Package 'rtpcr'

April 21, 2024

Type Package

Title qPCR Data Analysis

Version 1.0.5

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) or analysis of variance (for cases where more than two levels or factors exist) to calculate the fold change or relative expression. The functions also provide standard deviations and confidence limits for means and apply statistical mean comparisons. To facilitate using 'rtpcr', different datasets have been employed in the examples and the outputs are explained. An outstanding feature of 'rtpcr' package is providing publication-ready bar plots with various controlling arguments for experiments with up to three different factors. 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 agricolae, ggplot2, lme4, 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 2024-04-21 20:32:36 UTC

R topics documented:

`efficiency` *Efficiency, standard curves and related statistics*

Description

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

Usage

```
efficiency(x)
```

Arguments

`x`	a data frame

Details

The `efficiency` function calculates amplification efficiency of a target and a reference gene, and present the related standard curves along with the Slope, Efficiency, and R2 statistics. The function also compares the slopes of the two standard curves.

Value

A list including standard curves along with the Slope, Efficiency, and R2 statistics

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 of technical replicates in and output table appropriate 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

Multiple plot function

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

Producing multiple plots plate using ggplot objects

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 <- qpcrANOVA(data_1factor, numberOfrefGenes = 1)
p2 <- oneFACTORplot(out2,
                    width = 0.2,
                    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 (RE) from the* `qpcrANOVA` *output of a one-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 = 0.2,
  fill = "skyblue",
  y.axis.adjust = 0.5,
  y.axis.by = 2,
  errorbar = "se",
  show.letters = TRUE,
  letter.position.adjust = 0.1,
  ylab = "Relative Expression",
  xlab = "none",
  fontsize = 12,
  fontsizePvalue = 7,
  axis.text.x.angle = 0,
  axis.text.x.hjust = 0.5
)
```

Arguments

`res`	an object created by `qpcrANOVA(x)` 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 95% confidence interval as error bars.

Author(s)

Ghader Mirzaghaderi

Examples

```
# Before plotting, the result needs to be extracted as below:
res <- qpcrANOVA(data_1factor, numberOfrefGenes = 1)

# Bar plot
oneFACTORplot(res,
         width = 0.2,
         fill = "skyblue",
         y.axis.adjust = 0,
         y.axis.by = 0.2,
         errorbar = "se",
         show.letters = TRUE,
         letter.position.adjust = 0.05,
         ylab = "Relative Expression",
         xlab = "Factor Levels",
         fontsize = 12)
```

qpcrANCOVA *Fold change (FC) analysis using ANCOVA*

Description

ANCOVA (analysis of covariance) and ANOVA (analysis of variance) can be performed using qpcrANCOVA function, for uni- or multi-factorial experiment data. This function performs FC analysis even if there is only one factor (without covariate variable), although, for the data with only one factor, the analysis turns into ANOVA. The bar plot of the fold changes (FC) values along with the standard error (se) of confidence interval (ci) is also returned by the qpcrANCOVA function.

Usage

```
qpcrANCOVA(
  x,
  numberOfrefGenes,
  analysisType = "ancova",
  mainFactor.column,
  mainFactor.level.order,
  block = NULL,
  width = 0.5,
  fill = "#BFEFFF",
  y.axis.adjust = 1,
  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"
)
```

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. Please refer to the vignette for preparing your data frame correctly.
numberOfrefGenes	number of reference genes which is 1 or 2 (Up to two reference genes can be handled).
analysisType	should be one of "ancova" or "anova".

`mainFactor.column`
: the factor for which FC is calculated for its levels. The remaining factors (if any) are considered as covariate(s).

`mainFactor.level.order`
: a vector of main factor level names. The first level in the vector is used 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.

`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`.

Details

The `qpcrANCOVA` function applies both ANCOVA and ANOVA analysis to the data of a uni- or multi-factorial experiment, although for the data with only one factor, the analysis turns to ANOVA. ANCOVA is basically appropriate when the levels of a factor are also 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 even ANOVA based on a factorial experiment using `qpcrANCOVA`. This 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 pair-wise errors (square roots of pooled variances of each pair of samples) are also returned by the `qpcrANCOVA` function. There is also a function called `oneFACTORplot` which returns RE values and related plot for a one-factor-experiment with more than two levels. Along with the ANCOVA, the `qpcrANCOVA` also performs a full model factorial analysis of variance. If there is covariate variable(s), before ANCOVA analysis, it is better to run ANOVA based on a factorial design to see if the main factor and covariate(s) interaction is significant or not. If the pvalue of the interaction effect is smaller than 0.05, then the interaction between the main factor and covariate is significant, suggesting that ANCOVA is not appropriate in this case.

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 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

```
qpcrANCOVA(data_1factor,
           numberOfrefGenes = 1,
           mainFactor.column = 1,
           mainFactor.level.order = c("L1", "L2", "L3"),
           fill = c("#CDC673", "#EEDD82"),
```

```
             analysisType = "ancova",
             fontsizePvalue = 5,
             y.axis.adjust = 0.1)

qpcrANCOVA(data_2factor,
           numberOfrefGenes = 1,
           mainFactor.column = 2,
           mainFactor.level.order = c("D0", "D1", "D2"),
           fill = c("#79CDCD", "#B4EEB4"),
           analysisType = "ancova",
           fontsizePvalue = 5,
           y.axis.adjust = 0.4)


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


df <- meanTech(Lee_etal2020qPCR, groups = 1:3)
df2 <- df[df$factor1 == "DSWHi",][-1]
qpcrANCOVA(df2,
           mainFactor.column = 1,
           mainFactor.level.order = c("D7", "D12", "D15","D18"),
           numberOfrefGenes = 1,
           analysisType = "ancova",
           fontsizePvalue = 5,
           y.axis.adjust = 0.1)


qpcrANCOVA(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)

addline_format <- function(x,...){gsub('\\s','\n',x)}
order <- unique(data_2factor$Drought)
qpcrANCOVA(data_1factor,
```

```
  numberOfrefGenes = 1,
  mainFactor.column = 1,
  mainFactor.level.order = c("L1","L2","L3"),
  width = 0.5,
  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")))
```

---

qpcrANOVA *Relative efficiency (RE) analysis using ANOVA*

Description

Analysis of Variance of relative efficiency (RE) values based on a completely randomized design (CRD). Even there are more than a factor in the experiment, it is still possible to apply CRD analysis on the factor-level combinations as treatments. Analysis of variance based on factorial design or analysis of covariance can be performed using qpcrANCOVA function.

Usage

```
qpcrANOVA(
  x,
  numberOfrefGenes,
  block = NULL,
 p.adj = c("none", "holm", "hommel", "hochberg", "bonferroni", "BH", "BY", "fdr")
)
```

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.
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.

`p.adj` Method for adjusting p values (see p.adjust)

Details

The `qpcrANOVA` function performs analysis of variance (ANOVA) of relative efficiency (RE) values based on a completely randomized design (CRD). It is suitable when relative expression (RE) analysis between different treatment combinations (in a Uni- or multi-factorial experiment) is desired. If there are more than a factor in the experiment, it is still possible to apply CRD analysis on the factor-level combinations as treatments. For this, a column of treatment combinations is made first as a grouping factor Fold change analysis based on factorial design or analysis of covariance for the can be performed using qpcrANCOVA.

Value

A list with 5 elements:

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

lm The output of linear model analysis including ANOVA tables based on factorial experiment and completely randomized design (CRD).

ANOVA_factorial ANOVA table based on factorial arrangement

ANOVA_CRD ANOVA table based on CRD

Result The result table including treatments and factors, RE (Relative Expression), LCL, UCL, letter grouping and standard deviation of relative expression.

Post_hoc_Test Post hoc test of FC (Fold Change), pvalue, significance and confidence interval (LCL, UCL).

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 analysis
qpcrANOVA(
     data_3factor,
     numberOfrefGenes = 1,
     p.adj = "none")


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

---

qpcrTTEST *fold change (FC) 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 example 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.
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 (FC) expression and returns related statistics for any number of target genes along with one or two reference gene(s), that have been evaluated under control and treatment conditions. Sampling may be paired or unpaired. 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. Unpaired and paired samples are commonly analyzed using unpaired and paired t-test, respectively.

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 and the pvalues from compairing fold change between treated and non-treated conditions

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(Taylor_etal2019,
              numberOfrefGenes = 2,
              var.equal = TRUE,
              y.axis.adjust = -0.7,
              y.axis.by = 0.5)
```

qpcrTTESTplot *Bar plot of the average fold change (FC) of the target genes*

Description

Bar plot of the average fold change (FC) 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.
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 paired or unpaired. 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. Unpaired and paired samples are commonly analyzed using unpaired and paired t-test, respectively.

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 (RE) from the* qpcrANOVA *output of a a three-factorial experiment data*

Description

Bar plot of the relative expression (RE) 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 = "se",
  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 = 7,
  show.letters = TRUE,
  axis.text.x.angle = 0,
  axis.text.x.hjust = 0.5
)
```

Arguments

res	an object created by qpcrANOVA(x) 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 significance and the 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 <- qpcrANOVA(data_3factor, numberOfrefGenes = 1)
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),
    xlab = "condition")



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



# 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 (RE) from the* `qpcrANOVA` *output of a two-factorial experiment data*

Description

Bar plot of the relative expression (RE) 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 = "se",
  show.letters = TRUE,
  show.points = FALSE,
  letter.position.adjust = 0.1,
  ylab = "Relative Expression",
  xlab = "none",
  legend.position = c(0.09, 0.8),
  fontsize = 12,
  fontsizePvalue = 7,
  axis.text.x.angle = 0,
  axis.text.x.hjust = 0.5
)
```

Arguments

`res`	an object created by `qpcrANOVA(x)` 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.
show.points	show points
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 significance and the 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 <- qpcrANOVA(data_2factor, numberOfrefGenes = 1)
res
```

```
# 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",
   show.letters = TRUE,
   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",
   fontsize = 12,
   show.letters = FALSE,
   show.points = TRUE,
   show.errorbars = FALSE)
```

Index