Package 'rtpcr'

January 8, 2026

Type Package

Title qPCR Data Analysis

Version 2.1.1

Description Tools for qPCR data analysis using Delta Ct and Delta Delta Ct methods, including t-tests, ANOVA, ANCOVA, repeated-measures models, and publication-ready visualizations. The package supports multiple target, and multiple reference genes, and uses a calculation framework adopted from Ganger et al. (2017) <doi:10.1186/s12859-017-1949-5> and Taylor et al. (2019) <doi:10.1016/j.tibtech.2018.12.002>, covering both the Livak and Pfaffl methods.

URL https://mirzaghaderi.github.io/rtpcr/, https://github.com/mirzaghaderi/rtpcr

License GPL-3

Imports multcomp, ggplot2, lmerTest, purrr, reshape2, tidyr, dplyr, grid, emmeans

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

NeedsCompilation no

Depends R (>= 3.5.0)

Suggests knitr, rmarkdown, multcompView

VignetteBuilder knitr

Author Ghader Mirzaghaderi [aut, cre, cph]

Maintainer Ghader Mirzaghaderi <gh.mirzaghaderi@uok.ac.ir>

Repository CRAN

Date/Publication 2026-01-08 05:20:07 UTC

Contents

`ANOVA_DCt` *Delta Ct ANOVA analysis*

Description

Performs ΔC_T analysis for target genes by applying ΔC_T method to each target gene. Target genes must be provided as paired efficiency (E) and Ct columns followed by the the reference gene(s) columns. See "Input data structure and column arrangement" in vignettes for details about data structure.

Usage

```
ANOVA_DCt(
  x,
  numOfFactors,
  numberOfrefGenes,
  block,
  alpha = 0.05,
  p.adj = "none",
  analyseAllTarget = TRUE
)
```

Arguments

`x`	A data frame containing experimental design columns, target gene E/Ct column pairs, and reference gene E/Ct column pairs. Reference gene columns must be located at the end of the data frame.
`numOfFactors`	Integer. Number of experimental factor columns (excluding `rep` and optional `block`).
`numberOfrefGenes`	Integer. Number of reference genes. Each reference gene must be represented by two columns (E and Ct).
`block`	Character or `NULL`. Name of the blocking factor column. When a qPCR experiment is done in multiple qPCR plates, variation resulting from the plates may interfere with the actual amount of gene expression. One solution is to conduct each plate as a randomized block so that at least one replicate of each treatment and control is present on a plate. Block effect is usually considered as random and its interaction with any main effect is not considered.

`alpha` statistical level for comparisons

`p.adj` Method for p-value adjustment. See `p.adjust`.

`analyseAllTarget`
Logical or character. If `TRUE` (default), all detected target genes are analysed. Alternatively, a character vector specifying the names (names of their Efficiency columns) of target genes to be analysed.

Value

An object containing expression table, lm models, ANOVA tables, residuals, raw data and ANOVA table for each gene.

ΔC_T **combined expression table** `object$combinedResults`

ANOVA table for treatments `object$perGene$gene_name$ANOVA_T`

ANOVA table factorial `object$perGene$gene_name$ANOVA_factorial`

lm ANOVA for tratments `object$perGene$gene_name$lm_T`

lm ANOVA factorial `object$perGene$gene_name$lm_factorial`

Residuals `resid(object$perGene$gene_name$lm_T)`

Examples

```
data <- read.csv(system.file("extdata", "data_3factor.csv", package = "rtpcr"))
res <- ANOVA_DCt(
  data,
  numOfFactors = 3,
  numberOfrefGenes = 1,
  block = NULL)
```

`ANOVA_DDCt` *Delta Delta Ct ANOVA analysis*

Description

Apply $\Delta\Delta C_T$ analysis to each target gene in the input data frame. Target and reference genes must be provided as paired efficiency (E) and Ct columns located after the experimental design columns. columns.

Usage

```
ANOVA_DDCt(
  x,
  numOfFactors,
  numberOfrefGenes,
  mainFactor.column,
  block,
  analysisType = "anova",
```

```
  mainFactor.level.order = NULL,
  p.adj = "none",
  plot = FALSE,
  plotType = "RE",
  analyseAllTarget = TRUE
)
```

Arguments

x	A data frame containing experimental design columns, target gene E/Ct column pairs, and reference gene E/Ct column pairs. Reference gene columns must be located at the end of the data frame.
numOfFactors	Integer. Number of experimental factor columns (excluding rep and optional block).
numberOfrefGenes	Integer. Number of reference genes. Each reference gene must be represented by two columns (E and Ct).
mainFactor.column	Column index or name of the factor for which relative expression is calculated. When analysisType = "ancova", remaining factors are treated as covariates.
block	Character or NULL. Name of the blocking factor column. When a qPCR experiment is done in multiple qPCR plates, variation resulting from the plates may interfere with the actual amount of gene expression. One solution is to conduct each plate as a randomized block so that at least one replicate of each treatment and control is present on a plate. Block effect is usually considered as random and its interaction with any main effect is not considered.
analysisType	Character string specifying the analysis type; one of "anova" (default) or "ancova".
mainFactor.level.order	Optional character vector specifying the order of levels for the main factor. If NULL, the first observed level is used as the calibrator. If provided, the first element of the vector is used as the calibrator level.
p.adj	Method for p-value adjustment. See p.adjust.
plot	Logical; if FALSE, per gene-plots are not generated.
plotType	Plot scale to use: "RE" for relative expression or "log2FC" for log2 fold change.
analyseAllTarget	Logical or character. If TRUE (default), all target genes are analysed. Alternatively, a character vector specifying the names (names of their Efficiency columns) of target genes to be analysed.

Details

$\Delta\Delta C_T$ analysis is performed for the mainFactor.column based on a full model factorial experiment by default. However, if ancova, the analysisType argument, analysis of covariance is performed for the levels of the mainFactor.column and the other factors are treated as covariates. if the interaction between the main factor and the covariate is significant, ANCOVA is not appropriate. ANCOVA is basically used when a factor is affected by uncontrolled quantitative covariate(s). For

example, suppose that wDCt of a target gene in a plant is affected by temperature. The gene may also be affected by drought. Since we already know that temperature affects the target gene, we are interested to know if the gene expression is also altered by the drought levels. We can design an experiment to understand the gene behavior at both temperature and drought levels at the same time. The drought is another factor (the covariate) that may affect the expression of our gene under the levels of the first factor i.e. temperature. The data of such an experiment can be analyzed by ANCOVA or using ANOVA based on a factorial experiment. ANCOVA is done even there is only one factor (without covariate or factor variable).

Value

An object containing expression table, lm models, residuals, raw data and ANOVA table for each gene.

$\Delta\Delta C_T$ **combined expression table** `object$combinedFoldChange`

ANOVA table `object$perGene$gene_name$ANOVA_table`

lm ANOVA `object$perGene$gene_name$lm_ANOVA`

lm ANCOVA `object$perGene$gene_name$lm_ANCOVA`

Residuals `resid(object$perGene$gene_name$lm_ANOVA)`

log2FC_Plot `object$perGene$gene_name$log2FC_Plot`

RE_Plot `object$perGene$gene_name$RE_Plot`

Examples

```
data1 <- read.csv(system.file("extdata", "data_2factorBlock3ref.csv", package = "rtpcr"))
ANOVA_DDCt(x = data1,
           numOfFactors = 2,
           numberOfrefGenes = 2,
           block = "block",
           mainFactor.column = 2,
           plot = FALSE,
           p.adj = "none")

data2 <- read.csv(system.file("extdata", "data_1factor_one_ref.csv", package = "rtpcr"))
ANOVA_DDCt(
           x = data2,
           numOfFactors = 1,
           numberOfrefGenes = 1,
           block = NULL,
           mainFactor.column = 1,
           plot = FALSE,
           p.adj = "none")
```

`efficiency` *Amplification efficiency statistics and standard curves*

Description

The `efficiency` function calculates amplification efficiency (E) and related statistics, including slope and coefficient of determination (R^2), and generates standard curves for qPCR assays.

Usage

```
efficiency(df)
```

Arguments

`df`	A data frame containing dilution series and corresponding Ct values. The first column should represent dilution levels, and the remaining columns should contain Ct values for different genes.

Details

Amplification efficiency is estimated from standard curves generated by regressing Ct values against the logarithm of template dilution. For each gene, the function reports the slope of the standard curve, amplification efficiency (E), and R^2 as a measure of goodness of fit. The function also provides graphical visualization of the standard curves.

Value

A list with the following components:

efficiency A data frame containing slope, amplification efficiency (E), and R^2 statistics for each gene.

Slope_compare A table comparing slopes between genes.

plot A `ggplot2` object showing standard curves for all genes.

Author(s)

Ghader Mirzaghaderi

Examples

```
# Load example efficiency data
data <- read.csv(system.file("extdata", "data_efficiency.csv", package = "rtpcr"))

# Calculate amplification efficiency and generate standard curves
efficiency(data)
```

Means_DDCt *Delta Ct or Delta Delta Ct pairwise comparisons using a fitted model*

Description

Performs relative expression (fold change) analysis based on the ΔC_T or $\Delta\Delta C_T$ methods using a fitted model object produced by `ANOVA_DCt()`, `ANOVA_DDCt()` or `REPEATED_DDCt()`.

Usage

```
Means_DDCt(model, specs, p.adj = "none")
```

Arguments

`model`	A fitted model object (typically an `lmer` or `lm` object) created by `ANOVA_DCt()`, `ANOVA_DDCt()` or `REPEATED_DDCt()`.
`specs`	A character string or character vector specifying the predictors or combinations of predictors over which relative expression values are desired. This argument follows the specification syntax used by `emmeans::emmeans()` (e.g., `"Factor"`, `"Factor1 | Factor2"`).
`p.adj`	Character string specifying the method for adjusting p-values. See `p.adjust` for available options.

Details

The `Means_DDCt` function performs pairwise comparisons of relative expression values fo all combinations using estimated marginal means derived from a fitted model. For ANOVA models, relative expression values can be obtained for main effects, interactions, and sliced (simple) effects. For ANCOVA models returned by the **rtpcr** package, only simple effects are supported.

Internally, this function relies on the **emmeans** package to compute marginal means and contrasts, which are then back-transformed to fold change values using the $\Delta\Delta C_T$ framework.

Value

A data frame containing estimated relative expression values, confidence intervals, p-values, and significance levels derived from the fitted model.

Author(s)

Ghader Mirzaghaderi

Examples

```
# Obtain a fitted model from ANOVA_DDCt
res <- ANOVA_DDCt(
  data_3factor,
  numOfFactors = 3,
  numberOfrefGenes = 1,
  mainFactor.column = 1,
  block = NULL)

# Relative expression values for Type main effect
Means_DDCt(res$perGene$E_PO$lm_ANOVA, specs = "Type")

# Relative expression values for Concentration main effect
Means_DDCt(res$perGene$E_PO$lm_ANOVA, specs = "Conc")

# Relative expression values for Concentration sliced by Type
Means_DDCt(res$perGene$E_PO$lm_ANOVA, specs = "Conc | Type")

# Relative expression values for Concentration sliced by Type and SA
Means_DDCt(res$perGene$E_PO$lm_ANOVA, specs = "Conc | Type * SA")




data <- read.csv(system.file("extdata", "data_3factor.csv", package = "rtpcr"))
res <- ANOVA_DCt(
  data,
  numOfFactors = 3,
  numberOfrefGenes = 1,
  block = NULL)

# lm <- res$perGene$PO$lm_factorial
# Means_DDCt(lm, specs = "Type * Conc * SA", p.adj = "none")
```

---

meanTech *Calculate mean of technical replicates*

Description

Computes the arithmetic mean of technical replicates for each sample or group. This is often performed before ANOVA or other statistical analyses to simplify comparisons between experimental groups.

Usage

```
meanTech(x, groups)
```

Arguments

x	A raw data frame containing technical replicates.
groups	An integer vector or character vector specifying the column(s) to group by before calculating the mean of technical replicates.

Details

The meanTech function calculates the mean of technical replicates based on one or more grouping columns. This reduces the dataset to a single representative value per group, facilitating downstream analysis such as fold change calculation or ANOVA.

Value

A data frame with the mean of technical replicates for each group.

Author(s)

Ghader Mirzaghaderi

Examples

```
# Example input data frame with technical replicates
data1 <- read.csv(system.file("extdata", "data_withTechRep.csv", package = "rtpcr"))

# Calculate mean of technical replicates using first four columns as groups
meanTech(data1, groups = 1:4)

# Another example using different dataset and grouping columns
data2 <- read.csv(system.file("extdata", "Lee_etal2020qPCR.csv", package = "rtpcr"))
meanTech(data2, groups = 1:3)
```

---

multiplot *Combine multiple ggplot objects into a single layout*

Description

The multiplot function arranges multiple ggplot2 objects into a single plotting layout with a specified number of columns.

Usage

```
multiplot(..., cols = 1)
```

Arguments

...	One or more ggplot2 objects.
cols	Integer specifying the number of columns in the layout.

Details

Multiple ggplot2 objects can be provided either as separate arguments via .... The function uses the grid package to control the layout.

Value

A grid object displaying multiple plots arranged in the specified layout.

Author(s)

Pedro J. (adapted from https://gist.github.com/pedroj/ffe89c67282f82c1813d)

Examples

```
# Example using output from TTEST_DDCt
data1 <- read.csv(system.file("extdata", "data_1factor_one_ref.csv", package = "rtpcr"))
a <- TTEST_DDCt(
  data1,
  numberOfrefGenes = 1,
  plotType = "log2FC")
p1 <- a$plot

# Example using output from ANOVA_DCt
data2 <- read.csv(system.file("extdata", "data_1factor.csv", package = "rtpcr"))
out2 <- ANOVA_DCt(
  data2,
  numOfFactors = 1,
  numberOfrefGenes = 1,
  block = NULL)

df <- out2$combinedResults

p2 <- plotFactor(
  df,
  x_col = "SA",
  y_col = "RE",
  Lower.se_col = "Lower.se.RE",
  Upper.se_col = "Upper.se.RE",
  letters_col = "sig",
  letters_d = 0.1,
  col_width = 0.7,
  err_width = 0.15,
  fill_colors = "skyblue",
  alpha = 1,
  base_size = 16)

# Combine plots into a single layout
multiplot(p1, p2, cols = 2)

multiplot(p1, p2, cols = 1)
```

`plotFactor` *Bar plot of gene expression for 1-, 2-, or 3-factor experiments*

Description

Creates a bar plot of relative gene expression (fold change) values from 1-, 2-, or 3-factor experiments, including error bars and statistical significance annotations.

Usage

```
plotFactor(
  data,
  x_col,
  y_col,
  Lower.se_col,
  Upper.se_col,
  group_col = NULL,
  facet_col = NULL,
  letters_col = NULL,
  letters_d = 0.2,
  col_width = 0.8,
  err_width = 0.15,
  dodge_width = 0.8,
  fill_colors = NULL,
  alpha = 1,
  base_size = 12,
  legend_position = "right",
  ...
)
```

Arguments

`data`	Data frame containing expression results
`x_col`	Character. Column name for x-axis
`y_col`	Character. Column name for bar height
`Lower.se_col`	Character. Column name for lower SE
`Upper.se_col`	Character. Column name for upper SE
`group_col`	Character. Column name for grouping bars (optional)
`facet_col`	Character. Column name for faceting (optional)
`letters_col`	Character. Column name for significance letters (optional)
`letters_d`	Numeric. Vertical offset for letters (default `0.2`)
`col_width`	Numeric. Width of bars (default `0.8`)
`err_width`	Numeric. Width of error bars (default `0.15`)
`dodge_width`	Numeric. Width of dodge for grouped bars (default `0.8`)

fill_colors	Optional vector of fill colors
alpha	Numeric. Transparency of bars (default 1)
base_size	Numeric. Base font size for theme (default 12)
legend_position	Character or numeric vector. Legend position (default right)
...	Additional ggplot2 layer arguments

Value

ggplot2 plot object

Author(s)

Ghader Mirzaghaderi

Examples

```
data <- read.csv(system.file("extdata", "data_2factorBlock.csv", package = "rtpcr"))
res <- ANOVA_DCt(data,
    numOfFactors = 2,
    block = "block",
    numberOfrefGenes = 1)

df <- res$combinedResults

p1 <- plotFactor(
  data = df,
  x_col = "factor2",
  y_col = "RE",
  group_col = "factor1",
  Lower.se_col = "Lower.se.RE",
  Upper.se_col = "Upper.se.RE",
  letters_col = "sig",
  letters_d = 0.2,
  fill_colors = c("aquamarine4", "gold2"),
  alpha = 1,
  col_width = 0.7,
  dodge_width = 0.7,
  base_size = 16,
  legend_position = c(0.2, 0.8))

p1



data <- read.csv(system.file("extdata", "data_3factor.csv", package = "rtpcr"))
#Perform analysis first
res <- ANOVA_DCt(
  data,
  numOfFactors = 3,
  numberOfrefGenes = 1,
```

```
  block = NULL)

df <- res$combinedResults
 df
 # Generate three-factor bar plot
 p <- plotFactor(
  df,
  x_col = "SA",
  y_col = "log2FC",
  group_col = "Type",
  facet_col = "Conc",
  Lower.se_col = "Lower.se.log2FC",
  Upper.se_col = "Upper.se.log2FC",
  letters_col = "sig",
  letters_d = 0.3,
  col_width = 0.7,
  dodge_width = 0.7,
  fill_colors = c("blue", "brown"),
  base_size = 14,
  alpha = 1,
  legend_position = c(0.1, 0.2))
p
library(ggplot2)
p + theme(
  panel.border = element_rect(color = "black", linewidth = 0.5))
```

REPEATED_DDCt *Delta Delta Ct ANOVA analysis on repeated measure data*

Description

REPEATED_DDCt function performs $\Delta\Delta C_T$ method analysis of observations repeatedly taken over different time courses. Data may be obtained over time from a uni- or multi-factorial experiment. Target genes must be provided as paired efficiency (E) and Ct columns followed by the E/Ct column pairs of reference genes.

Usage

```
REPEATED_DDCt(
  x,
  numOfFactors,
  numberOfrefGenes,
  repeatedFactor,
  calibratorLevel,
  block,
  p.adj = "none",
  plot = FALSE,
  analyseAllTarget = TRUE
)
```

Arguments

x	input data frame in which the first column is id, followed by the factor column(s) which include at least time. The first level of time in data frame is used as calibrator or reference level. Additional factor(s) may also be present. Other columns are efficiency and Ct values of target and reference genes. In the id column, a unique number is assigned to each individual from which samples have been taken over time, for example see data_repeated_measure_1, all the three number 1 indicate one individual which has been sampled over three different time courses. See example data sets or refer vignettes, section "Input data structure" for details.
numOfFactors	Integer. Number of experimental factor columns (excluding optional block).
numberOfrefGenes	Integer. Number of reference genes. Each reference gene must be represented by two columns (E and Ct).
repeatedFactor	Character string specifying the factor for which fold changes are analysed (commonly "time").
calibratorLevel	A level of repeatedFactor to be used as the calibrator (reference level) which is the reference level or sample that all others are compared to. Examples are untreated or time 0.
block	Character or NULL. Name of the blocking factor column. When a qPCR experiment is done in multiple qPCR plates, variation resulting from the plates may interfere with the actual amount of gene expression. One solution is to conduct each plate as a randomized block so that at least one replicate of each treatment and control is present on a plate. Block effect is usually considered as random and its interaction with any main effect is not considered.
p.adj	Method for p-value adjustment. See p.adjust.
plot	Logical; if FALSE, plots are not produced.
analyseAllTarget	Logical or character. If TRUE (default), all detected target genes are analysed. Alternatively, a character vector specifying the names (names of their Efficiency columns) of target genes to be analysed.

Details

Column layout requirements for x:

- Target gene columns: E/Ct column pairs located between design and reference columns
- Reference gene columns: E/Ct column pairs located at the end

Value

An object containing expression table, lm models, residuals, raw data and ANOVA table for each gene.

$\Delta\Delta C_T$ **combined expression table** object$Relative_Expression_table

ANOVA table `object$perGene$gene_name$ANOVA_table`

lm ANOVA `object$perGene$gene_name$lm`

Residuals `resid(object$perGene$gene_name$lm)`

log2FC_Plot `object$perGene$gene_name$log2FC_Plot`

RE_Plot `object$perGene$gene_name$RE_Plot`

Examples

```
data1 <- read.csv(system.file("extdata", "data_repeated_measure_1.csv", package = "rtpcr"))
REPEATED_DDCt(
  data1,
  numOfFactors = 1,
  numberOfrefGenes = 1,
  repeatedFactor = "time",
  calibratorLevel = "1",
  block = NULL)



data2 <- read.csv(system.file("extdata", "data_repeated_measure_2.csv", package = "rtpcr"))
REPEATED_DDCt(
  data2,
  numOfFactors = 2,
  numberOfrefGenes = 1,
  repeatedFactor = "time",
  calibratorLevel = "1",
  block = NULL,
  p.adj = "none",
  plot = FALSE,
  analyseAllTarget = TRUE)
```

TTEST_DDCt *Delta Delta Ct method t-test analysis*

Description

The `TTEST_DDCt` function performs fold change expression analysis based on the $\Delta\Delta C_T$ method using Student's t-test. It supports analysis of one or more target genes evaluated under two experimental conditions (e.g. control vs treatment).

Usage

```
TTEST_DDCt(
  x,
  numberOfrefGenes,
  Factor.level.order = NULL,
  paired = FALSE,
```

```
  var.equal = TRUE,
  p.adj = "none",
  order = "none",
  plotType = "RE"
)
```

Arguments

x	A data frame containing experimental conditions, biological replicates, and amplification efficiency and Ct values for target and reference genes. The number of biological replicates must be equal across genes. If this is not true, or there are NA values use ANODA_DDCt function for independent samples or REPEATED_DDCt for paired samples. See the package vignette for details on the required data structure.
numberOfrefGenes	Integer specifying the number of reference genes used for normalization.
Factor.level.order	Optional character vector specifying the order of factor levels. If NULL, the first level of the factor column is used as the calibrator.
paired	Logical; if TRUE, a paired t-test is performed.
var.equal	Logical; if TRUE, equal variances are assumed and a pooled variance estimate is used. Otherwise, Welch's t-test is applied.
p.adj	Method for p-value adjustment. One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", or "none". See p.adjust.
order	Optional character vector specifying the order of genes in the output plot.
plotType	Plot scale to use: "RE" for relative expression or "log2FC" for log2 fold change.

Details

Relative expression values are computed using one or more reference genes for normalization. Both paired and unpaired experimental designs are supported.

Paired samples in quantitative PCR refer to measurements collected from the same individuals under two different conditions (e.g. before vs after treatment), whereas unpaired samples originate from different individuals in each condition. Paired designs allow within-individual comparisons and typically reduce inter-individual variability.

The function returns numerical summaries as well as bar plots based on either relative expression (RE) or log2 fold change (log2FC).

Value

A list with the following components:

Result Table containing RE values, log2FC, p-values, significance codes, confidence intervals, standard errors, and lower/upper SE limits.

RE_Plot Bar plot of relative expression values.

log2FC_Plot Bar plot of log2 fold change values.

Author(s)

Ghader Mirzaghaderi

References

Livak, K. J. and Schmittgen, T. D. (2001). Analysis of Relative Gene Expression Data Using Real-Time Quantitative PCR and the Double Delta CT Method. *Methods*, 25(4), 402–408. doi:10.1006/meth.2001.1262

Ganger, M. T., Dietz, G. D., and Ewing, S. J. (2017). A common base method for analysis of qPCR data and the application of simple blocking in qPCR experiments. *BMC Bioinformatics*, 18, 1–11.

Yuan, J. S., Reed, A., Chen, F., and Stewart, N. (2006). Statistical Analysis of Real-Time PCR Data. *BMC Bioinformatics*, 7, 85.

Examples

```
# Example data structure
data1 <- read.csv(system.file("extdata", "data_1factor_one_ref.csv", package = "rtpcr"))

# Unpaired t-test
TTEST_DDCt(
  data1,
  paired = FALSE,
  var.equal = TRUE,
  numberOfrefGenes = 1)

# With amplification efficiencies
data2 <- read.csv(system.file("extdata", "data_1factor_one_ref_Eff.csv", package = "rtpcr"))

TTEST_DDCt(
  data2,
  paired = FALSE,
  var.equal = TRUE,
  numberOfrefGenes = 1)

# Two reference genes
data3 <- read.csv(system.file("extdata", "data_1factor_Two_ref.csv", package = "rtpcr"))
TTEST_DDCt(
  data3,
  numberOfrefGenes = 2,
  var.equal = TRUE,
  p.adj = "BH")
```

Index