Package 'rtpcr'

July 7, 2024

Type Package

Title qPCR Data Analysis

Version 2.0.0

Description Various methods are employed for statistical analysis and graphical presentation of real-time PCR (quantitative PCR or qPCR) data. 'rtpcr' handles amplification efficiency calculation, statistical analysis and graphical representation of real-time PCR data based on up to two reference genes. By accounting for amplification efficiency values, 'rtpcr' was developed using a general calculation method described by 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. Based on the experimental conditions, the functions of the 'rtpcr' package use t-test (for experiments with a two-level factor), analysis of variance (ANOVA), analysis of covariance (ANCOVA) or analysis of repeated measure data to calculate the fold change (FC, Delta Delta Ct method) or relative expression (RE, Delta Ct method). The functions further provide standard errors and confidence intervals for means, apply statistical mean comparisons and present significance. To facilitate function application, different data sets were used as examples and the outputs were explained. 'rtpcr' package also provides bar plots using various controlling arguments. The 'rtpcr' package is user-friendly and easy to work with and provides an applicable resource for analyzing real-time PCR data.

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

License GPL-3

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

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

NeedsCompilation yes

Depends R (>= 3.5.0)

Suggests knitr, rmarkdown

VignetteBuilder knitr

Author Ghader Mirzaghaderi [aut, cre, cph]

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

Repository CRAN

Date/Publication 2024-07-07 10:30:06 UTC

Contents

`efficiency` *Slope, R2 and Efficiency (E) statistics and standard curves*

Description

The `efficiency` function calculates amplification efficiency and returns related statistics and standard curves.

Usage

```
efficiency(df)
```

Arguments

`df`	a data frame of dilutions and Ct of genes. First column is dilutions and other columns are Ct values for different genes.

Details

The `efficiency` function calculates amplification efficiency of genes, and present the Slope, Efficiency, and R2 statistics.

Value

A list 3 elements.

efficiency Slope, R2 and Efficiency (E) statistics

Slope_compare slope comparison table

plot standard curves

Author(s)

Ghader Mirzaghaderi

Examples

```
# locate and read the sample data
data_efficiency

# Applying the efficiency function
efficiency(data_efficiency)
```

`meanTech` *Calculating mean of technical replicates*

Description

Calculating arithmetic mean of technical replicates for subsequent ANOVA analysis

Usage

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

Arguments

`x`	A raw data frame including technical replicates.
`groups`	grouping columns based on which the mean technical replicates are calculated.

Details

The meanTech calculates mean of technical replicates. Arithmetic mean of technical replicates can be calculated in order to simplify the statistical comparison between sample groups.

Value

A data frame with the mean of technical replicates.

Author(s)

Ghader Mirzaghaderi

Examples

```
# See example input data frame:
data_withTechRep

# Calculating mean of technical replicates
meanTech(data_withTechRep, groups = 1:4)

# Calculating mean of technical replicates
meanTech(Lee_etal2020qPCR, groups = 1:3)
```

---

multiplot *Multiple plot function*

Description

`multiplot` function combines multiple ggplot objects into a single plate.

Usage

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

Arguments

`...`	ggplot objects can be passed in ... or to plotlist (as a list of ggplot objects)
`cols`	Number of columns in the panel

Details

Combining multiple ggplot objects into a single plate.

Value

A multiple-plots plate

Author(s)

gist.github.com/pedroj/ffe89c67282f82c1813d

Examples

```
p1 <- qpcrTTESTplot(data_ttest,
                    numberOfrefGenes = 1,
                    ylab = "Average Fold Change (FC)",
                    width = 0.3)


out2 <- qpcrANOVARE(data_1factor, numberOfrefGenes = 1, block = NULL)$Result
p2 <- oneFACTORplot(out2,
                    width = 0.4,
                    fill = "skyblue",
                    y.axis.adjust = 0.5,
                    y.axis.by = 1,
                    errorbar = "ci",
                    show.letters = TRUE,
                    letter.position.adjust = 0.1,
                    ylab = "Relative Expression (RE)",
                    xlab = "Factor Levels",
                    fontsize = 12)

multiplot(p1, p2, cols=2)

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

`oneFACTORplot` *Bar plot of the relative gene expression (ΔC_T method) from the* `qpcrANOVARE` *output of a single-factor experiment data*

Description

Bar plot of the relative expression of a gene along with the standard error (se), 95% confidence interval (ci) and significance. `oneFACTORplot` is mainly useful for a one-factor experiment with more than two levels.

Usage

```
oneFACTORplot(
  res,
  width = 0.4,
  fill = "skyblue",
  y.axis.adjust = 0.5,
  y.axis.by = 2,
  errorbar,
  show.letters = TRUE,
  letter.position.adjust = 0.1,
```

```
  ylab = "Relative Expression",
  xlab = "none",
  fontsize = 12,
  fontsizePvalue = 5,
  axis.text.x.angle = 0,
  axis.text.x.hjust = 0.5
)
```

Arguments

`res`	an FC data frame object created by `qpcrANOVARE(x)$Result` function on a one factor data such as `data_1factor`.
`width`	a positive number determining bar width.
`fill`	specify a fill color.
`y.axis.adjust`	a negative or positive number for reducing or increasing the length of the y axis.
`y.axis.by`	determines y axis step length.
`errorbar`	Type of error bar, can be `se` or `ci`.
`show.letters`	a logical variable. If TRUE, mean grouping letters are added to the bars.
`letter.position.adjust`	adjust the distance between the grouping letters to the error bars.
`ylab`	the title of the y axis.
`xlab`	the title of the x axis.
`fontsize`	size of all fonts of the plot.
`fontsizePvalue`	font size of the pvalue labels
`axis.text.x.angle`	angle of x axis text
`axis.text.x.hjust`	horizontal justification of x axis text

Details

The `oneFACTORplot` function generates the bar plot of the average fold change for target genes along with the significance and the 95% confidence interval as error bars.

Value

Bar plot of the average fold change for target genes along with the significance and the standard error or 95% confidence interval as error bars.

Author(s)

Ghader Mirzaghaderi

Examples

```
# Before plotting, the result needs to be extracted as below:
res <- qpcrANOVARE(data_1factor, numberOfrefGenes = 1, block = NULL)$Result

# Bar plot
oneFACTORplot(res,
         width = 0.4,
         fill = "skyblue",
         y.axis.adjust = 1,
         y.axis.by = 0.2,
         errorbar = "se",
         show.letters = TRUE,
         letter.position.adjust = 0.05,
         ylab = "Relative Expression",
         xlab = "Factor Levels",
         fontsize = 12)
```

---

qpcrANOVAFC *Fold change ($\Delta\Delta C\_T$ method) analysis using ANOVA and ANCOVA*

Description

Fold change ($\Delta\Delta C_T$ method) analysis of qPCR data can be done using ANOVA (analysis of variance) and ANCOVA (analysis of covariance) by the qpcrANOVAFC function, for uni- or multi-factorial experiment data. The bar plot of the fold changes (FC) values along with the standard error (se) and confidence interval (ci) is returned by the qpcrANOVAFC function.

Usage

```
qpcrANOVAFC(
  x,
  numberOfrefGenes,
  mainFactor.column,
  analysisType = "anova",
  mainFactor.level.order = NULL,
  block,
  width = 0.5,
  fill = "#BFEFFF",
  y.axis.adjust = 2,
  y.axis.by = 1,
  letter.position.adjust = 0.1,
  ylab = "Fold Change",
  xlab = "none",
  fontsize = 12,
  fontsizePvalue = 7,
  axis.text.x.angle = 0,
```

```
  axis.text.x.hjust = 0.5,
  x.axis.labels.rename = "none",
  p.adj = "none",
  errorbar = "se",
  plot = TRUE
)
```

Arguments

x	a data frame of condition(s), biological replicates, efficiency (E) and Ct values of target and reference genes. Each Ct value in the data frame is the mean of technical replicates. **NOTE:** Each line belongs to a separate individual reflecting a non-repeated measure experiment). See vignette, section "data structure and column arrangement" for details.
numberOfrefGenes	number of reference genes which is 1 or 2 (up to two reference genes can be handled).
mainFactor.column	the factor for which fold change expression is calculated for its levels. If ancova is selected as analysisType, the remaining factors (if any) are considered as covariate(s).
analysisType	should be one of "ancova" or "anova". Default is "anova".
mainFactor.level.order	NULL (default) or a vector of main factor level names. If NULL, the first level of the mainFactor.column is used as calibrator. If a vector of main factor levels (in any order) is specified, the first level in the vector is used as calibrator. Calibrator is the reference level or sample that all others are compared to. Examples are untreated of time 0. The FC value of the reference or calibrator level is 1 because it is not changed compared to itself.
block	column name of the block if there is a blocking factor (for correct column arrangement see example data.). 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 complete 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.
width	a positive number determining bar width.
fill	specify the fill color for the columns in the bar plot. If a vector of two colors is specified, the reference level is differentialy colored.
y.axis.adjust	a negative or positive value for reducing or increasing the length of the y axis.
y.axis.by	determines y axis step length
letter.position.adjust	adjust the distance between the signs and the error bars.
ylab	the title of the y axis
xlab	the title of the x axis

`fontsize`	font size of the plot
`fontsizePvalue`	font size of the pvalue labels
`axis.text.x.angle`	angle of x axis text
`axis.text.x.hjust`	horizontal justification of x axis text
`x.axis.labels.rename`	a vector replacing the x axis labels in the bar plot
`p.adj`	Method for adjusting p values
`errorbar`	Type of error bar, can be `se` or `ci`.
`plot`	if `FALSE`, prevents the plot.

Details

Fold change ($\Delta\Delta C_T$ method) analysis of qPCR data can be done using ANOVA (analysis of variance) and ANCOVA (analysis of covariance) by the `qpcrANOVAFC` function, for uni- or multi-factorial experiment data. If there are more than one factor, FC value calculations for the 'mainFactor.column' and the statistical analysis is performed based on a full model factorial experiment by default. However, if 'ancova' is defined for the 'analysisType' argument, FC values are calculated for the levels of the 'mainFactor.column' and the other factors are used as covariate(s) in the analysis of variance, but we should consider ANCOVA table: if the interaction between the main factor and covariate is significant, ANCOVA is not appropriate in this case. 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 using `qpcrANOVAFC`. `qpcrANOVAFC` function performs FC analysis even there is only one factor (without covariate or factor variable). Bar plot of fold changes (FC) values along with the standard errors are also returned by the `qpcrANOVAFC` function.

Value

A list with 7 elements:

Final_data Input data frame plus the weighted Delat Ct values (wDCt)

lm_ANOVA lm of factorial analysis-tyle

lm_ANCOVA lm of ANCOVA analysis-type

ANOVA_table ANOVA table

ANCOVA_table ANCOVA table

FC Table Table of FC values, significance and confidence interval and standard error with the lower and upper limits for the main factor levels.

Bar plot of FC values Bar plot of the fold change values for the main factor levels.

Author(s)

Ghader Mirzaghaderi

References

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

Ganger, MT, Dietz GD, and Ewing SJ. 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, Joshua S, Ann Reed, Feng Chen, and Neal Stewart. 2006. Statistical Analysis of Real-Time PCR Data. BMC Bioinformatics 7 (85). doi:10.1186/1471-2105-7-85.

Examples

```
qpcrANOVAFC(data_1factor, numberOfrefGenes = 1, mainFactor.column = 1, block = NULL,
fill = c("#CDC673", "#EEDD82"), fontsizePvalue = 5, y.axis.adjust = 0.1)


qpcrANOVAFC(data_2factor, numberOfrefGenes = 1, mainFactor.column = 2, block = NULL,
analysisType = "ancova", fontsizePvalue = 5, y.axis.adjust = 1)


# Data from Lee et al., 2020, Here, the data set contains technical
# replicates so we get mean of technical replicates first:
df <- meanTech(Lee_etal2020qPCR, groups = 1:3)
qpcrANOVAFC(df, numberOfrefGenes = 1, analysisType = "ancova", block = NULL,
mainFactor.column = 2, fill = c("skyblue", "#BFEFFF"), y.axis.adjust = 0.05)


qpcrANOVAFC(data_2factorBlock,  numberOfrefGenes = 1, mainFactor.column = 1,
mainFactor.level.order = c("S", "R"), block = "block",
fill = c("#CDC673", "#EEDD82"), analysisType = "ancova",
fontsizePvalue = 7, y.axis.adjust = 0.1, width = 0.35)


df <- meanTech(Lee_etal2020qPCR, groups = 1:3)
df2 <- df[df$factor1 == "DSWHi",][-1]
qpcrANOVAFC(df2, mainFactor.column = 1, fontsizePvalue = 5, y.axis.adjust = 2,
block = NULL, numberOfrefGenes = 1, analysisType = "anova")


addline_format <- function(x,...){gsub('\\s','\n',x)}
qpcrANOVAFC(data_1factor, numberOfrefGenes = 1, mainFactor.column = 1,
block = NULL, fill = c("skyblue","#79CDCD"), y.axis.by = 1,
letter.position.adjust = 0, y.axis.adjust = 1, ylab = "Fold Change",
fontsize = 12, x.axis.labels.rename = addline_format(c("Control",
                                        "Treatment_1 vs Control",
                                        "Treatment_2 vs Control")))
```

qpcrANOVARE *Relative expression (ΔC_T method) analysis using ANOVA*

Description

Analysis of variance of relative expression (ΔC_T method) values for all factor level combinations in the experiment in which the expression level of a reference gene is used as normalizer.

Usage

```
qpcrANOVARE(x, numberOfrefGenes, block, alpha = 0.05, adjust = "none")
```

Arguments

x	a data frame consisting of condition columns, target gene efficiency (E), target Gene Ct, reference gene efficiency and reference gene Ct values, respectively. Each Ct in the data frame is the mean of technical replicates. Complete amplification efficiencies of 2 was assumed in the example data for all wells but the calculated efficienies can be used instead. **NOTE:** Each line belongs to a separate individual reflecting a non-repeated measure experiment). See vignette, section "data structure and column arrangement" for details.
numberOfrefGenes	number of reference genes (1 or 2). Up to two reference genes can be handled.
block	column name of the blocking factor (for correct column arrangement see example data.). 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 complete 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	significance level
adjust	method for adjusting p-values

Details

The qpcrANOVARE function performs analysis of variance (ANOVA) of relative expression (RE) values for all factor level combinations as treatments using the expression level of a reference gene is used as normalizer. To get a reliable result, the expression of the reference gene needs to be constant across all test samples and it expression should not be affected by the experimental treatment under study.

Value

A list with 4 elements:

Final_data The row data plus weighed delta Ct (wDCt) values.

lm The output of linear model analysis including ANOVA tables

ANOVA ANOVA table based on CRD

Result The result table including treatments and factors, RE (Relative Expression), LCL, UCL, letter display for pair-wise comparisons and standard error with the lower and upper limits.

Author(s)

Ghader Mirzaghaderi

References

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

Ganger, MT, Dietz GD, and Ewing SJ. 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, Joshua S, Ann Reed, Feng Chen, and Neal Stewart. 2006. Statistical Analysis of Real-Time PCR Data. BMC Bioinformatics 7 (85). doi:10.1186/1471-2105-7-85.

Examples

```
# If the data include technical replicates, means of technical replicates
# should be calculated first using meanTech function.
# Applying ANOVA
qpcrANOVARE(data_3factor, numberOfrefGenes = 1, block = NULL)


qpcrANOVARE(data_2factorBlock, block = "Block", numberOfrefGenes = 1)
```

`qpcrMeans` *Fold change ($\Delta\Delta C\_T$ method) analysis using a model*

Description

Fold change ($\Delta\Delta C_T$ method) analysis using a model object produced by the `qpcrANOVAFC` or `qpcrREPEATED`.

Usage

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

Arguments

`model`	an 'lmer' fitted model object created by qpcrANOVAFC or qpcrREPEATED functions
`specs`	A character vector specifying the names of the predictors over which FC values are desired
`p.adj`	Method for adjusting p values

Details

The qpcrMeans function performs fold change ($\Delta\Delta C_T$ method) analysis using a model produced by the qpcrANOVAFC or qpcrREPEATED. The values can be returned for any effects in the model including simple effects, interactions and slicing if an ANOVA model is used, but ANCOVA models returned by rtpcr package only include simple effects.

Value

Table of FC values, significance and confidence interval.

Author(s)

Ghader Mirzaghaderi

Examples

```
# Returning fold change values from a fitted model.
# Firstly, result of `qpcrANOVAFC` or `qpcrREPEATED` is
# acquired which includes a model object:
res <- qpcrANOVAFC(data_3factor, numberOfrefGenes = 1, mainFactor.column = 1, block = NULL)

# Returning fold change values of Type levels from a fitted model:
qpcrMeans(res$lm_ANOVA, specs = "Type")

# Returning fold change values of Conc levels from a fitted model:
qpcrMeans(res$lm_ANOVA, specs = "Conc")

# Returning fold change values of Conc levels sliced by Type:
qpcrMeans(res$lm_ANOVA, specs = "Conc | Type")

# Returning fold change values of Conc levels sliced by Type*SA:
qpcrMeans(res$lm_ANOVA, specs = "Conc | (Type*SA)")
```

---

qpcrREPEATED *Fold change ($\Delta\Delta C\_T$ method) analysis of repeated measure qPCR data*

Description

qpcrREPEATED function performs fold change ($\Delta\Delta C_T$ method) analysis of observations repeatedly taken over different time courses. Data may be obtained over time from a uni- or multi-factorial experiment. The bar plot of the fold changes (FC) values along with the standard error (se) or confidence interval (ci) is also returned by the qpcrREPEATED function.

Usage

```
qpcrREPEATED(
  x,
  numberOfrefGenes,
  factor,
  block,
  width = 0.5,
  fill = "#BFEFFF",
  y.axis.adjust = 2,
  y.axis.by = 1,
  ylab = "Fold Change",
  xlab = "none",
  fontsize = 12,
  fontsizePvalue = 7,
  axis.text.x.angle = 0,
  axis.text.x.hjust = 0.5,
  x.axis.labels.rename = "none",
  letter.position.adjust = 0,
  p.adj = "none",
  errorbar = "se",
  plot = 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. **NOTE:** 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 `vignette`, section "data structure and column arrangement" for details.
numberOfrefGenes	number of reference genes which is 1 or 2 (Up to two reference genes can be handled). as reference or calibrator which is the reference level or sample that all others are compared to. Examples are untreated of time 0. The FC value of the reference or calibrator level is 1 because it is not changed compared to itself. If NULL, the first level of the main factor column is used as calibrator.
factor	the factor for which the FC values is analysed. The first level of the specified factor in the input data frame is used as calibrator.
block	column name of the block if there is a blocking factor (for correct column arrangement see example data.). Block effect is usually considered as random and its interaction with any main effect is not considered.
width	a positive number determining bar width in the output bar plot.

`fill`	specify the fill color for the columns in the bar plot. If a vector of two colors is specified, the reference level is differentialy colored.
`y.axis.adjust`	a negative or positive value for reducing or increasing the length of the y axis.
`y.axis.by`	determines y axis step length
`ylab`	the title of the y axis
`xlab`	the title of the x axis
`fontsize`	font size of the plot
`fontsizePvalue`	font size of the pvalue labels
`axis.text.x.angle`	angle of x axis text
`axis.text.x.hjust`	horizontal justification of x axis text
`x.axis.labels.rename`	a vector replacing the x axis labels in the bar plot
`letter.position.adjust`	adjust the distance between the signs and the error bars.
`p.adj`	Method for adjusting p values
`errorbar`	Type of error bar, can be `se` or `ci`.
`plot`	if `FALSE`, prevents the plot.

Details

The `qpcrREPEATED` function performs fold change (FC) analysis of observations repeatedly taken over time. The intended factor (could be time or any other factor) is defined for the analysis by the `factor` argument, then the function performs FC analysis on its levels so that the first levels (as appears in the input data frame) is used as reference or calibrator. The function returns FC values along with confidence interval and standard error for the FC values.

Value

A list with 5 elements:

Final_data Input data frame plus the weighted Delat Ct values (wDCt)

lm lm of factorial analysis-tyle

ANOVA_table ANOVA table

FC Table Table of FC values, significance, confidence interval and standard error with the lower and upper limits for the selected factor levels.

Bar plot of FC values Bar plot of the fold change values for the main factor levels.

Author(s)

Ghader Mirzaghaderi

Examples

```
qpcrREPEATED(data_repeated_measure_1,
            numberOfrefGenes = 1,
            factor = "time", block = NULL)

qpcrREPEATED(data_repeated_measure_2,
             numberOfrefGenes = 1,
             factor = "time", block = NULL)
```

---

qpcrTTEST *Fold change ($\Delta\Delta C\_T$ method) analysis of target genes using t-test*

Description

t.test based analysis of the fold change expression for any number of target genes.

Usage

```
qpcrTTEST(x, numberOfrefGenes, paired = FALSE, var.equal = TRUE)
```

Arguments

x	a data frame of 4 columns including Conditions, E (efficiency), Gene and Ct values (see examples below). Biological replicates needs to be equal for all Genes. Each Ct value is the mean of technical replicates. Complete amplification efficiencies of 2 is assumed here for all wells but the calculated efficienies can be used instead. See vignette for details about "data structure and column arrangement".
numberOfrefGenes	number of reference genes. Up to two reference genes can be handled.
paired	a logical indicating whether you want a paired t-test.
var.equal	a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.

Details

The qpcrTTEST function applies a t.test based analysis to calculate fold change ($\Delta\Delta C_T$ method) expression and returns related statistics for any number of target genes that have been evaluated under control and treatment conditions. Sampling may be paired or unpaired. One or two reference genes can be used. Unpaired and paired samples are commonly analyzed using unpaired and paired t-test, respectively. **NOTE:** Paired samples in quantitative PCR refer to two sample data that are collected from one set of individuals at two different conditions, for example before and after a treatment or at two different time points. While for unpaired samples, two sets of individuals are used: one under untreated and the other set under treated condition. Paired samples allow to compare gene expression changes within the same individual, reducing inter-individual variability.

Value

A list of two elements:

Row_data The row data including Genes and weighed delta Ct (wDCt) values.

Result Output table including the Fold Change values, lower and upper confidence interval, pvalue and standard error with the lower and upper limits.

For more information about the test procedure and its arguments, refer `t.test`, and `lm`. If the residuals of the model do not follow normal distribution and variances between the two groups are not homoGene, `wilcox.test` procedure may be concidered

Author(s)

Ghader Mirzaghaderi

References

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

Ganger, MT, Dietz GD, and Ewing SJ. 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, Joshua S, Ann Reed, Feng Chen, and Neal Stewart. 2006. Statistical Analysis of Real-Time PCR Data. BMC Bioinformatics 7 (85). doi:10.1186/1471-2105-7-85.

Examples

```
# See the sample data structure
data_ttest

# Getting t.test results
qpcrTTEST(data_ttest,
   paired = FALSE,
   var.equal = TRUE,
   numberOfrefGenes = 1)



qpcrTTEST(Taylor_etal2019,
          numberOfrefGenes = 2,
          var.equal = TRUE)
```

qpcrTTESTplot *Bar plot of the average fold change ($\Delta\Delta C\_T$ method) of target genes*

Description

Bar plot of the fold change ($\Delta\Delta C_T$ method) values for for any number of target genes under a two-level conditional experimental (e.g. control and treatment).

Usage

```
qpcrTTESTplot(
  x,
  order = "none",
  numberOfrefGenes,
  paired = FALSE,
  var.equal = TRUE,
  width = 0.5,
  fill = "skyblue",
  y.axis.adjust = 0,
  y.axis.by = 2,
  letter.position.adjust = 0.3,
  ylab = "Average Fold Change",
  xlab = "none",
  fontsize = 12,
  fontsizePvalue = 7,
  axis.text.x.angle = 0,
  axis.text.x.hjust = 0.5,
  errorbar = "se"
)
```

Arguments

x	a data frame. The data frame consists of 4 columns belonging to condition levels, E (efficiency), genes and Ct values, respectively. Each Ct in the following data frame is the mean of technical replicates. Complete amplification efficiencies of 2 is assumed here for all wells but the calculated efficienies can be used we well. We use this data set for fold change expression analysis of the target genes in treatment condition compared to normal condition. See vignette for details.
order	a vector determining genes order on the output graph.
numberOfrefGenes	number of reference genes. Up to two reference genes can be handled.
paired	a logical indicating whether you want a paired t-test.
var.equal	a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.

`width`	a positive number determining bar width.
`fill`	specify the fill color for the columns of the bar plot.
`y.axis.adjust`	a negative or positive value for reducing or increasing the length of the y axis.
`y.axis.by`	determines y axis step length
`letter.position.adjust`	adjust the distance between the signs and the error bars.
`ylab`	the title of the y axis
`xlab`	the title of the x axis
`fontsize`	fonts size of the plot
`fontsizePvalue`	font size of the pvalue labels
`axis.text.x.angle`	angle of x axis text
`axis.text.x.hjust`	horizontal justification of x axis text
`errorbar`	Type of error bar, can be `se` or `ci`.

Details

The `qpcrTTESTplot` function applies a t.test based analysis to any number of target genes along with one or two reference gene(s), that have been evaluated under control and treatment conditions. It returns the bar plot of the fold change (FC) values for target genes along with the 95% CI and significance. Sampling may be unpaired or paired. Unpaired and paired samples are commonly analyzed using unpaired and paired t-test, respectively.Paired samples in quantitative PCR refer to two sample data that are collected from one set of individuals at two different conditions, for example before and after a treatment or at two different time points. While for unpaired samples, two sets of individuals are used: one under untreated and the other set under treated condition. Paired samples allow to compare gene expression changes within the same individual, reducing inter-individual variability.

Value

Bar plot of the average fold change for target genes along with the significance and the 95 percent CI as error bars.

Author(s)

Ghader Mirzaghaderi

Examples

```
# See a sample data frame
data_ttest


qpcrTTESTplot(data_ttest,
              numberOfrefGenes = 1,
              errorbar = "ci")
```

```
# Producing the plot
qpcrTTESTplot(data_ttest,
              numberOfrefGenes = 1,
              order = c("C2H2-01", "C2H2-12", "C2H2-26"),
              paired = FALSE,
              var.equal = TRUE,
              width = 0.5,
              fill = "skyblue",
              y.axis.adjust = 0,
              y.axis.by = 2,
              letter.position.adjust = 0.3,
              ylab = "Fold Change in Treatment vs Control",
              xlab = "Gene",
              errorbar = "se")
```

`threeFACTORplot` *Bar plot of the relative gene expression (ΔC_T method) from the* `qpcrANOVARE` *output of a a three-factorial experiment data*

Description

Bar plot of the relative expression (ΔC_T method) of a gene along with the confidence interval and significance

Usage

```
threeFACTORplot(
  res,
  arrangement = c(1, 2, 3),
  bar.width = 0.5,
  fill = "Reds",
  xlab = "none",
  ylab = "Relative Expression",
  errorbar,
  y.axis.adjust = 0.5,
  y.axis.by = 2,
  letter.position.adjust = 0.3,
  legend.title = "Legend Title",
  legend.position = c(0.4, 0.8),
  fontsize = 12,
  fontsizePvalue = 5,
  show.letters = TRUE,
  axis.text.x.angle = 0,
  axis.text.x.hjust = 0.5
)
```

Arguments

res	the FC data frame created by qpcrANOVARE(x)$Result function on a three factorial data such as data_3factor example data frame.
arrangement	order based on the columns in the output table (e.g. c(2,3,1) or c(1,3,2)) affecting factor arrangement of the output graph.
bar.width	a positive number determining bar width.
fill	a color vector specifying the fill color for the columns of the bar plot. One of the palettes in display.brewer.all (e.g. "Reds" or "Blues", ...) can be applied.
xlab	the title of the x axis
ylab	the title of the y axis
errorbar	Type of error bar, can be se or ci.
y.axis.adjust	a negative or positive number for reducing or increasing the length of the y axis.
y.axis.by	determines y axis step length
letter.position.adjust	adjust the distance between the grouping letters to the error bars
legend.title	legend title
legend.position	a two digit vector specifying the legend position.
fontsize	all fonts size of the plot
fontsizePvalue	font size of the pvalue labels
show.letters	a logical variable. If TRUE, mean grouping letters are added to the bars.
axis.text.x.angle	angle of x axis text
axis.text.x.hjust	horizontal justification of x axis text

Details

The threeFACTORplot function generates the bar plot of the average fold change for target genes along with the significance, standard error (se) and the 95% confidence interval (ci).

Value

Bar plot of the average fold change for target genes along with the standard error or 95% confidence interval as error bars.

Author(s)

Ghader Mirzaghaderi

Examples

```
#' # See a sample data frame
data_3factor

# Before plotting, the result needs to be extracted as below:
res <- qpcrANOVARE(data_3factor, numberOfrefGenes = 1, block = NULL)$Result
res

# Arrange the first three colunms of the result table.
# This determines the columns order and shapes the plot output.
threeFACTORplot(res, arrangement = c(3, 1, 2), errorbar = "se",
    xlab = "condition")


threeFACTORplot(res, arrangement = c(1, 2, 3), bar.width = 0.5, fill = "Greys",
xlab = "Genotype", ylab = "Relative Expression", errorbar = "se")


# Reordering factor levels to a desired order.
res$Conc <- factor(res$Conc, levels = c("L","M","H"))
res$Type <- factor(res$Type, levels = c("S","R"))

# Producing the plot
threeFACTORplot(res, arrangement = c(2, 3, 1), bar.width = 0.5,
fill = "Reds", xlab = "Drought", ylab = "Relative Expression",
errorbar = "se", legend.title = "Genotype", legend.position = c(0.2, 0.8))


# When using ci as error, increase the
# y.axis.adjust value to see the plot correctly!
threeFACTORplot(res, arrangement = c(2, 3, 1), bar.width = 0.8, fill = "Greens",
xlab = "Drought", ylab = "Relative Expression", errorbar = "ci",
y.axis.adjust = 1, y.axis.by = 2, letter.position.adjust = 0.6,
legend.title = "Genotype", fontsize = 12, legend.position = c(0.2, 0.8),
show.letters = TRUE)
```

`twoFACTORplot` *Bar plot of the relative gene expression (ΔC_T method) from the* `qpcrANOVARE` *output of a two-factorial experiment data*

Description

Bar plot of the relative expression (ΔC_T method) of a gene along with the standard error (se), 95% confidence interval (ci) and significance

Usage

```
twoFACTORplot(
  res,
  x.axis.factor,
  group.factor,
  width = 0.5,
  fill = "Blues",
  y.axis.adjust = 0.5,
  y.axis.by = 2,
  show.errorbars = TRUE,
  errorbar,
  show.letters = TRUE,
  letter.position.adjust = 0.1,
  ylab = "Relative Expression",
  xlab = "none",
  legend.position = c(0.09, 0.8),
  fontsize = 12,
  fontsizePvalue = 5,
  axis.text.x.angle = 0,
  axis.text.x.hjust = 0.5
)
```

Arguments

`res`	the FC data frame created by `qpcrANOVARE(x)$Result` function on a two factor data such as `data_2factor`.
`x.axis.factor`	x-axis factor.
`group.factor`	grouping factor.
`width`	a positive number determining bar width.
`fill`	specify the fill color vector for the columns of the bar plot. One of the palettes in `display.brewer.all` (e.g. "Reds" or "Blues", ...) can be applied.
`y.axis.adjust`	a negative or positive number for reducing or increasing the length of the y axis.
`y.axis.by`	determines y axis step length.
`show.errorbars`	show errorbars
`errorbar`	Type of error bar, can be `se` or `ci`.
`show.letters`	a logical variable. If TRUE, mean grouping letters are added to the bars.
`letter.position.adjust`	adjust the distance between the grouping letters to the error bars.
`ylab`	the title of the y axis.
`xlab`	the title of the x axis.
`legend.position`	a two digit vector specifying the legend position.
`fontsize`	size of all fonts of the plot.
`fontsizePvalue`	font size of the pvalue labels

`axis.text.x.angle`
: angle of x axis text

`axis.text.x.hjust`
: horizontal justification of x axis text

Details

The `twoFACTORplot` function generates the bar plot of the average fold change for target genes along with the significance, standard error (se) and the 95% confidence interval (ci) as error bars.

Value

Bar plot of the average fold change for target genes along with the standard error or 95% confidence interval as error bars.

Author(s)

Ghader Mirzaghaderi

Examples

```
# See a sample data frame
data_2factor

# Before generating plot, the result table needs to be extracted as below:
res <- qpcrANOVARE(data_2factor, numberOfrefGenes = 1, block = NULL)$Result

# Plot of the 'res' data with 'Genotype' as grouping factor
twoFACTORplot(res,
   x.axis.factor = Drought,
   group.factor = Genotype,
   width = 0.5,
   fill = "Greens",
   y.axis.adjust = 1,
   y.axis.by = 2,
   ylab = "Relative Expression",
   xlab = "Drought Levels",
   letter.position.adjust = 0.2,
   legend.position = c(0.2, 0.8),
   errorbar = "se")


# Plotting the same data with 'Drought' as grouping factor
twoFACTORplot(res,
              x.axis.factor = Genotype,
              group.factor = Drought,
              xlab = "Genotype",
              fill = "Blues",
              fontsizePvalue = 5,
              errorbar = "ci")
```

```
# Combining FC results of two different genes:
a <- qpcrREPEATED(data_repeated_measure_1,
                  numberOfrefGenes = 1,
                  factor = "time", block = NULL, plot = FALSE)

b <- qpcrREPEATED(data_repeated_measure_2,
                  factor = "time",
                  numberOfrefGenes = 1, block = NULL, plot = FALSE)

a1 <- a$FC_statistics_of_the_main_factor
b1 <- b$FC_statistics_of_the_main_factor
c <- rbind(a1, b1)
c$gene <- factor(c(1,1,1,2,2,2))
c

twoFACTORplot(c, x.axis.factor = contrast,
              group.factor = gene, fill = 'Reds',
              ylab = "FC", axis.text.x.angle = 45,
              errorbar = "se", y.axis.adjust = 1,
              axis.text.x.hjust = 1)
```

Index