Package 'RQdeltaCT'

April 17, 2024

Type Package

Title Relative Quantification of Gene Expression using Delta Ct Methods

Version 1.3.0

Description The commonly used methods for relative quantification of gene expression levels obtained in real-time PCR (Polymerase Chain Reaction) experiments are the delta Ct methods, encompassing 2^-dCt and 2^-ddCt methods, originally proposed by Kenneth J. Livak and Thomas D. Schmittgen (2001) <doi:10.1006/meth.2001.1262>. The main idea is to normalise gene expression values using endogenous control gene, present gene expression levels in linear form by using the 2^-(value)^ transformation, and calculate differences in gene expression levels between groups of samples (or technical replicates of a single sample). The 'RQdeltaCT' package offers functions that cover both methods for comparison of either independent groups of samples or groups with paired samples, together with importing expression datasets, performing multi-step quality control of data, enabling numerous data visualisations, enrichment of the standard workflow with additional useful analyses (correlation analysis, Receiver Operating Characteristic analysis, logistic regression), and conveniently export obtained results in table and image formats. The package has been designed to be friendly to non-experts in R programming.

URL <https://github.com/Donadelnal/RQdeltaCT>

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Depends R (>= 2.10)

Imports tidyverse, dplyr, ggplot2, magrittr, tidyr, tidyselect, coin, ctrlGene, ggsignif, Hmisc, corrplot, ggpmisc, pROC, oddsratio, stats, graphics, grDevices, utils, pheatmap, GGally

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Daniel Zalewski [aut, cre] (<https://orcid.org/0000-0002-2254-2009>)

Maintainer Daniel Zalewski <daniel.piotr.zalewski@gmail.com>

Repository CRAN

Date/Publication 2024-04-17 15:50:02 UTC

R topics documented:

control_boxplot_gene *control_boxplot_gene*

Description

This function creates boxplot that illustrate distribution of data in each gene.

Usage

```
control_boxplot_gene(
  data,
  sel.Gene = "all",
  pairwise.FCh = FALSE,
  coef = 1.5,
  by.group = TRUE,
  colors = c("#66c2a5", "#fc8d62"),
  axis.title.size = 11,
  axis.text.size = 12,
  x.axis.title = "Gene",
  y.axis.title = "value",
  legend.title = "Group",
  legend.title.size = 11,
  legend.text.size = 11,
  legend.position = "right",
  plot.title = "",
  plot.title.size = 14,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "control_boxplot_genes"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions. Also, a data frame with fold change values returned from RQ_dCt() and RQ_ddCt() functions run in a pairwise mode can be used.
sel.Gene	Character vector with names of genes to include, or "all" (default) to use all genes.
pairwise.FCh	Logical: If fold change values returned from RQ_dCt() and RQ_ddCt() functions in a pairwise approach are used as data, this parameter should be set to TRUE, otherwise to FALSE (default).
coef	Numeric: how many times of interquartile range should be used to determine range point for whiskers. Default to 1.5.
by.group	Logical: if TRUE, distributions will be drawn by compared groups of samples.

`colors` Character vector containing colors for groups, length of one (when by.group = FALSE) or equal to the number of groups (when by.group = TRUE).

`axis.title.size` Integer: font size of axis titles. Default to 11.

`axis.text.size` Integer: font size of axis text. Default to 12.

`x.axis.title` Character: title of x axis. Default to "Gene".

`y.axis.title` Character: title of y axis. Default to "value".

`legend.title` Character: title of legend. Default to "Group".

`legend.title.size` Integer: font size of legend title. Default to 11.

`legend.text.size` Integer: font size of legend text. Default to 11.

`legend.position` Position of the legend, can be "top", "right" (default), "bottom", "left", or "none" (no legend). See description for legend.position in ggplot2::theme() function.

`plot.title` Character: title of plot. Default to "".

`plot.title.size` Integer: font size of plot title. Default to 14.

`save.to.tiff` Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.

`dpi` Integer: resolution of saved .tiff file. Default to 600.

`width` Numeric: width (in cm) of saved .tiff file. Default to 15.

`height` Numeric: height (in cm) of saved .tiff file. Default to 15.

`name.tiff` Character: name of saved .tiff file, without ".tiff" name of extension. Default to "control_boxplot_genes".

Value

Object with boxplot illustrating distribution of data for each gene. Created plot is also displayed on the graphic device.

Examples

```
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
control.boxplot.gene <- control_boxplot_gene(data.dCt)
```

control_boxplot_sample

control_boxplot_sample

Description

Boxplot that illustrate distribution of data in each sample. This function is helpful to identify outlier samples.

Usage

```
control_boxplot_sample(
  data,
  sel.Sample = "all",
  pairwise.FCh = FALSE,
  coef = 1.5,
  colors = c("#66c2a5", "#fc8d62"),
  x.axis.title = "Sample",
  y.axis.title = "value",
  axis.title.size = 11,
  axis.text.size = 12,
  legend.title = "Group",
  legend.title.size = 11,
  legend.text.size = 11,
  legend.position = "right",
  plot.title = "",
  plot.title.size = 14,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "control_boxplot_samples"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions. Also, a data frame with fold change values returned from RQ_dCt() and RQ_ddCt() functions run in a pairwise mode can be used.
sel.Sample	Character vector with names of samples to include, or "all" (default) to use all samples.
pairwise.FCh	Logical: If fold change values returned from RQ_dCt() and RQ_ddCt() functions in a pairwise approach are used as data, this parameter should be set to TRUE, otherwise to FALSE (default).
coef	Numeric: how many times of interquartile range should be used to determine range point for whiskers. Default to 1.5.

colors	Character vector containing colors for compared groups. Numbers of colors must be equal to number of groups. Default to c("#66c2a5", "#fc8d62"). If pairwise.FCh parameter is set to TRUE, one color is required (data contain no groups).
x.axis.title	Character: title of x axis. Default to "Sample".
y.axis.title	Character: title of y axis. Default to "value".
axis.title.size	Integer: font size of axis titles. Default to 11.
axis.text.size	Integer: font size of axis text. Default to 12.
legend.title	Character: title of legend. Default to "Group".
legend.title.size	Integer: font size of legend title. Default to 11.
legend.text.size	Integer: font size of legend text. Default to 11.
legend.position	Position of the legend, can be "top", "right" (default), "bottom", "left", or "none" (no legend). See description for legend.position in ggplot2::theme() function.
plot.title	Character: title of plot. Default to "".
plot.title.size	Integer: font size of plot title. Default to 14.
save.to.tiff	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
dpi	Integer: resolution of saved .tiff file. Default to 600.
width	Numeric: width (in cm) of saved .tiff file. Default to 15.
height	Numeric: height (in cm) of saved .tiff file. Default to 15.
name.tiff	Character: name of saved .tiff file, without ".tiff" name of extension. Default to "control_boxplot_samples".

Value

Object with a boxplot illustrating distribution of data in each sample. Created plot is also displayed on the graphic device.

Examples

```
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
control.boxplot.sample <- control_boxplot_sample(data.dCt)
```

control_cluster_gene *control_cluster_gene*

Description

This function performs hierarchical clustering of genes based on the data, useful to gain insight into similarity in expression of analyzed genes.

Usage

```
control_cluster_gene(
  data,
  method.dist = "euclidean",
  sel.Sample = "all",
  method.clust = "average",
  pairwise.FCh = FALSE,
  x.axis.title = "Genes",
  y.axis.title = "Height",
  label.size = 0.8,
  plot.title = "",
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "control_clust_genes"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions. Also, table with fold change values returned from RQ_dCt() and RQ_ddCt() functions can be used, but in such situation a pairwise.FCh parameter must be set to TRUE.
method.dist	Character: name of method used for calculation of distances, derived from stats::dist() function, must be one of "euclidean" (default) , "maximum", "manhattan", "canberra", "binary" or "minkowski".
sel.Sample	Character vector with names of samples to include, or "all" (default) to use all samples.
method.clust	Character: name of used method for agglomeration, derived from stats::hclust() function, must be one of "ward.D", "ward.D2", "single", "complete", "average" (default), "mcquitty", "median" or "centroid".
pairwise.FCh	Logical: If fold change values returned from RQ_dCt() and RQ_ddCt() functions in a pairwise approach are used as data, this parameter should be set to TRUE, otherwise to FALSE (default).
x.axis.title	Character: title of x axis. Default to "Genes".
y.axis.title	Character: title of y axis. Default to "Height".

label.size	Numeric: size of text labels. Default to 0.8.
plot.title	Character: title of plot. Default to "".
save.to.tiff	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
dpi	Integer: resolution of saved .tiff file. Default to 600.
width	Numeric: width (in cm) of saved .tiff file. Default to 15.
height	Numeric: height (in cm) of saved .tiff file. Default to 15.
name.tiff	Character: name of saved .tiff file, without ".tiff" name of extension. Default to "control_clust_genes".

Value

Plot with hierarchical clustering of genes, displayed on the graphic device.

Examples

```
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
control_cluster_gene(data.dCt)
```

---

control_cluster_sample

control_cluster_sample

Description

This function performs hierarchical clustering of samples based on the data, useful to identify outlier samples.

Usage

```
control_cluster_sample(
  data,
  sel.Gene = "all",
  method.dist = "euclidean",
  method.clust = "average",
  pairwise.FCh = FALSE,
  x.axis.title = "Samples",
  y.axis.title = "Height",
  label.size = 0.8,
  plot.title = "",
```

```
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "control_clust_samples"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions. Also, table with fold change values returned from RQ_dCt() and RQ_ddCt() functions can be used, but in such situation a pairwise.FCh parameter must be set to TRUE.
sel.Gene	Character vector with names of genes to include, or "all" (default) to use all genes.
method.dist	Character: name of method used for calculation of distances, derived from stats::dist() function, must be one of "euclidean" (default) , "maximum", "manhattan", "canberra", "binary" or "minkowski".
method.clust	Character: name of used method for agglomeration, derived from stats::hclust() function, must be one of "ward.D", "ward.D2", "single", "complete", "average" (default), "mcquitty", "median" or "centroid".
pairwise.FCh	Logical: If fold change values returned from RQ_dCt() and RQ_ddCt() functions in a pairwise approach are used as data, this parameter should be set to TRUE, otherwise to FALSE (default).
x.axis.title	Character: title of x axis. Default to "Samples".
y.axis.title	Character: title of y axis. Default to "Height".
label.size	Numeric: size of text labels. Default to 0.8.
plot.title	Character: title of plot. Default to "".
save.to.tiff	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
dpi	Integer: resolution of saved .tiff file. Default to 600.
width	Numeric: width (in cm) of saved .tiff file. Default to 15.
height	Numeric: height (in cm) of saved .tiff file. Default to 15.
name.tiff	Character: name of saved .tiff file, without ".tiff" name of extension. Default to "control_clust_samples".

Value

Plot with hierarchical clustering of samples, displayed on the graphic device.

Examples

```
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                       remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
```

```
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
control_cluster_sample(data.dCt)
```

`control_Ct_barplot_gene` *control_Ct_barplot_gene*

Description

Gene-wide quality control of raw Ct values across groups by illustrating numbers of Ct values labelled as reliable or not by using reliability criteria (see function parameters).

Usage

```
control_Ct_barplot_gene(
  data,
  flag.Ct = "Undetermined",
  maxCt = 35,
  flag = "Undetermined",
  colors = c("#66c2a5", "#fc8d62"),
  x.axis.title = "",
  y.axis.title = "Number",
  axis.title.size = 11,
  axis.text.size = 10,
  legend.title = "Reliable Ct value?",
  legend.title.size = 11,
  legend.text.size = 11,
  legend.position = "top",
  plot.title = "",
  plot.title.size = 14,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "Ct_control_barplot_for_genes"
)
```

Arguments

`data`	Object returned from read_Ct_long() or read_Ct_wide() function, or data frame containing column named "Sample" with sample names, column named "Gene" with gene names, column named "Ct" with raw Ct values, column named "Group" with group names. Optionally, data frame can contain column named "Flag" with flag information (e.g. "Undetermined" and "OK"), which will be used for reliability assessment.
`flag.Ct`	Character of a flag used for undetermined Ct values. Default to "Undetermined".

maxCt	Numeric, a maximum of Ct value allowed. Default to 35.
flag	Character of a flag used in the Flag column for values which are unreliable. Default to "Undetermined".
colors	Character vector length of two, containing colors for Ct values which are labelled as reliable (first element of vector) or not (second element of vector).
x.axis.title	Character: title of x axis. Default to "".
y.axis.title	Character: title of y axis. Default to "Number".
axis.title.size	Integer: font size of axis titles. Default to 11.
axis.text.size	Integer: font size of axis text. Default to 10.
legend.title	Character: title of legend. Default to "Reliable Ct value?".
legend.title.size	Integer: font size of legend title. Default to 12.
legend.text.size	Integer: font size of legend text. Default to 11.
legend.position	Position of the legend, can be "top" (default), "right", "bottom", "left", or "none" (no legend). See description for legend.position parameter in ggplot2::theme() function.
plot.title	Character: title of plot. Default to "".
plot.title.size	Integer: font size of plot title. Default to 14.
save.to.tiff	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
dpi	Integer: resolution of saved .tiff file. Default to 600.
width	Numeric: width (in cm) of saved .tiff file. Default to 15.
height	Numeric: height (in cm) of saved .tiff file. Default to 15.
name.tiff	Character: name of saved .tiff file, without ".tiff" name of extension. Default to "Ct_control_barplot_for_genes".

Details

This function does not perform data filtering, but only counts Ct values labelled as reliable or not and presents them graphically. Could be useful to identify genes with low number of reliable Ct values.

Value

List containing plot and table with counts of reliable and non reliable Ct values in genes. Additional information about returned table is also printed to help the user to properly interpret returned table. Plot is also displayed on the graphic device.

Examples

```
library(tidyverse)
data(data.Ct)
gene.Ct.control <- control_Ct_barplot_gene(data.Ct)
gene.Ct.control[[2]]
```

`control_Ct_barplot_sample` *control_Ct_barplot_sample*

Description

Sample-wide quality control of raw Ct values by illustrating the numbers of Ct values labelled as reliable or not by using reliability criteria (see function parameters).

Usage

```
control_Ct_barplot_sample(
  data,
  flag.Ct = "Undetermined",
  maxCt = 35,
  flag = "Undetermined",
  colors = c("#66c2a5", "#fc8d62"),
  x.axis.title = "",
  y.axis.title = "Number",
  axis.title.size = 11,
  axis.text.size = 10,
  plot.title = "",
  plot.title.size = 14,
  legend.title = "Reliable Ct value?",
  legend.title.size = 11,
  legend.text.size = 11,
  legend.position = "top",
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "Ct_control_barplot_for_samples"
)
```

Arguments

`data`	Object returned from read_Ct_long() or read_Ct_wide() function, or data frame containing column named "Sample" with sample names, column named "Gene" with gene names, column named "Ct" with raw Ct values, and column named "Group" with group names. Optionally, data frame could contain column named

	"Flag" with flag information (e.g. "Undetermined" and "OK"), which will be used for reliability assessment.
`flag.Ct`	Character of a flag used for undetermined Ct values. Default to "Undetermined".
`maxCt`	Numeric, a maximum of Ct value allowed. Default to 35.
`flag`	Character of a flag used in the Flag column for values which are unreliable. Default to "Undetermined".
`colors`	Character vector length of two, containing colors for Ct values that were labelled as reliable (first element of vector) or not (second element of vector).
`x.axis.title`	Character: title of x axis. Default to "".
`y.axis.title`	Character: title of y axis. Default to "Number".
`axis.title.size`	Integer: font size of axis titles. Default to 11.
`axis.text.size`	Integer: font size of axis text. Default to 10.
`plot.title`	Character: title of plot. Default to "".
`plot.title.size`	Integer: font size of plot title. Default to 14.
`legend.title`	Character: title of legend. Default to "Reliable Ct value?".
`legend.title.size`	Integer: font size of legend title. Default to 12.
`legend.text.size`	Integer: font size of legend text. Default to 11.
`legend.position`	Position of the legend, can be set to "top" (default), "right", "bottom", "left", or "none" (no legend). See description for legend.position parameter in ggplot2::theme() function.
`save.to.tiff`	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
`dpi`	Integer: resolution of saved .tiff file. Default to 600.
`width`	Numeric: width (in cm) of saved .tiff file. Default to 15.
`height`	Numeric: height (in cm) of saved .tiff file. Default to 15.
`name.tiff`	Character: name of saved .tiff file, without ".tiff" name of extension. Default to "Ct_control_barplot_for_samples".

Details

This function labels Ct values as reliable or not using given reliability criteria, counts them, and presents them graphically. Results are useful to identify samples with low numbers of reliable Ct values. This function does not perform data filtering.

Value

List containing plot and table with counts of reliable and no reliable Ct values in samples. Additional information about returned table is also printed to help the user to properly interpret returned table. Plot is also displayed on the graphic device.

Examples

```
library(tidyverse)
data(data.Ct)
sample.Ct.control <- control_Ct_barplot_sample(data.Ct)
sample.Ct.control[[2]]
```

`control_pca_gene` *control_pca_gene*

Description

This function performs principal component analysis (PCA) for genes and generates plot illustrating spatial arrangement of genes using two first PCA components. This plot allows to gain insight into similarity in expression of analyzed genes. PCA analysis can not deal with missing values, thus all genes with at least one missing value are removed from data before analysis.

Usage

```
control_pca_gene(
  data,
  sel.Sample = "all",
  pairwise.FCh = FALSE,
  point.size = 4,
  point.shape = 19,
  alpha = 0.7,
  label.size = 3,
  hjust = 0.5,
  vjust = -1,
  color = "black",
  axis.title.size = 11,
  axis.text.size = 10,
  legend.text.size = 11,
  legend.title = "Group",
  legend.title.size = 11,
  legend.position = "right",
  plot.title = "",
  plot.title.size = 14,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "control_pca_genes"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions. Also, table with fold change values returned from RQ_dCt() and RQ_ddCt() functions can be used, but in such situation a pairwise.FCh parameter must be set to TRUE.
sel.Sample	Character vector with names of samples to include, or "all" (default) to use all samples.
pairwise.FCh	Logical: If fold change values returned from RQ_dCt() and RQ_ddCt() functions in a pairwise approach are used as data, this parameter should be set to TRUE, otherwise to FALSE (default).
point.size	Numeric: size of points. Default to 4.
point.shape	Integer: shape of points. Default to 19.
alpha	Numeric: transparency of points, a value between 0 and 1. Default to 0.7.
label.size	Numeric: size of points labels (names of samples). Default to 3.
hjust	Numeric: horizontal position of points labels. Default to 0.
vjust	Numeric: vertical position of points labels. Default to -1.
color	Character: color used for points.
axis.title.size	Integer: font size of axis titles. Default to 11.
axis.text.size	Integer: font size of axis text. Default to 10.
legend.text.size	Integer: font size of legend text. Default to 11.
legend.title	Character: title of legend. Default to "Group".
legend.title.size	Integer: font size of legend title. Default to 11.
legend.position	Position of the legend, can be "top", "right" (default), "bottom", "left", or "none" (no legend). See description for legend.position parameter in ggplot2::theme() function.
plot.title	Character: title of plot. Default to "".
plot.title.size	Integer: font size of plot title. Default to 14.
save.to.tiff	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
dpi	Integer: resolution of saved .tiff file. Default to 600.
width	Numeric: width (in cm) of saved .tiff file. Default to 15.
height	Numeric: height (in cm) of saved .tiff file. Default to 15.
name.tiff	character: name of saved .tiff file, without ".tiff" name of extension. Default to "control_pca_genes".

Value

Object with plot. The plot is also displayed on the graphic device.

Examples

```
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
control_pca_gene(data.dCt)
```

`control_pca_sample` *control_pca_sample*

Description

This function performs principal component analysis (PCA) for samples and generate plot that illustrate spatial arrangement of samples based on the two first components. This plot is useful to identify outlier samples. PCA analysis can not deal with missing values, thus all samples with at least one missing value are removed from data before analysis.

Usage

```
control_pca_sample(
  data,
  sel.Gene = "all",
  pairwise.FCh = FALSE,
  point.size = 4,
  point.shape = 19,
  alpha = 0.7,
  colors = c("#66c2a5", "#fc8d62"),
  label.size = 3,
  hjust = 0.5,
  vjust = -1,
  axis.title.size = 11,
  axis.text.size = 10,
  legend.text.size = 11,
  legend.title = "Group",
  legend.title.size = 11,
  legend.position = "right",
  plot.title = "",
  plot.title.size = 14,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "control_pca_samples"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions. Also, table with fold change values returned from RQ_dCt() and RQ_ddCt() functions can be used, but in such situation a pairwise.FCh parameter must be set to TRUE.
sel.Gene	Character vector with names of genes to include, or "all" (default) to use all genes.
pairwise.FCh	Logical: If fold change values returned from RQ_dCt() and RQ_ddCt() functions in a pairwise approach are used as data, this parameter should be set to TRUE, otherwise to FALSE (default).
point.size	Numeric: size of points. Default to 4.
point.shape	Integer: shape of points. Default to 19.
alpha	Numeric: transparency of points, a value between 0 and 1. Default to 0.7.
colors	Character vector containing colors for compared groups. The number of colors must be equal to the number of groups. Default to c("#66c2a5", "#fc8d62").
label.size	Numeric: size of points labels (names of samples). Default to 3.
hjust	Numeric: horizontal position of points labels. Default to 0.
vjust	Numeric: vertical position of points labels. Default to -1.
axis.title.size	Integer: font size of axis titles. Default to 11.
axis.text.size	Integer: font size of axis text. Default to 10.
legend.text.size	Integer: font size of legend text. Default to 11.
legend.title	Character: title of legend. Default to "Group".
legend.title.size	Integer: font size of legend title. Default to 11.
legend.position	Position of the legend, can be "top", "right" (default), "bottom", "left", or "none" (no legend). See description for legend.position parameter in ggplot2::theme() function.
plot.title	Character: title of plot. Default to "".
plot.title.size	Integer: font size of plot title. Default to 14.
save.to.tiff	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
dpi	Integer: resolution of saved .tiff file. Default to 600.
width	Numeric: width (in cm) of saved .tiff file. Default to 15.
height	Numeric: height (in cm) of saved .tiff file. Default to 15.
name.tiff	character: name of saved .tiff file, without ".tiff" name of extension. Default to "control_pca_samples".

Value

Object with plot. The plot is also displayed on the graphic device.

Examples

```
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
control_pca_sample(data.dCt)
```

`corr_gene` *corr_gene*

Description

This function performs correlation analysis of genes based on the data, useful to gain insight into relationships between analyzed genes.

Usage

```
corr_gene(
  data,
  sel.Gene = "all",
  type = "upper",
  method = "pearson",
  add.coef = "black",
  order = "hclust",
  hclust.method = "average",
  size = 0.6,
  coef.size = 0.6,
  p.adjust.method = "BH",
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "corr_genes",
  save.to.txt = FALSE,
  name.txt = "corr_genes"
)
```

Arguments

`data`	Object returned from make_Ct_ready() or delta_Ct() functions.
`sel.Gene`	Character vector with names of genes to include, or "all" (default) to use all genes.

type	Character: type of displayed matrix, must be one of the 'full' (full matrix), 'upper' (upper triangular, default) or 'lower' (lower triangular).
method	Character: type of correlations to compute, can be "pearson" (default) for Pearson's correlation coefficients or "spearman" for Spearman's rank correlation coefficients.
add.coef	If correlation coefficients should be add to the plot, specify color of coefficients (default to "black"). If NULL, correlation coefficients will not be printed.
order	Character: method used for ordering the correlation matrix, inherited from corrplot::corrplot() function. Must be one of the "original" (original order), "AOE" (angular order of the eigenvectors), "FPC" (first principal component order), "hclust" (hierarchical clustering order, default), or "alphabet" (alphabetical order).
hclust.method	Character: name of method used for hclust agglomeration, must be one of "ward", ward.D", "ward.D2", "single", "complete", "average" (default), "mcquitty", "median" or "centroid".
size	Numeric: size of variable names and numbers in legend. Default to 0.6.
coef.size	Numeric: size of correlation coefficients. Default to 0.6.
p.adjust.method	Character: p value correction method for multiple testing, one of the "holm", "hochberg", "hommel", "bonferroni", "BH" (default), "BY","fdr", or "none". See documentation for stats::p.adjust() function for details.
save.to.tiff	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
dpi	Integer: resolution of saved .tiff file. Default to 600.
width	Numeric: width (in cm) of saved .tiff file. Default to 15.
height	Numeric: height (in cm) of saved .tiff file. Default to 15.
name.tiff	Character: name of saved .tiff file, without ".tiff" name of extension. Default to "corr_genes".
save.to.txt	Logical: if TRUE, correlation results (sorted by absolute values of correlation coefficients in descending order) will be saved to .txt file. Default to FALSE.
name.txt	character: name of saved .txt file, without ".txt" name of extension.. Default to "corr_genes".

Value

Plot illustrating correlation matrix (displayed on the graphic device) and data frame with computed correlation coefficients and p values.

Examples

```
library(tidyverse)
library(Hmisc)
library(corrplot)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
```

```
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
corr.genes <- corr_gene(data.dCt)
head(corr.genes)
```

`corr_sample` *corr_sample*

Description

This function performs correlation analysis of samples based on the data. Results are useful to gain insight into relationships between analyzed samples.

Usage

```
corr_sample(
  data,
  sel.Sample = "all",
  type = "upper",
  method = "pearson",
  add.coef = "black",
  order = "hclust",
  hclust.method = "average",
  size = 0.6,
  coef.size = 0.6,
  p.adjust.method = "BH",
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "corr_samples",
  save.to.txt = FALSE,
  name.txt = "corr_samples"
)
```

Arguments

`data`	Object returned from make_Ct_ready() or delta_Ct() functions.
`sel.Sample`	Character vector with names of samples to include, or "all" (default) to use all samples.
`type`	Character: type of displayed matrix, must be one of the 'full' (full matrix), 'upper' (upper triangular, default) or 'lower' (lower triangular).
`method`	Character: type of correlations to compute, can be "pearson" (default) for Pearson's correlation coefficients or "spearman" for Spearman's rank correlation coefficients.

add.coef	If correlation coefficients should be add to the plot, specify color of coefficients (default to "black"). If NULL, correlation coefficients will not be printed.
order	Character: method used for ordering the correlation matrix, inherited from corrplot::corrplot() function. Must be one of the "original" (original order), "AOE" (angular order of the eigenvectors), "FPC" (first principal component order), "hclust" (hierarchical clustering order, default), or "alphabet" (alphabetical order).
hclust.method	Character: name of method used for hclust agglomeration, must be one of "ward", ward.D", "ward.D2", "single", "complete", "average" (default), "mcquitty", "median" or "centroid".
size	Numeric: size of variable names and numbers in legend. Default to 0.6.
coef.size	Numeric: size of correlation coefficients. Default to 0.6.
p.adjust.method	Character: p value correction method for multiple testing, one of the "holm", "hochberg", "hommel", "bonferroni", "BH" (default), "BY","fdr", or "none". See documentation for stats::p.adjust() function for details.
save.to.tiff	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
dpi	Integer: resolution of saved .tiff file. Default to 600.
width	Numeric: width (in cm) of saved .tiff file. Default to 15.
height	Numeric: height (in cm) of saved .tiff file. Default to 15.
name.tiff	Character: name of saved .tiff file, without ".tiff" name of extension. Default to "corr_samples".
save.to.txt	Logical: if TRUE, correlation results (sorted by absolute values of correlation coefficients in descending order) will be saved to .txt file. Default to FALSE.
name.txt	character: name of saved .txt file, without ".txt" name of extension.. Default to "corr_samples".

Value

Plot illustrating correlation matrix (displayed on the graphic device) and data.frame with computed correlation coefficients and p values.

Examples

```
library(tidyverse)
library(Hmisc)
library(corrplot)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
corr.samples <- corr_sample(data.CtF.ready)
head(corr.samples)
```

data.Ct — *Gene expression dataset for RQdeltaCT package.*

Description

Dataset that contain table with gene expression data obtained for 19 genes and 64 samples using qPCR experiments with TaqMan assays. Samples are divided into two groups: Disease (40 samples) and Control (24 samples). This dataset is a part of data used in article: Zalewski, D.; Chmiel, P.; Kołodziej, P.; Borowski, G.; Feldo, M.; Kocki, J.; Bogucka-Kocka, A. Dysregulations of Key Regulators of Angiogenesis and Inflammation in Abdominal Aortic Aneurysm. Int. J. Mol. Sci. 2023, 24, 12087. https://doi.org/10.3390/ijms241512087

Usage

```
data.Ct
```

Format

A data frame with 1288 rows and 5 variables:

`Sample` character COLUMN_DESCRIPTION

`Gene` character COLUMN_DESCRIPTION

`Ct` character COLUMN_DESCRIPTION

`Group` character COLUMN_DESCRIPTION

`Flag` character COLUMN_DESCRIPTION

Source

Dataset is attached to the 'RQdeltaCT' package and can be loaded using 'data(data.Ct)' command.

data.Ct.pairwise — *Gene expression dataset for RQdeltaCT package. Siutable for pairwise analysis.*

Description

Dataset that contain table with gene expression data obtained for 18 genes and 42 samples using qPCR experiments with TaqMan assays. Samples are divided into two groups: Before (21 samples) and After (21 samples). This dataset is created based on the real data used in article: Zalewski, D.; Chmiel, P.; Kołodziej, P.; Borowski, G.; Feldo, M.; Kocki, J.; Bogucka-Kocka, A. Dysregulations of Key Regulators of Angiogenesis and Inflammation in Abdominal Aortic Aneurysm. Int. J. Mol. Sci. 2023, 24, 12087. https://doi.org/10.3390/ijms241512087

Usage

```
data.Ct.pairwise
```

Format

A data frame with 756 rows and 4 variables:

`Sample` character COLUMN_DESCRIPTION

`Gene` character COLUMN_DESCRIPTION

`Ct` character COLUMN_DESCRIPTION

`Group` character COLUMN_DESCRIPTION

Source

Dataset is attached to the 'RQdeltaCT' package and can be loaded using 'data(data.Ct.pairwise)' command.

`delta_Ct` *delta_Ct*

Description

This function calculates delta Ct (dCt) values by subtracting Ct values of reference gene or genes from Ct values of remaining genes. Obtained dCt values can be further transformed by using 2^-dCt formula (if transform == TRUE).

Usage

```
delta_Ct(
  data,
  ref,
  normalise = TRUE,
  transform = FALSE,
  save.to.txt = FALSE,
  name.txt = "data_dCt"
)
```

Arguments

`data`	Data object returned from make_Ct_ready function.
`ref`	Character vector with name of one or more reference genes.
`normalise`	Logical: if TRUE, data normalisation will be done using reference gene/genes (provided in ref parameter).
`transform`	Logical: if TRUE, calculated dCt values will be transformed using 2^-dCt formula. Default to FALSE.
`save.to.txt`	Logical: if TRUE, returned data will be saved to .txt file. Default to FALSE.
`name.txt`	Character: name of saved .txt file, without ".txt" name of extension. Default to "data_dCt".

Value

Data frame with dCt values.

Examples

```
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
head(data.dCt)
data.dCt.exp <- delta_Ct(data.CtF.ready, ref = "Gene8", transform = TRUE)
head(data.dCt.exp)
```

FCh_plot *FCh_plot*

Description

This function creates barplot that illustrate fold change values with indicating of significance by different colors of bars.

Usage

```
FCh_plot(
  data,
  use.p = TRUE,
  mode,
  p.threshold = 0.05,
  use.FCh = FALSE,
  FCh.threshold = 2,
  use.sd = FALSE,
  sel.Gene = "all",
  bar.width = 0.8,
  signif.show = FALSE,
  signif.labels,
  signif.length = 0.2,
  signif.dist = 0.5,
  y.exp.low = 0.1,
  y.exp.up = 0.1,
  angle = 0,
  rotate = FALSE,
  colors = c("#66c2a5", "#fc8d62"),
  border.color = "black",
  x.axis.title = "",
```

```
  y.axis.title = "log10(Fold change)",
  axis.title.size = 11,
  axis.text.size = 10,
  legend.text.size = 11,
  legend.title = "Selected as significant?",
  legend.title.size = 11,
  legend.position = "top",
  plot.title = "",
  plot.title.size = 14,
  dpi = 600,
  width = 15,
  height = 15,
  save.to.tiff = FALSE,
  name.tiff = "FCh_plot"
)
```

Arguments

data	Object returned from RQ_dCt() or RQ_ddCt() functions.
use.p	Logical: if TRUE, p value threshold will be used to label gene as significant.
mode	Character: which p value should be used? One of the "t" (p values from Student's t test), "t.adj" (adjusted p values from Student's t test), "mw" (p values from Mann-Whitney U test),"mw.adj" (adjusted p values from Mann-Whitney U test), "depends" (if data in both compared groups were considered as derived from normal distribution (p value from Shapiro_Wilk test > 0.05) - p values from Student's t test will be used for significance assignment, otherwise p values from Mann-Whitney U test will be used), "depends.adj" (if data in both compared groups were considered as derived from normal distribution (p value from Shapiro_Wilk test > 0.05) - adjusted p values from Student's t test will be used for significance assignment, otherwise adjusted p values from Mann-Whitney U test will be used), and "user" that can be used the user intend to use another p values, e.g. obtained from other statistical test. In such situation, before run FCh_plot function, the user should prepare data frame object named "user" that contains two columns, the first of them with Gene names and the second with p values. The order of columns must be kept as described.
p.threshold	Numeric: threshold of p values for statistical significance. Default to 0.05.
use.FCh	Logical: if TRUE, the criterion of fold change will be also used for significance assignment of genes.
FCh.threshold	Numeric: threshold of fold change values used for significance assignment of genes. If is set to 2 (default), genes with 2-fold changed expression (increased or decreased) between groups will be assigned as significant.
use.sd	Logical: if TRUE, errorbars with standard deviations will be added to the plot.
sel.Gene	Character vector with names of genes to include, or "all" (default) to use all names of genes.
bar.width	numeric: width of bars.
signif.show	Logical: if TRUE, labels for statistical significance will be added to the plot. Default to FALSE.

`signif.labels` Character vector with statistical significance labels (e.g. "ns.","***", etc.). The number of elements must be equal to the number of genes used for plotting. All elements in the vector must be different; therefore, symmetrically white spaces to repeated labels must be add to the same labels, e.g. "ns.", " ns. ", " ns. ".

`signif.length` Numeric: length of horizontal bars under statistical significance labels, values from 0 to 1.

`signif.dist` Numeric: distance between errorbar and statistical significance labels.

`y.exp.low, y.exp.up` Numeric: space between data on the plot and lower or upper axis. Useful to add extra space for statistical significance labels when faceting = TRUE.

`angle` Integer: value of angle in which names of genes are displayed. Default to 0.

`rotate` Logical: if TRUE, bars will be arranged horizontally. Deafault to FALSE.

`colors` Character vector length of one (when use.p = FALSE) or two (when use.p = TRUE), containing colors for significant and no significant genes.

`border.color` Character: color for borders of bars.

`x.axis.title` Character: title of x axis. Default to "Gene".

`y.axis.title` Character: title of y axis. Default to "value".

`axis.title.size` Integer: font size of axis titles. Default to 11.

`axis.text.size` Integer: font size of axis text. Default to 10.

`legend.text.size` Integer: font size of legend text. Default to 11.

`legend.title` Character: title of legend. Default to "Group".

`legend.title.size` Integer: font size of legend title. Default to 11.

`legend.position` Position of the legend, can be one of "top" (default), "right", "bottom", "left", or "none" (no legend). See description for legend.position in ggplot2::theme() function.

`plot.title` Character: title of plot. Default to "".

`plot.title.size` Integer: font size of plot title. Default to 14.

`dpi` Integer: resolution of saved .tiff file. Default to 600.

`width` Numeric: width (in cm) of saved .tiff file. Default to 15.

`height` Numeric: height (in cm) of saved .tiff file. Default to 15.

`save.to.tiff` Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.

`name.tiff` Character: name of saved .tiff file, without ".tiff" name of extension. Default to "FCh_plot".

Value

List containing object with barplot and data frame with results. Created plot is also displayed on graphic device.

Examples

```
library(ggsignif)
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
data.dCtF <- filter_transformed_data(data.dCt, remove.Sample = c("Control11"))
results.ddCt <- RQ_ddCt(data.dCtF, "Disease", "Control")

signif.labels <- c("****",
                   "**",
                   "ns.",
                   " ns. ",
                   "  ns.  ",
                   "   ns.   ",
                   "    ns.    ",
                   "     ns.     ",
                   "      ns.      ",
                   "       ns.       ",
                   "        ns.        ",
                   "         ns.         ",
                   "          ns.          ",
                   "           ns.           ",
                   "***")
FCh.plot <- FCh_plot(results.ddCt,
                     mode = "depends",
                     use.FCh = TRUE,
                     FCh.threshold = 2.5,
                     signif.labels = signif.labels,
                     angle = 30)
head(FCh.plot[[2]])

# with user p values - calculated using stats::wilcox.test() function:
user <- data.dCt %>%
pivot_longer(cols = -c(Group, Sample), names_to = "Gene", values_to = "dCt") %>%
  group_by(Gene) %>%
  summarise(MW_test_p = wilcox.test(dCt ~ Group)$p.value, .groups = "keep")

FCh.plot <- FCh_plot(results.ddCt,
                     mode = "user",
                     use.FCh = TRUE,
                     FCh.threshold = 2,
                     signif.labels = signif.labels,
                     angle = 30)
head(FCh.plot[[2]])
```

---

filter_Ct *filter_Ct*

Description

This function filters Ct data according to the used filtering criteria (see parameters).

Usage

```
filter_Ct(
  data,
  flag.Ct = "Undetermined",
  maxCt = 35,
  flag = c("Undetermined"),
  remove.Gene = c(""),
  remove.Sample = c(""),
  remove.Group = c("")
)
```

Arguments

data	Object returned from read_Ct_long() or read_Ct_wide() function, or data frame containing column named "Sample" with sample names, column named "Gene" with gene names, column named "Ct" with raw Ct values, column named "Group" with group names. Optionally, data frame can contain column named "Flag" with flag information (e.g. "Undetermined" and "OK"), which will be used for filtering.
flag.Ct	Character of a flag used for undetermined Ct values, default to "Undetermined".
maxCt	Numeric, a maximum of Ct value allowed.
flag	Character: flag used in Flag column for values which should be filtered out, default to "Undetermined".
remove.Gene	Character: vector with names of genes to remove from data
remove.Sample	Character: vector with names of samples to remove from data
remove.Group	Character: vector with names of groups to remove from data

Value

Data frame with filtered data.

Examples

```
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))

dim(data.Ct)
dim(data.CtF)
```

filter_transformed_data *filter_transformed_data*

Description

This function filters transformed Ct data (filtered Ct, delta Ct, and 2^-dCt data) according to the used filtering criteria (see parameters).

Usage

```
filter_transformed_data(
  data,
  remove.Gene = c(""),
  remove.Sample = c(""),
  remove.Group = c("")
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions.
remove.Gene	Character: vector with names of genes to remove from data.
remove.Sample	Character: vector with names of samples to remove from data.
remove.Group	Character: vector with names of groups to remove from data.

Value

Data frame with filtered data.

Examples

```
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
data.dCtF <- filter_transformed_data(data.dCt, remove.Sample = c("Control11"))

dim(data.dCt)
dim(data.dCtF)
```

find_ref_gene *find_ref_gene*

Description

This function assess gene expression stability by calculation the following parameters: minimum, maximum, standard deviation, variance, and stability measures from NormFinder and geNorm algorithms. It also presents C values graphically on a line plot. This function is helpful to select the best reference gene for normalization of Ct values.

Usage

```
find_ref_gene(
  data,
  groups,
  candidates,
  colors,
  norm.finder.score = TRUE,
  genorm.score = TRUE,
  line.width = 1,
  angle = 0,
  x.axis.title = "",
  y.axis.title = "Ct",
  axis.title.size = 11,
  axis.text.size = 10,
  legend.title = "",
  legend.title.size = 11,
  legend.text.size = 11,
  legend.position = "top",
  plot.title = "",
  plot.title.size = 14,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "Ct_reference_gene_selection"
)
```

Arguments

data	Object returned from make_Ct_ready() functions.
groups	Character vector with names of groups used for analysis. If all groups should be included to the analysis, groups parameter should be set to "all".
candidates	Character: vector of names of genes - candidates for gene reference.
colors	Character: vector of colors for genes, the number of colors should be equal to the number of candidate genes.

`norm.finder.score`
: Logical: if TRUE, NormFinder stability score will be calculated. Default to TRUE.

`genorm.score`
: Logical: if TRUE, geNorm stability score will be calculated. Default to TRUE.

`line.width`
: Numeric: width of lines drawn in the plot. Default to 1.

`angle`
: Integer: value of angle in which names of genes are displayed. Default to 0.

`x.axis.title`
: Character: title of x axis. Default to "".

`y.axis.title`
: Character: title of y axis. Default to "Ct".

`axis.title.size`
: Integer: font size of axis titles. Default to 11.

`axis.text.size`
: Integer: font size of axis text. Default to 10.

`legend.title`
: Character: title of legend. Default to "".

`legend.title.size`
: Integer: font size of legend title. Default to 11.

`legend.text.size`
: Integer: font size of legend text. Default to 11.

`legend.position`
: Position of the legend, can be "top" (default), "right", "bottom", "left", or "none" (no legend). See description for legend.position in ggplot2::theme() function.

`plot.title`
: Character: title of plot. Default to "".

`plot.title.size`
: Integer: font size of plot title. Default to 14.

`save.to.tiff`
: Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.

`dpi`
: Integer: resolution of saved .tiff file. Default to 600.

`width`
: Numeric: width (in cm) of saved .tiff file. Default to 15.

`height`
: Numeric: height (in cm) of saved .tiff file. Default to 15.

`name.tiff`
: Character: name of saved .tiff file, without ".tiff" name of extension. Default to "Ct_reference_gene_selection".

Value

List containing an object with plot and a table with calculated parameters. Created plot is also displayed on the graphic device.

Examples

```
library(ctrlGene)
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
ref <- find_ref_gene(data.CtF.ready,
                     groups = c("Disease","Control"),
```

```
                    candidates = c("Gene4", "Gene8","Gene10","Gene16","Gene17", "Gene18"),
                   col = c("#66c2a5", "#fc8d62","#6A6599", "#D62728", "#1F77B4", "black"),
                      norm.finder.score = TRUE,
                      genorm.score = TRUE)
ref[[2]]
```

---

log_reg *log_reg*

Description

This function performs logistic regression analysis, computes odd ratio values, and presents them graphically.

Usage

```
log_reg(
  data,
  sel.Gene = "all",
  group.study,
  group.ref,
  increment,
  centerline = 1,
  ci = 0.95,
  log.axis = FALSE,
  x.axis.title = "Odds ratio",
  y.axis.title = "",
  axis.title.size = 11,
  axis.text.size = 10,
  legend.title = "p value",
  legend.text.size = 11,
  legend.title.size = 11,
  legend.position = "right",
  plot.title = "",
  plot.title.size = 14,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "OR_plot",
  save.to.txt = FALSE,
  name.txt = "OR_results"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions.
sel.Gene	Character vector with names of genes to include, or "all" (default) to use all genes.
group.study	Character: name of study group (group of interest).
group.ref	Character: name of reference group.
increment	Numeric or "mean": the change of expression for which odds ratio values are calculated. If increment = 1, odds ratio values regards for a one-unit increase in gene expression (more suitable where data are not transformed using 2^-value formula). If increment = "mean", odds ratio values are calculated for the situation where gene expression increases by mean of gene expression levels in all samples (more suitable where data were transformed using 2^-value formula).
centerline	Numeric: position of vertical centerline on the plot. Default to 1.
ci	Numeric: confidence level used for computation of confidence interval. Default to 0.95.
log.axis	Logical: if TRUE, axis with odds ratio values will be in log10 scale. Default to FALSE.
x.axis.title	Character: title of x axis. Default to "Gene".
y.axis.title	Character: title of y axis. Default to "value".
axis.title.size	Integer: font size of axis titles. Default to 11.
axis.text.size	Integer: font size of axis text. Default to 10.
legend.title	Character: title of legend. Default to "Group".
legend.text.size	Integer: font size of legend text. Default to 11.
legend.title.size	Integer: font size of legend title. Default to 11.
legend.position	Position of the legend, can be one of "top" (default), "right", "bottom", "left", or "none" (no legend). See description for legend.position in ggplot2::theme() function.
plot.title	Character: title of plot. Default to "".
plot.title.size	Integer: font size of plot title. Default to 14.
save.to.tiff	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
dpi	Integer: resolution of saved .tiff file. Default to 600.
width	Numeric: width (in cm) of saved .tiff file. Default to 15.
height	Numeric: height (in cm) of saved .tiff file. Default to 15.
name.tiff	Character: name of saved .tiff file, without ".tiff" name of extension. Default to "OR_plot".
save.to.txt	Logical: if TRUE, returned table with results will be saved to .txt file. Default to FALSE.
name.txt	Character: name of saved .txt file, without ".txt" name of extension. Default to "OR_results".

Value

A list that contains an object with plot and data frame with calculated results. Created plot is also displayed on the graphic device.

Examples

```
library(tidyverse)
library(oddsratio)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                       remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready,
                     ref = "Gene8")
log.reg.results <- log_reg(data.dCt,
                           sel.Gene = c("Gene1","Gene16","Gene19","Gene20"),
                           group.study = "Disease",
                           group.ref = "Control",
                           increment = 1)
```

`make_Ct_ready` *make_Ct_ready*

Description

This function collapses technical replicates (if present in data) by means and (optionally) imputes missing data by means calculated separately for each group. This function also prepares Ct data for further steps of analysis.

Usage

```
make_Ct_ready(
  data,
  imput.by.mean.within.groups,
  save.to.txt = FALSE,
  name.txt = "Ct_ready"
)
```

Arguments

`data`	Data object returned from read_Ct_long(), read_Ct_wide() or filter_Ct() function, or data frame containing column named "Sample" with sample names, column named "Gene" with gene names, column named "Ct" with raw Ct values (must be numeric), column named "Group" with group names. Presence of any other columns is allowed, but they will not be used by this function.

imput.by.mean.within.groups
: Logical: if TRUE, missing values will be imputed by means calculated separately for each group. This parameter can influence results, thus to draw more user attention on this parameter, no default value was set.

save.to.txt
: Logical: if TRUE, returned data will be saved to .txt file. Default to FALSE.

name.txt
: Character: name of saved .txt file, without ".txt" name of extension. Default to "Ct_ready".

Value

Data frame with prepared data. Information about number and percentage of missing values is also printed.

Examples

```
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
head(data.CtF.ready)
```

---

norm_finder *norm_finder*

Description

This function calculates stability scores using NormFinder algorithm (https://www.moma.dk/software/normfinder) published in https://aacrjournals.org/cancerres/article/64/15/5245/511517/Normalization-of-Real-Time-Quantitative-Reverse. This function is internally used by other RQdeltaCT package function, find_ref_gene(); therefore norm_finder() function does not need to be used separately.

Usage

```
norm_finder(
  data,
  candidates,
  save.to.txt = FALSE,
  name.txt = "NormFinder_results"
)
```

Arguments

data	Object returned from make_Ct_ready() functions.
candidates	Character: vector of names of genes - candidates for reference gene.
save.to.txt	Logical: if TRUE, returned table with results will be saved to .txt file. Default to FALSE.
name.txt	Character: name of saved .txt file, without ".txt" name of extension. Default to "norm_finder_results".

Value

Table with calculated stability score; the lowest value the best candidate for reference gene.

Examples

```
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
reference.stability.nF <- norm_finder(data.CtF.ready,
                                      candidates = c("Gene4",
                                                     "Gene8",
                                                     "Gene10",
                                                     "Gene16",
                                                     "Gene17",
                                                     "Gene18"))
```

---

parallel_plot *parallel_plot*

Description

This function illustrates expression values in a pairwise samples as series of lines connected across each axis. This function can be used only for a pairwise data.

Usage

```
parallel_plot(
  data,
  sel.Gene = "all",
  scale = "globalminmax",
  alpha = 0.7,
  custom.colors = FALSE,
  order = "anyClass",
  colors,
  linewidth = 1,
```

```
  show.points = TRUE,
  x.axis.title = "",
  y.axis.title = "value",
  axis.title.size = 11,
  axis.text.size = 10,
  legend.text.size = 11,
  legend.title = "Gene",
  legend.title.size = 11,
  legend.position = "top",
  plot.title = "",
  plot.title.size = 14,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "parallel_plot"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions.
sel.Gene	Character vector with names of genes to include, or "all" (default) to use all genes.
scale	Character: a scale used for data presentation, one of the passed to ggparcoord() function. Generally, scaling is not required since variables are the same units. Default to "globalminmax (no scaling).
alpha	Numeric: transparency of lines, a value between 0 and 1. Default to 0.7.
custom.colors	Logical: if custom vector colors for genes is provided (and passed to the colors parameter), it should be set to TRUE. For default colors, use custom.colors = FALSE (default).
order	Character: method for groups ordering, one of the used in the ggparcoord() function. Default to 'anyClass'. Must either be a vector of column indices (obligatory if only one gene is plotted), starting from 3 (e.g., for two groups it can be c(3,4) or c(4,3)), or one of 'skewness', 'allClass', 'anyClass' (default), as well as scagnostic measures available in the 'scagnostics' package (must be loaded): 'Outlying', 'Skewed', 'Clumpy', 'Sparse', 'Striated', 'Convex', 'Skinny', 'Stringy', 'Monotonic'.
colors	Character vector containing custom colors for genes. The number of colors must be equal to the number of presented genes. Must be provided if custom.colors = TRUE.
linewidth	Numeric: width of lines. Default to 1.
show.points	Logical: if TRUE (default), points will be also shown.
x.axis.title	Character: title of x axis. Default to "".
y.axis.title	Character: title of y axis. Default to "value".
axis.title.size	Integer: font size of axis titles. Default to 11.

`axis.text.size` Integer: font size of axis text. Default to 10.

`legend.text.size`
Integer: font size of legend text. Default to 11.

`legend.title` Character: title of legend for groups. Default to "Gene".

`legend.title.size`
Integer: font size of legend title. Default to 11.

`legend.position`
Position of the legend, can be "top", "right" (default), "bottom", "left", or "none" (no legend). See description for legend.position parameter in ggplot2::theme() function.

`plot.title` Character: title of plot. Default to "".

`plot.title.size`
Integer: font size of plot title. Default to 14.

`save.to.tiff` Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.

`dpi` Integer: resolution of saved .tiff file. Default to 600.

`width` Numeric: width (in cm) of saved .tiff file. Default to 15.

`height` Numeric: height (in cm) of saved .tiff file. Default to 15.

`name.tiff` character: name of saved .tiff file, without ".tiff" name of extension. Default to "pca_and_kmeans".

Value

An object with plot. Created plot is also displayed on the graphic device.

Examples

```
library(tidyverse)
library(GGally)
data(data.Ct.pairwise)
data.CtF.pairwise <- filter_Ct(data = data.Ct.pairwise,
                               flag.Ct = "Undetermined",
                               maxCt = 35,
                               flag = c("Undetermined"),
                             remove.Gene = c("Gene9", "Gene2","Gene5", "Gene11","Gene1"))
data.CtF.ready.pairwise <- make_Ct_ready(data = data.CtF.pairwise,
                                         imput.by.mean.within.groups = TRUE)
data.dCt.pairwise <- delta_Ct(data = data.CtF.ready.pairwise,
                              ref = "Gene4")
parallel.plot <- parallel_plot(data = data.dCt.pairwise,
                               sel.Gene = c("Gene8","Gene19"))
```

pca_kmeans *pca_kmeans*

Description

This function performs principal component analysis (PCA) together with k means analysis for samples, and generate plot that illustrate spatial arrangement of samples based on the two first components and with assignment to k means clusters. PCA analysis can not deal with missing values, thus all samples with at least one missing value are removed from data before analysis.

Usage

```
pca_kmeans(
  data,
  do.k.means = TRUE,
  k.clust = 2,
  clust.names = c("Cluster1", "Cluster2"),
  sel.Gene = "all",
  point.size = 4,
  point.shape = c(19, 17),
  alpha = 0.7,
  point.color = c("#66c2a5", "#fc8d62"),
  add.sample.labels = FALSE,
  label.size = 3,
  hjust = 0,
  vjust = -1,
  axis.title.size = 11,
  axis.text.size = 10,
  legend.text.size = 11,
  legend.title.group = "Group",
  legend.title.cluster = "Cluster",
  legend.title.size = 11,
  legend.position = "right",
  plot.title = "",
  plot.title.size = 14,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "pca_and_kmeans"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions.
do.k.means	Logical: if TRUE (default), k means analysis will be performed.

`k.clust` Integer: number of clusters for k means analysis. Default to 2.

`clust.names` Character vector with names of clusters, must be equal to the number of clusters specified in the k.clust parameter.

`sel.Gene` Character vector with names of genes to include, or "all" (default) to use all genes.

`point.size` Numeric: size of points. Default to 4.

`point.shape` Integer: shape of points. If do.k.means = TRUE, the number of provided values must be equal to the number of cluster (k.clust). Default to c(19, 17).

`alpha` Numeric: transparency of points, a value between 0 and 1. Default to 0.7.

`point.color` Character vector containing colors for compared groups. The number of colors must be equal to the number of groups. Default to c("#66c2a5", "#fc8d62").

`add.sample.labels` Logical: if TRUE, points labels (names of samples) will be added. Default to FALSE.

`label.size` Numeric: size of points labels (names of samples). Default to 3.

`hjust` Numeric: horizontal position of points labels. Default to 0.

`vjust` Numeric: vertical position of points labels. Default to -1.

`axis.title.size` Integer: font size of axis titles. Default to 11.

`axis.text.size` Integer: font size of axis text. Default to 10.

`legend.text.size` Integer: font size of legend text. Default to 11.

`legend.title.group` Character: title of legend for groups. Default to "Group".

`legend.title.cluster` Character: title of legend for k means clusters. Default to "Clusters".

`legend.title.size` Integer: font size of legend title. Default to 11.

`legend.position` Position of the legend, can be "top", "right" (default), "bottom", "left", or "none" (no legend). See description for legend.position parameter in ggplot2::theme() function.

`plot.title` Character: title of plot. Default to "".

`plot.title.size` Integer: font size of plot title. Default to 14.

`save.to.tiff` Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.

`dpi` Integer: resolution of saved .tiff file. Default to 600.

`width` Numeric: width (in cm) of saved .tiff file. Default to 15.

`height` Numeric: height (in cm) of saved .tiff file. Default to 15.

`name.tiff` character: name of saved .tiff file, without ".tiff" name of extension. Default to "pca_and_kmeans".

Value

A list containing object with plot and, if do.k.means = TRUE, a confusion matrix that show classification performance of k means method. Created plot is also displayed on the graphic device.

Examples

```
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
pca_kmeans(data.dCt, sel.Gene = c("Gene1","Gene16","Gene19","Gene20"))
```

---

read_Ct_long *read_Ct_long*

Description

Imports a long-format table with Ct values.

Usage

```
read_Ct_long(
  path,
  sep,
  dec,
  skip = 0,
  column.Sample,
  column.Gene,
  column.Ct,
  column.Group,
  add.column.Flag = FALSE,
  column.Flag
)
```

Arguments

path	Path to a .txt or csv. file with long-type table with Ct values. This table must contain at least 4 columns, separately for sample names, gene names, Ct values and group names (these columns will be imported by this function). Imported table could also contain a column with a flag information, which could be optionally imported (see add.col.Flag and col.Flag parameters).
sep	Character of a field separator in imported file.
dec	Character used for decimal points in Ct values.

skip	Integer: number of lines of the data file to skip before beginning to read data. Default to 0.
column.Sample	Integer: number of column with sample names.
column.Gene	Integer: number of column with gene names.
column.Ct	Integer: number of column with Ct values.
column.Group	Integer: number of column with group names.
add.column.Flag	Logical: if data contains a column with flag information which should also be imported, this parameter should be set to TRUE. Default to FALSE.
column.Flag	Integer: number of column with flag information. Should be specified if add.col.Flag = TRUE. This column should contain a character-type values (e.g. "Undetermined" and "OK"), however, other types of values are allowed (e.g. numeric), but must be converted to character or factor after importing data (see examples).

Value

Data.frame in long format ready for analysis.

Examples

```
path <- system.file("extdata",
                    "data_Ct_long.txt",
                    package = "RQdeltaCT")

library(tidyverse)
data.Ct <- read_Ct_long(path = path,
                        sep = "\t",
                        dec = ".",
                        skip = 0,
                        add.column.Flag = TRUE,
                        column.Sample = 1,
                        column.Gene = 2,
                        column.Ct = 5,
                        column.Group = 9,
                        column.Flag = 4)
str(data.Ct)

data.Ct <- mutate(data.Ct,
                  Flag = ifelse(Flag < 1, "Undetermined", "OK"))
str(data.Ct)
```

---

read_Ct_wide *read_Ct_wide*

Description

This function imports Ct data in a wide-format table with sample names given in columns.

Usage

```
read_Ct_wide(path.Ct.file, path.design.file, sep, dec)
```

Arguments

path.Ct.file	Path to wide-format table in .txt or csv. format with Ct values. This table must contain gene names in the first column, and sample names in the first row (genes by rows and samples by columns).
path.design.file	Path to table in .txt or csv. file that contains two columns: column named "Sample" with names of samples and column named "Group" with names of groups assigned to samples. The names of samples in this file must correspond to the names of columns in the file with Ct values.
sep	Character of a field separator in both imported files.
dec	Character used for decimal points in Ct values.

Details

This function needs two files to import: a wide-format table with Ct values and an additional file with group names (see parameters path.Ct.file and path.design.file for further details on tables structure). Both files are merged to return a long-format table ready for analysis. All parameters must be specified; there are no default values.

Value

Data frame in long format ready to analysis.

Examples

```
path.Ct.file <- system.file("extdata",
                            "data_Ct_wide.txt",
                            package = "RQdeltaCT")
path.design.file <- system.file("extdata",
                                "data_design.txt",
                                package = "RQdeltaCT")

library(tidyverse)
data.Ct <- read_Ct_wide(path.Ct.file = path.Ct.file,
                        path.design.file = path.design.file,
                        sep ="\t",
                        dec = ".")
str(data.Ct)
```

results_barplot *results_barplot*

Description

This function creates a barplot that illustrate mean and sd values of genes. Faceting and adding custom labels of statistical significance are available. This function is useful to present results for finally selected genes.

Usage

```
results_barplot(
  data,
  sel.Gene = "all",
  bar.width = 0.8,
  signif.show = FALSE,
  signif.labels,
  signif.length = 0.2,
  signif.dist = 0.2,
  faceting = FALSE,
  facet.row,
  facet.col,
  y.exp.low = 0.1,
  y.exp.up = 0.2,
  angle = 0,
  rotate = FALSE,
  colors = c("#66c2a5", "#fc8d62"),
  x.axis.title = "",
  y.axis.title = "value",
  axis.title.size = 11,
  axis.text.size = 10,
  legend.text.size = 11,
  legend.title = "Group",
  legend.title.size = 11,
  legend.position = "top",
  plot.title = "",
  plot.title.size = 14,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "results_barplot"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions.

`sel.Gene` Character vector with names of genes to include, or "all" (default) to use all genes.

`bar.width` Numeric: width of bars.

`signif.show` Logical: if TRUE, labels for statistical significance will be added to the plot. Default to FALSE.

`signif.labels` Character vector with statistical significance labels (e.g. "ns.","***", etc.). The number of elements must be equal to the number of genes used for plotting. All elements in the vector must be different; therefore, symmetrically white spaces to repeated labels must be add to the same labels, e.g. "ns.", " ns. ", " ns. ".

`signif.length` Numeric: length of horizontal bars under statistical significance labels, values from 0 to 1.

`signif.dist` Numeric: distance between errorbar and statistical significance labels. Can be in y axis units (if faceting = FALSE) or fraction of y axis value reached by errorbar (mean + sd value) (if faceting = TRUE).

`faceting` Logical: if TRUE (default), plot will be drawn with facets with free scales.

`facet.row, facet.col`
Integer: number of rows and columns to arrange facets.

`y.exp.low, y.exp.up`
Numeric: space between data on the plot and lower or upper axis. Useful to add extra space for statistical significance labels when faceting = TRUE.

`angle` Integer: value of angle in which names of genes are displayed. Default to 0.

`rotate` Logical: if TRUE, boxplots will be arranged horizontally. Default to FALSE.

`colors` Character vector length of one (when by.group = FALSE) or two (when by.group = TRUE), containing colors for groups. The numbers of colors must be equal to the number of groups. Default to c("#66c2a5", "#fc8d62").

`x.axis.title` Character: title of x axis. Default to "Gene".

`y.axis.title` character: title of y axis. Default to "value".

`axis.title.size`
Integer: font size of axis titles. Default to 11.

`axis.text.size` Integer: font size of axis text. Default to 10.

`legend.text.size`
Integer: font size of legend text. Default to 11.

`legend.title` Character: title of legend. Default to "Group".

`legend.title.size`
Integer: font size of legend title. Default to 11.

`legend.position`
Position of the legend, can be "top" (default), "right", "bottom", "left", or "none" (no legend). See description for legend.position in ggplot2::theme() function.

`plot.title` Character: title of plot. Default to "".

`plot.title.size`
Integer: font size of plot title. Default to 14.

`save.to.tiff` Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.

dpi	Integer: resolution of saved .tiff file. Default to 600.
width	Numeric: width (in cm) of saved .tiff file. Default to 15.
height	Numeric: height (in cm) of saved .tiff file. Default to 15.
name.tiff	Character: name of saved .tiff file, without ".tiff" name of extension. Default to "results_barplot".

Value

Object with plot. Created plot will be also displayed on graphic device.

Examples

```
library(ggsignif)
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
data.dCtF <- filter_transformed_data(data.dCt, remove.Sample = c("Control11"))
results_barplot(data.dCtF,
                sel.Gene = c("Gene1","Gene16","Gene19","Gene20"),
                signif.labels = c("****","*","***"," * "),
                angle = 30,
                signif.dist = 1.05,
                facet.row = 1,
                facet.col = 4,
                y.exp.up = 0.1,
                y.axis.title = bquote(~2^-dCt))
```

---

results_boxplot *results_boxplot*

Description

This function creates boxplot that illustrate distribution of data for selected genes. It is similar to control_boxplot_gene() function; however, some new options are added, including gene selection, faceting, adding mean labels to boxes, and adding statistical significance labels. This function can be used to present results for finally selected genes.

Usage

```
results_boxplot(
  data,
  coef = 1.5,
  sel.Gene = "all",
```

```
  by.group = TRUE,
  signif.show = FALSE,
  signif.labels,
  signif.length = 0.2,
  signif.dist = 0.2,
  faceting = TRUE,
  facet.row,
  facet.col,
  y.exp.low = 0.1,
  y.exp.up = 0.2,
  angle = 0,
  rotate = FALSE,
  add.mean = TRUE,
  add.mean.size = 2,
  add.mean.color = "black",
  colors = c("#66c2a5", "#fc8d62"),
  x.axis.title = "",
  y.axis.title = "value",
  axis.title.size = 11,
  axis.text.size = 10,
  legend.text.size = 11,
  legend.title = "Group",
  legend.title.size = 11,
  legend.position = "top",
  plot.title = "",
  plot.title.size = 14,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "results_boxplot"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions.
coef	Numeric: how many times of interquartile range should be used to determine range point for whiskers. Default to 1.5.
sel.Gene	Character vector with names of genes to include, or "all" (default) to use all genes.
by.group	Logical: if TRUE (default), distributions will be drawn by compared groups of samples.
signif.show	Logical: if TRUE, labels for statistical significance will be added to the plot. Default to FALSE.
signif.labels	Character vector with statistical significance labels (e.g. "ns.","***", etc.). The number of elements must be equal to the number of genes used for plotting. All elements in the vector must be different; therefore, symmetrically white spaces to repeated labels must be add to the same labels, e.g. "ns.", " ns. ", " ns. ".

`signif.length` Numeric: length of horizontal bars under statistical significance labels, values from 0 to 1.

`signif.dist` Numeric: distance between the highest value and statistical significance labels. Can be in y axis units (if faceting = FALSE) or fraction of y axis value reached by the highest value (if faceting = TRUE).

`faceting` Logical: if TRUE (default), plot will be drawn with facets with free scales.

`facet.row, facet.col`
Integer: number of rows and columns to arrange facets.

`y.exp.low, y.exp.up`
Numeric: space between data on the plot and lower and upper axis. Useful to add extra space for statistical significance labels when faceting = TRUE.

`angle` Integer: value of angle in which names of genes are displayed. Default to 0.

`rotate` Logical: if TRUE, boxplots will be arranged horizontally. Default to FALSE.

`add.mean` Logical: if TRUE, mean points will be added to boxes as squares. Default to TRUE.

`add.mean.size` Numeric: size of squares indicating means. Default to 2.

`add.mean.color` Character: color of squares indicating means. Default to "black".

`colors` Character vector length of one (when by.group = FALSE) or more (when by.group = TRUE), containing colors for groups. The number of colors must be equal to the number of groups. Default to c("#66c2a5", "#fc8d62").

`x.axis.title` Character: title of x axis. Default to "Gene".

`y.axis.title` Character: title of y axis. Default to "value".

`axis.title.size`
Integer: font size of axis titles. Default to 11.

`axis.text.size` Integer: font size of axis text. Default to 10.

`legend.text.size`
Integer: font size of legend text. Default to 11.

`legend.title` Character: title of legend. Default to "Group".

`legend.title.size`
Integer: font size of legend title. Default to 11.

`legend.position`
Position of the legend, can be "top" (default), "right", "bottom", "left", or "none" (no legend). See description for legend.position in ggplot2::theme() function.

`plot.title` Character: title of plot. Default to "".

`plot.title.size`
Integer: font size of plot title. Default to 14.

`save.to.tiff` Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.

`dpi` Integer: resolution of saved .tiff file. Default to 600.

`width` Numeric: width (in cm) of saved .tiff file. Default to 15.

`height` Numeric: height (in cm) of saved .tiff file. Default to 15.

`name.tiff` Character: name of saved .tiff file, without ".tiff" name of extension. Default to "results_boxplot".

Value

Object with plot. Created plot will be also displayed on the graphic device.

Examples

```
library(ggsignif)
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
data.dCtF <- filter_transformed_data(data.dCt, remove.Sample = c("Control11"))
results_boxplot(data.dCtF,
                sel.Gene = c("Gene1","Gene16","Gene19","Gene20"),
                signif.labels = c("****","*","***"," * "),
                angle = 30,
                signif.dist = 1.05,
                facet.row = 1,
                facet.col = 4,
                y.exp.up = 0.1,
                y.axis.title = bquote(~2^-dCt))
```

---

results_heatmap *results_heatmap*

Description

This function creatse heatmap with hierarchical clustering.

Usage

```
results_heatmap(
  data,
  sel.Gene = "all",
  dist.row = "euclidean",
  dist.col = "euclidean",
  clust.method = "average",
  col.groups,
 colors = c("navy", "#313695", "#4575B4", "#74ADD1", "#ABD9E9", "#E0F3F8", "#FFFFBF",
   "#FEE090", "#FDAE61", "#F46D43", "#D73027", "#C32B23", "#A50026", "#8B0000",
   "#7E0202", "#000000"),
  show.colnames = TRUE,
  show.rownames = TRUE,
  border.color = NA,
  fontsize = 10,
```

```
  fontsize.col = 10,
  fontsize.row = 10,
  angle.col = 0,
  cellwidth = NA,
  cellheight = NA,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "heatmap_results"
)
```

Arguments

`data`	Object returned from make_Ct_ready() or delta_Ct() functions.
`sel.Gene`	Character vector with names of genes to include, or "all" (default) to use all genes.
`dist.row, dist.col`	Character: name of method used for calculation of distances between rows or columns, derived from stats::dist() function, must be one of "euclidean" (default) , "maximum", "manhattan", "canberra", "binary" or "minkowski".
`clust.method`	Character: name of used method for agglomeration, derived from stats::hclust() function, must be one of "ward.D", "ward.D2", "single", "complete", "average" (default), "mcquitty", "median" or "centroid".
`col.groups`	A named list with colors for groups annotation (see example).
`colors`	Vector with colors used to fill created heatmap.
`show.colnames, show.rownames`	Logical: of TRUE, names of columns (sample names) and rows (gene names) will be shown. Both default to TRUE.
`border.color`	Character: color of cell borders on heatmap. If set to NA (default) no border will be drawn.
`fontsize`	Numeric: global fontsize of heatmap. Default to 10.
`fontsize.col, fontsize.row`	Numeric: fontsize of colnames and rownames. Default to 10.
`angle.col`	Integer: angle of the column labels, one of the 0, 45, 90, 270, and 315.
`cellwidth, cellheight`	Numeric: width and height of individual cell. Both default to NA. These parameters are useful in situations where margins are too small and the plot is cropped (column names and annotation legend are sometimes partially hidden). Specification of this parameter allows to adjust size of the plot and solve this problem.
`save.to.tiff`	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
`dpi`	Integer: resolution of saved .tiff file. Default to 600.
`width`	Numeric: width (in cm) of saved .tiff file. Default to 15.
`height`	Numeric: height (in cm) of saved .tiff file. Default to 15.

`name.tiff` Character: name of saved .tiff file, without ".tiff" name of extension. Default to "heatmap_results".

Value

Heatmap with hierarchical clustering, displayed on the graphic device (if save.to.tiff = FALSE) or saved to .tiff file (if save.to.tiff = TRUE).

Examples

```
library(tidyverse)
library(pheatmap)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                       remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
# Remember to firstly create named list with colors for groups annotation:
colors.for.groups = list("Group" = c("Disease"="firebrick1","Control"="green3"))
# Vector of colors to fill the heatmap can be also specified to fit the user needings:
colors <- c("navy","navy","#313695","#4575B4","#74ADD1","#ABD9E9",
            "#E0F3F8","#FFFFBF","#FEE090","#FDAE61","#F46D43",
            "#D73027","#C32B23","#A50026","#8B0000",
            "#7E0202","#000000")
results_heatmap(data.dCt,
                sel.Gene = "all",
                col.groups = colors.for.groups,
                colors = colors,
                show.colnames = FALSE,
                show.rownames = TRUE,
                fontsize = 11,
                fontsize.row = 11)
```

`results_volcano` *results_volcano*

Description

This function creates volcano plot that illustrate the arrangement of genes based on fold change values and p values. Significant genes can be pointed out using specified p value and fold change thresholds, and highlighted on the plot by color and (optionally) isolated by thresholds lines.

Usage

```
results_volcano(
  data,
  mode,
```

```
  p.threshold = 0.05,
  FCh.threshold = 2,
  sel.Gene = "all",
  point.size = 4,
  point.shape = 19,
  alpha = 0.7,
  colors = c("#66c2a5", "#fc8d62"),
  add.thr.lines = TRUE,
  linewidth = 0.25,
  linetype = "dashed",
  x.axis.title = "log2(fold change)",
  y.axis.title = "-log10(p value)",
  axis.title.size = 11,
  axis.text.size = 10,
  legend.text.size = 11,
  legend.title = "Selected as significant?",
  legend.title.size = 11,
  legend.position = "top",
  plot.title = "",
  plot.title.size = 14,
  dpi = 600,
  width = 15,
  height = 15,
  save.to.tiff = FALSE,
  name.tiff = "results_volcano"
)
```

Arguments

data	Object returned from RQ_dCt() or RQ_ddCt() functions.
mode	Character: which p value should be used? One of the "t" (p values from Student's t test), "t.adj" (adjusted p values from Student's t test), "mw" (p values from Mann-Whitney U test),"mw.adj" (adjusted p values from Mann-Whitney U test), "depends" (if data in both compared groups were considered as derived from normal distribution (p value from Shapiro_Wilk test > 0.05) - p values from Student's t test will be used for significance assignment, otherwise p values from Mann-Whitney U test will be used), "depends.adj" (if data in both compared groups were considered as derived from normal distribution (p value from Shapiro_Wilk test > 0.05) - adjusted p values from Student's t test will be used for significance assignment, otherwise adjusted p values from Mann-Whitney U test will be used), and "user" that can be used the user intend to use another p values, e.g. obtained from other statistical test. In such situation, before run FCh_plot function, the user should prepare data frame object named "user" that contains two columns, the first of them with Gene names and the second with p values. The order of columns must be kept as described.
p.threshold	Numeric: threshold of p values for statistical significance.
FCh.threshold	Numeric: threshold of fold change values used for significance assignment of genes. If is set to 2 (default), genes with 2-fold changed expression (increased

	or decreased) between groups will be assigned as significant.
`sel.Gene`	Character vector with names of genes to include, or "all" (default) to use all names of genes.
`point.size`	Numeric: size of points. Default to 4.
`point.shape`	Integer: shape of points. Default to 19..
`alpha`	Numeric: transparency of points, a value between 0 and 1. Default to 0.7.
`colors`	Character vector length of two, containing colors for significant and no significant genes.
`add.thr.lines`	Logical, if TRUE (default), threshold lines will be added to the plot.
`linewidth`	Numeric: width of the added threshold lines. Default to 0.25.
`linetype`	Character: type of the added threshold lines. One of the "solid", "dashed" (default), "dotted", "dotdash", "longdash", "twodash", and "blank".
`x.axis.title`	Character: title of x axis. Default to "Gene".
`y.axis.title`	Character: title of y axis. Default to "value".
`axis.title.size`	Integer: font size of axis titles. Default to 11.
`axis.text.size`	Integer: font size of axis text. Default to 10.
`legend.text.size`	Integer: font size of legend text. Default to 11.
`legend.title`	Character: title of legend. Default to "Group".
`legend.title.size`	Integer: font size of legend title. Default to 11.
`legend.position`	Position of the legend, can be one of "top" (default), "right", "bottom", "left", or "none" (no legend). See description for legend.position in ggplot2::theme() function.
`plot.title`	Character: title of plot. Default to "".
`plot.title.size`	Integer: font size of plot title. Default to 14.
`dpi`	Integer: resolution of saved .tiff file. Default to 600.
`width`	Numeric: width (in cm) of saved .tiff file. Default to 15.
`height`	Numeric: height (in cm) of saved .tiff file. Default to 15.
`save.to.tiff`	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
`name.tiff`	Character: name of saved .tiff file, without ".tiff" name of extension. Default to "FCh_plot".

Value

List containing object with barplot and data frame with results. Created plot is also displayed on graphic device.

Examples

```
library(ggsignif)
library(tidyverse)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
data.dCtF <- filter_transformed_data(data.dCt, remove.Sample = c("Control11"))
results.dCt <- RQ_dCt(data.dCtF, "Disease", "Control")

RQ.volcano <- results_volcano(data = results.dCt,
                         mode = "depends",
                         p.threshold = 0.05,
                         FCh.threshold = 2)
head(RQ.volcano[[2]])
```

ROCh *ROCh*

Description

This function is designed to perform Receiver Operating Characteristic (ROC) analysis based on the gene expression data. This kind of analysis is useful to further examine performance of samples classification into two groups.

Usage

```
ROCh(
  data,
  sel.Gene = "all",
  groups,
  panels.row,
  panels.col,
  text.size = 1.1,
  print.auc = TRUE,
  print.auc.size = 0.8,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "ROC_plot",
  save.to.txt = FALSE,
  name.txt = "ROC_results"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions.
sel.Gene	Character vector with names of genes to include, or "all" (default) to use all genes.
groups	Character vector length of two with names of two compared groups.
panels.row, panels.col	Integer: number of rows and columns to arrange panels with plots.
text.size	Numeric: size of text on the plot. Default to 1.1.
print.auc	Logical: if TRUE, AUC values with confidence interval will be added to the plot. Default to TRUE.
print.auc.size	Numeric: size of AUC text on the plot. Default to 0.8.
save.to.tiff	Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.
dpi	Integer: resolution of saved .tiff file. Default to 600.
width	Numeric: width (in cm) of saved .tiff file. Default to 15.
height	Numeric: height (in cm) of saved .tiff file. Default to 15.
name.tiff	Character: name of saved .tiff file, without ".tiff" name of extension. Default to "ROC_plot".
save.to.txt	Logical: if TRUE, returned table with results will be saved to .txt file. Default to FALSE.
name.txt	Character: name of saved .txt file, without ".txt" name of extension. Default to "ROC_results".

Value

Data frame with ROC parameters including AUC, threshold, specificity, sensitivity, accuracy, positive predictive value, negative predictive value, and Youden's J statistic. Plot with ROC curves can be saved to .tiff file and opened from the working directory (will not be displayed on the graphic device).

Examples

```
library(tidyverse)
library(pROC)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
data.dCtF <- filter_transformed_data(data.dCt, remove.Sample = c("Control11"))
roc_parameters <- ROCh(data.dCtF, sel.Gene = c("Gene1","Gene16","Gene19","Gene20"),
                       groups = c("Disease","Control"),
                       panels.row = 2,
                       panels.col = 2)
```

RQ_dCt *RQ_dCt*

Description

This function performs relative quantification of gene expression using 2^-dCt method.

Usage

```
RQ_dCt(
  data,
  group.study,
  group.ref,
  do.tests = TRUE,
  pairwise = FALSE,
  alternative = "two.sided",
  p.adjust.method = "BH",
  save.to.txt = FALSE,
  name.txt = "results_dCt"
)
```

Arguments

data	Data object returned from delta_Ct() or filter_transformed_data() function.
group.study	Character: name of study group (group of interest).
group.ref	Character: name of reference group.
do.tests	Logical: if TRUE, statistical significance of differences between compared groups will be calculated using Student's t test and Mann-Whitney U test. Default to TRUE.
pairwise	Logical: if TRUE, a pairwise analysis will be performed (see details). Default to FALSE.
alternative	Character: alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
p.adjust.method	Character: p value correction method for multiple testing, one of the "holm", "hochberg", "hommel", "bonferroni", "BH" (default), "BY","fdr", or "none". See documentation for stats::p.adjust() function for details.
save.to.txt	Logical: if TRUE, returned table with results will be saved to .txt file. Default to FALSE.
name.txt	Character: name of saved .txt file, without ".txt" name of extension. Default to "results_dCt".

Details

This function calculates: 1. Means (returned in columns with the "_mean" pattern) and standard deviations (returned in columns with the "_sd" pattern) of 2^-dCt transformed dCt values for each analyzed gene across compared groups. 2. P values of normality test (Shapiro_Wilk test) performed on 2^-dCt values across compared groups (returned in columns with the "_norm_p" pattern). 3. Fold change values (returned in "FCh" column) calculated for each gene by dividing mean of 2^-dCt values in study group by mean of 2^-dCt values in reference group. 4. Statistics (returned in column with the "_test_stat" pattern) and p values (returned in column with "_test_p" pattern) of differences in 2^-dCt values between study group and reference group using Student's t test and Mann-Whitney U test. 5. P values adjusted for multiple testing using a selected method.

Value

Data frame with results (if pairwise = FALSE) or, if pairwise = TRUE, the list object with two elements: a table with the results and the second table with fold change values calculated individually for each sample.

Examples

```
library(tidyverse)
library(coin)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8", transform = TRUE)
results.dCt <- RQ_dCt(data.dCt,
                            group.study = "Disease",
                            group.ref = "Control")
head(results.dCt)
```

`RQ_ddCt` *RQ_ddCt*

Description

This function performs relative quantification of gene expression using 2^-ddCt method.

Usage

```
RQ_ddCt(
  data,
  group.study,
  group.ref,
  do.tests = TRUE,
  pairwise = FALSE,
```

```
  alternative = "two.sided",
  p.adjust.method = "BH",
  save.to.txt = FALSE,
  name.txt = "ddCt_RQ_results"
)
```

Arguments

data	Data object returned from delta_Ct() function.
group.study	Character: name of study group (group of interest).
group.ref	Character: name of reference group.
do.tests	Logical: if TRUE, statistical significance of delta delta Ct values between compared groups is calculated using Student's t test and Mann-Whitney U test. Default to TRUE.
pairwise	Logical: set to TRUE if a pairwise analysis should be done.
alternative	Character: alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
p.adjust.method	Character: p value correction method for multiple testing, one of the "holm", "hochberg", "hommel", "bonferroni", "BH" (default), "BY","fdr", or "none". See documentation for stats::p.adjust() function for details.
save.to.txt	Logical: if TRUE, returned table with results is saved to .txt file. Default to FALSE.
name.txt	Character: name of saved .txt file, without ".txt" name of extension. Default to "results_ddCt".

Details

This function performs: 1. Calculation of means (returned in columns with the "_mean" pattern) and standard deviations (returned in columns with the "_sd" pattern) of delta Ct values for analyzed genes across compared groups. 2. Normality tests (Shapiro_Wilk test) of delta Ct values of analyzed genes across compared groups and returned p values are stored in columns with the "_norm_p" pattern. 3. Calculation of differences in mean delta Ct values of genes between compared groups, obtaining delta delta Ct values (returned in"ddCt" column). 4. Calculation of fold change values (returned in "FCh" column) for each gene by exponentiation of ddCt values using 2^-ddCt formula. 5. Statistical testing of differences between study group and reference group using Student's t test and Mann-Whitney U test. Resulted statistics (in column with "_test_stat" pattern) and p values (in column with "_test_p" pattern) are returned. 6. P values adjusted for multiple testing using a selected method.

Value

Data frame with relative quantification results.

Examples

```
library(tidyverse)
library(coin)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
results.ddCt <- RQ_ddCt(data.dCt, "Disease", "Control")
head(results.ddCt)
```

---

single_pair_gene *single_pair_gene*

Description

This function generates scatter plot with linear regression line for two specified genes.

Usage

```
single_pair_gene(
  data,
  x,
  y,
  by.group = TRUE,
  point.size = 4,
  point.shape = 19,
  point.alpha = 0.7,
  colors = c("#66c2a5", "#fc8d62"),
  axis.title.size = 11,
  axis.text.size = 10,
  legend.title = "Group",
  legend.title.size = 11,
  legend.text.size = 11,
  legend.position = "right",
  plot.title = "",
  plot.title.size = 14,
  labels = TRUE,
  label = c("eq", "R2", "p"),
  label.position.x = c(1, 1),
  label.position.y = c(1, 0.95),
  small.p = FALSE,
  small.r = FALSE,
  p.digits = 3,
  rr.digits = 2,
```

```
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "_single_pair_plot"
)
```

Arguments

data	Object returned from make_Ct_ready() or delta_Ct() functions.
x, y	Characters: names of genes to use.
by.group	Logical: if TRUE (default), relationships will be shown separately for compared groups.
point.size	Numeric: size of points. Default to 4.
point.shape	Integer: shape of points. Default to 19.
point.alpha	Numeric: transparency of points, a value between 0 and 1. Default to 0.7.
colors	Character vector containing colors for compared groups. The number of colors must be equal to the number of groups. Default to c("#66c2a5", "#fc8d62").
axis.title.size	Integer: font size of axis titles. Default to 11.
axis.text.size	Integer: font size of axis text. Default to 10.
legend.title	Character: title of legend. Default to "Group".
legend.title.size	Integer: font size of legend title. Default to 11.
legend.text.size	Integer: font size of legend text. Default to 11.
legend.position	Position of the legend, can be "top", "right" (default), "bottom", "left", or "none" (no legend). See description for legend.position parameter in ggplot2::theme() function.
plot.title	Character: title of plot. Default to "".
plot.title.size	Integer: font size of plot title. Default to 14.
labels	Logical: if TRUE (default), a regression statistics will be added to the plot using ggpimsc::stat_poly_eq() function.
label	Character: specifies regression statistics to add, names specified by ggpimsc::stat_poly_eq() function must be used. Default to c("eq", "R2", "p") for regression equation, coefficient of determination and p value, respectively.
label.position.x, label.position.y	Numeric: coordinates for position of regression statistics. If by.group = TRUE, a vector length of groups number can be provided to avoid overlapping. See description of label.x and label.y parameters from ggpimsc::stat_poly_eq() function.

`small.p, small.r`
: Logical, if TRUE, p character in p value label and r character in coefficient of determination label will be lowercase. Default to FALSE.

`rr.digits, p.digits`
: Integer: number of digits after the decimal point in coefficient of determination and p value in labels. Default to 2 for rr.digits and 3 for p.digits.

`save.to.tiff`
: Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.

`dpi`
: Integer: resolution of saved .tiff file. Default to 600.

`width`
: Numeric: width (in cm) of saved .tiff file. Default to 15.

`height`
: Numeric: height (in cm) of saved .tiff file. Default to 15.

`name.tiff`
: Character: name of saved .tiff file, without ".tiff" name of extension. Default to "Gene1_Gene2_single_pair_plot".

Value

Object with plot. Created plot is also displayed on the graphic device.

Examples

```
library(tidyverse)
library(ggpmisc)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
single_pair_gene(data.dCt, "Gene16", "Gene17")
```

---

single_pair_sample *single_pair_sample*

Description

This function generates scatter plot with linear regression line for two specified samples.

Usage

```
single_pair_sample(
  data,
  x,
  y,
  pairwise.data = FALSE,
  by.group = FALSE,
  point.size = 4,
  point.shape = 19,
```

```
  point.alpha = 0.7,
  colors = c("#66c2a5", "#fc8d62"),
  axis.title.size = 11,
  axis.text.size = 10,
  legend.title = "Group",
  legend.title.size = 11,
  legend.text.size = 11,
  legend.position = "right",
  plot.title = "",
  plot.title.size = 14,
  labels = TRUE,
  label = c("eq", "R2", "p"),
  label.position.x = 1,
  label.position.y = 1,
  small.p = FALSE,
  small.r = FALSE,
  p.digits = 3,
  rr.digits = 2,
  save.to.tiff = FALSE,
  dpi = 600,
  width = 15,
  height = 15,
  name.tiff = "samples_single_pair_plot"
)
```

Arguments

`data`	Object returned from make_Ct_ready() or delta_Ct() functions.
`x, y`	Characters: names of samples to use.
`pairwise.data`	Logical: set to TRUE if a pairwise data is used, default to FALSE.
`by.group`	Logical: If TRUE (only for pairwise analysis), relationships will be shown separately for groups. Default to FALSE.
`point.size`	Numeric: size of points. Default to 4.
`point.shape`	Integer: shape of points. Default to 19.
`point.alpha`	Numeric: transparency of points, a value between 0 and 1. Default to 0.7.
`colors`	Character vector containing colors for compared groups. Use only if pairwise.data = TRUE. The number of colors must be equal to the number of groups. Default to c("#66c2a5", "#fc8d62").
`axis.title.size`	Integer: font size of axis titles. Default to 11.
`axis.text.size`	Integer: font size of axis text. Default to 10.
`legend.title`	Character: title of legend. Default to "Group".
`legend.title.size`	Integer: font size of legend title. Default to 11.
`legend.text.size`	Integer: font size of legend text. Default to 11.

`legend.position`
: Position of the legend, can be "top", "right" (default), "bottom", "left", or "none" (no legend). See description for legend.position parameter in ggplot2::theme() function.

`plot.title`
: Character: title of plot. Default to "".

`plot.title.size`
: Integer: font size of plot title. Default to 14.

`labels`
: Logical: if TRUE (default), a regression statistics will be added to the plot using ggpimsc::stat_poly_eq() function.

`label`
: Character: specifies regression statistics to add, names specified by ggpimsc::stat_poly_eq() function must be used. Default to c("eq", "R2", "p") for regression equation, coefficient of determination and p value, respectively.

`label.position.x, label.position.y`
: Numeric: coordinates for position of regression statistics. See description of label.x and label.y parameters from ggpimsc::stat_poly_eq() function.

`small.p, small.r`
: Logical, if TRUE, p character in p value label and r character in coefficient of determination label will be lowercase. Default to FALSE.

`rr.digits, p.digits`
: Integer: number of digits after the decimal point in coefficient of determination and p value in labels. Default to 2 for rr.digits and 3 for p.digits.

`save.to.tiff`
: Logical: if TRUE, plot will be saved as .tiff file. Default to FALSE.

`dpi`
: Integer: resolution of saved .tiff file. Default to 600.

`width`
: Numeric: width (in cm) of saved .tiff file. Default to 15.

`height`
: Numeric: height (in cm) of saved .tiff file. Default to 15.

`name.tiff`
: Character: name of saved .tiff file, without ".tiff" name of extension. Default to "Sample1_Sample2_single_pair_plot".

Value

Object with plot. Created plot is also displayed on the graphic device.

Examples

```
library(tidyverse)
library(ggpmisc)
data(data.Ct)
data.CtF <- filter_Ct(data.Ct,
                      remove.Gene = c("Gene2","Gene5","Gene6","Gene9","Gene11"),
                      remove.Sample = c("Control08","Control16","Control22"))
data.CtF.ready <- make_Ct_ready(data.CtF, imput.by.mean.within.groups = TRUE)
data.dCt <- delta_Ct(data.CtF.ready, ref = "Gene8")
single_pair_sample(data.dCt, "Disease6", "Control17")
```

Index