

Package: qPCRtools (via r-universe)

October 8, 2026

Version 2.0.0

Title Tools for qPCR

Description Provides tools for quantitative PCR (qPCR) data analysis, including standard curve calculation with amplification efficiency, expression level calculation using multiple methods (2-dCt, 2-ddCt, standard curve, and RqPCR), and RNA volume calculation for reverse transcription.

URL <https://github.com/lixiang117423/qPCRtools>

BugReports <https://github.com/lixiang117423/qPCRtools/issues>

License MIT + file LICENSE

Encoding UTF-8

VignetteBuilder knitr

Suggests knitr, rmarkdown

Imports broom, dplyr, ggplot2, ggpmisc, ggthemes, kableExtra, magrittr, multcomp, rstatix, stats, tibble, tidyr

RoxygenNote 7.3.3

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libpng-dev libuv1-dev libxml2-dev

Repository <https://lixiang117423.r-universe.dev>

Date/Publication 2026-06-10 05:57:44 UTC

RemoteUrl <https://github.com/lixiang117423/qpcrtools>

RemoteRef HEAD

RemoteSha eac6168b5d491d2ba758a771fe013e0f3ea42f9f

Contents

CalCurve	2
CalExp2dCt	3
CalExp2ddCt	4
CalExpCurve	5

CalExpRqPCR	7
CalRTable	8
hello	9
Index	10

CalCurve	<i>Standard Curve Calculation</i>
----------	-----------------------------------

Description

Calculate the standard curve and obtain the amplification efficiency of primer(s). Based on the amplification efficiency, we can determine which method to use for expression level calculation.

Usage

```
CalCurve(  
  cq_table,  
  concen_table,  
  highest_concen,  
  lowest_concen,  
  dilution = 4,  
  by_mean = TRUE  
)
```

Arguments

cq_table	A data frame containing position and Cq values. Must have columns: Position, Gene, Cq.
concen_table	A data frame containing position and concentration. Must have columns: Position, Conc.
highest_concen	Numeric. The highest concentration for calculation.
lowest_concen	Numeric. The lowest concentration for calculation.
dilution	Numeric. Dilution factor of cDNA template (default: 4).
by_mean	Logical. Calculate by mean Cq value or not (default: TRUE).

Value

A list containing:	
table	Data frame with standard curve parameters per gene (Formula, Slope, Intercept, R2, P.value, max.Cq, min.Cq, E, Date)
figure	ggplot object of the standard curve

Author(s)

Xiang LI <lixiang117423@gmail.com>

Examples

```
## Not run:
df.1.path <- system.file("examples", "calsc.cq.txt", package = "qPCRtools")
df.2.path <- system.file("examples", "calsc.info.txt", package = "qPCRtools")
df.1 <- read.table(df.1.path, header = TRUE)
df.2 <- read.table(df.2.path, header = TRUE)
res <- CalCurve(
  cq_table = df.1,
  concen_table = df.2,
  lowest_concen = 4,
  highest_concen = 4096,
  dilution = 4,
  by_mean = TRUE
)
res[["table"]]
res[["figure"]]

## End(Not run)
```

CalExp2dCt

*Calculate Expression Using 2-dCt Method***Description**

Calculate relative gene expression using the 2-dCt method with a reference gene for normalization.

Usage

```
CalExp2dCt(cq_table, design_table, ref_gene = "Actin")
```

Arguments

<code>cq_table</code>	A data frame containing position and Cq values. Must have columns: Position, Gene, Cq.
<code>design_table</code>	A data frame containing position and group information. Must have columns: Position, Group, BioRep.
<code>ref_gene</code>	Character. The name of the reference gene (default: "Actin").

Value

A data frame with expression values, including columns: position, cq, group, gene, biorep, mean.cq, expre, n, mean.expre, sd.expre, se.expre.

Author(s)

Xiang LI <lixiang117423@gmail.com>

Examples

```
## Not run:
df1.path <- system.file("examples", "dct.cq.txt", package = "qPCRtools")
df2.path <- system.file("examples", "dct.design.txt", package = "qPCRtools")
cq_table <- read.table(df1.path, sep = ",", header = TRUE)
design_table <- read.table(df2.path, sep = ",", header = TRUE)
res <- CalExp2dCt(cq_table, design_table, ref_gene = "Actin")
head(res)

## End(Not run)
```

CalExp2ddCt

Calculate Expression Using 2-ddCt Method

Description

Calculate relative gene expression using the 2-ddCt method with a reference gene and reference group for normalization. Supports statistical testing and outlier removal.

Usage

```
CalExp2ddCt(
  cq_table,
  design_table,
  ref_gene = "OsUBQ",
  ref_group = "CK",
  stat_method = "t.test",
  remove_outliers = TRUE,
  fig_type = "box",
  fig_ncol = NULL
)
```

Arguments

<code>cq_table</code>	A data frame containing position and Cq values. Must have columns: Position, Gene, Cq.
<code>design_table</code>	A data frame containing position and group information. Must have columns: Position, Group, BioRep.
<code>ref_gene</code>	Character. The name of the reference gene (default: "OsUBQ").
<code>ref_group</code>	Character. The name of the reference/control group (default: "CK").
<code>stat_method</code>	Character. Statistical method for group comparison. One of "t.test", "wilcox.test", or "anova" (default: "t.test").
<code>remove_outliers</code>	Logical. Remove outliers using IQR method (default: TRUE).
<code>fig_type</code>	Character. Plot type: "box" for boxplot, "bar" for barplot (default: "box").
<code>fig_ncol</code>	Integer. Number of columns in facet plot (default: NULL).

Value

A list containing:

table	Data frame with expression values and statistics
figure	ggplot object

Author(s)

Xiang LI <lixiang117423@gmail.com>

Examples

```
## Not run:
df1.path <- system.file("examples", "ddct.cq.txt", package = "qPCRtools")
df2.path <- system.file("examples", "ddct.design.txt", package = "qPCRtools")
cq_table <- read.table(df1.path, header = TRUE)
design_table <- read.table(df2.path, header = TRUE)
res <- CalExp2ddCt(
  cq_table,
  design_table,
  ref_gene = "OsUBQ",
  ref_group = "CK",
  stat_method = "t.test",
  remove_outliers = TRUE,
  fig_type = "box",
  fig_ncol = NULL
)
res[["table"]]
res[["figure"]]

## End(Not run)
```

CalExpCurve

Calculate Expression Using Standard Curve

Description

Calculate relative gene expression using a standard curve method with optional reference gene correction and statistical testing.

Usage

```
CalExpCurve(
  cq_table,
  curve_table,
  design_table,
  correction = TRUE,
```

```

    ref_gene = "OsUBQ",
    stat_method = "t.test",
    ref_group = "CK",
    fig_type = "box",
    fig_ncol = NULL
  )

```

Arguments

cq_table	A data frame containing position and Cq values. Must have columns: Position, Gene, Cq.
curve_table	A data frame with standard curve parameters per gene. Must have columns: Gene, Slope, Intercept, max.Cq, min.Cq.
design_table	A data frame containing position and group information. Must have columns: Position, Treatment, Gene.
correction	Logical. Correct expression by reference gene (default: TRUE).
ref_gene	Character. The name of the reference gene (default: "OsUBQ").
stat_method	Character. Statistical method for group comparison. One of "t.test", "wilcox.test", or "anova" (default: "t.test").
ref_group	Character. The name of the reference/control group (default: "CK").
fig_type	Character. Plot type: "box" for boxplot, "bar" for barplot (default: "box").
fig_ncol	Integer. Number of columns in facet plot (default: NULL).

Value

A list containing:

table	Data frame with expression values and statistics
figure	ggplot object

Author(s)

Xiang LI <lixiang117423@gmail.com>

Examples

```

## Not run:
df1.path <- system.file("examples", "cal.exp.curve.cq.txt", package = "qPCRtools")
df2.path <- system.file("examples", "cal.expre.curve.sdc.txt", package = "qPCRtools")
df3.path <- system.file("examples", "cal.exp.curve.design.txt", package = "qPCRtools")
cq_table <- read.table(df1.path, header = TRUE)
curve_table <- read.table(df2.path, sep = "\t", header = TRUE)
design_table <- read.table(df3.path, header = TRUE)
res <- CalExpCurve(
  cq_table, curve_table, design_table,
  correction = TRUE,
  ref_gene = "OsUBQ",
  stat_method = "t.test",

```

```

    ref_group = "CK",
    fig_type = "box",
    fig_ncol = NULL
)
res[["table"]]
res[["figure"]]

## End(Not run)

```

CalExpRqPCR

Calculate Expression Using RqPCR Method

Description

Calculate relative gene expression using the RqPCR method with amplification efficiency correction. Can auto-select reference genes using the GeNorm algorithm when ref_gene is NULL.

Usage

```

CalExpRqPCR(
  cq_table,
  design_table,
  ref_gene = NULL,
  ref_group = "CK",
  stat_method = "t.test",
  fig_type = "box",
  fig_ncol = NULL
)

```

Arguments

cq_table	A data frame containing position and Cq values. Must have columns: Position, Gene, Cq, BioRep, TechRep, Eff.
design_table	A data frame containing position and group information. Must have columns: Position, Group, BioRep, TechRep, Eff.
ref_gene	Character. The name(s) of reference gene(s). If NULL, reference genes are auto-selected via GeNorm (default: NULL).
ref_group	Character. The name of the reference/control group (default: "CK").
stat_method	Character. Statistical method for group comparison. One of "t.test", "wilcox.test", or "anova" (default: "t.test").
fig_type	Character. Plot type: "box" for boxplot, "bar" for barplot (default: "box").
fig_ncol	Integer. Number of columns in facet plot (default: NULL).

Value

A list containing:

table	Data frame with expression values and statistics
figure	ggplot object

Author(s)

Xiang LI <lixiang117423@gmail.com>

Examples

```
## Not run:
df1.path <- system.file("examples", "cal.expre.rqpcr.cq.txt", package = "qPCRtools")
df2.path <- system.file("examples", "cal.expre.rqpcr.design.txt", package = "qPCRtools")
cq_table <- read.table(df1.path, header = TRUE)
design_table <- read.table(df2.path, header = TRUE)
res <- CalExpRqPCR(
  cq_table, design_table,
  ref_gene = NULL,
  ref_group = "CK",
  stat_method = "t.test",
  fig_type = "box",
  fig_ncol = NULL
)
res[["table"]]
res[["figure"]]

## End(Not run)
```

CalRTable

Calculate RNA Volume for Reverse Transcription

Description

The first step of qPCR is usually the preparation of cDNA. This function calculates the volume of RNA needed for reverse transcription based on RNA concentration.

Usage

```
CalRTable(data, template, rna_weight = 1)
```

Arguments

data	A data frame containing sample names and concentration values (default unit: ng/uL). Must have columns: sample, concentration.
template	A data frame containing reverse transcription information. Must have a column called 'all'.
rna_weight	Numeric. RNA weight required for reverse transcription in micrograms (default: 1).

Value

A data frame with calculated RNA and water volumes for each sample.

Author(s)

Xiang LI <lixiang117423@gmail.com>

Examples

```
## Not run:
df.1.path <- system.file("examples", "crtv.data.txt", package = "qPCRtools")
df.2.path <- system.file("examples", "crtv.template.txt", package = "qPCRtools")
df.1 <- read.table(df.1.path, sep = "\t", header = TRUE)
df.2 <- read.table(df.2.path, sep = "\t", header = TRUE)
result <- CalRTTable(data = df.1, template = df.2, rna_weight = 2)
head(result)

## End(Not run)
```

hello	<i>Hello, World!</i>
-------	----------------------

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

Index

CalCurve, [2](#)
CalExp2dCt, [3](#)
CalExp2ddCt, [4](#)
CalExpCurve, [5](#)
CalExpRqPCR, [7](#)
CalRTable, [8](#)

hello, [9](#)