An introduction to the medrc package

Daniel Gerhard

April 8, 2014

Contents

1 Motivation

The analysis of dose-response experiments using nonlinear models allows inference about the model parameters and the prediction of new response values. A common way to summarize the effect of an increasing dose level on the response is deriving parameters from the dose-response curve, like the effective dose $ED(p)$. Also the comparison of several curves by selectivity indices, like the relative potency, is available. For risk estimation in toxicology the estimation of benchmark dose (BMD) levels is a further important topic that is based on the nonlinear modeling of dose-response curves.

The drc package (Ritz and Streibig, 2005) allows the simultaneous fitting of several non-linear regression models, providing a common parameterization for several models and searching automatically for starting values. Several functions are available for model-averaging and inference of derived parameters.

When analyzing dose-response curves, the observed data is often obtained from experiments with hierarchical designs, where the responses can be assigned to several known clusters. Instead of just assuming a single residual error in the dose-response model, the variability within and between clusters can be modeled. Either additional correlation parameters can be introduced to structure the residual error by generalized nonlinear least squares estimation or a distribution function for the cluster effects can be assumed separately to the distribution of the residuals in a mixed model framework.

The nlme package (Pinheiro and Bates, 2000) provides functions to estimate parameters and variance components in nonlinear mixed models by (restricted) maximum likelihood. Some functions to parameterize the non-linear curve are already available, but it is also possible to provide your own modeling function.

The medrc combines the automated dose-response modeling framework of the package drc with the nonlinear mixed estimation framework of the package nlme. Thereby, additional random effects can be introduced to the dose-response models with the unified parameterization of drc, with the availability of inference for derived parameters also for hierarchical models.

2 The medrm function

Following the notation of Davidian and Giltinan (1995, 2003), a nonlinear regression model with a single hierarchical level can be defined in two stages that parameterize the intra- and inter-curve specific variation, respectively.

Stage 1: For the ith individual ($i = 1, \dots, m$), we assume the following nonlinear regression model:

$$y_{ij} = f(x_{ij}, \boldsymbol{\beta}_i) + \epsilon_{ij} \tag{1}$$

where $\{y_{i1}, \dots, y_{in_i}\}$ and $\{x_{i1}, \dots, x_{in_i}\}$ denote the vectors of response values and dose levels, respectively. The population mean dose-response curve is characterized by the dose-response function f through curve-specific effect $\boldsymbol{\beta}_i$ (a $q \times 1$ vector). The residual error ϵ_i is assumed to be mean-zero normally distributed with variance-covariance matrix $\boldsymbol{\Lambda}_i$, which in practice often is assumed to be $\sigma^2 I_{n_i}$ or $diag(\sigma_1^2, \dots, \sigma_{n_i}^2)$.

Stage 2: The inter-curve variation is captured by splitting the curve-specific effect $\boldsymbol{\beta}_i$ into components describing the systematic and random variation between curves:

$$\boldsymbol{\beta}_i = \boldsymbol{A}_i\boldsymbol{\beta} + \boldsymbol{B}_i\boldsymbol{b}_i$$

where $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ denote the fixed-effects and random-effects design matrices, respectively, $\boldsymbol{\beta}$ denotes the fixed-effects parameters (a $p \times 1$ vector with $p \leq q$), and the b_i's denote the curve-specific random effects. The random effects can be assumed to follow a mean-zero normal distribution with a variance-covariance matrix denoted $\boldsymbol{G}$, which usually simply is the unstructured matrix.

We restrict our model to a fixed effects parameterization of independent curves, using a dummy 0 and 1 coded design matrix $\boldsymbol{A}_i$, and allowing only random intercepts additively on the fixed effects parameters to represent the subject variability. This hierarchical model will cover many experimental settings in bioassay analysis and toxicology, like the comparison of several treatments in a dose-response experiment, where the responses are repeated measurements on the same experimental units or the experimental design consists of several blocks.

Function arguments

To fit a hierarchical dose-response model with medrc, the medrm function can be used with following main arguments:

form A formula with the name of the response y_{ij} on the left and the name of the dose variable x_{ij} on the right hand side

curveid A formula with the name of a factor, which divides the dataset into several clusters, specifying the structure of the fixed effects design matrix $\boldsymbol{A}_i$. For each cluster a different fixed effect curve is assumed. On the left hand side, the names of parameters can be given, separated by a + symbol, which are assumed to be different across the curves. Hence, only a subset of parameters may describe the difference between fixed effect curves. If these parameter names are omitted, or curveid is set to NULL, only a single curve is fitted.

data A data.frame object with the dose-response data

fct A model function of package drc specifying the dose-response function $f()$ with a set of parameters $\boldsymbol{\beta}_i$. Some predefined functions are shown in the following section.

random The definition of random effects, similar to the random argument in the function nlme, as a definition of the random effect design matrix $\boldsymbol{B}_i$. This can be, for example, a formula with parameter names on the left hand side, like in the curveid argument, and 1|name of clusters defining factor on the right hand side.

The 1 can also be substituted by an additional covariate. For a more flexible definition, a list of different pdClasses objects of nlme can be given.

start Starting values for the nlme function. If NULL, the initial values for the fixed effects are found automatically, using the self start functionality of package drc.

2.1 Dose-response curve parameterization

In drc a number of different dose-response models are available. Each model can be simplified, by fixing a parameter to a specific values instead of estimating it from the data.

The parameters are defined in a unified way with

b steepness of the curve

c lower asymptote

d upper asymptote

e location of the inflection point

f asymmetry parameter

5-parameter logistic L.5()

$$f(x) = c + \frac{d - c}{(1 + \exp(b(x - e)))^f}$$

5-parameter log-logistic LL.5()

$$f(x) = c + \frac{d - c}{(1 + \exp(b(\log(x) - \log(e))))^f}$$

4 parameter Weibull W1.4() and W2.4()

$$f(x) = c + (d - c)\exp(-\exp(b(\log(x) - \log(e))))$$

or

$$f(x) = c + (d - c)(1 - \exp(-\exp(b(\log(x) - \log(e)))))$$

4 parameter log-Normal LN.4()

$$f(x) = c + (d - c)(\Phi(b(\log(x) - \log(e))))$$

gompertz()

$$f(x) = c + (d - c)(\exp(-\exp(b(x - e))))$$

3 parameter Michaelis-Menten MM.3()

$$f(x, (c, d, e)) = c + \frac{d - c}{1 + (e/x)}$$

Further functions

More functions are available, like fractional polynomial-logistic models fplogistic(p1, p2), or the Brain-Cousens hormesis model braincousens().

3 Inference for derived parameters

Instead of interpreting the model parameters directly, we can set focus on derived parameters, like the effective dose at a specific level $ED(p)$ or the ratio of two effective doses, known as relative potency. The effective dose $ED(p)$ is defined as the solution to the following inverse regression problem:

$$f(ED(p), \boldsymbol{\beta}) = \frac{p}{100} f(\infty, \boldsymbol{\beta}_i) + \left(1 - \frac{p}{100}\right) f(0, \boldsymbol{\beta}_i)$$

By definition $ED(p)$ values are relative quantities, relative to the lower and upper limits β_2 and β_3, which corresponds to $f(0, \boldsymbol{\beta}_i)$ and $f(\infty, \boldsymbol{\beta}_i)$, respectively, if $\beta_1 < 0$ (otherwise β_2 and β_3 swap places). The function `ED()` of package *drc* can be directly used with an medrc object to calculate the effective dose at (multiple) response levels conditional on random effects being equal to zero. Inference for the $ED(p)$ parameters, like hypotheses tests or corresponding confidence intervals, are available by using the Delta-method to approximate the variance-covariance of the derived parameters.

Another important concept used for summarizing dose-response data is the benchmark dose methodology. Ritz et al. (2013) proposed an operational definition of the benchmark dose concept for continuous endpoints, allowing for the incorporation of an a priori specified background level p_0 and benchmark response of interest BMR. The resulting BMD is obtained by solving the following equation:

$$\left[\Phi^{-1}(1 - p_0) - \Phi^{-1}\{1 - (p_0 + BMR)\}\right] \frac{\sigma}{f(\infty, \boldsymbol{\beta}_i) - f(0, \boldsymbol{\beta}_i)} = \frac{f(BMD, \boldsymbol{\beta}_i) - f(0, \boldsymbol{\beta}_i)}{f(\infty, \boldsymbol{\beta}_i) - f(0, \boldsymbol{\beta}_i)}. \tag{2}$$

By ignoring the variability in the ratio $\sigma\{f(\infty, \boldsymbol{\beta}_i) - f(0, \boldsymbol{\beta}_i)\}^{-1}$ this definition implies that benchmark doses may be derived in the same way as ordinary effective doses defined above by specifying the non-standard response level. The function `BMD()` enables the computation of the BMD and lower confidence bounds (BMDL) for a medrc object, given the benchmark risk and a background risk level.

3.1 Inference for marginalized parameters

As all derived parameters, discussed in the previous section, are functions of the nonlinear model predictions, a population average interpretation is available when these derived parameters are based on the marginal predictions. Hence, the estimates for marginal derived parameters, like the effective dose, are directly available by obtaining marginal predictions.

The marginal expectation for a model with a single fixed effect curve and a single matrix of random effects is equal to

$$E\left\{f(x_{ij}, \boldsymbol{\beta}_i)\right\} = \int \cdots \int f(x_{ij}, (\beta_1 + b_1, \ldots, \beta_p + b_p))\ \Phi(b_1, \ldots, b_p, \boldsymbol{G})\ db_1 \cdots db_p$$

where $\Phi()$ denotes a multivariate Gaussian density with mean vector $\mathbf{0}$ and covariance structure $\boldsymbol{G}$.

As in nonlinear regression correlated random effects are usually the case, a change of variable can be performed, transforming the dependent random effects $\boldsymbol{b}_i = \boldsymbol{\Omega}\boldsymbol{u}_i$ into a vector of spherical random effects $\boldsymbol{u}_i \sim N(\mathbf{0}, \boldsymbol{I})$. The matrix $\boldsymbol{\Omega}$ is a left factor of the covariance matrix $\boldsymbol{G} = \boldsymbol{\Omega}\boldsymbol{\Omega}'$, using a Choleski decomposition.

The multiple integral can be approximated by the weighted sum

$$E\left\{f(x_{ij}, \boldsymbol{\beta}_i)\right\} \approx \sum_{n=1}^{N} w_n f(x_{ij}, (\beta_1 + \xi_{rn}, \ldots, \beta_p + \xi_{rn})), \quad \text{with} \quad w_n = \prod_{r=1}^{p} w_{rn}$$

using numerical quadrature, with a $(p \times N)$ grid of nodes $\boldsymbol{\Psi}$ and corresponding weights w_{rn}, based on independent standard normal distribution functions (Smyth, 1998); the quadrature nodes are transformed by $\boldsymbol{\xi} = \boldsymbol{\Psi}\widehat{\boldsymbol{\Omega}}$ to reflect the scales and covariance of the random effects. The solution to the multiple integral can be thought of as an average of the individual curves, but instead of using the specific random effect predictions, this average is based on the empirical (multivariate) normal distribution of the random effects specified by the random effect covariances. Standard errors for derived parameters are obtained by applying the delta method, using a numerically calculated gradient.

Analogously to the `ED()` and `BMD()` functions, the `EDmarg()` and `BMDmarg()` functions are available for calculating marginalized estimates for the derived parameters. (The marginalization is only available for medrc objects with a simple random effects structure, allowing only with random intercepts for possibly nested random effects.)

3.2 Model-averaged estimates

By specifying the nonlinear function $f(x_{ij}, \boldsymbol{\beta}_i)$ the principle shape of the curve is treated as known. Without any prior knowledge about the progress of the dose-response curve it is reasonable to choose from a larger set of dose-response models instead of assuming a single fixed function. With a set of candidate models a specific dose-response relationship can be selected based on the data, or model averaging approaches can be used to incorporate the model uncertainty into the parameter inference.

The framework of package drc allows to fit several dose-response models to the same data in an automated fashion, with several, predefined dose-response curves, all of them composed of a similar set of similar defined parameters with lower and upper asymptotes, steepness, inflection points, etc. As these model functions can be directly used in medrc, the formula interfaces enable the composition of a set of models with different fixed and random effect parameterizations and different dose-response shapes.

Model-averaged inference for the effective dose is available by the function `mmaED()`, allowing the input of several medrc model objects. By default the model parameters are estimated by maximum likelihood; it is not adviced to use the model averaging approach with REML estimates. Furthermore, only models with changes in the fixed effect parameterization OR the random effect structure should be combined.

4 Case Studies

4.1 Vinclozolin example

Nellemann et al. (2003) carried out experiments to assess the in vitro effects of the fungicide vinclozolin. The data were obtained using an androgen receptor reporter gene assay, which was repeated six times (on different days). Each assay resulted in concentration-response data with nine concentrations (in μM) of the fungicide, and the response measured was chemiluminescence (in luminescence units), so the same nine concentrations were used in all six assays. However, in one assay, only eight concentrations were used.

The dataset is available in the package drc.

```
data(vinclozolin)
```

4.1.1 3 parameter log-logistic model

Assuming a lower asymptote at 0 for the control, a three parameter log-logistic model can be assumed, estimating the upper asymptote, location of the inflection point, and steepness of the curve. The assay effect is treated as a normally distributed random effect, summarizing the between-assay variability by a (3×3) covariance matrix.

```
m <- medrm(effect ~ conc, data=vinclozolin,
           random=b + d + e ~ 1|exper,
           fct=LL.3(), start=c(0.5, 2000, 0.05))
print(m)
```

```
##
## A 'drc' model.
##
## Call:
## medrm(form = effect ~ conc, data = vinclozolin, fct = LL.3(),     random = b + d + e ~ 1 | exper, st
##
## Coefficients:
##         b          d          e
##     0.538   1987.311      0.101
```

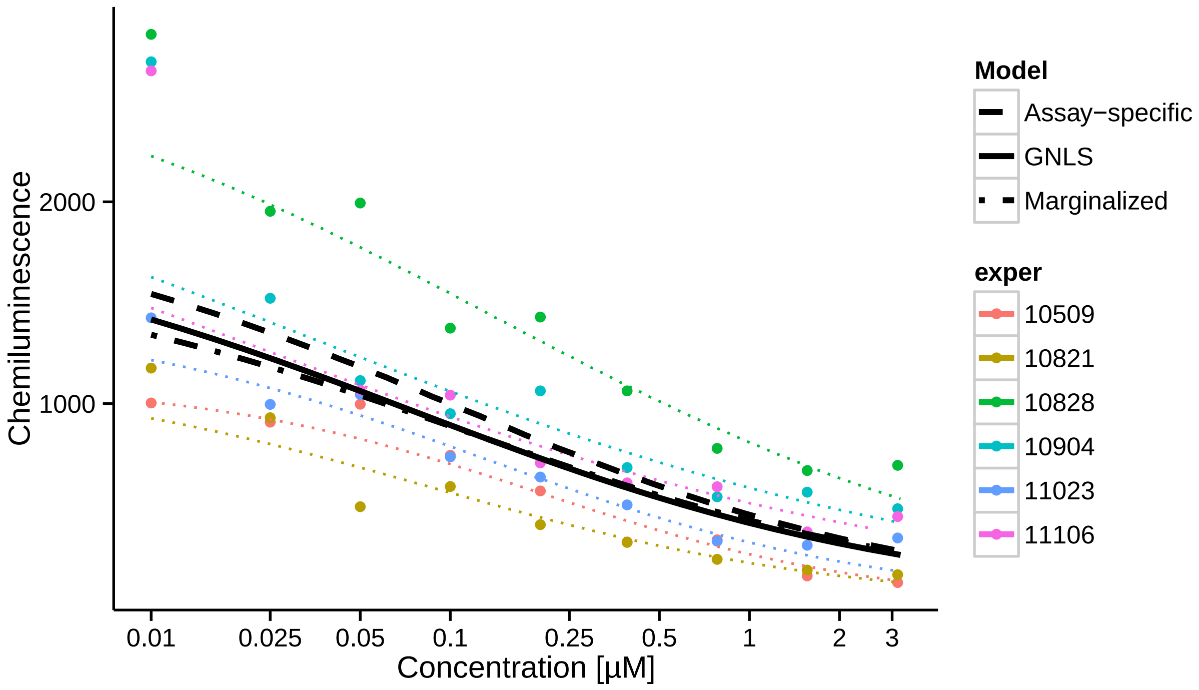
Instead of assay-specific parameters in a nonlinear mixed model, population-averaged estimates can be obtained by a generalized nonlinear least-squares fit, assuming a compound-symmetry structure for the residuals to model the between assay variability.

```
mg <- glsdrm(effect ~ conc, data=vinclozolin,
             fct=LL.3(), start=c(0.5, 2000, 0.05),
             correlation=corCompSymm(form=~1|exper),
             control=gnlsControl(tolerance=0.01))
```

```
print(mg)

##
## A 'drc' model.
##
## Call:
## glsdrm(form = effect ~ conc, data = vinclozolin, fct = LL.3(),     correlation = corCompSymm(form =
##
## Coefficients:
##        b          d          e
## 5.07e-01  1.93e+03  7.51e-02
```

The predicted curves can easily be visualized by the plot method for the model objects. But to compare the assay-specific, marginalized, and marginal predictions a user defined plot can be constructed, using the predict method to obtain the desired predicted values for new dose levels.

```
nd <- expand.grid(conc=exp(seq(log(0.01), log(3.2), length=25)))
nd1 <- nd2 <- nd3 <- nd
nd1$p <- predict(m, type="marginal", newdata=nd)
nd1$model <- "Marginalized"
nd2$p <- predict(mg, newdata=nd)
nd2$model <- "GNLS"
nd3$p <- m$fct$fct(nd$conc, rbind(coefficients(m)))
nd3$model <- "Assay-specific"
nd <- rbind(nd1, nd2, nd3)
nd4 <- expand.grid(conc=exp(seq(log(0.01), log(3.2), length=25)),
                   exper=unique(vinclozolin$exper))
nd4$p <- predict(m, newdata=nd4)
vinclozolin2 <- vinclozolin
vinclozolin2$conc[vinclozolin2$conc == 0] <- 0.01

ggplot(vinclozolin2, aes(x=conc, y=effect, group=exper, colour=exper)) +
  geom_point() +
  geom_line(data=nd4, aes(y=p, colour=exper), linetype=3) +
  geom_line(data=nd, aes(y=p, linetype=model, group=NULL, colour=NULL), size=1.1) +
  coord_trans(x="log") +
  theme_classic() +
  scale_linetype_manual("Model", values=c(2,1,4)) +
  ylab("Chemiluminescence") +
  xlab("Concentration [µM]") +
  scale_x_continuous(breaks=c(0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 1, 2, 3),
                     labels=c(0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 1, 2, 3))
```



4.1.2 Effective dose estimation

The effective doses can be estimated at several response levels, conditional on random effects being equal to zero, using the function `ED()`.

```
ED(m, respLev = c(10, 25, 50, 75, 90))

##
## Estimated effective doses
##
##      Estimate Std. Error
## 1:10  0.00171       0.00
## 1:25  0.01316       0.01
## 1:50  0.10136       0.03
## 1:75  0.78063       0.14
## 1:90  6.01224       1.95
```

For this example, not an assay-specific effective dose is of interest, but a population-averaged estimate of the $ED(p)$. Hence, either the `EDmarg()` function can be applied, or the `ED()` function together with the GNLS model.

```
EDmarg(m, respLev = c(10, 25, 50, 75, 90))

##
## Estimated effective doses
##
##      Estimate Std. Error
## 1:10 0.000119       0.00
## 1:25 0.003225       0.01
## 1:50 0.061922       0.05
## 1:75 0.640921       0.33
## 1:90 6.270124       4.13

ED(mg, respLev = c(10, 25, 50, 75, 90))

##
## Estimated effective doses
##
##      Estimate Std. Error
## 1:10 0.000987       0.00
## 1:25 0.008609       0.00
## 1:50 0.075078       0.03
## 1:75 0.654753       0.39
## 1:90 5.710099       5.39
```

4.2 Spinach example

Streibig and Dayan (1999) investigated the inhibition of photosynthesis in response to two synthetic photosystem II inhibitors, the herbicides diuron and bentazon. In an experiment, the effect of oxygen consumption of thylakoid membranes (chloroplasts) from spinach was measured after incubation with the synthetic inhibitors in five assays, three treated with bentazon and two with diuron. For each assay six increasing herbicide concentrations were applied together with a negative control, using different dose ranges on a logarithmic scale for the two treatments based on preliminary experiments to encompass the whole dose-response range.

The dataset is available in the package drc.

```
data(spinach)
spinach$CURVE <- as.factor(spinach$CURVE)
```

4.2.1 4 parameter log-logistic model

For the comparison of the two herbicides, two dose-response curves are fitted under assumption of a four parameter log-logistic model with a separate set of fixed effects coefficients for each treatment. Random effects are included for each of the four parameters to model the between assay variability. Using the information of the between assay variability by the additional distributional assumptions of the random intercepts is especially advantageous, as the dose levels for the two herbicides do not cover the same dose range.

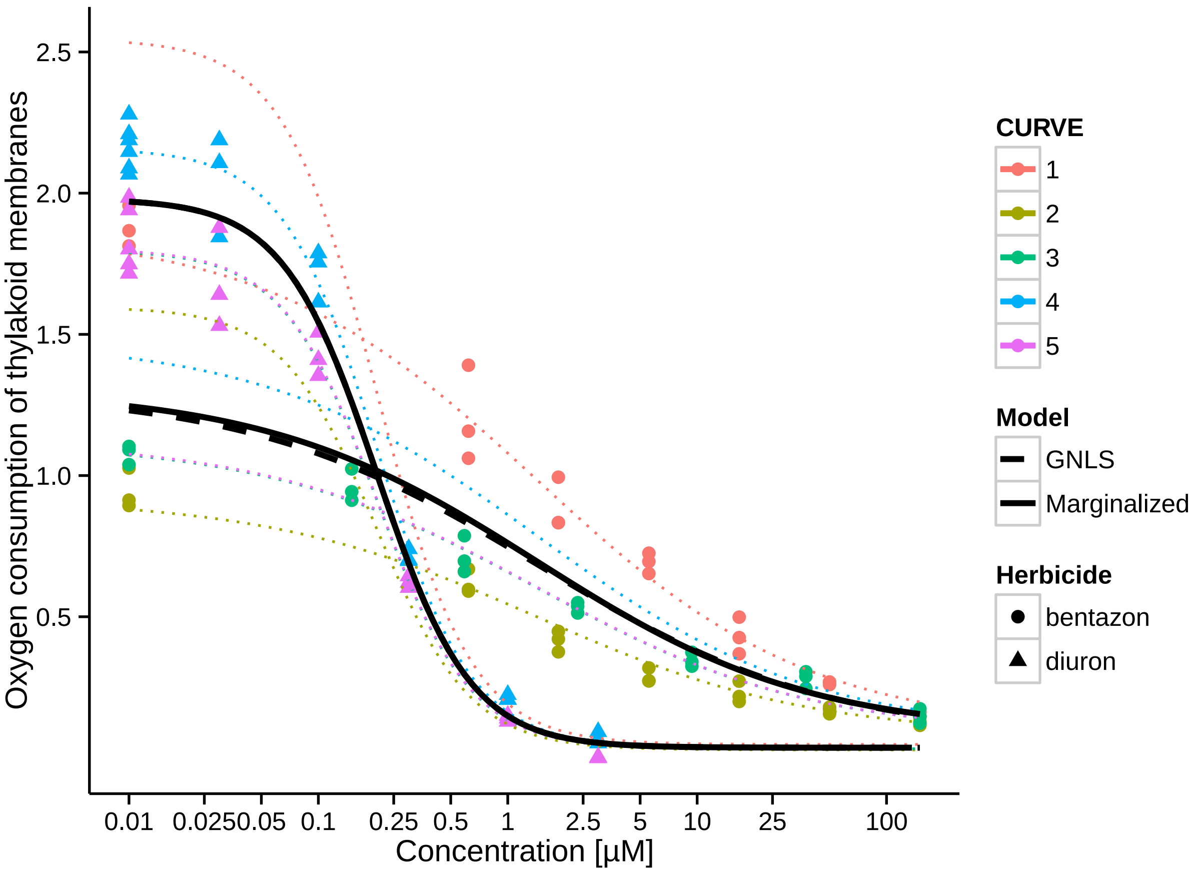
As for the Vinclozolin example, either a nonlinear mixed model or a generalized nonlinear least-squares model can be fitted.

```
### NLME
spm <- medrm(SLOPE ~ DOSE,
           curveid=b + c + d + e ~ HERBICIDE,
           data=spinach,
           random=b + c + d + e ~ 1|CURVE,
           fct=LL.4())

print(spm)

##
## A 'drc' model.
##
## Call:
## medrm(form = SLOPE ~ DOSE, curveid = b + c + d + e ~ HERBICIDE,     data = spinach, fct = LL.4(), ra
##
## Coefficients:
## b.bentazon    b.diuron  c.bentazon    c.diuron  d.bentazon    d.diuron
##     0.6041      1.7508      0.0862      0.0363      1.3036      1.9800
## e.bentazon    e.diuron
##     1.4330      0.2031

### GNLS
spmg <- glsdrm(SLOPE ~ DOSE,
           curveid=b + c + d + e ~ HERBICIDE,
           data=spinach,
           correlation=corCompSymm(form=~1|CURVE),
           fct=LL.4())

print(spmg)
```

```
##
## A 'drc' model.
##
## Call:
## glsdrm(form = SLOPE ~ DOSE, curveid = b + c + d + e ~ HERBICIDE,     data = spinach, fct = LL.4(), c
##
## Coefficients:
## b.bentazon    b.diuron  c.bentazon    c.diuron  d.bentazon    d.diuron
##     0.5656      1.7504      0.0680      0.0361      1.3008      1.9799
## e.bentazon    e.diuron
##     1.4446      0.2032
```

Like for the Vinclozolin example a user-defined graphic can be constructed, comparing the fitted GNLS curves with the marginalized predictions.

```
snd <- expand.grid(DOSE=exp(seq(log(0.01), log(150), length=100)),
                   HERBICIDE=levels(spinach$HERBICIDE))
snd1 <- snd2 <- snd
snd1$p <- predict(spm, type="marginal", newdata=snd)
snd1$model <- "Marginalized"
snd2$p <- predict(spmg, newdata=snd)
snd2$model <- "GNLS"
snd <- rbind(snd1, snd2)
snd3 <- expand.grid(DOSE=exp(seq(log(0.01), log(150), length=100)),
                    HERBICIDE=levels(spinach$HERBICIDE),
                    CURVE=as.factor(1:5))
snd3$p <- predict(spm, newdata=snd3)
spinach2 <- spinach
spinach2$DOSE[spinach2$DOSE == 0] <- 0.01

ggplot(spinach2, aes(x=DOSE, y=SLOPE, group=HERBICIDE, shape=HERBICIDE, colour=CURVE)) +
  geom_point(size=2.5) +
  geom_line(data=snd3, aes(y=p, group=CURVE:HERBICIDE), linetype=3) +
  geom_line(data=snd, aes(y=p, linetype=model, group=HERBICIDE:as.factor(snd$model),
                          shape=NULL, colour=NULL), size=1.1) +
  coord_trans(x="log") +
  theme_classic() +
  scale_linetype_manual("Model", values=c(2,1,4)) +
  scale_shape_discrete("Herbicide") +
  ylab("Oxygen consumption of thylakoid membranes") +
  xlab("Concentration [µM]") +
  scale_x_continuous(breaks=c(0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 1,2.5, 5,10,25,100),
                     labels=c(0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 1,2.5, 5,10,25,100))
```



4.2.2 Inference for the relative potency

The two fixed effect curves are compared by the ratio of effective dose estimates. As in this case also the marginal estimates are of interest, the marginalization approach can be similarly applied to the ratio of two parameters by extending the use of the delta method to calculate corresponding standard errors. Some tuning of the `SImarg()` function is needed, specifying a range of possible SI estimates by rfinterval.

```
SImarg(spm, c(10, 25, 50, 75, 90), percMat = matrix(1:5, 5, 2), interval = "delta",
    rfinterval = c(0, 100))

##
## Estimated ratios of effect doses
## (Delta method-based confidence interval(s))
##
##                         Estimate  Lower  Upper
## bentazon/diuron:10/10     0.651 -2.923   4.23
## bentazon/diuron:25/25     2.144 -2.600   6.89
## bentazon/diuron:50/50     7.056 -0.850  14.96
## bentazon/diuron:75/75    23.220  4.718  41.72
## bentazon/diuron:90/90    76.415 12.476 140.35

SI(spmg, c(10, 25, 50, 75, 90), percMat = matrix(1:5, 5, 2), interval = "delta")

##
## Estimated ratios of effect doses
## (Delta method-based confidence interval(s))
##
##                         Estimate   Lower Upper
## bentazon/diuron:10/10     0.513  -0.173   1.2
## bentazon/diuron:25/25     1.909   0.521   3.3
## bentazon/diuron:50/50     7.110   2.221  12.0
## bentazon/diuron:75/75    26.478  -7.214  60.2
## bentazon/diuron:90/90   98.601 -97.797 295.0
```

4.3 CellTiter - Blue Cell Viability Assay example

Neurotoxicity was tested by a cell viability assay SH-SY5Y cells for increasing concentrations of acrylamide (Frimat et al., 2010). The experiment was carried out over three days using a total of seven well plates, resulting in seven dose–response curves, measuring fluorescence at 11 or 12 doses and seven or eight replicates (672 observations in total).

```
data(ctb)
ctb$day <- as.factor(ctb$day)
ctb$dayplate <- as.factor(with(ctb, paste(day, plate, sep = "/")))
```

4.3.1 5 parameter log-logistic model with a fixed lower asymptote at 0

For the dose-response relationship a five-parameter log-logistic model is assumed, where the lower asymptote is fixed at zero under the premise of no cell viability at highly toxic dose levels. To fix the parameter of the lower asymptote, a fixed argument is in the `LL.5()` function call. The day effects and the variability of plates are modeled as hierarchical random effects, additively on the four parameters: steepness, upper asymptote, inflection point, and curve asymmetry. Additional to a large between day variability, an increase in residual variance can be observed with higher fluorescence measurements; this fact is represented in the model by including an exponential variance function, modeling the change in scale with increasing concentration levels.

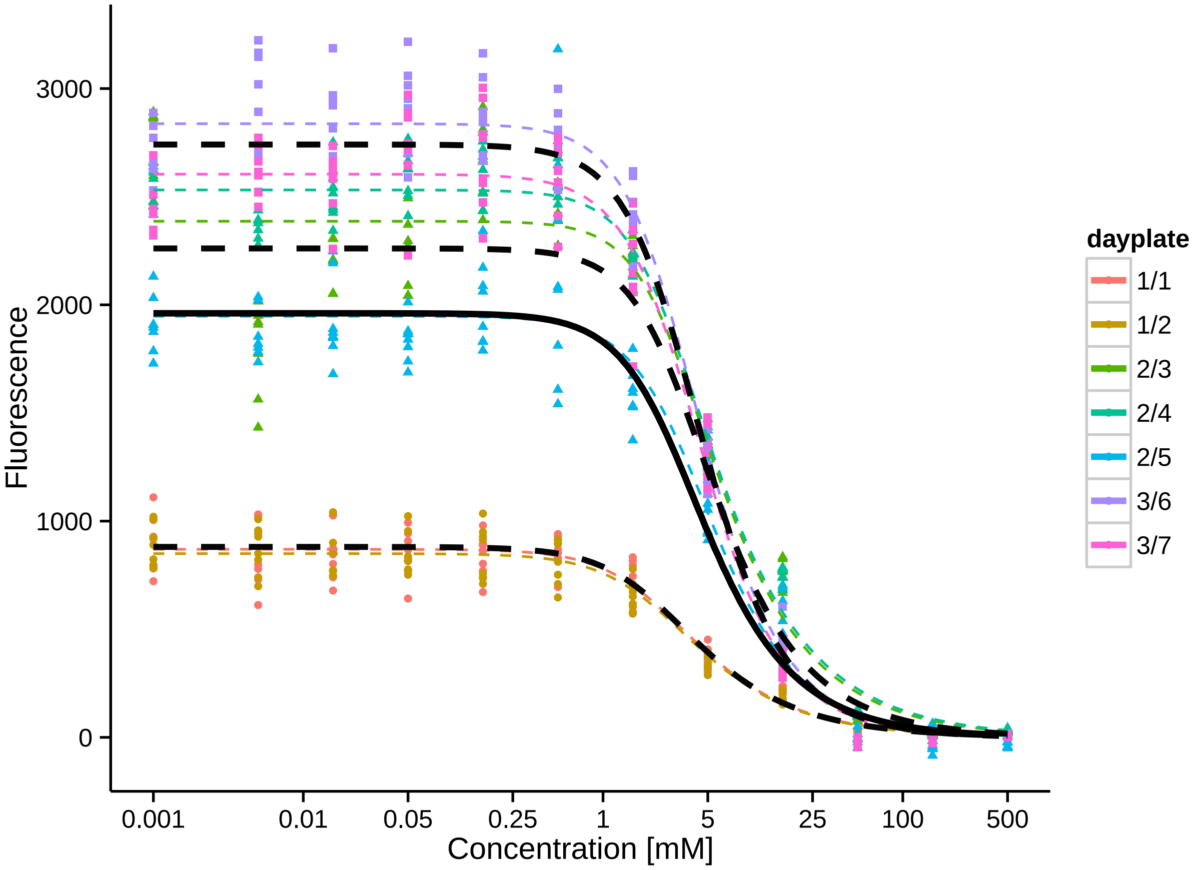
For convergence of the parameter estimation algorithm, user-defined starting values have to be provided for the fixed effects and also for a part of the random effects. These startingvalues are obtained by fitting separate nonlinear least-squares models for each random effect level.

```
# fixed effect starting values
fix <- coefficients(drm(fluorescence ~ conc, fct=LL.5(fixed=c(NA, 0, NA, NA, NA)), data=ctb))
# random effect starting values for days
rday <- drm(fluorescence ~ conc, curveid=day, fct=LL.5(fixed=c(NA, 0, NA, NA, NA)), data=ctb)
cmatday <- matrix(coefficients(rday), ncol=4)
mday <- apply(cmatday, 2, mean)
names(mday) <- c("b", "d", "e", "f")
rmatday <- t(apply(cmatday, 1, function(x) x-mday))
rownames(rmatday) <- levels(ctb$day)
colnames(rmatday) <- c("b", "d", "e", "f")

## medrm fit
mv <- medrm(fluorescence  ~ conc, fct=LL.5(fixed=c(NA, 0, NA, NA, NA)),
            data=ctb, random=b+d+e+f~1|day/plate,
            weights=varExp(form = ~conc|plate),
            start=list(fixed=fix, random=list(day=rmatday)),
            control=nlmeControl(maxIter=200))

summary(mv)
```

```
## Nonlinear mixed-effects model fit by maximum likelihood
##   Model: fluorescence ~ drcfunction(conc, b, d, e, f)
##  Data: data
##    AIC  BIC logLik
##   8224 8368  -4080
##
## Random effects:
##  Formula: list(b ~ 1, d ~ 1, e ~ 1, f ~ 1)
##  Level: day
##  Structure: General positive-definite, Log-Cholesky parametrization
##   StdDev   Corr
## b   0.1137 b      d      e
## d 792.4372  0.800
## e   0.4510  0.878  0.989
## f   0.0744 -0.030  0.576  0.452
##
##  Formula: list(b ~ 1, d ~ 1, e ~ 1, f ~ 1)
##  Level: plate %in% day
##  Structure: General positive-definite, Log-Cholesky parametrization
##          StdDev    Corr
## b           0.13170 b      d      e
## d         192.53892  0.689
## e           0.28000 -0.707 -1.000
## f           0.07804 -0.937 -0.898  0.909
## Residual 187.95763
##
## Variance function:
##  Structure: Exponential of variance covariate, different strata
##  Formula: ~conc | plate
##  Parameter estimates:
##         1         2         3         4         5         6         7
## -0.020699 -0.019855 -0.006667 -0.006856 -0.003358 -0.007158 -0.008127
## Fixed effects: b + d + e + f ~ 1
##    Value Std.Error  DF t-value p-value
## b    1.9       0.1 637  12.542       0
## d 1960.9     464.9 637   4.218       0
## e    2.9       0.3 637   8.761       0
## f    0.5       0.1 637   8.115       0
##  Correlation:
##   b      d      e
## d  0.372
## e -0.053  0.731
## f -0.648  0.314  0.612
##
## Standardized Within-Group Residuals:
##     Min      Q1     Med      Q3     Max
## -5.0732 -0.7041 -0.1431  0.4903  6.7450
##
## Number of Observations: 647
## Number of Groups:
##             day plate %in% day
##               3              7
```

Also for this model the predicted curves on the plate, day, and population-average level can be displayed.

```
cnd <- expand.grid(conc=exp(seq(log(0.001), log(500), length=100)))
cnd1 <- cnd
cnd1$p <- predict(mv$fit, level=0, newdata=cnd)
cnd2 <- expand.grid(conc=exp(seq(log(0.001), log(500), length=100)),
                    day=as.factor(levels(ctb$day)))
cnd2$p <- as.vector(apply(coef(mv$fit, level=1), 1,
                          function(x) mv$fct$fct(cnd$conc, rbind(x))))
cnd3 <- expand.grid(conc=exp(seq(log(0.001), log(500), length=100)),
                    dayplate=levels(ctb$dayplate))
cnd3$p <- as.vector(apply(coef(mv$fit, level=2), 1,
                          function(x) mv$fct$fct(cnd$conc, rbind(x))))
ctb2 <- ctb
ctb2$conc[ctb2$conc == 0] <- 0.001

ggplot(ctb2, aes(x=conc, y=fluorescence, group=dayplate, shape=day, colour=dayplate)) +
  geom_point(size=1.5) +
  geom_line(data=cnd3, aes(y=p, group=dayplate, shape=NULL), linetype=2) +
  geom_line(data=cnd2, aes(y=p, group=day, colour=NULL), size=1.1, linetype=2, colour="grey2") +
  geom_line(data=cnd1, aes(y=p, group=NULL, shape=NULL, colour=NULL), size=1.2) +
  coord_trans(x="log") +
  theme_classic() +
  guides(shape="none") +
  ylab("Fluorescence") +
  xlab("Concentration [mM]") +
  scale_x_continuous(breaks=c(0.001, 0.01, 0.05, 0.25, 1, 5,25,100,500),
                     labels=c(0.001, 0.01, 0.05, 0.25, 1, 5,25,100,500))
```



4.3.2 Benchmark dose calculation

The benchmark dose and the benchmark dose lower confidence limit can be calculated, marginalizing the estimate to obtain a population-average interpretation. Multiple benchmark risk levels are assumed and a background level of 0.05. To save some computation time, the Gauss quadrature is reduced to only two nodes for each random effect. As some BMDLs are computed near the boundary of a concentration at 0 μM, the interval limits are calculated on a logarithmic scale. The confidence level is specified at 0.9, to maintain an error level of 0.05 for the lower confidence limit (ignoring the upper limit).

```
BMDmarg(mv, respLev = c(5, 10, 25, 50), nGQ = 2, interval = "tfls", level = 0.9)

##
## Estimated benchmark doses
## (To and from log scale-based confidence interval(s))
##
##      Estimate Std. Error   Lower  Upper
## 1:5   0.69932    2.68151 0.00126 387.12
## 1:10  0.94316    2.20118 0.02018  44.08
## 1:25  1.36836    1.80236 0.15628  11.98
## 1:50  1.85279    1.61182 0.44205   7.77
```

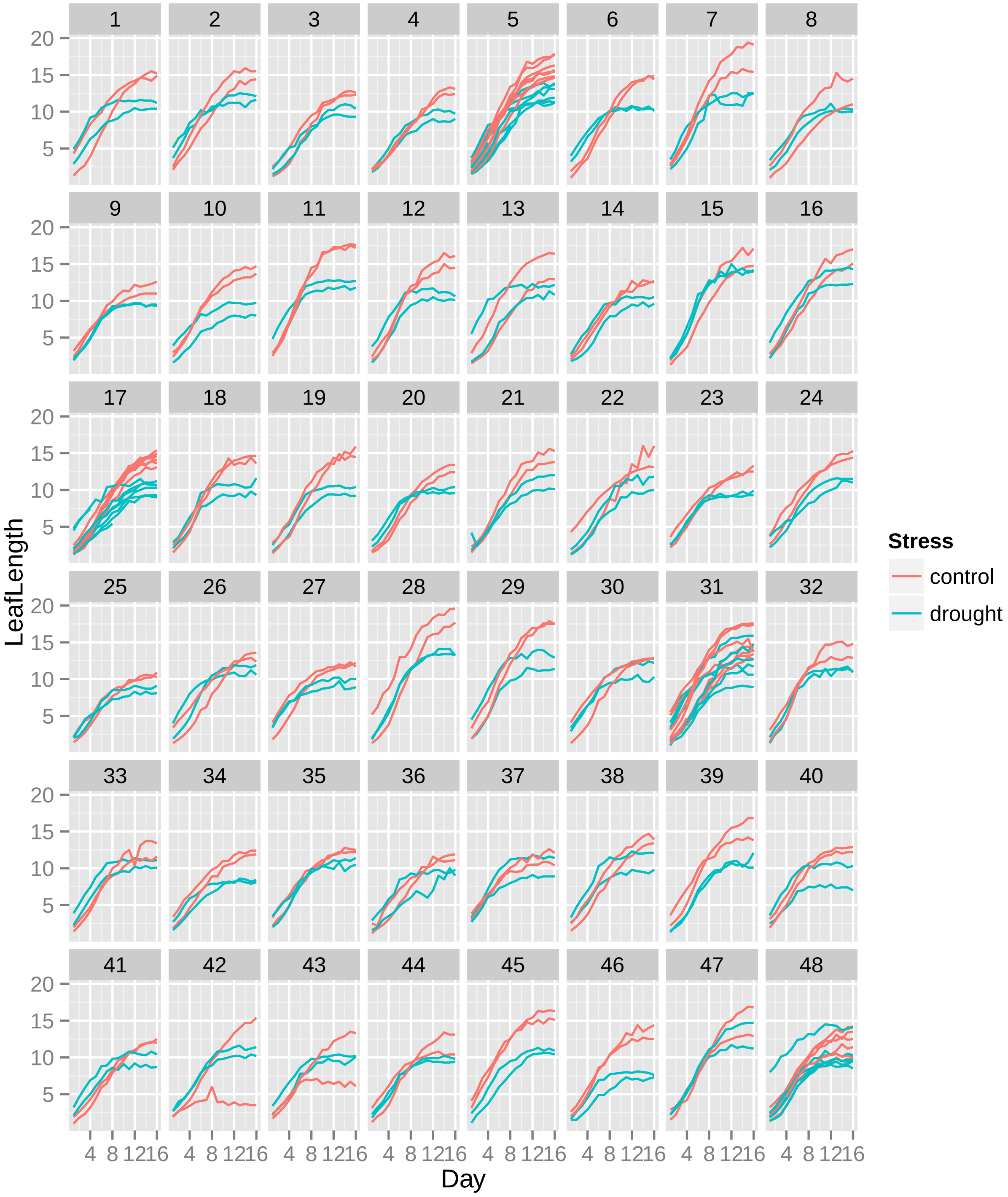
4.4 Broccoli example

The effect of drought stress on Brassica oleracea is investigated, selecting drought stress resistant varieties out of a population of different DH genotypes. The screening study was carried out on 48 DH lines developed from F1 plants of a cross between the rapid cycling chinese kale (*Brassica oleracea* var. *alboglabra* (L.H. Bailey) Musil) and broccoli (*Brassica oleracea* var. *italica* Plenck). Two stress treatments (not watered and a watered control) are randomly assigned to 4 plants per genotype (2 per treatment) resulting in 192 plants in total. For the genotypes 5, 17, 31, 48, additional 12 plants (6 per treatment) are included into the completely randomized design, which results in a total of 240 plants. For each plant the length of the youngest leaf at the beginning of the experiment is measured daily for a period of 16 days. For the additional 12 plants of the 4 genotypes the leaf water potential was measured as a secondary endpoint (omitted here); due to these destructive measurements some dropouts occur.

```
data(broccoli)
str(broccoli)
```

```
## 'data.frame': 3689 obs. of  5 variables:
##  $ LeafLength: num  1.4 1.2 2.5 1.8 1.7 1.4 5.1 5.5 3.1 1.8 ...
##  $ ID        : Factor w/ 240 levels "1","2","3","4",..: 38 62 35 91 76 108 149 237 223 239 ...
##  $ Stress    : Factor w/ 2 levels "control","drought": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Genotype  : Factor w/ 48 levels "1","2","3","4",..: 17 17 17 17 17 17 31 31 31 31 ...
##  $ Day       : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
ggplot(broccoli, aes(x=Day, y=LeafLength, group=ID, colour=Stress)) +
  geom_line() +
  facet_wrap(~ Genotype, ncol=8)
```



There are two individuals for genotypes 42 and 43 with growth curves, which are not representative for the population of interest, as their leaf growth has stopped in favor of early flowering. These curves are removed from the dataset.

```
bro <- droplevels(subset(broccoli, ID != "110" & ID != "125"))
```

4.4.1 5 parameter logistic model

Let us assume that the growth curves for each individual follow a 4 parameter log-logistic model with different parameters for each stress treatment except the lower asymptote. The model function can be conveniently defined as a predefined function of package drc. The parameterization of different fixed effect effect vectors per treatment level can be defined by a curveid argument, with the name of a fixed effect factor in the dataset on the right hand side of a formula, and the parameters to change for different curves on the left hand side.

The definition of random effects follow in a similar way. The random genotype effects are parameterized by a variance component for the upper asymptote, and the location of the inflection point. To model the dependency of observations measured at the same plant over time, a second set of random effects is assumed within the genotype effects to model the variance between individuals within genotypes.

For a simplification we ignore the real aim of the study to detect genotype specific stress effects and just model an additive stress effect for each genotype. The same lower asymptote (parameter c) is assumed for both stress treatment levels, as at the beginning of the experiment, all plants are watered at the same level.

```
m5pl <- medrm(LeafLength ~ Day, data=bro,
              fct=L.5(),
              curveid=b + d + e + f ~ Stress,
              random=d + e ~ 1|Genotype/ID)

print(m5pl)

##
## A 'drc' model.
##
## Call:
## medrm(form = LeafLength ~ Day, curveid = b + d + e + f ~ Stress,     data = bro, fct = L.5(), random
##
## Coefficients:
## b.control  b.drought  c.control  c.drought  d.control  d.drought
##    -0.322     -0.491      0.473      0.473     14.462     10.707
## e.control  e.drought  f.control  f.drought
##     2.624      4.364      2.101      0.881
```

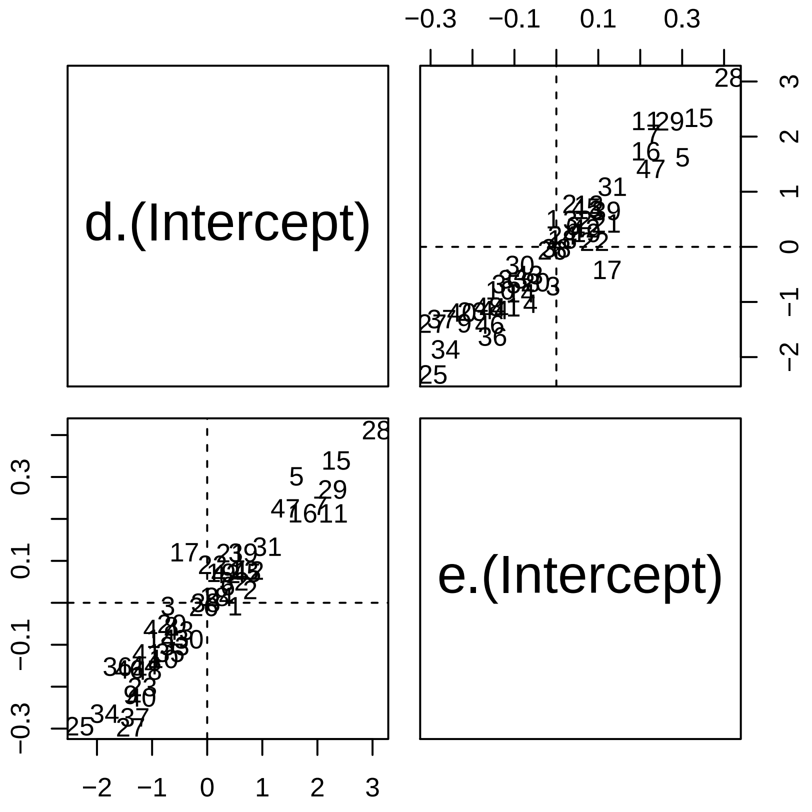
With the summary method, the summary output of the nlme component can be accessed. There are also some methods, like VarCorr, to access the specific nlme slots. For any special information about the nlme estimation, the nlme object is stored in a list slot named fit.

```
VarCorr(m5pl)

##                 Variance                      StdDev Corr
## Genotype =      pdLogChol(list(d ~ 1,e ~ 1))
## d.(Intercept) 1.86689                         1.3663 d.(In)
## e.(Intercept) 0.06088                         0.2467 0.726
## ID =            pdLogChol(list(d ~ 1,e ~ 1))
## d.(Intercept) 1.82951                         1.3526 d.(In)
## e.(Intercept) 1.28191                         1.1322 0.131
## Residual      0.10657                         0.3265

# same as
VarCorr(m5pl$fit)

##                 Variance                      StdDev Corr
## Genotype =      pdLogChol(list(d ~ 1,e ~ 1))
## d.(Intercept) 1.86689                         1.3663 d.(In)
## e.(Intercept) 0.06088                         0.2467 0.726
## ID =            pdLogChol(list(d ~ 1,e ~ 1))
## d.(Intercept) 1.82951                         1.3526 d.(In)
## e.(Intercept) 1.28191                         1.1322 0.131
## Residual      0.10657                         0.3265
```

The predicted random effects can be directly accessed by the function ranef().

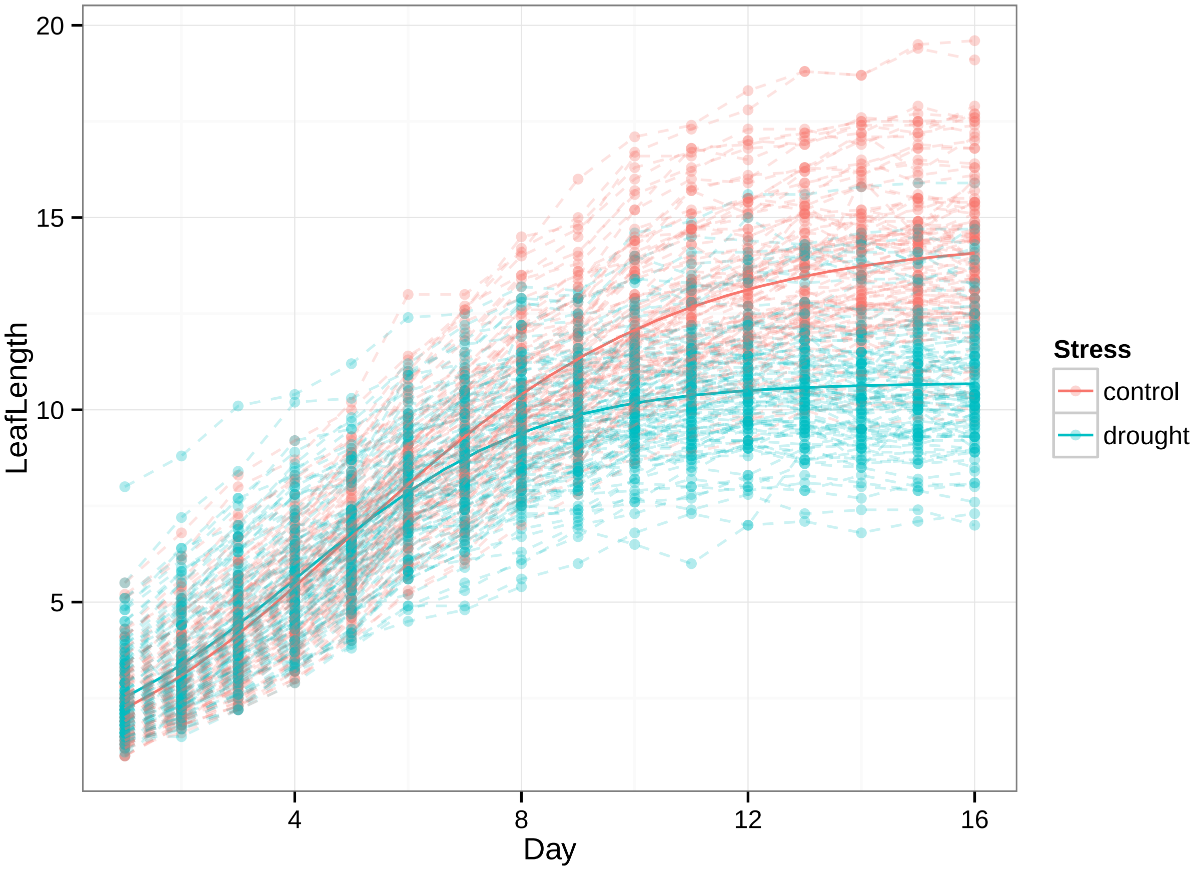
```
re <- ranef(m5pl)[[1]]
head(re)

##   d.(Intercept) e.(Intercept)
## 1        0.5049     -0.007898
## 2        0.7811      0.031048
## 3       -0.7137     -0.007855
## 4       -1.0287     -0.061984
## 5        1.6206      0.301007
## 6         0.3744       0.039826
```

```
panellab <- function(x, y, ...){
  abline(h=0, lty=2)
  abline(v=0, lty=2)
  text(x, y, rownames(re))
}
pairs(re, panel = panellab)
```



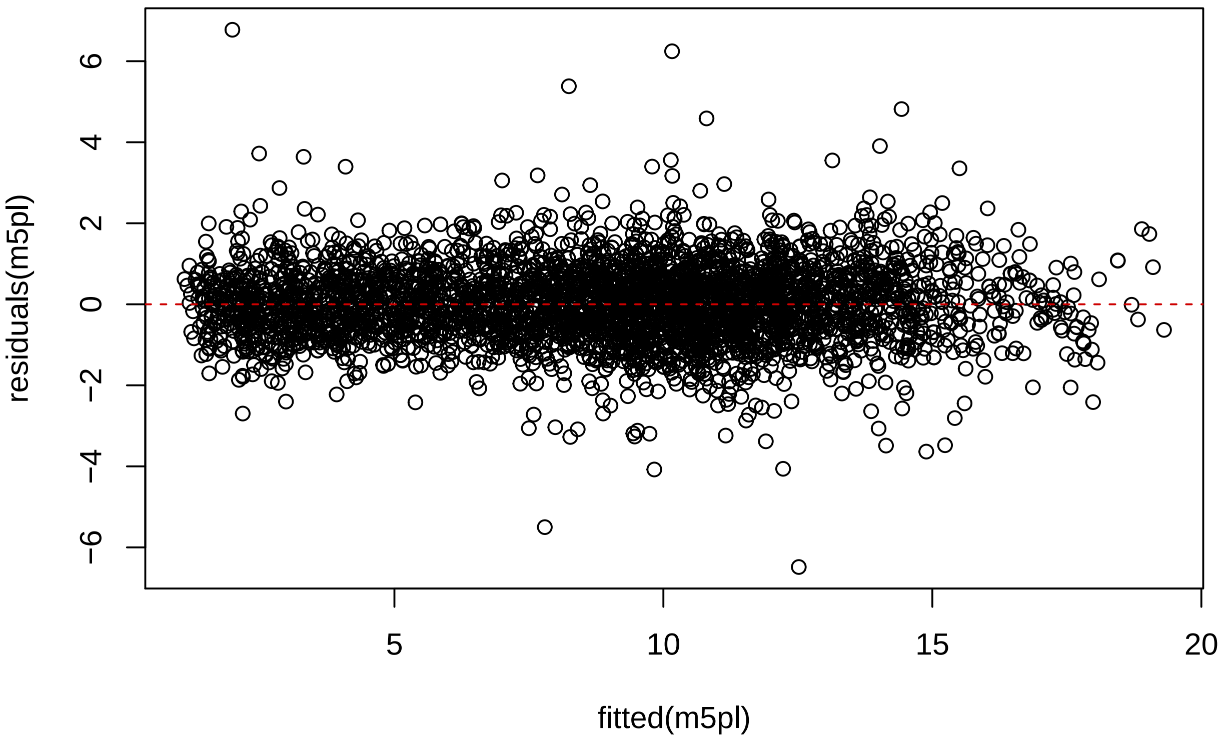
The growth curves based just on the fixed effects of the model can also be plotted. This ggplot object can be extended by some further ggplot2 functions. A plot of curves, conditional on the random effects, is not yet implemented when several fixed effects curves are present.

```
plot(m5pl) +
  geom_line(data=bro, aes(group=ID), linetype=2, alpha=0.2) +
  theme_bw()
```



Other diagnostic graphics, like residuals vs. fitted values, are also easily available.

```
plot(residuals(m5pl) ~ fitted(m5pl))
abline(h = 0, lty = 2, col = "red3")
```



4.4.2 Sets of dose-response models

A set of different log-logistic and Weibull models can be fitted for the broccoli data to capture the uncertainty of the growth curve shape.

```
# 3 parameter logistic with lower asymptote fixed at 0
mod1 <- medrm(LeafLength ~ Day, data=bro,
              fct=L.3(),
              curveid=b + d + e ~ Stress,
              random=d + e ~ 1|Genotype/ID)
# 4 parameter logistic
# with the same lower asymptote for both stress treatments
mod2 <- medrm(LeafLength ~ Day, data=bro,
              fct=L.4(fixed=c(NA, 5, NA, NA)),
              curveid=b + d + e ~ Stress,
              random=d + e ~ 1|Genotype/ID)
# 4 parameter Weibull model
mod3 <- medrm(LeafLength ~ Day, data=bro,
              fct=W1.4(),
              curveid=b + d + e ~ Stress,
              random=d + e ~ 1|Genotype/ID)
# 2nd parameterization of 4 parameter Weibull model
mod4 <- medrm(LeafLength ~ Day, data=bro,
              fct=W2.4(),
              curveid=b + d + e ~ Stress,
              random=d + e ~ 1|Genotype/ID)
# even a 4 parameter logistic with same parameters
# for both stress treatments is available with a onesided curveid formula
mod5 <- medrm(LeafLength ~ Day, data=bro,
              fct=L.4(),
              curveid= ~ Stress,
              random=d + e ~ 1|Genotype/ID)
# or a 4p-log-logistic model with a lower asymptote fixed at 5
mod6 <- medrm(LeafLength ~ Day, data=bro,
              fct=L.4(fixed=c(NA, 5, NA, NA)),
              curveid=d + e ~ Stress,
              random=d + e ~ 1|Genotype/ID)
```

The different curves can be displayed by the function mmplot().

```
mmplot(mod1, mod2, mod3, mod4, mod5, mod6, ndose=50)
```

4.4.3 Parameter inference

As the medrc class provides a coef and vcov method, simultaneous confidence intervals and multiple tests for the fixed effects are directly provided by the add-on packages multcomp. By multiple contrast tests, the comparison of drought stress to the control can be performed for each fixed effect parameter (except the lower asymptote).

```
library(multcomp)
K <- rbind("drought-control | b"=c(-1, 1,  0, 0,  0, 0,  0, 0,  0, 0),
           "drought-control | d"=c( 0, 0,  0, 0, -1, 1,  0, 0,  0, 0),
           "drought-control | e"=c( 0, 0,  0, 0,  0, 0, -1, 1,  0, 0),
           "drought-control | f"=c( 0, 0,  0, 0,  0, 0,  0, 0, -1, 1))
gg <- glht(m5pl, linfct=K)
summary(gg)

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: medrm(form = LeafLength ~ Day, curveid = b + d + e + f ~ Stress,
##     data = bro, fct = L.5(), random = d + e ~ 1 | Genotype/ID)
##
## Linear Hypotheses:
##                            Estimate Std. Error z value Pr(>|z|)
## drought-control | b == 0  -0.1694     0.0113  -14.95  < 1e-04 ***
## drought-control | d == 0  -3.7550     0.1835  -20.46  < 1e-04 ***
## drought-control | e == 0   1.7395     0.4559    3.82  0.00044 ***
## drought-control | f == 0  -1.2206     0.2467   -4.95  < 1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

4.4.4 Inference for derived parameters

The growth curve of the broccoli leaf lengths can be summarized by the effective dose, or in our case by the estimated day, at which 50% or any other percentage of the full length of a leaf is reached. Inference based on the fixed effects can be directly obtained by using the function ED in the drc package.

```
ED(m5pl, respLev = c(25, 50, 75), interval = "delta")

##
## Estimated effective doses
## (Delta method-based confidence interval(s))
##
##             Estimate Std. Error Lower Upper
## control:25     2.835      0.147 2.547  3.12
## control:50     5.542      0.125 5.297  5.79
## control:75     8.584      0.128 8.333  8.83
## drought:25     1.632      0.164 1.312  1.95
## drought:50     3.997      0.125 3.751  4.24
## drought:75    6.298      0.120 6.064  6.53
```

Comparisons of these ED parameters can be made with the drc function SI.

```
SI(m5pl, percVec = c(25, 25), interval = "delta")

##
## Estimated ratios of effect doses
## (Delta method-based confidence interval(s))
##
##                        Estimate Lower Upper
## control/drought:25/25      1.74  1.45  2.03
```

When a set of models are available, the model-averaged ED estimates can be estimated by the medrc function mmaED(). The model specific estimates are weighted by the corresponding information criteria for each model, e.g. using Akaike weights. It is recommended to use maximum likelihood estimation instead of REML, and not mixing simultaneous changes of fixed and random effect structures between the models.

```
mmaED(mod1, mod2, mod3, mod4, mod5, mod6, respLev = c(25, 50, 75), interval = "kang")
```

4.4.5 Marginal effective dose estimation

We might want to marginalize the ED of the broccoli model conditional on the estimated variance components by numerical integration methods. The medrc package provides functions EDmarg and SImarg, but assuming only simple random effect structures.

```
EDmarg(m5pl, respLev = c(25, 50, 75), interval = "delta", nGQ = 3)

##
## Estimated effective doses
## (Delta method-based confidence interval(s))
##
##            Estimate Std. Error Lower Upper
## control:25    2.762      0.111 2.544  2.98
## control:50    5.599      0.125 5.353  5.84
## control:75    8.743      0.203 8.345  9.14
## drought:25    1.522      0.116 1.294  1.75
## drought:50    4.034      0.131 3.776  4.29
## drought:75    6.490      0.204 6.089  6.89

SImarg(m5pl, percVec = c(25, 25), interval = "delta", nGQ = 3)

##
## Estimated ratios of effect doses
## (Delta method-based confidence interval(s))
##
##                        Estimate Lower Upper
## control/drought:25/25      1.81  1.52  2.11
```

In this case there are only small changes compared to the fixed effects estimates; a difference can be observed, when the variance component for the steepness, the location of the inflection point increases, or a variance component for the asymmetry parameter is introduced into the model.

References

Davidian, M. and Giltinan, D. M. (1995). *Nonlinear Models for Repeated Measurement Data*. Chapman \& Hall/CRC Monographs on Statistics \& Applied Probability, London.

Davidian, M. and Giltinan, D. M. (2003). Nonlinear models for repeated measurement data: An overview and update. *Journal of Agricultural, Biological, and Environmental Statistics*, 8(4):387–419.

Frimat, J.-P., Sisnaiske, J., Subbiah, S., Menne, H., Godoy, P., Lampen, P., Leist, M., Franzke, J., Hengstler,

J. G., van Thriel, C., and West, J. (2010). The network formation assay: a spatially standardized neurite outgrowth analytical display for neurotoxicity screening. *Lab on a Chip*, 10(6):701–709.

Nellemann, C., Majken, D., Lam, H. R., and Vinggaard, A. M. (2003). The Combined Effects of Vinclozolin and Procymidone Do Not Deviate from Expected Additivity in Vitro and in Vivo. *Toxicological Sciences*, 71(2):251–262.

Pinheiro, J. C. and Bates, D. M. (2000). *Mixed-Effects Models in S and S-Plus*. Statistics and Computing. Springer.

Ritz, C., Gerhard, D., and Hothorn, L. A. (2013). A Unified Framework for Benchmark Dose Estimation Applied to Mixed Models and Model Averaging. *Statistics in Biopharmaceutical Research*, 5(1):79–90.

Ritz, C. and Streibig, J. C. (2005). Bioassay analysis using R. *Journal of Statistical Software*, 12(5):1–22.

Smyth, G. K. (1998). Numerical integration. In Armitage, P. and Colton, T., editors, *Encyclopedia of Biostatistics*, pages 3088–3095. Wiley, London.

Streibig, J. C. and Dayan, F. E. (1999). Joint action of natural and synthetic photosystem II inhibitors. *Pesticide Science*, 146(June 1998):137–146.