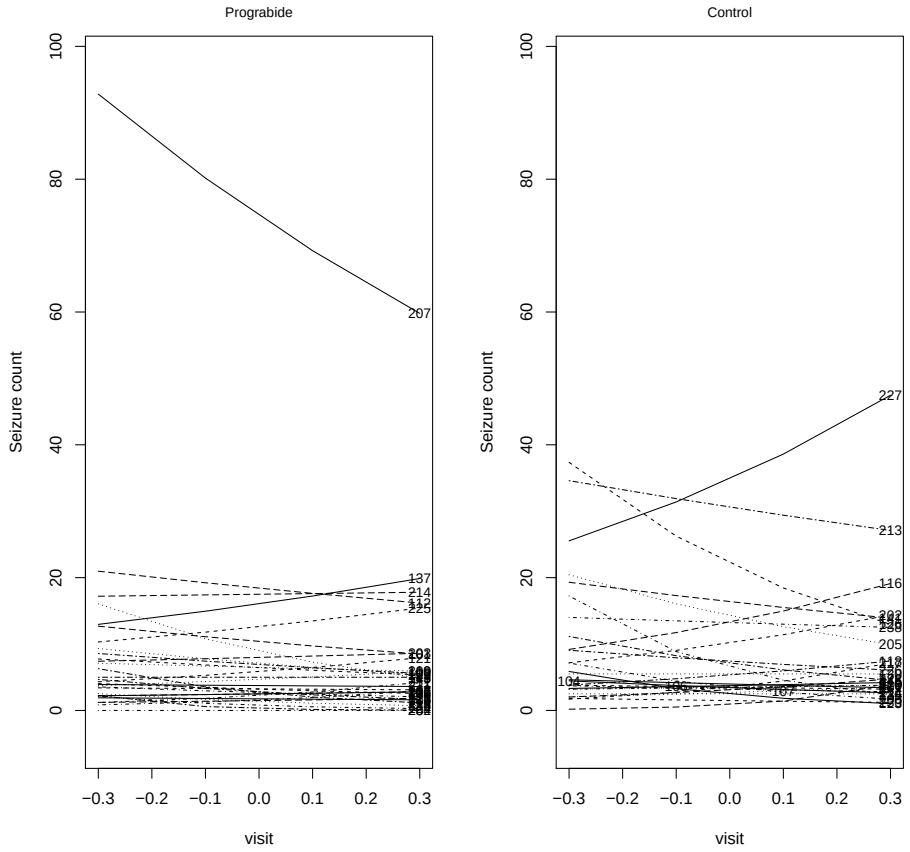

Figure 1: Individual mean profiles of subjects for `seiz_indR2`.

4.2. Bolus data

The `bolus` dataset is a longitudinal study on the number of bolus of drug that a patient request to control their own pain relief following surgery. The dataset has the number of requests per interval in 12 successive four-hourly intervals following abdominal surgery for 65 patients in a clinical trial to compare two bolus/lock-out combinations Henderson and Shimakura (2003). This dataset was also analysed by Weiss (2005). The variables present in `bolus` dataset are the patient identifier `id`, the indicator `group` whether the patient takes 1mg (`group = "1mg"`) or 2mg (`group = "2mg"`) of morphine, the observation `time` and the number of requests of analgesic doses taken by hospital patients in 12 successive four-hourly intervals `y`.

```
R> data("bolus", package = "cold")
R> str(bolus)
```

```
'data.frame':        780 obs. of  4 variables:
 $ id   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ group: Factor w/ 2 levels "1mg","2mg": 2 2 2 2 2 2 2 2 2 2 ...
 $ time : int  1 2 3 4 5 6 7 8 9 10 ...
 $ y    : int  5 2 2 5 2 4 0 2 4 4 ...
```

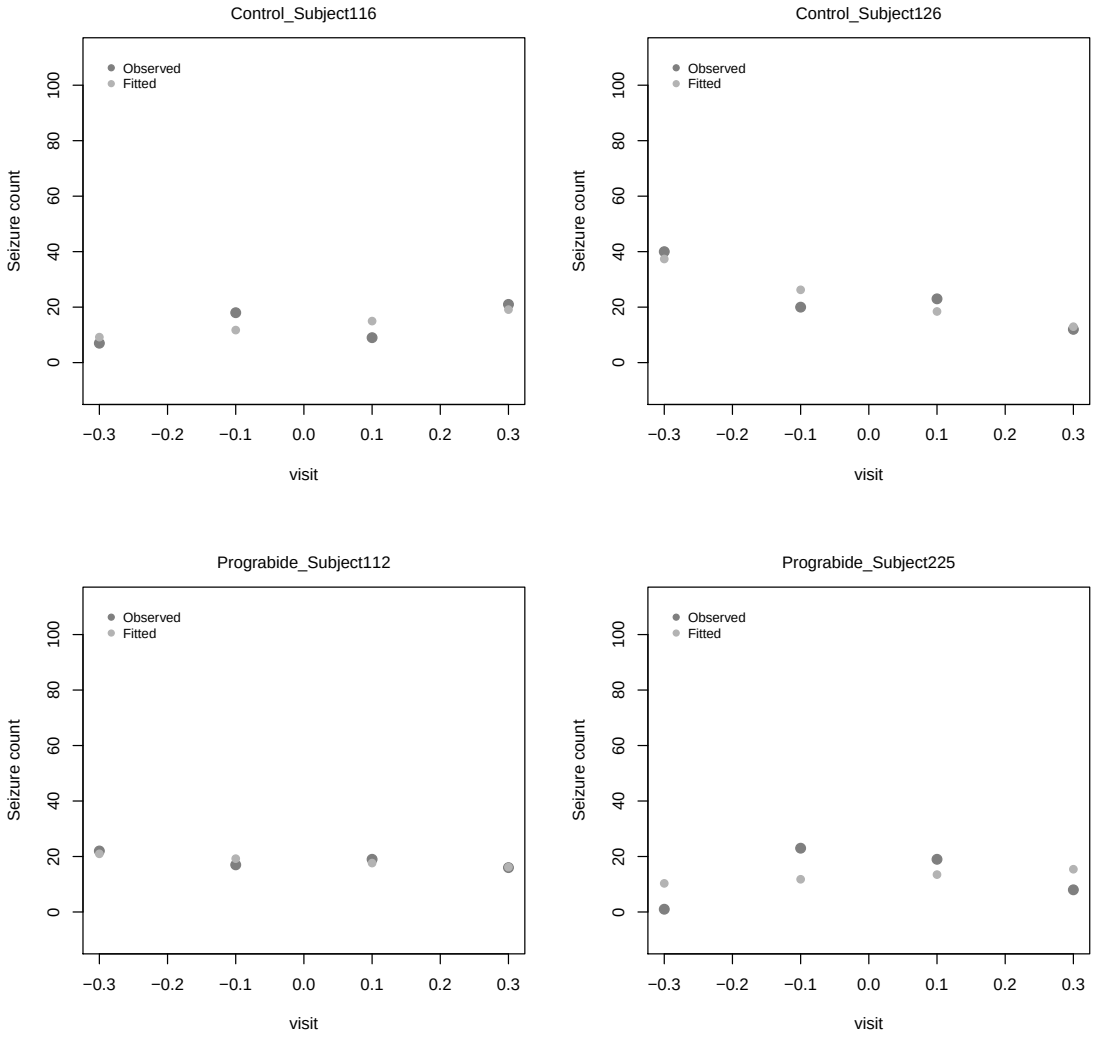

Figure 2: Observed and individual mean profiles for two subjects in each treatment for `seiz_indR2`.

We start the analysis declaring the level `2mg` of the factor `group` as the reference class:

```
R> options(contrasts = c("contr.treatment", ordered = "contr.poly"))
R> contrasts(bolus$group)
```

```
    2mg
1mg   0
2mg   1
```

```
R> bolus$group <- relevel(factor(bolus$group), ref = "2mg")
R> contrasts(bolus$group)
```

```
    1mg
2mg   0
1mg   1
```

The model with two random effects is given by

$$\ln(\theta_{it}^b) = \beta_0 + \beta_1 \texttt{group}_i + \beta_2 \texttt{time1}_i + \beta_3 \texttt{group}_i \times \texttt{time1}_i + b_{0i} + b_{1i}\texttt{time1}_i, \qquad (16)$$

and was fitted to the `bolus` data where `time1` = `time`-6, for `time` $= 1, ..., 12$, codded for numerical convenience of the integration procedure for models with random effects and added to `bolus` dataset:

```
R> bolus$time1 <- bolus$time - 6
```

For the subject specific models that involves two random effects, the fit was performed using Monte Carlo methods (`integration = "MC"`). The function `cold()` was called to fit the model presented in Equation 16 using a dependence structure `indR2` via the statement:

```
R> bol_indR2 <- cold(y ~ time1 * group,  random = ~ 1 + time1, data = bolus,
+    time = "time1", dependence = "indR2", integration = "MC")
R> summary(bol_indR2)

Call:
cold(formula = y ~ time1 * group, random = ~ 1 + time1, data = bolus,
    time = "time1", dependence = "indR2", integration = "MC")

Number of profiles in the dataset:  65
Number of profiles used in the fit:  65
Dependence structure: indR2
Log likelihood:  -2215.781
AIC:  4443.562

Fixed effects:
                  Estimate Std. Error z value  p-value
(Intercept)     1.54288238 0.08978924  17.183 0.000000
time1          -0.07602076 0.01603544  -4.741 0.000002
group1mg        0.26212841 0.12401815   2.114 0.034547
time1:group1mg  0.02031507 0.02144403   0.947 0.343459

Random effects:
               Variance
(Intercept) 0.259513219
time1       0.005908598

Message:  0
```

All the parameters estimates, except the interaction, are significant at 5% level. This shows, among other things, that time effect is present and that a difference exists between the two groups. To explore further this point, a similar model was fitted but without the interaction term via

```
R> bol_indR2a <- cold(y ~ time1 + group, random = ~ 1 + time1, data = bolus,
+    time = "time1", dependence = "indR2", integration = "MC")
```

and the `anova` function was used to compare the two models

```
R> anova(bol_indR2a, bol_indR2)

Data:  bolus

Model1:  y ~ time1 + group
dependence = indR2
Model2:  y ~ time1 * group
dependence = indR2

               AIC      BIC    logLik  Deviance df p-value
 Model1   4442.515 4465.812 -2216.258
 Model2   4443.562 4471.518 -2215.781     0.953  1  0.3289
```

Based on the results of the likelihood ratio test given by the `anova()` function (p value=0.3289) the model without interaction (`bol_indR2a`) is not rejected at the level of significance 5%.

In order to investigate if an `AR1` dependence structure is more appropriate, the additive model was fitted:

```
> bol_AR1R2a <- cold(y ~ time1 + group,  random = ~ 1 + time1, data = bolus,
+   time = "time1", dependence = "AR1R2", integration = "MC",
+   M = 12000)
> summary(bol_AR1R2a)

Call:
cold(formula = y ~ time1 + group, random = ~ 1 + time1, data = bolus,
    time = "time1", dependence = "AR1R2", integration = "MC", M = 12000)

Number of profiles in the dataset:  65
Number of profiles used in the fit:  65
Dependence structure: AR1R2
Log likelihood:  -2213.333
AIC:  4438.666

Fixed effects:
               Estimate Std. Error z value  p-value
(Intercept)  1.52422030 0.09578845  15.912 0.000000
time1       -0.06521342 0.01090329  -5.981 0.000000
group1mg     0.27364983 0.12984007   2.108 0.035066

Estimated correlation parameter:
      Estimate Std. Error z value  p-value
rho 0.04967921 0.02407635   2.063 0.039074

Random effects:
               Variance
```

```
(Intercept) 0.258340121
time1       0.005776967

Message:  0
```

All the parameters estimates are significant at 5% level. Although we have a significant `rho`, the likelihood ratio test was used to compare the two dependence structures (`indR2` and `AR1R2`) using `anova()` function:

```
R> anova(bol_indR2a, bol_AR1R2a)

Data:  bolus

Model1:  y ~ time1 + group
dependence = indR2
Model2:  y ~ time1 + group
dependence = AR1R2

               AIC      BIC    logLik  Deviance df p-value
 Model1   4442.515 4465.812 -2216.258
 Model2   4438.666 4466.622 -2213.333     5.849  1 0.01558
```

Based on the p value (0.01558), the model with `dependence = "indR2"` was rejected at 5% significance level.

Taking into account the random effect variance value of the slope in model `bol_AR1R2a`, a random intercept model `bol_AR1Ra` with the same dependence structure was fitted. The `anova()` function was used to compare both models,

```
> anova(bol_AR1Ra, bol_AR1R2a)

Data:  bolus

Model1:  y ~ time1 + group
dependence = AR1R
Model2:  y ~ time1 + group
dependence = AR1R2

               AIC      BIC    logLik  Deviance df p-value
 Model1   4583.753 4607.050 -2286.877
 Model2   4438.666 4466.622 -2213.333   147.087  1       0
```

As mentioned in Section 3.7 the p value of `anova()` is computed following the naive approach (Pinheiro and Bates (2000)). Based on the p value (0), the model with `dependence = "AR1R"` was rejected. Hence, our analysis concludes that the model `bol_AR1R2a` is preferable for the `bolus` data set.

The individual mean profiles for "typical" subjects ($b_{0i} = 0$ and $b_{1i} = 0$) for both treatments are display in Figure 3 and can be produced using `plot` method by:

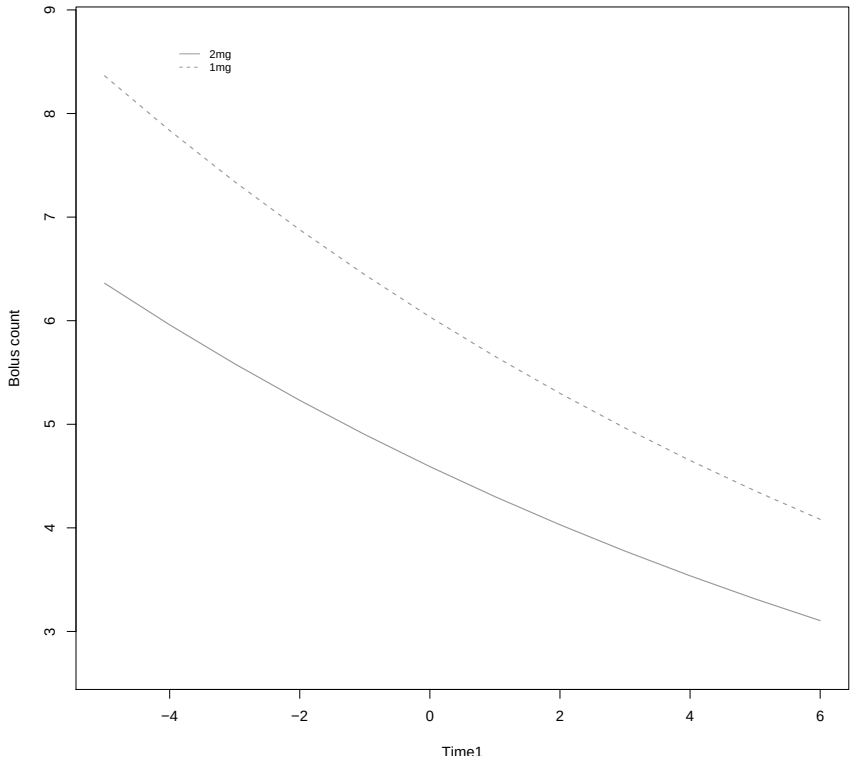

Figure 3: Individual mean profiles of "typical" subjects for `bol_AR1R2a`.

```
R> plot(bol_AR1R2a, factor = group, xlab = "Time1", ylab = "Bolus count")
```

The individual mean profiles presented in Figure 4 can be produced by:

```
R> op <- par(mfrow = c(1, 2))
R> plot(bol_AR1R2a, which = 2, ident = TRUE, subSET = group == "1mg",
+  xlab = "Time1", ylab = "Bolus count", main = "1mg")
R> plot(bol_AR1R2a, which = 2, ident = TRUE, subSET = group == "2mg",
+  xlab = "Time1", ylab = "Bolus count", main = "2mg")
R> par(op)
```

To display in the same plot observed and individual mean profiles as illustrated in Figure 5 we used the settings bellow:

```
R> op <- par(mfrow = c(2, 2))
R> plot(bol_AR1R2a, which = 3, subSET = (id == c(10)), xlab = "Time1",
+    ylab = "Bolus count", main = "2mg_Subject10")
R> plot(bol_AR1R2a, which = 3, subSET = (id == c(26)), xlab = "Time1",
+    ylab = "Bolus count", main = "2mg_Subject26")
R> plot(bol_AR1R2a, which = 3, subSET = (id == c(33)), xlab = "Time1",
+    ylab = "Bolus count", main = "1mg_Subject33")
R> plot(bol_AR1R2a, which = 3, subSET = (id == c(56)), xlab = "Time1",
+    ylab = "Bolus count", main = "1mg_Subject56")
R> par(op)
```

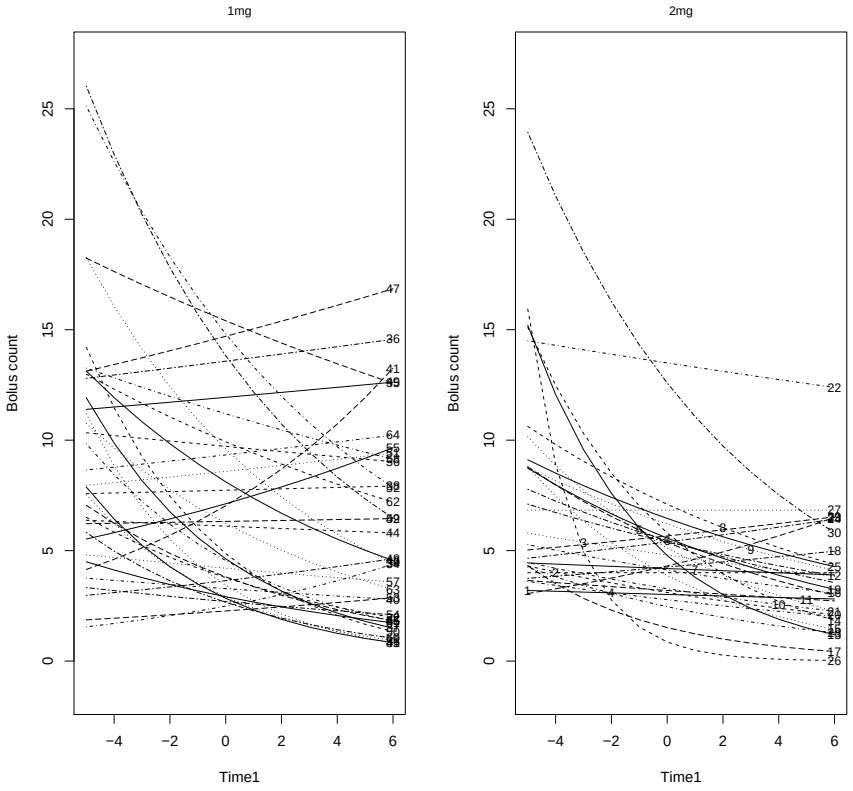

Figure 4: Individual mean profiles of `bolus` data for `bol_AR1R2a`.

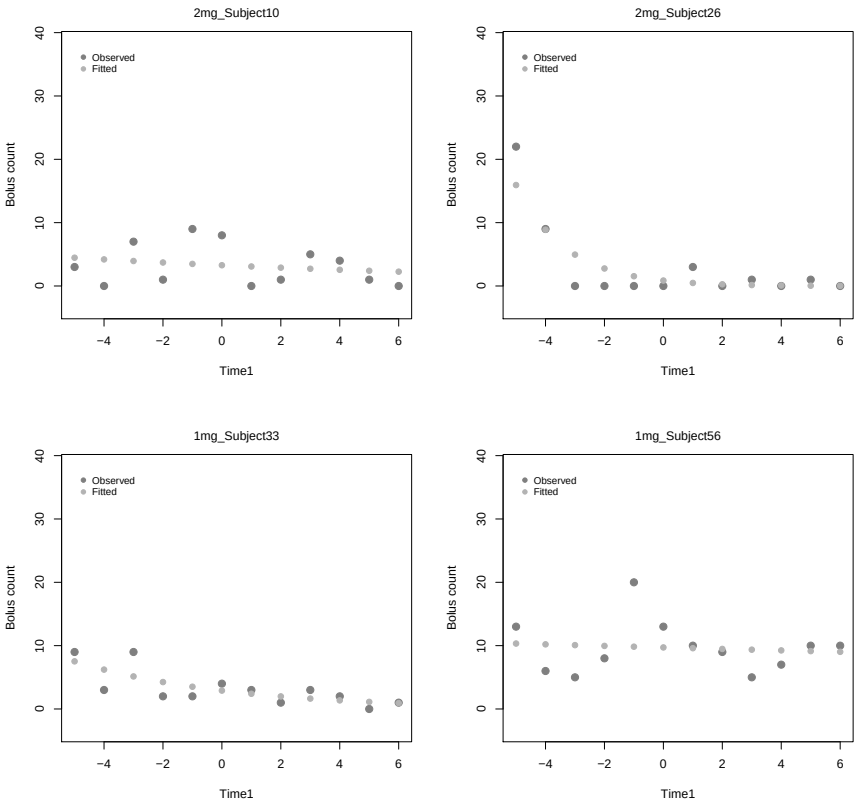

Figure 5: Observed and individual mean profiles of two subjects in each group for `bol_AR1R2a`.

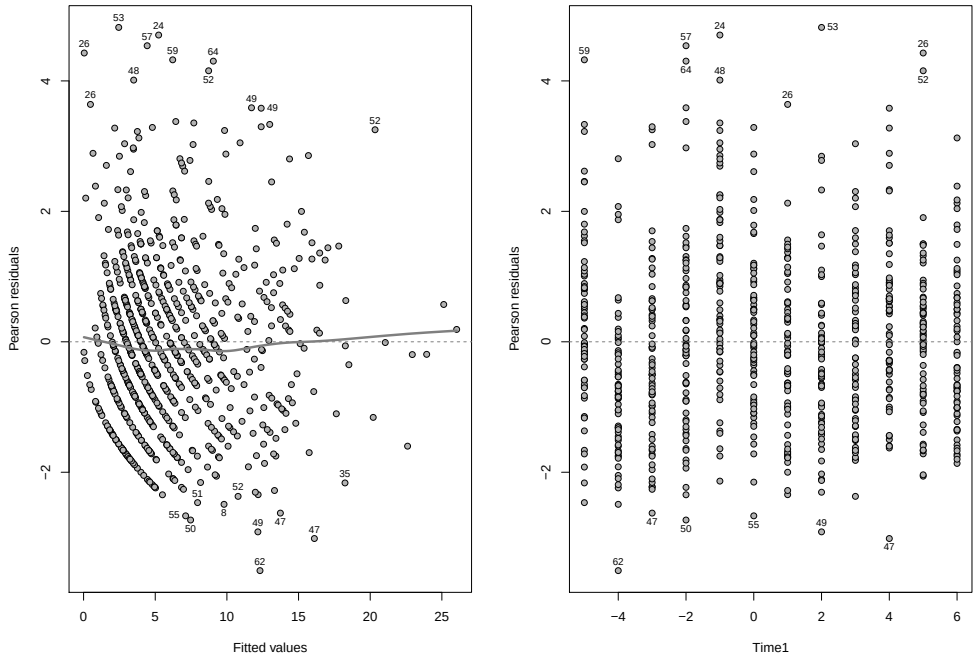

Figure 6: Residual plots of bolus data for `bol_AR1R2a`.

The functions `fitted()` and `resid()`, available in the **cold** package, can be used to perform a brief residual analysis as illustrated in Figure 6 via the statement:

```
R> attach(bolus)
R> resp <- resid(bol_AR1R2a)
R> mu <- fitted(bol_AR1R2a)
R> op <- par(mfrow = c(1, 2))
R> plot(mu, resp, pch = 21, bg = "grey70", xlab = "Fitted values",
+    ylab = "Pearson residuals")
R> abline(h = 0, lty = 2, col = "grey50")
R> identify(mu, resp, id, cex = 0.75)
R> lines(lowess(resp ~ mu), lwd = 3, col = "grey50")
R> plot(time1, resp, pch = 21, bg = "grey70", xlab = "Time1",
+    ylab = "Pearson residuals")
R> abline(h = 0, lty = 2, col = "grey50")
R> identify(time1, resp, id, cex = 0.75)
R> par(op)
```

Graphical diagnosis reported in Figure 6 suggests that no pattern is detected.

In addition, the results obtained for the additive model with `dependence = "indR"`, not showed here and taking into account that `time` variable was codded, are in agreement with the ones presented by Weiss (2005).

5. Some considerations on cold integration approach

The aim of this section is to give some guidelines to the user for the choice of the `integration` approach in **cold** package for models with one and two-random effects. For illustration, `seizure` data was used.

5.1. One random effect

For random intercept models **cold** uses, by default, an adaptive Gauss-Kronrod quadrature. The number of Gauss-Kronrod quadrature points can be select by setting the `key` argument of `coldIntegrate()` function as described in Section 3.6. In order to illustrate the effects of changing the number of quadrature points the random intercept model given by Equation 14 was fitted for both, `independence` and `AR1` structure, considering all the options of `key` argument. The results are displayed in Table 4.

Considerations on Gauss-Kronrod quadrature points

Taking into account the results displayed for the random intercept model, Table 4, the main conclusions are: (i) For `key = 1` and `key = 2`, in both `dependence` structures, the changes in estimates suggest that the choice of these values is not recommended. (ii) For the `independence` structure the results show that from `key = 3` the parameters estimates and respective standard errors are very similar; (iii) For the `AR1` dependence structure the results show that, to obtain similar values for the parameters estimates, is needed to increase the

`Independence`	`key = 1`	`key = 2`	`key = 3`	`key = 4`	`key = 5`	`key = 6`
Fixed effects						
`Intercept`	−1.31 (1.24)	−1.33 (0.98)	−1.34 (1.18)	−1.34 (1.18)	−1.34 (1.18)	−1.34 (1.18)
`lbase`	0.87 (0.11)	0.88 (0.13)	0.88 (0.13)	0.88 (0.13)	0.88 (0.13)	0.88 (0.13)
`trt`	−1.02 (0.33)	−0.93 (0.43)	−0.93 (0.37)	−0.93 (0.40)	−0.93 (0.40)	−0.93 (0.40)
`lage`	0.48 (0.37)	0.48 (0.30)	0.48 (0.35)	0.48 (0.35)	0.48 (0.35)	0.48 (0.35)
`v4`	−0.16 (0.05)	−0.16 (0.05)	−0.16 (0.05)	−0.16 (0.05)	−0.16 (0.05)	−0.16 (0.05)
`lbase:trt`	0.40 (0.13)	0.33 (0.19)	0.34 (0.18)	0.34 (0.20)	0.34 (0.20)	0.34 (0.20)
Random effects						
`intercept`	0.26	0.25	0.25	0.25	0.25	0.25
`AR1`	`key = 1`	`key = 2`	`key = 3`	`key = 4`	`key = 5`	`key = 6`
Fixed effects						
`Intercept`	0.26 (1.98)	−1.08 (1.03)	−1.16 (1.15)	−1.15 (1.17)	−1.08 (1.18)	−1.06 (1.19)
`lbase`	0.98 (0.15)	0.89 (0.11)	0.96 (0.13)	0.95 (0.13)	0.92 (0.13)	0.92 (0.13)
`trt`	−0.15 (0.46)	−0.59 (0.31)	−0.74 (0.42)	−0.64 (0.40)	−0.82 (0.40)	−0.83 (0.40)
`lage`	−0.04 (0.59)	0.36 (0.30)	0.41 (0.34)	0.38 (0.34)	0.39 (0.35)	0.39 (0.35)
`v4`	−0.17 (0.05)	−0.17 (0.06)	−0.16 (0.05)	−0.18 (0.05)	−0.19 (0.06)	−0.20 (0.06)
`lbase:trt`	−0.11 (0.19)	0.16 (0.14)	0.23 (0.22)	0.20 (0.20)	0.27 (0.20)	0.27 (0.20)
Correlation						
`rho`	5.0×10^{-10}	5.3×10^{-9}	2.9×10^{-8}	2.9×10^{-8}	3.4×10^{-8}	2.7×10^{-8}
Random effects						
`intercept`	0.35	0.18	0.24	0.25	0.25	0.25

Table 4: Parameter estimates and standard errors of `seizure` data for a random intercept model with `independence` and `AR1` structure for different number of quadrature points.

Indepencence	M = 6000	M = 8000	M = 10000	M = 12000	cubature
Fixed effects					
Intercept	−1.23 (1.24)	−1.28 (1.11)	−1.31 (1.14)	−1.38 (1.12)	−1.36 (1.18)
lbase	0.89 (0.11)	0.91 (0.12)	0.89 (0.12)	0.87 (0.16)	0.89 (0.13)
trt	−0.94 (0.36)	−0.82 (0.38)	−0.96 (0.45)	−0.98 (0.39)	−0.93 (0.40)
lage	0.43 (0.37)	0.44 (0.33)	0.46 (0.34)	0.49 (0.33)	0.47 (0.35)
visit	−0.28 (0.16)	−0.27 (0.15)	−0.25 (0.16)	−0.27 (0.15)	−0.26 (0.16)
lbase:trt	0.35 (0.17)	0.27 (0.19)	0.35 (0.24)	0.36 (0.21)	0.34 (0.20)
Random effects					
intercept	0.25	0.25	0.25	0.25	0.25
visit	0.51	0.53	0.53	0.51	0.53
AR1	**M = 6000**	**M = 8000**	**M = 10000**	**M = 12000**	**cubature**
Fixed effects					
Intercept	−1.35 (1.27)	−1.44 (1.16)	−1.47 (1.15)	−1.53 (1.13)	−1.53 (1.15)
lbase	0.91 (0.11)	1.04 (0.12)	0.88 (0.12)	0.88 (0.16)	0.88 (0.13)
trt	−0.72 (0.37)	−0.86 (0.38)	−1.08 (0.37)	−0.93 (0.38)	−0.87 (0.39)
lage	0.45 (0.37)	0.43 (0.34)	0.51 (0.34)	0.53 (0.34)	0.53 (0.34)
visit	−0.37 (0.15)	−0.30 (0.14)	−0.27 (0.15)	−0.36 (0.15)	−0.33 (0.16)
lbase:trt	0.25 (0.17)	0.32 (0.19)	0.41 (0.17)	0.32 (0.20)	0.32 (0.20)
Correlation					
rho	3.3×10^{-8}	1.0×10^{-7}	2.3×10^{-8}	7.3×10^{-8}	1.8×10^{-7}
Random effects					
intercept	0.26	0.25	0.25	0.25	0.23
visit	0.41	0.37	0.43	0.46	0.59

Table 5: Parameter estimates and standard errors of `seizure` data for a two-random effects model for `independence` and `AR1` structure for different number of iterations.

number of quadrature points at least to `key = 5`.

Increasing the number of quadrature points requires more evaluations of the integrand in Equation 11, however the time needed for the estimating procedure convergence for `key = 5` and for `key = 6` is nearly the same, hence we set the default to `key = 6`. Despite that `QUADPACK` only allows one random effect, this choice to the default for `integration` argument was based on the considerable increase in speed to get convergence when compared with `cubature` and `MC`.

5.2. Two random effects

In **cold** package to fit a two random effects model the user has to define the `integration` argument to `cubature` or `MC`. To illustrate the behavior of **cold** approaches for models with two random effects the model in Equation 15 was fitted for both, `independence` and `AR1` structure, using `integration = "cubature"` and `integration = "MC"` considering several `M` values. The results obtained are presented in Table 5.

Considerations on number of iterations of Monte Carlo

Monte Carlo approach appears to be a good alternative to `cubature` for two random effects models although it is difficult to establish a rule for the appropriate number of iterations. However, based on the results displayed in Table 5, our suggestions are: (i) For the `independence`

Integration	AGH	AGH	QUADPACK	MC	cubature
Package	**lme4**	**glmmML**	**cold**	**cold**	**cold**
Fixed effects					
Intercept	−1.34 (1.18)	−1.34 (1.18)	−1.34 (1.18)	−1.37 (1.18)	−1.34 (1.18)
lbase	0.88 (0.13)	0.88 (0.13)	0.88 (0.13)	0.88 (0.17)	0.88 (0.13)
trt	−0.93 (0.40)	−0.93 (0.40)	−0.93 (0.40)	−0.89 (0.39)	−0.93 (0.40)
lage	0.48 (0.35)	0.48 (0.35)	0.48 (0.35)	0.49 (0.34)	0.48 (0.35)
v4	−0.16 (0.05)	−0.16 (0.05)	−0.16 (0.05)	−0.16 (0.05)	−0.16 (0.05)
lbase:trt	0.34 (0.20)	0.34 (0.20)	0.34 (0.20)	0.32 (0.20)	0.34 (0.20)
Random effects					
intercept	0.25	0.25	0.25	0.25	0.25

Table 6: Parameters estimates and standard errors of `seizure` data for random intercept model and `independence` structure.

structure the results suggests that `M` value should be at least 6000; (ii) For the `AR1` structure the results suggests that `M` value should be at least 12000.

If the time to reach convergence is not an issue the user should use `integration = "cubature"`.

6. Comparison with other packages

In this section **cold** package is compared with other R packages (**lme4** and **glmmML**) which can be employed for count longitudinal analysis. For illustration, `seizure` data was used once more. One must have in mind that **lme4** and **glmmML** only allow an `independence` structure and that **glmmML** only allows a random effect in the intercept.

6.1. One random effect

The random intercept model given in Equation 14 was fitted using functions, `glmer()`, `glmmML()` and `cold()` of pakages **lme4**, **glmmML** and **cold**, respectively. For `glmer()` function, adaptive Gauss-Hermite approximation to the log-likelihood was used with `nAGQ = 10` and `nAGQ = 25`, for both values of `nAGQ` considered the same results were obtained. For `glmmML()` function, the Gauss-Hermite quadrature was used setting `method = "ghq"` and the number of quadrature points used was `n.points = 8` (default) and `n.points = 16`, the same results were obtained for both number of quadrature points. For `cold()` function, the three `integration` options (`QUADPACK`, `MC` and `cubature`) were considered with the respective defaults settings.

The results displayed in Table 6 allow to conclude that: (i) For random intercept model, `QUADPACK` (`k = 6`) and `cubature` produce the same results obtained for **lme4** and **glmmML**; (ii) The results obtained for `MC`, considering the default number of iterations (`M = 6000`), are not exactly the same but agree closely.

Note that the results for `QUADPACK` presented in Table 6 correspond to `key = 6` for the independence model in Table 4.

6.2. Two random effects

The `independence` model with two random effects given in Equation 15 was also used to

Integration	`Laplace`	`MC`	`cubature`
Package	**lme4**	**cold**	**cold**
Fixed effects			
`Intercept`	−1.36 (1.17)	−1.23 (1.24)	−1.36 (1.18)
`lbase`	0.89 (0.13)	0.89 (0.11)	0.89 (0.13)
`trt`	−0.93 (0.40)	−0.94 (0.36)	−0.93 (0.40)
`lage`	0.47 (0.34)	0.43 (0.37)	0.47 (0.35)
`visit`	−0.26 (0.16)	−0.28 (0.16)	−0.26 (0.16)
`lbase:trt`	0.34 (0.20)	0.35 (0.17)	0.34 (0.20)
Random effects			
`intercept`	0.25	0.25	0.25
`visit`	0.53	0.51	0.53

Table 7: Parameters estimates and standard errors of `seizure` data for a two-random effects model and `independence` structure.

compare **cold** with **lme4**. For `glmer()` function, only Laplace approximation is available which corresponds to `nAGQ = 1`. For `cold()` function, `MC (M = 6000)` and `cubature integration` approaches were used. The results displayed in Table 7 allow us to conclude that the `cubature` approach gives the same results of **lme4** package, however, as mentioned in Section 3.5, this approach can be computationally intensive and Monte Carlo approach appears to be a good alternative.

7. Closing remarks

The **cold** package was presented for the analysis of count longitudinal data. The package was built around its main function `cold()` which performs the fit of models by maximizing the log-likelihood according to the serial dependence structure. Serial dependence of first order autoregressive model, intercept and slope random effects as well as missing values are allowed. The theory used for model fitting was summarized briefly. The most important arguments of `cold()` function were described and discussed. A substantial computational burden is involved by the numerical integration connected to the random effects of Section 2.3, but this is not heavier than other formulations which incorporate random effects in discrete longitudinal data when a similar exact numerical integration is performed. We have illustrated the package **cold** through the use of two real dataset.

Acknowledgments

The authors thank to the referees and the Editors-in-Chief for a number of comments which have lead to a much improved presentation of the material. This work was partially financed by national funds through FCT-Fundação para a Ciência e a Tecnologia under the project UIDB/00006/2020.

References

Al-Osh MA, Alzaid AA (1987). "First-Order Integer-Valued Autoregressive (INAR(1)) Process." *Journal of Time Series Analysis*, **8**(3), 261–275. doi:10.1111/j.1467-9892.1987.tb00438.x.

Azzalini A (1986). "Alcuni Risultati Sull'Effetto Dell'Autocorrelazione." In DM Cifarelli (ed.), *Scritti in Onore Di Francesco Brambilla*, pp. 1–9. Edizioni di Bocconi Comunicazione.

Azzalini A (1994). "Logistic Regression and Other Discrete Data Models for Serially Correlated Observations." *Journal of the Italian Statistical Society*, **3**, 169–179. doi:10.1007/bf02589225.

Azzalini A, Chiogna M (1997). "S-PLUS Tools for the Analysis of Repeated Measurements Data." *Computational Statistics*, **12**, 53–66.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using **lme4**." *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.

Breslow NE, Clayton DG (1993). "Approximate Inference in Generalized Linear Mixed Models." *Journal of the American Statistical Association*, **88**(421), 9–25. doi:10.1080/01621459.1993.10594284.

Broström G, Holmberg H (2018). ***glmmML**: Generalized Linear Models with Clustering*. R package version 1.0.3, URL https://CRAN.R-project.org/package=glmmML.

Carey VJ (2019). ***gee**: Generalized Estimation Equation Solver*. R package version 4.13-20, URL http://CRAN.R-project.org/package=gee.

Diggle PJ, Heagerty P, Liang KY, Zeger SL (2002). *Analysis of Longitudinal Data*. Oxford University Press, Oxford.

Fitzmaurice GM, Davidian M, Verbeke G, Molenberghs G (2009). *Longitudinal Data Analysis*. Chapman & Hall/CRC, Boca Raton.

Fitzmaurice GM, Laird NM, Ware JH (2004). *Applied Longitudinal Analysis*. John Wiley & Sons, New York. doi:10.1002/9781119513469.

Gonçalves MH (2002). *Likelihood Methods for Discrete Longitudinal Data*. Ph.D. thesis, University of Lisbon.

Gonçalves MH, Cabral MS (2021). ***cold**: A Package for Count Longitudinal Data*. R package version 2.0-2, URL https://CRAN.R-project.org/package=cold.

Gonçalves MH, Cabral MS, Azzalini A (2012). "The R Package **bild** for the Analysis of Binary Longitudinal Data." *Journal of Statistical Software*, **46**(9), 1–17. doi:10.18637/jss.v046.i09.

Gonçalves MH, Cabral MS, Azzalini A (2020). ***bild**: A Package for BInary Longitudinal Data*. R package version 1.2-0, URL https://CRAN.R-project.org/package=bild.

Gonçalves MH, Cabral MS, de Villa MCR, Escrich E, Solanas M (2007). "Likelihood Approach for Count Data in Longitudinal Experiments." *Computational Statistics & Data Analysis*, **51**(12), 6511–6520. doi:10.1016/j.csda.2007.03.002.

Henderson R, Shimakura S (2003). "A Serially Correlated Gamma Frailty Model for Longitudinal Count Data." *Biometrika*, **90**(2), 355–366. doi:10.1093/biomet/90.2.355.

Højsgaard S, Halekoh U, Yan J (2006). "The R Package **geepack** for Generalized Estimating Equations." *Journal of Statistical Software*, **15**(2), 1–11. doi:10.18637/jss.v015.i02.

Jowaheer V, Sutradhar BC (2002). "Analysing Longitudinal Count Data with Overdispersion." *Biometrika*, **89**, 389–399. doi:10.1093/biomet/89.2.389.

Liang KY, Zeger SL (1986). "Longitudinal Data Analysis Using Generalized Linear Models." *Biometrika*, **73**(1), 13–22. doi:10.1093/biomet/73.1.13.

Little RJA, Rubin DB (1987). *Statistical Analysis with Missing Data*. John Wiley & Sons, New York.

McDaniel LS, Henderson NC, Rathouz PJ (2013). "Fast Pure R Implementation of GEE: Application of the **Matrix** Package." *The R Journal*, **5**, 181–187. doi:10.32614/rj-2013-017.

McKenzie E (1985). "Some Simple Models for Discrete Variate Time Series." *Water Resources Bulletin*, **21**(4), 645–650. doi:10.1111/j.1752-1688.1985.tb05379.x.

McKenzie E (1988). "Some ARMA Models for Dependent Sequences of Poisson Counts." *Advances in Applied Probability*, **20**, 822–835. doi:10.2307/1427362.

Molenberghs G, Verbeke G, Demetrio CGB (2007). "An Extended Random-Effects Approach to Modelling Repeated, Overdispersed Count Data." *Lifetime Data Analysis*, **13**, 513–531. doi:10.1007/s10985-007-9064-y.

Narasimhan B, Johnson SG, Hahn T, Bouvier A, Kiêu K (2020). ***cubature**: Adaptive Multivariate Integration over Hypercubes*. R package version 2.0.4.1, URL https://CRAN.R-project.org/package=cubature.

Pinheiro JC, Bates DM (2000). *Mixed-Effects Models in S and S-PLUS*. Springer-Verlag, New York. doi:10.1007/b98882.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Self SG, Liang KY (1987). "Asymptotic Properties of Maximum Likelihood Estimators and Likelihood Ratio Tests Under Nonstandard Conditions." *Journal of the American Statistical Association*, **82**, 605–610. doi:10.1080/01621459.1987.10478472.

Steutel FW, Harn KV (1979). "Discrete Analogues of Self-Decomposability and Stability." *The Annals of Probability*, **7**(5), 893–899. doi:10.1214/aop/1176994950.

Stram DO, Lee JW (1994). "Variance Components Testing in the Longitudinal Mixed Effects Model." *Biometrics*, **50**, 1171–1177. doi:10.2307/2533455.

Thall PF, Vail SC (1990). "Some Covariance Models for Longitudinal Count Data with Overdispersion." *Biometrics*, **46**, 657–671. doi:10.2307/2532086.

Weiss RE (2005). *Modeling Longitudinal Data*. Springer-Verlag, New York. doi:10.1007/0-387-28314-5.

Zeger SL (1988). "A Regression Model for Time Series of Counts." *Biometrika*, **75**(4), 621–629. doi:10.1093/biomet/75.4.621.

A. Derivatives of the log-likelihood function

A.1. Marginal model

The derivatives of the logarithm of the likelihood function given in Equation 8 were obtained having in mind a sequence of data $y_{i1}, \ldots, y_{iT}$, corresponding to the subject i. To compute these derivatives denote by ℓ_t the contribution to the log-likelihood of a generic individual at the observation time t,

$$\ell_t = \mathsf{P}(Y_t = y_t | Y_{t-m} = y_{t-m}) = \log p_{y_{t-m}, y_t}.$$

So, for $t \geq 2$ the use of the chain rule to differentiate the corresponding term, ℓ_t gives the derivatives

$$\frac{\partial \ell_t}{\partial \beta} = \frac{\partial \ell_t}{\partial p_{y_{t-m},y_t}} \frac{\partial p_{y_{t-m},y_t}}{\partial v_{t,m}} \left(\frac{\partial v_{t,m}}{\partial \theta_t} \frac{\partial \theta_t}{\partial \beta} + \frac{\partial v_{t,m}}{\partial \theta_{t-m}} \frac{\partial \theta_{t-m}}{\partial \beta} \right),$$

$$\frac{\partial \ell_t}{\partial \rho} = \frac{\partial \ell_t}{\partial \rho^m} \frac{\partial \rho^m}{\partial \rho} = \frac{\partial \ell_t}{\partial p_{y_{t-m},y_t}} \left(\frac{\partial p_{y_{t-m},y_t}}{\partial \rho^m} + \frac{\partial p_{y_{t-m},y_t}}{\partial v_{t,m}} \frac{\partial v_{t,m}}{\partial \rho^m} \right) \frac{\partial \rho^m}{\partial \rho}.$$

Furthermore, different expressions are required for the case $t = 1$.

Given the algebraic work required to obtain explicit expressions of the gradient of the log-likelihood, it is completely unfeasible to develop analogous results for the Hessian matrix. Therefore, the observed information matrix must be computed via numerical differentiation of the first derivatives.

A.2. Random intercept model

The main steps to compute the derivatives of the logarithm of the likelihood function given in Equation 11, for the subject i, are presented below. The derivatives of the log-likelihood function with respect to β are given by

$$\frac{\partial \ell^R(\beta, \rho, \omega)}{\partial \beta} = \sum_{i=1}^{n} \frac{\dfrac{\partial L_i^R(\beta, \rho, \omega)}{\partial \beta}}{L_i^R(\beta, \lambda, \omega)}, \tag{17}$$

hence,

$$\frac{\partial \ell^R(\beta, \rho, \omega)}{\partial \beta} = \sum_{i=1}^{n} \frac{\displaystyle\int_{\mathbb{R}} \frac{\partial \ell_i^F(\beta^{b_i}, \rho | b_i)}{\partial \beta^{b_i}} \exp\left(\ell_i^F(\beta^{b_i}, \rho | b_i) - \frac{b_i^2}{2e^{\omega}} \right) db_i}{\displaystyle\int_{\mathbb{R}} \exp\left(\ell_i^F(\beta^{b_i}, \rho | b_i) - \frac{b_i^2}{2e^{\omega}} \right) db_i}. \tag{18}$$

The derivatives of the log-likelihood function with respect to ρ, $\dfrac{\partial \ell^R(\beta, \rho, \omega)}{\partial \rho}$, are computed by a similar expression to Equation 18, where we replace $\dfrac{\partial \ell_i^F(\beta^{b_i}, \rho | b_i)}{\partial \beta^{b_i}}$ by $\dfrac{\partial \ell_i^F(\beta^{b_i}, \rho | b_i)}{\partial \rho}$, whose derivatives are computed using the expressions presented in Section A.1.

The derivatives of the log-likelihood function with respect to ω, where $\omega = \log(\sigma_b^2)$, for the ith subject are computed using:

$$\frac{\partial \ell^R(\beta, \lambda, \omega)}{\partial \omega} = \sum_{i=1}^{n} \frac{\dfrac{\partial L_i^R(\beta, \lambda, \omega)}{\partial \omega}}{L_i^R(\beta, \lambda, \omega)}, \tag{19}$$

Hence, the log-likelihood function with respect to ω is

$$\frac{\partial \ell^R(\beta,\rho,\omega)}{\partial \omega} = \sum_{i=1}^{n} \frac{\int_{\mathbb{R}} \exp\left(\ell_i^F(\beta^{b_i},\rho|b_i) - \frac{b_i^2}{2e^{\omega}}\right)\left(\frac{b_i^2 - e^{\omega}}{2e^{2\omega}}\right) db_i \; e^{\omega}}{\int_{\mathbb{R}} \exp\left(\ell_i^F(\beta^{b_i},\rho|b_i) - \frac{b_i^2}{2e^{\omega}}\right) db_i}.$$

See Gonçalves (2002) for complete details about these results.

A.3. Two-dimensional random effects model

The main steps to compute the derivatives of the logarithm of the likelihood function given in Equation 12, for the subject i, are presented below. The derivatives of the log-likelihood function with respect to β are computed using Equation 17 and given by

$$\frac{\partial \ell^R(\beta,\rho,\Omega)}{\partial \beta} = \sum_{i=1}^{n} \frac{\int_{\mathbb{R}}\int_{\mathbb{R}} \frac{\partial \ell_i^F(\beta^{b_i},\rho|b_i)}{\partial \beta^{b_i}} \exp\left[\ell_i^F(\beta^{b_i},\rho|b_i) - \frac{1}{2}\left(\frac{b_{0i}^2}{e^{\omega_0}} + \frac{b_{1i}^2}{e^{\omega_1}}\right)\right] db_{0i} db_{1i}}{\int_{\mathbb{R}}\int_{\mathbb{R}} \exp\left[\ell_i^F(\beta^{b_i},\rho|b_i) - \frac{1}{2}\left(\frac{b_{0i}^2}{e^{\omega_0}} + \frac{b_{1i}^2}{e^{\omega_1}}\right)\right] db_{0i} db_{1i}}. \quad (20)$$

The derivatives of the log-likelihood function with respect to ρ, $\frac{\partial \ell^R(\beta,\rho,\Omega)}{\partial \rho}$, are computed by a similar expression to Equation 20, where we replace $\frac{\partial \ell_i^F(\beta^{b_i},\rho|b_i)}{\partial \beta^{b_i}}$ by $\frac{\partial \ell_i^F(\beta^{b_i},\rho|b_i)}{\partial \rho}$, whose derivatives are computed using the expressions presented in Section A.1.

Hence, the derivatives of the log-likelihood function with respect to ω_0 for n individuals can be computed by:

$$\frac{\partial \ell^R(\beta,\rho,\Omega)}{\partial \omega_0} = \sum_{i=1}^{n} \frac{\int_{\mathbb{R}}\int_{\mathbb{R}} \exp\left[\ell_i^F(\beta^{b_i},\rho|b_i) - \frac{1}{2}\left(\frac{b_{0i}^2}{e^{\omega_0}} + \frac{b_{1i}^2}{e^{\omega_1}}\right)\right]\left(\frac{b_{0i}^2 - e^{\omega_0}}{2e^{2\omega_0}}\right) db_{0i} db_{1i} e^{\omega_0}}{\int_{\mathbb{R}}\int_{\mathbb{R}} \exp\left[\ell_i^F(\beta^{b_i},\rho|b_i) - \frac{1}{2}\left(\frac{b_{0i}^2}{e^{\omega_0}} + \frac{b_{1i}^2}{e^{\omega_1}}\right)\right] db_{0i} db_{1i}}.$$

A similar procedure is used to obtain the derivatives of the log-likelihood function with respect to ω_1 for n individuals.

Again, the reader is referred to Gonçalves (2002) for complete details about these results.

B. Implementation details of Monte Carlo Methods

Monte Carlo methods are an alternative procedure to evaluate numerically high-dimensional integrals when random effects are incorporated in the linear predictor. To this end, the integral in Equation 11 can be approximated by

$$L_i^R(\beta,\rho,\omega) = E_{b_i}\{L_i^F(\beta^{b_i},\rho|b_i)\} \simeq \frac{1}{M}\sum_{j=1}^{M} L_i^F(\beta^{b_i},\rho|b_{ij}) = \frac{1}{M}\sum_{j=1}^{M} \exp\left(\ell_i^F(\beta^{b_i},\rho|b_{ij})\right), \quad (21)$$

where M is a large integer and $b_{ij} \sim N(0,\sigma_b^2)$ and $\ell_i^F(\beta^{b_i},\rho|b_i)$ can be computed using logarithm of the likelihood function given in Equation 8.

The log-likelihood for the n subjects is

$$\ell^R(\beta, \rho, \omega) \simeq \sum_{i=1}^{n} \log \left[\frac{1}{M} \sum_{j=1}^{M} \exp\left(\ell_i^F(\beta^{b_i}, \rho | b_{ij})\right) \right].$$

The main steps to compute the derivatives of $\ell_i^R(\beta, \rho, \omega) = \log L_i^R(\beta, \rho, \omega)$ with respect to each parameter β, ρ and ω are presented below. The derivatives of the log-likelihood function with respect to β are computed using Equation 17 where the denominator is given in Equation 21, and the numerator is

$$\begin{aligned}
\frac{\partial L_i^R(\beta, \rho, \omega)}{\partial \beta} \simeq & \frac{\partial}{\partial \beta} \Big\{ \frac{1}{M} \sum_{j=1}^{M} \exp\left(\ell_i^F(\beta^{b_i}, \rho | b_{ij})\right) \Big\} \\
= & \frac{1}{M} \sum_{j=1}^{M} \Big\{ \frac{\partial \ell_i^F(\beta^{b_i}, \rho | b_{ij})}{\partial \beta^{b_i}} \exp\left(\ell_i^F(\beta^{b_i}, \rho | b_{ij})\right) \Big\},
\end{aligned}$$

where $\dfrac{\partial \ell_i^F(\beta^{b_i}, \rho | b_{ij})}{\partial \beta^{b_i}}$ are computed using the expressions presented in Section A.1.

The derivatives of the log-likelihood function with respect to ρ, $\dfrac{\partial L_i^R(\beta, \rho, \sigma_b^2)}{\partial \rho}$, are computed by an analogous procedure.

The derivatives of the log-likelihood function with respect to ω, are computed using the scheme used in Section A.1 where the denominator of Equation 19 is computed by Equation 21 and the numerator by

$$\begin{aligned}
\frac{\partial L_i^R(\beta, \rho, \omega)}{\partial \omega} = & \int_{\mathbb{R}} \exp\left(\ell_i^F(\beta^{b_i}, \rho | b_i)\right) \left(\frac{b_i^2 - \sigma_b^2}{2\sigma_b^4} \right) \frac{1}{\sqrt{2\pi}\sigma_b} \exp\left(- \frac{b_i^2}{2\sigma_b^2} \right) db_i e^{\omega} \\
= & E_{b_i} \Big\{ \exp\left(\ell_i^F(\beta^{b_i}, \rho | b_i)\right) \left(\frac{b_i^2 - e^{\omega}}{2e^{2\omega}} \right) \Big\} e^{\omega} \\
\simeq & \frac{1}{M} \Big\{ \sum_{j=1}^{M} \exp\left(\ell_i^F(\beta^{b_i}, \rho | b_{ij})\right) \left(\frac{b_i^2 - e^{\omega}}{2e^{2\omega}} \right) \Big\} e^{\omega}.
\end{aligned}$$

As for the two-dimensional random effects and higher dimensions, the procedure to use Monte Carlo methods is similar to the one presented here for the random intercept model, keeping in mind that the number of parameters increases as we increase the dimension of the vector of random effects.

For the random intercept model, Monte Carlo methods performs very poorly, in terms of computational time, when comparing with numerical integration; in this situation, numerical integration is preferred since it gives essentially exact results and it is much faster than Monte Carlo methods.

Affiliation:

M. Helena Gonçalves
Centro de Estatística e Aplicações da Universidade de Lisboa
Departamento de Matemática, FCT, Universidade do Algarve
Gambelas, Portugal
E-mail: mhgoncal@ualg.pt

M. Salomé Cabral
Centro de Estatística e Aplicações da Universidade de Lisboa
Departamento de Estatística e Investigação Operacional, FCUL
Lisboa, Portugal
E-mail: salome@fc.ul.pt

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
August 2021, Volume 99, Issue 3 *Submitted:* 2019-07-31
doi:10.18637/jss.v099.i03 *Accepted:* 2020-09-08