A short introduction to the qpcrnlme package

Daniel Gerhard

June 25, 2013

1 Data Examples

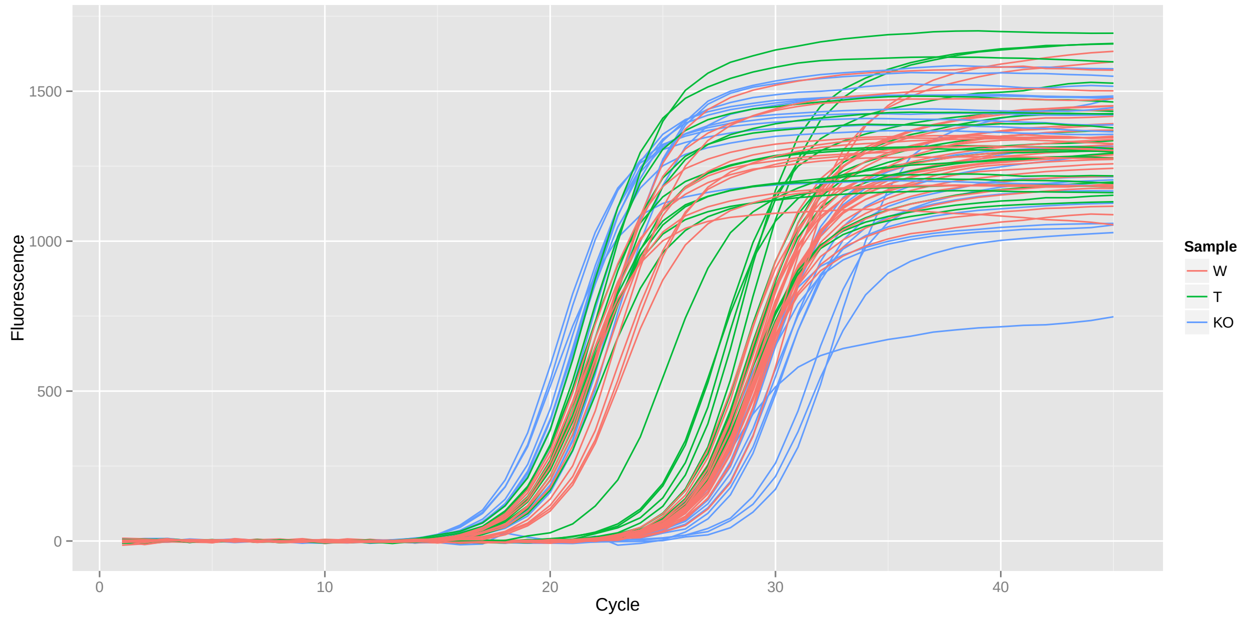
1.1 KOTWTcurves

For this experiment an insertion has been introduced into the genome of different rice lines. It is assumed that because of this insertion, the expression of the gene 10g30610 is changed. By using the quantitative RT-PCR methodology the changes in gene expression of the target gene is analysed between two different insertions and the wild type. `Content`: ID of biological replicates; `Target`: gene of interest and EF (elongation factor) gene of control; `Samples`: rice lines with an insertion (KO, T) and wild type (W).

```
library(qpcrnlme)
data(KOTWTcurves)
str(KOTWTcurves)
```

```
## 'data.frame': 3780 obs. of  6 variables:
##  $ Well        : Factor w/ 84 levels "A01","A02","A03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Content     : Factor w/ 12 levels "Unkn-01","Unkn-02",..: 1 1 1 2 2 2 3 3 3 4 ...
##  $ Target      : Factor w/ 2 levels "10g30610","EF": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Sample      : Factor w/ 3 levels "KO","T","W": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Cycle       : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Fluorescence: num  -2.06 2.5 4.81 5.75 4.08 ...
```

```
KOTWTcurves$Sample <- factor(KOTWTcurves$Sample, levels = c("W", "T", "KO"))
ggplot(KOTWTcurves, aes(x = Cycle, y = Fluorescence, colour = Sample, group = Well)) +
    geom_line()
```



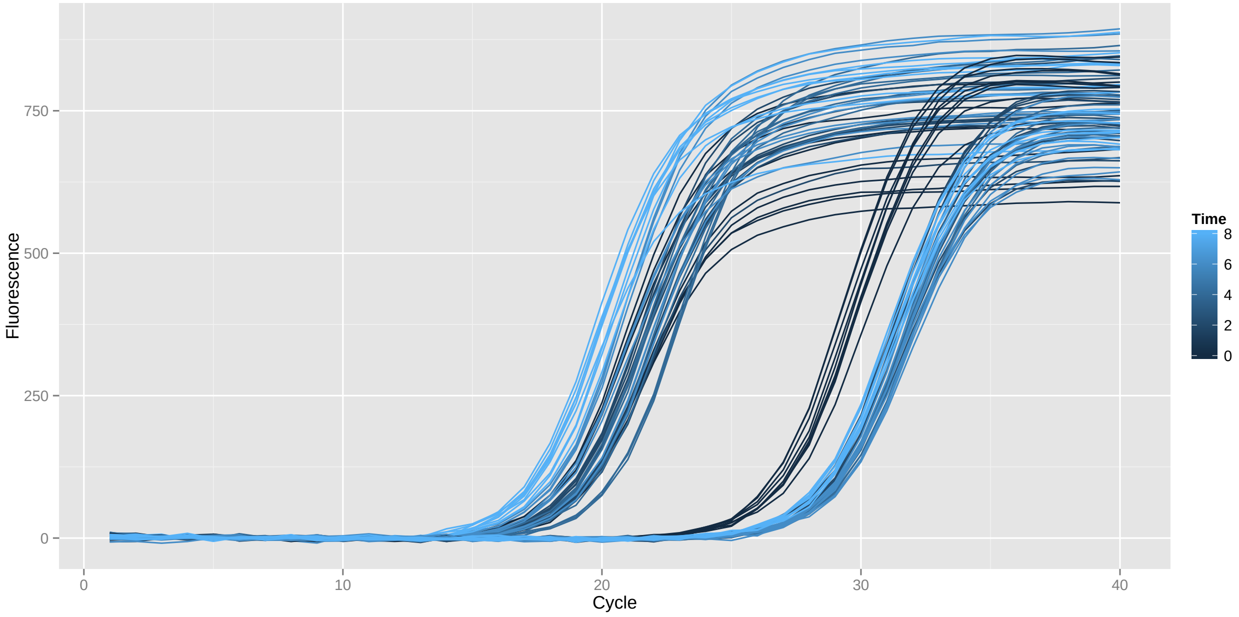
1.2 pt6c

In this experiment rice (*Oryza sativa* L.) plants were starved five days for phosphate resulting in an upregulation of the phosphate transporter PT6 to increase the uptake of the limited nutrient. Resupply of phosphate decreased the expression of PT6 after 0, 2, 4, 6, and 8 hours (variable `Time`). At each time point the expression of PT6 and the reference gene eEf (variable `Target`) were observed for three biological replicates (variable `Content`), which are thought of as representatives of the underlying biological system. For each biological replicate and gene the fluorescence intensity (variable `Fluorescence`) was measured over 40 PCR cycles (variable `Cycle`) for three technical replicates (variable `Well`).

```
data(pt6c)
str(pt6c)

## 'data.frame': 3600 obs. of  6 variables:
##  $ Cycle       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Fluorescence: num  3.42 3.077 2.707 -0.777 0.352 ...
##  $ Well        : Factor w/ 90 levels "A07","A08","A09",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Content     : Factor w/ 15 levels "Unkn-01","Unkn-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Time        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Target      : Factor w/ 2 levels "eEf","PT6": 1 1 1 1 1 1 1 1 1 1 ...

ggplot(pt6c, aes(x = Cycle, y = Fluorescence, colour = Time, group = Well)) +
    geom_line()
```



You may want to skip the model description and directly continue with the analysis of the data example in Section 4.

2 Modeling RT-PCR Data

We consider fluorescence intensities $\{y_{ijk}\}$ obtained from a hierarchical design using a number of biological replicates ($i = 1, \dots, I$), which are again divided into a number of technical replicates ($j = 1, \dots, J$), each with PCR cycle numbers c_k ($k = 1, \dots, K$).

2.1 Modeling the Fluorescence Curve

At the first stage we specify the nonlinear relationship for each individual fluorescence intensity:

$$y_{ijk} = f\left(c_k, \boldsymbol{\beta}_{ij} + \boldsymbol{u}_{ij}\right) + \epsilon_{ijk}, \tag{1}$$

assuming a specific nonlinear model function f depending on the cycle number c_k and R-dimensional vectors of fixed- and random-effects contributions $\boldsymbol{\beta}_{ij}$ and $\boldsymbol{u}_{ij}$, respectively. The error terms follow a normal distribution $N\left(0, \sigma^2\right)$.

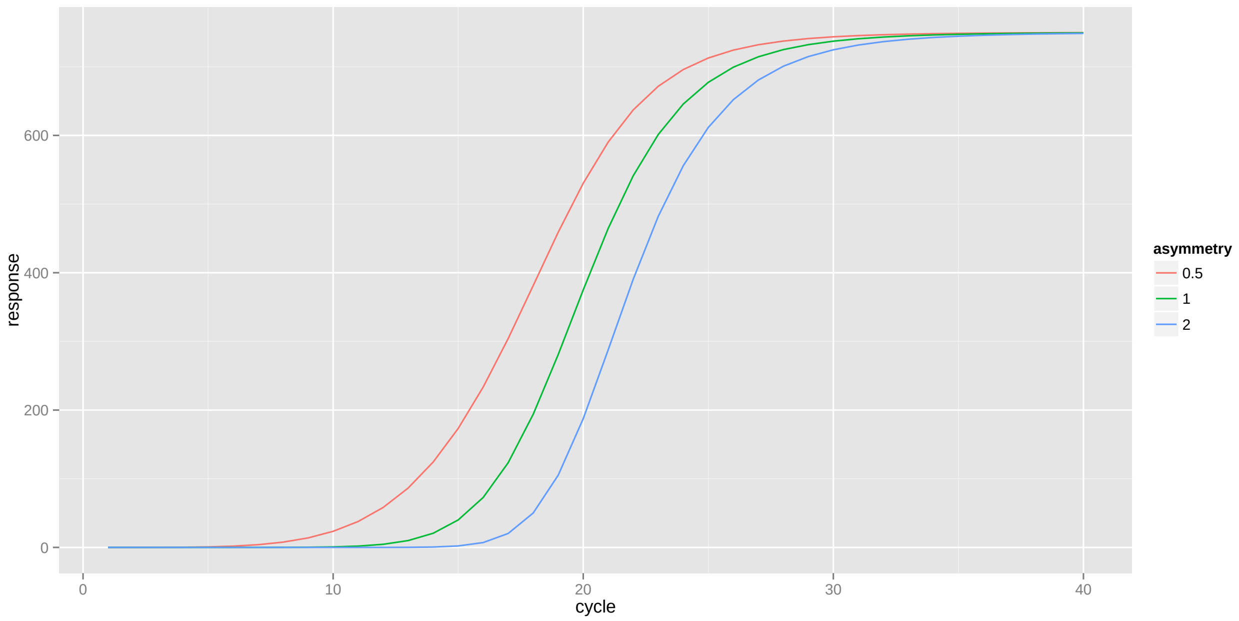
Until now, only the five-parameter log-logistic model is implemented in qpcrnlme:

$$
\begin{aligned}
f\left(c_k, \boldsymbol{\beta}_{ij}\right) =& f\{c_k, (\beta_{ij}^{(1)}, \ldots, \beta_{ij}^{(5)})\} \qquad (2)\\
=& \beta_{ij}^{(2)} + \frac{\beta_{ij}^{(3)} - \beta_{ij}^{(2)}}{(1+\exp[\beta_{ij}^{(1)}\{\log(c_j) - \log(\beta_{ij}^{(4)})\}])^{\beta_{ij}^{(5)}}}
\end{aligned}
$$

where β_{1ij} characterize the steepness in the s-shaped curve, β_{2ij} and β_{3ij} correspond to the lower and upper asymptotes, β_{4ij} denotes the approximate location of the inflection point, and β_{5ij} is an asymmetry parameter where positive values above or below 1 correspond to differences in curvature close to the lower and upper asymptotes.

This function can be evaluated by calling `logistic5()`. For example, the effect of the asymmetry parameter on the logistic function can be illustrated.

```r
cycle <- seq(1, 40, by = 1)
f05 <- llogistic5(cycle, b = -10, c = 0, d = 750, e = 20, f = 0.5)
f1 <- llogistic5(cycle, b = -10, c = 0, d = 750, e = 20, f = 1)
f2 <- llogistic5(cycle, b = -10, c = 0, d = 750, e = 20, f = 2)
asdat <- data.frame(cycle, response = c(f05, f1, f2), asymmetry = factor(rep(c(0.5,
    1, 2), each = length(cycle))))
ggplot(asdat, aes(x = cycle, y = response, colour = asymmetry)) + geom_line()
```



2.2 Hierarchical Structure of Biological and Technical Replicates

At the second stage we specify the decomposition of the model parameters at the level of the technical replicates. For the rth parameter in the vector $\boldsymbol{\beta}_{ij}$ the fixed- and random effects contributions are:

$$
\begin{aligned}
\boldsymbol{\beta}_{ij}^{(r)} &= \left(\boldsymbol{X}_T^{(r)}\right)_j \otimes \left(\boldsymbol{X}_B^{(r)}\right)_i \boldsymbol{\beta}, \\
\boldsymbol{u}_{ij}^{(r)} &= \left(\boldsymbol{Z}_B^{(r)}\right)_i \boldsymbol{u}_B^{(r)} + \left(\boldsymbol{Z}_T^{(r)}\right)_j \boldsymbol{u}_{Ti}^{(r)},
\end{aligned}
$$

where $\boldsymbol{X}_B^{(r)}$ $(I \times p_1)$ and $\boldsymbol{X}_T^{(r)}$ $(J \times p_2)$ denote the design matrices of the fixed-effects structures at the level of the biological and technical replicates, respectively (the subscript refers to specific rows in the matrices), and $\boldsymbol{\beta}$ denotes the $p_1 p_2$-dimensional fixed-effects parameter. Any kind of covariate information available at the level of the biological replicates could be included in the model. For our data example $\boldsymbol{X}_B^{(r)} = \boldsymbol{X}_B$ specifies a time trend that is assumed to be present in all r parameters, whereas $\boldsymbol{X}_T^{(r)} = \boldsymbol{X}_T$ groups technical replicates corresponding to the same gene into clusters (two clusters within each biological replicate as we consider two genes).

Similarly, $\boldsymbol{Z}_B^{(r)}$ $(I \times q_1)$ and $\boldsymbol{Z}_T^{(r)}$ $(J \times q_2)$ are the design matrices of the random effects associated with the biological and technical replicates within biological replicates, respectively. The random effects are assumed to follow normal distributions:

$$\boldsymbol{u}_B^{(r)} \sim N(\boldsymbol{0}, \boldsymbol{\Psi}_B^{(r)}),$$
$$\boldsymbol{u}_T^{(r)} = (\boldsymbol{u}_{T1}^{(r)}, \ldots, \boldsymbol{u}_{TI}^{(r)}) \sim N(0, \boldsymbol{\Psi}_T^{(r)}).$$

In principle $\boldsymbol{\Psi}_B^{(r)}$ and $\boldsymbol{\Psi}_T^{(r)}$ may be unstructured variance-covariance matrices, but we restrict ourselves to diagonal matrices that correspond to uncorrelated random effects, apart from letting $\boldsymbol{\Psi}_B^{(r)}$ be a block diagonal matrix with diagonal entries that are themselves diagonal matrices $\boldsymbol{\Psi}_{B1}^{(r)}, \ldots, \boldsymbol{\Psi}_{Bp_2}^{(r)}$ allowing different variance components for different groups [Davidian and Giltinan, 1995, pp. 122–124].

2.3 Implementation

To fit a hierarchical nonlinear model to the RT-PCR data, the qpcrnlme package makes use of the function `nlme` in the package `nlme`. For a fixed layout of the RT-PCR experiment, most of the model parameters are pre-specified, like a common upper asymptote for each gene and treatment, and the random effect structure for biological and technical replicates with different variance components for each gene. Starting values are found automatically by estimating several nonlinear models for each random effect level with the package drc; hence, no additional starting value has to be provided.

3 Marginal Cycle Number Estimation

It is established practice to evaluate real-time PCR data by means of the threshold cycle summary measure instead of directly interpreting the parameters in f. The threshold cycle is defined as the cycle number where the mean fluorescence level reaches a certain cutoff intensity t. Determining the threshold is an inverse regression problem that in case of a nonlinear regression model for a single replicate has the solution $c(t) = f^{-1}(t)$. For a nonlinear mixed model, the threshold cycle is obtained by solving the equation $E(f(c, \boldsymbol{\beta}_{0ij} + \boldsymbol{u}_0)) = t$ in c for some specific fixed-effects parameter configuration denoted $\boldsymbol{\beta}_{0ij}$. Solving the equation requires repeated evaluation of the integral by integrating out the random effects $\boldsymbol{u}_0$. The qpcrnlme package uses Gauss-Hermite quadrature to approximate these integrals with help of the package statmod.

The result of a RT-PCR data analysis with the qpcrnlme function is mainly a vector with marginal $c(t)$ estimates and the corresponding variance-covariances. Based on these estimates, specific linear combinations of the $c(t)$ values can be constructed. $\Delta\Delta c(t)$ values to compare treatment levels and genes can be calculated by function ddctcomp. For these derived parameters, hypotheses tests and confidence intervals are provided.

4 Treatment Comparisons

4.1 Pairwise-Comparisons to a Control

For a first evaluation, the two different insertion treatment groups in the KOTWTcurves dataset are compared to the wildtype group (W). The sample variable is a factor for this example, hence the design matrices for the fixed effects are structured in a way that for each gene-treatment level combination a single $c(t)$ value is estimated.

The model parameterization is defined by assigning the appropriate column names of the data.frame to the corresponding variable input argument, that is, a fluorescence response, a vector with cycle numbers, identifiers of treatment and gene factors, and factors specifying the structure of biological and technical replicates.

Some optional arguments can be changed:

- With the cutoff argument a $c(t)$ cutoff t can be specified. This value should lie within the range of the lower and upper asymptote.

- The nGQ defines the number of nodes and weights for the Gauss-Hermite approximation.

The marginal $c(t)$ estimates for each time group per gene are calculated by

```
ctest <- qpcr_nlme(response = "Fluorescence", cycle = "Cycle", gene = "Target",
    treatment = "Sample", brep = "Content", well = "Well", data = KOTWTcurves,
    cutoff = 100, nGQ = 1, verbose = FALSE)
print(ctest)
```

```
##
## c(t) estimates:
##                estimate std.err
## 10g30610:KO        29.6   0.380
## 10g30610:T         27.9   0.376
## 10g30610:W         28.4   0.361
## EF:KO              21.6   0.349
## EF:T               22.0   0.350
## EF:W               22.1   0.351
```

$\Delta\Delta c(t)$ comparisons to the control can be performed by the function ddctcomp. Besides the definition of a control gene and control treatment levels, following arguments can be modified:

- The conf.level argument specifies a confidence level of a $\Delta\Delta c(t)$ confidence interval.
- If the `ratio_ddct` argument has the value TRUE, ratios of $c(t)$ estimates are computed to define the $\Delta\Delta c(t)$, otherwise the difference of $c(t)$ values are calculated.
- If adjusted is TRUE, the family-wise error rate is controlled for all teatment comparisons. A single-step procedure is used, similar to the implementation in package multcomp, to obtain adjusted p-values and simultaneous confidence intervals. If adjusted equals FALSE, the per comparison error rate is used, assigning a separate type-I-error rate to each comparison.

```
ddctcomp(ctest, control_treatment = "W", control_gene = "EF", conf.level = 0.95,
    ratio_ddct = TRUE, adjusted = TRUE)
```

```
##
## delta delta c(t) estimates:
##                                 estimate std.err  lower upper p-value
## (W:EF / T:EF) / (W:103 / T:103)     0.9247  0.0271 0.8538  1.00   0.038 *
## (W:EF / KO:EF) / (W:103 / KO:103)   0.9352  0.0271 0.8645  1.01   0.071 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.2 All Pairwise-Comparisons

If no control treatment or control gene can be specified, the control lables can be set to NULL. In this case, all pairwise-comparison of treatment and/or gene groups are performed. With only two genes there will be no difference for the genewise comparisons to the comparisons to a control.

```
ddctcomp(ctest, control_treatment = NULL, control_gene = NULL, conf.level = 0.95,
    ratio_ddct = TRUE, adjusted = TRUE)
```

```
##
## delta delta c(t) estimates:
##                                 estimate std.err  lower upper p-value
## (W:103 / T:103) / (W:EF / T:EF)     1.0814  0.0317 0.9930  1.17   0.071 .
## (W:103 / KO:103) / (W:EF / KO:EF)   1.0693  0.0309 0.9830  1.16   0.117
## (T:103 / KO:103) / (T:EF / KO:EF)   0.9888  0.0288 0.9085  1.07   0.920
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.3 General Linear Contrasts

To provide the ability to define more general linear contrasts, as described for RT-PCR analysis in Steibel et al. [2009, p. 151], the R package multcomp can be used directly with the nlmect class. For example, the KO genotype can be compared with the average of the T and W group, for each single gene, and for the interaction contrast, comparing the differential expression of the treatments based on the comparison of gene 10g30610 with EF. Thus, for each $c(t)$ estimate in the `ctest` object, a contrast coefficient is assigned to compose the desired linear combination of $c(t)$ parameters. Simultaneous confidence intervals and adjusted p-values are given for each defined contrast.

```
library(multcomp)
K <- rbind("KO vs. ave(T, W) | 10g"=c(1, -0.5, -0.5, 0, 0, 0),
           "KO vs. ave(T, W) | EF"=c(0, 0, 0, 1, -0.5, -0.5),
           "KO vs. ave(T, W) | 10g vs EF"=c(1, -0.5, -0.5, -1, 0.5, 0.5))
gc <- glht(ctest, K)
summary(gc)

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Linear Hypotheses:
##                                    Estimate Std. Error z value Pr(>|z|)
## KO vs. ave(T, W) | 10g == 0          1.492      0.460    3.24   0.0031 **
## KO vs. ave(T, W) | EF == 0          -0.458      0.428   -1.07   0.5175
## KO vs. ave(T, W) | 10g vs EF == 0    1.950      0.628    3.10   0.0050 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

confint(gc)

##
##   Simultaneous Confidence Intervals
##
## Fit: NULL
##
## Quantile = 2.318
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##                                    Estimate lwr    upr
## KO vs. ave(T, W) | 10g == 0         1.492    0.425  2.559
## KO vs. ave(T, W) | EF == 0         -0.458   -1.449  0.534
## KO vs. ave(T, W) | 10g vs EF == 0  1.950    0.493  3.406
```

If instead of differences, the ratios of linear combinations of parameters are of interest, the gsci.ratio function in the package mratios can be applied, making use of Fiellers theorem to compute the simultaneous confidence intervals.

5 More general modeling options

If a more specific specification of the designmatrix of the fixed effects is needed, a formula interface to model the treatment effects per gene is available in the function `qpcr_nlme_formula`. Until now this interface is only restricted to apply the same design matrices for each of the nonlinear model parameters.

The pt6c dataset is used to illustrate the function `qpcr_nlme_formula`, which enables us to describe the change in differential expression over time by a regression model.

As an example, a cubic polynomial function is used to describe the differential gene-expression in time separately for each gene and each nonlinear model parameter except the upper asymptote.

```
poly4 <- qpcr_nlme_formula(response = "Fluorescence", cycle = "Cycle", gene = "Target",
    trtformula = ~poly(Time, 3, raw = TRUE), brep = "Content", well = "Well",
    data = pt6c, newdata = data.frame(Time = seq(1, 8, length = 5)), cutoff = 100,
    nGQ = 1, verbose = FALSE)
print(poly4)
```

```
##
## c(t) estimates:
##         estimate std.err
## eEf | 1     21.3  0.2078
## eEf | 2     21.5  0.2131
## eEf | 3     21.5  0.1869
## eEf | 4     21.0  0.2274
## eEf | 5     19.6  0.2574
## PT6 | 1     31.4  0.0711
## PT6 | 2     32.2  0.0733
## PT6 | 3     32.3  0.0665
## PT6 | 4     32.0  0.0807
## PT6 | 5     31.7  0.0900
```

6 Structure of a ddct Object

A ddct object contains all results of each modeling step. The content of this class of objects is shown for the comp2control results.

The output of the nonlinear mixed model is found in the nlme slot. Here, access to the estimated variance components, fixed- and random effects, fitted values, etc. is available.

```
print(ctest$nlme, correlation = FALSE)
```

```
## Nonlinear mixed-effects model fit by maximum likelihood
##   Model: response ~ llogistic5(cycle, b, c, d, e, f)
##   Data: dat
##   Log-likelihood: -14006
##   Fixed: list(b + c + e + f ~ gt - 1, d ~ 1)
## b.gt10g30610:KO  b.gt10g30610:T  b.gt10g30610:W       b.gtEF:KO
##        -18.4491        -17.0368        -16.2664        -15.4206
##         b.gtEF:T        b.gtEF:W c.gt10g30610:KO  c.gt10g30610:T
##        -14.1564        -13.9542          0.7219          0.9100
##  c.gt10g30610:W       c.gtEF:KO        c.gtEF:T        c.gtEF:W
##          1.0283          0.5636          0.7819          0.8942
## e.gt10g30610:KO  e.gt10g30610:T  e.gt10g30610:W       e.gtEF:KO
##         29.7328         28.0238         28.4537         21.8011
##         e.gtEF:T        e.gtEF:W f.gt10g30610:KO  f.gt10g30610:T
##         22.1852         22.2786          1.3144          1.3413
##  f.gt10g30610:W       f.gtEF:KO        f.gtEF:T        f.gtEF:W
##          1.5615          0.8812          1.0772          1.1302
##   d.(Intercept)
##       1331.8969
##
## Random effects:
##  Formula: list(d ~ gene - 1, e ~ gene - 1, b ~ gene - 1)
##  Level: brep
##  Structure: Diagonal
##         d.gene10g30610 d.geneEF e.gene10g30610 e.geneEF b.gene10g30610
## StdDev:          117.7    86.75         0.6823   0.6284         0.5519
##         b.geneEF
## StdDev: 0.001291
##
##  Formula: list(d ~ 1, e ~ 1, b ~ 1)
##  Level: well %in% brep
##  Structure: Diagonal
##             d e.(Intercept) b.(Intercept) Residual
## StdDev: 133.6        0.5062        0.4048    8.077
##
## Number of Observations: 3780
## Number of Groups:
##           brep well %in% brep
##             12             84
```

The marginal $c(t)$ estimates can be obtained from the slot ct, the corresponding variance-covariance matrix is available in the vcov slot.

```
print(ctest$ct)

##             estimate std.err
## 10g30610:KO    29.63  0.3795
## 10g30610:T     27.92  0.3762
## 10g30610:W     28.36  0.3605
## EF:KO          21.61  0.3488
## EF:T           22.02  0.3504
## EF:W           22.12  0.3507
```

```
print(ctest$vcov)

##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 1.440e-01 4.796e-07 4.134e-07 6.503e-07 5.861e-07 6.262e-07
## [2,] 4.796e-07 1.416e-01 1.225e-06 1.926e-06 1.736e-06 1.855e-06
## [3,] 4.134e-07 1.225e-06 1.300e-01 1.660e-06 1.497e-06 1.599e-06
## [4,] 6.503e-07 1.926e-06 1.660e-06 1.216e-01 2.354e-06 2.515e-06
## [5,] 5.861e-07 1.736e-06 1.497e-06 2.354e-06 1.228e-01 2.267e-06
## [6,] 6.262e-07 1.855e-06 1.599e-06 2.515e-06 2.267e-06 1.230e-01
```

References

Bates, D. M. and Watts, D. G. (1988). Nonlinear regression analysis and its applications, Wiley, New York.

Davidian, M. and Giltinan, D. M. (1995). Nonlinear Models for Repeated Measurement Data, Chapman and Hall, London.

Ritz, C. (2010). Towards a unified approach to dose-response modeling in ecotoxicology. *Environmental Toxicology & Chemistry* **29,** 220–229.

Spiess A. N., Feig C., Ritz C. (2008). Highly accurate sigmoidal fitting of real-time PCR data by introducing a parameter for asymmetry. *BMC Bioinformatics* **9,** 221.

Steibel J. P., Poletto R., Coussens P. M., Rosa G. J. M. (2009). A powerful and flexible linear mixed model framework for the analysis of relative quantification RT-PCR data. *Genomics* **94,** 146–152.