



# rtpcr: a package for statistical analysis and graphical presentation of qPCR data in R

Ghader Mirzaghaderi

Plant Production and Genetics, University of Kurdistan, Sanandaj, Iran

## ABSTRACT

**Background.** Quantitative real-time polymerase chain reaction (qRT-PCR or qPCR) is widely used in molecular biology research. Various analysis methods are employed to interpret qPCR data and measure mRNA levels of target genes under different experimental conditions.

**Results.** The rtpcr package was developed for amplification efficiency calculation, statistical analysis, and graphical presentation of qPCR data in R. It uses a general calculation methodology that accommodates up to two reference genes and amplification efficiency values, including the Pfaffl method. Depending on the experimental design, rtpcr functions apply a *t*-test (for experiments with a two-level factor), analysis of variance (ANOVA), or analysis of covariance (ANCOVA) (for experiments with more than two levels or factors) to calculate fold change (FC) or relative expression (RE) of a target gene. The functions also provide standard errors and confidence intervals for the means and support statistical mean comparisons. To facilitate usage, the package includes example datasets. It also offers ggplot-based visualizations with customizable arguments, allowing users to tailor the graphical output.

**Conclusions.** In summary, the rtpcr package is a useful and user-friendly tool for analyzing real-time PCR data from experiments involving up to three different factors. Built on a general methodology, it provides robust calculations and comprehensive graphical outputs, making it a valuable resource for researchers working with qPCR data.

**Subjects** Bioinformatics, Genetics, Molecular Biology, Computational Science

**Keywords** Quantitative PCR, Fold change, Relative expression, Data analysis, Bar plot

## BACKGROUND

Quantitative real-time polymerase chain reaction (qRT-PCR also known as qPCR), is a powerful analytical tool that is able to quantify nucleic acids using reference sequences. The technique's sensitivity, specificity, and broad quantification range make it the gold standard for the detection and quantification of DNA and RNA sequences. Nowadays, relative quantification using qPCR is primarily used in the field of molecular genetics, genomics and functional transcriptomics to perform gene expression analysis in biological experiments (*Harshitha & Arunraj, 2021*).

Submitted 19 February 2025

Accepted 15 September 2025

Published 13 October 2025

Corresponding author

Ghader Mirzaghaderi,  
mirzaghaderi@gmail.com

Academic editor

Giovanni Angiulli

Additional Information and  
Declarations can be found on  
page 10

DOI 10.7717/peerj.20185

© Copyright  
2025 Mirzaghaderi

Distributed under  
Creative Commons CC-BY 4.0

## OPEN ACCESS

qPCR utilizes the increased fluorescence from a reporter molecule as the template is exponentially amplified. SYBR Green is a commonly used chemical that binds to double-stranded DNA during the primer extension step of the polymerase reaction (Boulter *et al.*, 2016). Specialized oligonucleotide probes like TaqMan hybridization probes are also used as fluorescent reporters for targeted RT-qPCR assays (Navarro *et al.*, 2015).

Reliability of the qPCR results depends on the application of robust mathematical methodologies that ensure accurate data analysis and outputs. Various mathematical approaches have been developed for qPCR data analysis (Pabinger *et al.*, 2014; Ritz & Spiess, 2008). The basic method for expression estimation of a gene between conditions relies on the calculation of fold changes (FC) using the PCR amplification efficiency (E) and the threshold cycle (syn. crossing point or  $C_T$ ). Two basic mathematical methods are commonly used based on FC namely the Livak method and the Pfaffl method. The Livak approach (Livak & Schmittgen, 2001), also known as the  $2^{-\Delta\Delta CT}$  method, stands out for its simplicity and widespread use where the fold change (FC) expression of a target gene ( $2^{-\Delta\Delta CT}$ ) in Treatment condition (Tr) compared to Control condition (Co) is calculated according to the following Eq. (1):

$$\begin{aligned} FC &= \frac{2^{-(C_{T_{\text{target}}} - C_{T_{\text{ref}}})_{Tr}}}{2^{-(C_{T_{\text{target}}} - C_{T_{\text{ref}}})_{Co}}} \\ &= 2^{-[(C_{T_{\text{target}}} - C_{T_{\text{ref}}})_{Tr} - (C_{T_{\text{target}}} - C_{T_{\text{ref}}})_{Co}]} \\ &= 2^{-(\Delta CT_{Tr} - \Delta CT_{Co})}. \end{aligned} \quad (1)$$

Here,  $\Delta C_T$  is the difference between two  $C_T$  values (e.g.,  $C_{T_{\text{target}}} - C_{T_{\text{ref}}}$ ) in treatment or control condition, and target and ref are target gene and reference genes, respectively. This method assumes that both the target and reference genes are amplified with efficiencies close to 100%, allowing for the relative quantification of gene expression levels (Livak & Schmittgen, 2001).

On the other hand, the Pfaffl method (Pfaffl, Horgan & Dempfle, 2002) offers a more flexible approach by accounting for differences in amplification efficiencies between the target and the reference genes. This method adjusts the calculated expression ratio by incorporating the specific amplification efficiencies, thus provides a more accurate representation of the relative gene expression levels (Pfaffl, Horgan & Dempfle, 2002). The Pfaffl formula can be written as Eq. (2):

$$FC = \frac{E^{-(C_{T_{Tr}} - C_{T_{Co}})_{\text{target}}}}{E^{-(C_{T_{Tr}} - C_{T_{Co}})_{\text{ref}}}}. \quad (2)$$

Many statistical tools and analysis codes have been developed for the statistical analysis of qPCR data in a stand alone format (Yuan *et al.*, 2006), for the R platform (Ahmed & Kim, 2018; Li *et al.*, 2022; Ritz & Spiess, 2008) or SAS software (Yuan *et al.*, 2006). The rtpcr package is a comprehensive tool designed for the analysis of qRT-PCR data in R, providing a set of functions that allow researchers to perform various analyses on their qRT-PCR data using the cycle threshold ( $C_T$ ) and efficiency values (E) under different experimental conditions.

## IMPLEMENTATION

The rtpcr package was developed for the R environment (<http://www.r-project.org>) in the major operating systems and its source code, binary format, and under-development version are freely available at CRAN (<https://cran.r-project.org/web/packages/rtpcr/index.html>) and Github (<https://github.com/mirzaghaderi/rtpcr>). The package functions are mainly based on the calculation of efficiency-weighted  $\Delta C_T$  ( $w\Delta C_T$ ) values from target and reference gene  $C_T$  (Eq. (3)).  $w\Delta C_T$  values are weighted for the amplification efficiencies as below. Portions of this text were previously published as part of a preprint (Mirzaghaderi, 2024):

$$w\Delta C_T = \log_2(E_{target}) \cdot C_{T_{target}} - \log_2(E_{ref}) \cdot C_{T_{ref}}. \quad (3)$$

From the arithmetic mean  $w\Delta C_T$  values over biological replicates, relative expression (RE,  $\Delta C_T$  method) of a target gene can be calculated for each condition according to Eq. (4):

$$RE = 2^{-\overline{w\Delta C_T}}. \quad (4)$$

The rtpcr package considers efficiency values, thus the results match the Pfaffl method. If all input efficiency values are 2,  $2^{-\Delta\Delta C_T}$ -based results are returned. The average fold change (FC,  $\Delta\Delta C_T$  method) expression statistics and graphs are returned for the target gene based on Eq. (5):

$$FC = 2^{-(\overline{w\Delta C_T}_{Tr} - \overline{w\Delta C_T}_{Co})}. \quad (5)$$

Because both the relative expression and fold change expression follow a lognormal distribution (Derveaux, Vandesompele & Hellemans, 2010; McDavid et al., 2012), a normal distribution is expected for the  $w\Delta C_T$  or  $w\Delta\Delta C_T$  values making it possible to apply t-tests or analysis of variance to them.  $w\Delta C_T$  values can be statistically compared and standard deviations and confidence intervals are calculated, but the transformation  $y = 2^{-x}$  is applied in the final step to report the results. Here, a brief methodology is presented but details about the mathematical calculations and statistical analyses are available in Ganger, Dietz & Ewing (2017) and Ganger et al. (2020). In the rtpcr package, model creation and analysis of variance is done using the lmer (Bates et al., 2015) function of the lmerTest (Kuznetsova, Brockhoff & Christensen, 2017) package. lmerTest::lmer fits a linear mixed model and provides  $p$ -values for fixed effects in the analysis of variance (ANOVA) and summary output. For this, the biological replicate is served as a random effect. Means are compared using the emmeans function (Searle, Speed & Milliken, 1980) from the model. The standard error (standard deviation/sqrt(n)) for average FC or RE is calculated in different ways in literature. This statistic is calculated based on Taylor et al. (2019) by the rtpcr package according to Eq. (6) for average FC. For the standard error of the RE means, RE is used instead of FC.

$$Lower.se = 2^{\log_2(FC) - se_{w\Delta C_T}} \quad (6)$$

$$Upper.se = 2^{\log_2(FC) + se_{w\Delta C_T}}.$$

## RESULTS

The rtpcr package was developed for amplification efficiency calculation, statistical analysis, and graphical representation of qPCR data in R. It accepts up to two reference genes and amplification efficiency values. Based on the experimental conditions, the functions of the rtpcr package use a  $t$ -test, analysis of variance, or covariance (for cases with more than two factors) to calculate the fold change (FC) or relative expression (RE). The functions provide standard errors and confidence intervals for FC or RE means and apply statistical mean comparisons. Different sample data sets were added to the rtpcr package and included in the examples to facilitate the usage of the package functions. The rtpcr package further provides editable ggplots with various editing arguments. Some functionalities of the rtpcr package are presented in Fig. 1 and a more detailed representation of the rtpcr application is shown in Fig. S1.

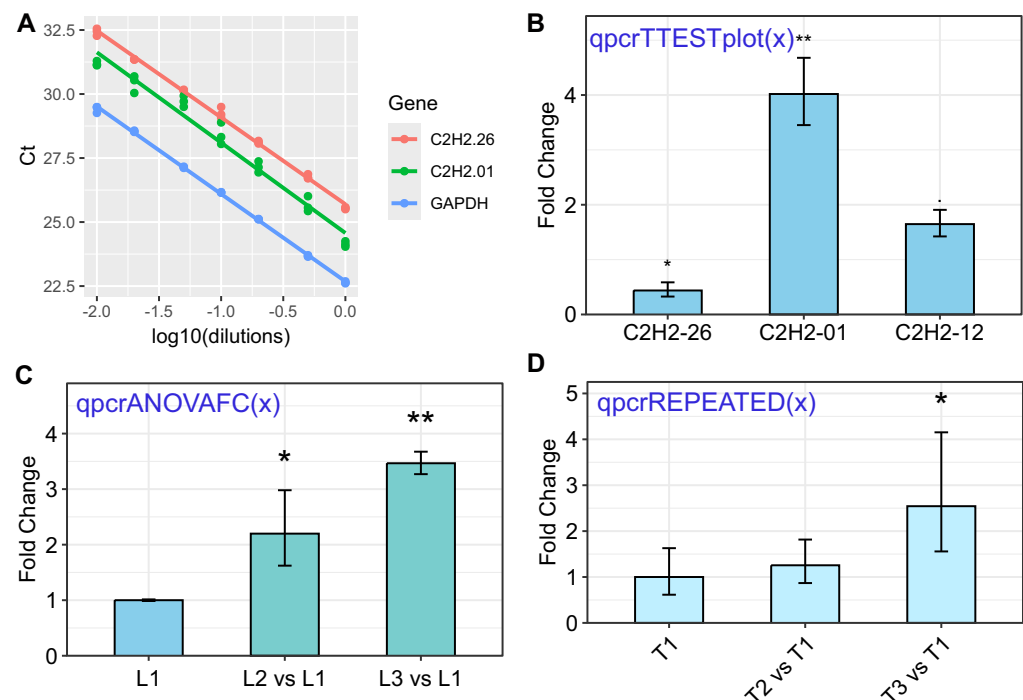
### Input data frame structure

The input data set should be prepared as shown in example data sets of the rtpcr package. The column structure and arrangement of the data frames should follow the format indicated in Tables 1 and 2, as shown in the package help and the vignettes (<https://cran.r-project.org/web/packages/rtpcr/vignettes/vignette.html>). To see example data sets included in the rtpcr package, you can run the following command in R: `data(package = "rtpcr")`. This will display a list of the available example data sets within the rtpcr package. You can then load a specific data set by running its name. Except for the  $t$ -test analysis which requires a specific data structure, factor columns should appear first in the data frame followed by blocking factor (if available), biological replicates, target gene efficiencies,  $C_T$  values of the target gene, the reference gene efficiencies, and the  $C_T$  values of the reference gene, respectively. The data frame can contain one to three factor columns, depending on the experimental design.

This recommended column structure ensures compatibility with the various analysis functions provided by the rtpcr package, such as the analysis of variance and analysis of covariance methods, which require the data to be organized in this specific way. For the  $t$ -test analysis, the data structure may differ slightly, as it needs to be in a format that allows for the comparison of two experimental conditions. The package documentation and examples will provide guidance on the appropriate data structure for this specific analysis.

### Amplification efficiency

The rtpcr package in R provides a comprehensive set of functions for the analysis of real-time PCR (qPCR) data. A brief explanation of the key functions is presented in Fig. S1. An important function in the rtpcr package is the 'efficiency' function. This function is used to calculate the amplification efficiency of genes based on the provided data. The input data frame for the 'efficiency' function should have a specific column structure, with the first column containing the dilution information, followed by the  $C_T$  value columns for each target gene. The 'efficiency' function takes this input data and calculates the amplification efficiency of the target genes. It presents the related standard curves, along with the slope, efficiency, and  $R^2$  statistics (Fig. 1A). Additionally, the function performs



**Figure 1** Some of the `rtPCR` package functionalities for the analysis of the qPCR data. The used functions are presented in blue. (A) Standard curve and the amplification efficiency analysis of three genes using the `'efficiency(x)'` function. (B) Average fold changes of three target genes relative to the control condition computed by unpaired *t*-tests via the `'qpcrTTESTplot'` function. (C) Plot of average Fold changes produced by the `'qpcrANOVAFC'` function where the level 1 condition (L1) has been selected as calibrator. (D) Fold change expression graph of a target gene produced by `'qpcrREPEATED'` function. Error bars can be set as standard deviation or confidence interval.

Full-size [DOI: 10.7717/peerj.20185/fig-1](https://doi.org/10.7717/peerj.20185/fig-1)

statistical pairwise comparisons of the slopes to determine if the amplification efficiencies of genes are significantly different.

### Fold Change (FC) analysis: *t*-test

The expression of a target gene under two different conditions, such as control and treatment, can be presented as the average fold change of the target gene in the treatment condition relative to the control condition. This analysis is performed by the `'qpcrTTEST'` and `'qpcrTTESTplot'` functions, which apply a *t*-test to any number of genes that have been evaluated under control and treatment conditions. The analysis can be done for both unpaired or paired samples. A *p*.adj argument has been added to `qpcrTTEST` and `qpcrTTESTplot` functions to adjust the *p*-values, in which the default adjustment method is set to "BH". An example of the resulting output of the `'qpcrTTEST'` function is shown in Table 3, which contains the target gene names, fold changes, confidence limits, *p*-values, standard errors (se), and lower and upper standard errors. The `'qpcrTTEST'` function includes the `'var.equal'` argument, which, if set to 'FALSE', performs the *t*-test under the unequal variances hypothesis. The `'qpcrTTESTplot'` function, which performs the *t*-test and displays the results in a bar plot, automatically adds appropriate significance

**Table 1** Column arrangement of the input data frame for use in the `rtpcr` package. To see example data sets, in the `rtpcr` package run `data` (package = "rtpcr"). Example data sets can be presented by running the name of each data set. `targetE` and `refE`: amplification efficiency columns for target and reference genes, respectively. `targetCt` and `refCt`: Ct columns for target and reference genes, respectively.

Analysis type	Column arrangement of the input data set	Name of the example data sets
Amplification efficiency	Dilutions - <code>targetCt</code> - <code>refCt</code>	<code>data_efficiency</code>
<i>t</i> -test (accepts multiple genes)	condition (control level first) - gene (ref gene(s) last) - efficiency - Ct	<code>data_ttest</code>
ANOVA or ANCOVA (Up to three factors)	<code>factor1</code> - rep - <code>targetE</code> - <code>targetCt</code> - <code>refE</code> - <code>refCt</code>	<code>data_1factor</code>
	<code>factor1</code> - <code>factor2</code> - rep - <code>targetE</code> - <code>targetCt</code> - <code>refE</code> - <code>refCt</code>	<code>data_2factor</code>
	<code>factor1</code> - <code>factor2</code> - <code>factor3</code> - rep - <code>targetE</code> - <code>targetCt</code> - <code>refE</code> - <code>refCt</code>	<code>data_3factor</code>
ANOVA or ANCOVA with blocking	<code>factor1</code> - block - rep - <code>targetE</code> - <code>targetCt</code> - <code>refE</code> - <code>refCt</code>	<code>data_2factorBlock</code>
	<code>factor1</code> - <code>factor2</code> - block - rep - <code>targetE</code> - <code>targetCt</code> - <code>refE</code> - <code>refCt</code>	
	<code>factor1</code> - <code>factor2</code> - <code>factor3</code> - block - rep - <code>targetE</code> - <code>targetCt</code> - <code>refE</code> - <code>refCt</code>	
With two reference genes	... rep - <code>targetE</code> - <code>targetCt</code> - <code>ref1E</code> - <code>ref1Ct</code> - <code>ref2E</code> - <code>ref2Ct</code>	
Calculating biological replicated	... biologicalRep - techicalRep - <code>Etarget</code> - <code>targetCt</code> - <code>Eref</code> - <code>refCt</code>	<code>data_withTechRep</code>
	... biolRep - techRep - <code>Etarget</code> - <code>targetCt</code> - <code>ref1E</code> - <code>ref1Ct</code> - <code>ref2E</code> - <code>ref2Ct</code>	

**Notes.**

For ANOVA and ANCOVA analysis, each line in the input data set belongs to a separate individual (reflecting a non-repeated measure experiment).

**Table 2** Repeated measure data structure and column arrangement required for the 'qpcrREPEATED' function. `targetE` and `refE`: amplification efficiency columns for target and reference genes, respectively. `targetCt` and `refCt`: Ct columns for target and reference genes, respectively. In the "id" column, a unique number is assigned to each individual, e.g. all the three number 1 indicate a single individual.

Column arrangement of the input data set	Name of the example data sets
id - time - <code>targetE</code> - <code>targetCt</code> - <code>ref1E</code> - <code>ref1Ct</code>	<code>data_repeated_measure_1</code>
id - time - <code>targetE</code> - <code>targetCt</code> - <code>ref1E</code> - <code>ref1Ct</code> - <code>ref2E</code> - <code>ref2Ct</code>	
id - treatment - time - <code>targetE</code> - <code>targetCt</code> - <code>ref1E</code> - <code>ref1Ct</code>	<code>data_repeated_measure_2</code>
id - treatment - time - <code>targetE</code> - <code>targetCt</code> - <code>ref1E</code> - <code>ref1Ct</code> - <code>ref2E</code> - <code>ref2Ct</code>	

**Notes.**

To see example data sets, in the `rtpcr` package run `data` (package = "rtpcr"). Example data sets can be presented by running the name of each data set.

signs ('\*\*', '\*', or '.') on top of the bars based on the *t*-test *p*-values (Fig. 1B). Similarly, in single- or multi-factorial experiments, fold change (FC) analysis can be performed for each of the factors using the 'qpcrANOVAFC' function. One of the factor levels can be selected as the reference level, allowing for the comparison of gene expression across different experimental conditions. These functions in the `rtpcr` package provide a comprehensive set of tools for researchers to analyze and interpret gene expression

**Table 3 The fold change (FC) analysis.** An example output table of fold change (FC) analysis of three genes evaluated under control and treatment conditions. The output was produced by the 'qpcrTTEST' function of the rtpcr package.

Gene	FC	LCL	UCL	p value	se	Lower.se	Upper.se
C2H2-26	0.4373	0.1926	0.9927	0.0488	0.4218	0.3264	0.5858
C2H2-01	4.0185	2.4598	6.5649	0.0014	0.2193	3.4518	4.6782
C2H2-12	1.6472	0.9595	2.8279	0.0624	0.2113	1.4228	1.907

**Notes.**

FC, The average fold change of the target gene in the treatment condition relative to the control condition; LCL, lower confidence limit of the FC; UCL, upper confidence limit; se, standard error of the fold change; Lower.se, lower standard error; Upper.se, upper standard error.

data from qPCR experiments, enabling them to draw meaningful conclusions about the differential expression of target genes under various experimental conditions.

### Fold change (FC) analysis: analysis of variance

If there is one factor with more than two levels or more than one factor in the experiment, the qpcrANOVAFC function can be used for the fold change analysis of the levels of each factor. A partial output of this function has been presented in Table 4. The qpcrANOVAFC function applies both ANOVA and analysis of covariance (ANCOVA) analysis to the data of a single-factor or a multi-factorial experiment. If there are multiple factors, the fold change (FC) value calculations for the mainFactor.column and the statistical analysis are performed based on a full model factorial experiment by default. However, if ANCOVA is defined for the analysisType argument, the FC values are calculated for the levels of the mainFactor.column, and the other factors are used as covariate(s) in the analysis. It is important to consider the output analysis of variance table, as if an interaction between the main factor and another factor or covariate is significant, statistical comparisons of FC values may not be appropriate between the levels of a factor alone. ANCOVA is used when a factor is affected by uncontrolled quantitative covariate(s). For example, suppose that the  $\Delta C_T$  of a target gene in a plant is affected by temperature. The gene may also be affected by drought. Since we already know that temperature affects the target gene, we are interested in understanding if the gene expression is also altered by the drought levels. We can design an experiment to study the gene behavior at both temperature and drought levels simultaneously. The drought is another factor (the covariate) that may affect the expression of our gene under the levels of the first factor, *i.e.*, temperature. The data of such an experiment can be analyzed using ANCOVA or ANOVA based on a factorial experiment using the qpcrANOVAFC function. The qpcrANOVAFC function performs FC analysis even if there is only one factor (without a covariate or factor variable). It also returns a bar plot of the FC values along with the standard errors.

The qpcrMeans function also performs FC analysis using a model produced by the qpcrANOVAFC or qpcrREPEATED functions. As an advantage, qpcrMeans function returns all the pairwise statistical comparisons for any effects in the model, including simple effects, interactions, and slicing from the ANOVA models. However, ANCOVA models returned by the rtpcr package only include simple effects.



**Table 4 The output table of fold change analysis.** An example output table of fold change analysis of one gene evaluated under three different levels of a factor. The output was produced by the ‘qpcrANOVAFC’ function of the rtpcr package applied on the ‘data\_1factor’ sample data. ‘L1’ has been selected as the check or reference level.

Contrast	FC	p value	sig	LCL	UCL	se	Lower.se	Upper.se
L1	1.0000	1.0000		0.0000	0.0000	0.0208	0.9857	1.0145
L2 vs L1	2.1987	0.0285	*	0.9514	5.0812	0.4388	1.6221	2.9803
L3 vs L1	3.4661	0.0061	**	1.4999	8.0101	0.0841	3.2698	3.6742

**Notes.**

FC, The average fold change of the target gene in the treatment condition compared to the control condition; sig, \* and \*\* shows significant FC expression at 0.05 and 0.01 significance levels, respectively; LCL, lower confidence limit of the FC; UCL, upper confidence limit; se, standard error of the FC; Lower.se, lower standard error; Upper.se, upper standard error.

**Table 5 The output table of relative expression.** An example output table of relative expression of a target gene evaluated under a two factorial experiment. The output produced by the ‘qpcrANOVARE’ function of the rtpcr package.

Factor1	Factor2	RE	LCL	UCL	se	Lower.se	Upper.se	Letters
S	0.5	2.9545	2.047	4.2644	0.0551	2.8438	3.0695	a
R	0.5	0.9837	0.6815	1.4198	0.0841	0.928	1.0427	b
S	0	0.7916	0.5485	1.1426	0.2128	0.683	0.9174	b
R	0.25	0.624	0.4323	0.9006	0.4388	0.4604	0.8458	bc
S	0.25	0.4126	0.2859	0.5956	0.254	0.346	0.492	cd

**Notes.**

RE, Relative expression of the target gene relative to the reference gene in each specific condition; LCL, lower confidence limit; UCL, upper confidence limit; se, standard error; Lower.se, lower standard error; Upper.se, upper standard error; Letters, Means that does not share a letter in common, have significant difference.

## Relative expression (RE) analysis

The ‘qpcrANOVARE’ function can perform ANOVA analysis of the relative gene expression ( $\Delta C_t$  method) for one- to three-factor experiments. The package generates relative statistics and ‘ggplot2’-derived graphs (Wilkinson, 2016). If available, the blocking factor can also be handled. The output of the ‘qpcrANOVARE’ function includes a table with relative expression values, grouping letters, standard deviations, and post-hoc testing of means along with the significance and confidence interval (Table 5). The standard deviation for each mean is derived from the back-transformed raw wDCt values of biological replicates.

An outstanding feature of the rtpcr package is providing publication-ready bar plots with various controlling arguments for a lot of graphical aspects of plots. A sample of the output plots are presented in Fig. 1 and Fig. S1. The RE table from the ‘qpcrANOVARE’ function can be used by plot functions to generate a bar plot. ‘qpcrTTESTplot’ and ‘qpcrANOVAFC’ functions also generate fold change plots directly from the raw data. The bar plots produced by the rtpcr package can further be edited by the ggplot2 functions.

## Checking residuals

If the residuals from a  $t$ -test or a linear model (lm) object do not show homogeneity of variances or are not normally distributed, the assumptions for the  $t$ -test and ANOVA analyses may be violated. In such cases, the statistical results might not be reliable. The qpcrTTEST and qpcrTTESTplot functions in the rtpcr package include the var.equal



argument. When set to FALSE, these functions perform the  $t$ -test under the unequal variance hypothesis, by which can address the issue of heterogeneous variances. However, in other cases, such as ANOVA analysis, it may be more appropriate to apply non-parametric tests instead of the standard ANOVA. Currently, non-parametric tests have not been built in the rtpcr package but some options include the Mann–Whitney test (a non-parametric alternative to the  $t$ -test that can be used to test the difference between the medians of two populations using independent samples) and the Kruskal-Wallis test (a non-parametric alternative to one-way ANOVA can be used to test the difference between the medians of multiple populations). These non-parametric tests do not rely on the assumptions of normality and homogeneity of variances, making them more robust when these assumptions are violated.

### Mean of the technical replicates

Calculating the mean of technical replicates and getting an output table appropriate for subsequent analysis in rtpcr can be done using the ‘meanTech’ function. For this, the input data set should follow the column arrangement of the ‘data\_withTechRep’ example data set. The grouping columns needs to be specified using the ‘groups’ argument of the ‘meanTech’ function.

## DISCUSSION

rtpcr is an open-source R package that covers a lot of aspects of qPCR data analysis including efficiency and fold change analysis, statistical comparisons and graph production from single and multi-factorial experiments with or without a blocking factor. It accepts one or two reference genes, performs post-hoc testing, and provides standard error and confidence limits.

Different R packages have been developed for the analysis of qPCR data with different capabilities (Ahmed & Kim, 2018; Pabinger et al., 2014; Yuan et al., 2006). For example, chipPCR (Rödiger, Burdukiewicz & Schierack, 2015) can handle high throughput qPCR data and qpcR performs sigmoidal model selection for the analysis of the real-time PCR data (Ritz & Spiess, 2008). Some packages such as chipPCR, qpcR and FPK-PCR (Lievens et al., 2012) calculate the  $C_T$  values from raw florescence data and some such as ddCT (Zhang, Ruschhaupt & Biczok, 2013), dpcR, EasyqpcR, HTqPCR (Dvinge & Bertone, 2009), NormqPCR (Perkins et al., 2012), qpcrNorm, pcr (Ahmed & Kim, 2018) and qPCRtools (Li et al., 2022) require  $C_T$  values for the analysis. These packages also differ in type of quantification and analysis, amplification efficiency handling, *post-hoc* comparison, error calculation, and graph presentation (Table 6). The main advantages of the rtpcr package over the other R packages are performing expression analysis based on both  $\Delta C_T$  and  $\Delta\Delta C_T$  methods for different experimental conditions and accounting for given amplification efficiencies matching the Pfaffl method. Using the rtpcr package, it is also possible to account for efficiency value for each primer or each reaction which has been recommended to get a more precise quantification result (Ruijter et al., 2021; Ruiz-Villalba, Ruijter & van den Hoff, 2021). The provided sample data frames for all the analysis types and a documentation enables all R users with minimum knowledge of data handling to

**Table 6** **rtpcr package functionality.** Comparing functionalities of the rtpcr package with available R packages for qPCR data analysis.

Package name	Efficiency calculation	efficiency values <sup>a</sup>	Fold change expression	Error calc.	Norm.	NA handling	Graphs	Stats.	Experimental designs <sup>b</sup>	MIQE <sup>c</sup>
chipPCR	+					+	+			+
ddCT		+	+		+		+	+		+
dpcR						+	+	+		+
EasyqpcR	+	+	+		+	+				+
HTqPCR			+		+	+	+	+	+	
NormqPCR		+	+		+	+				+
qpcR	+	+	+	+	+	+	+			+
qpcrNorm					+		+	+		
pcr	+			+	+		+	+		+
rtpcr	+	+	+	+	+		+	+	+	+

**Notes.**

<sup>a</sup>Can include PCR amplification efficiency values for each replicate in fold change or relative expression analysis.

<sup>b</sup>Data from multi-factorial and repeated measure designs can be handled.

<sup>c</sup>Follows MIQE (Minimum Information for Publication of Quantitative Real-Time PCR Experiments) approvals for reporting RT-qPCR results.

simply use the rtpcr package. In conclusion, the rtpcr was built based on a general method with both calculational and graphical output capabilities for analyzing qPCR  $C_T$  values from the experiments with up to three different factors.

## ADDITIONAL INFORMATION AND DECLARATIONS

### Funding

The author received no funding for this work.

### Competing Interests

The author declares that he has no competing interests.

### Author Contributions

- Ghader Mirzaghaderi conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.

### Data Availability

The following information was supplied regarding data availability:

The rtpcr package source is available at CRAN:

- <https://doi.org/10.32614/CRAN.package.rtpcr>.

### Supplemental Information

Supplemental information for this article can be found online at <http://dx.doi.org/10.7717/peerj.20185#supplemental-information>.

## REFERENCES

- Ahmed M, Kim DR. 2018. pcr: an R package for quality assessment, analysis and testing of qPCR data. *PeerJ* 6:e4473 DOI 10.7717/peerj.4473.
- Bates D, Mächler M, Bolker B, Walker S. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software* 67:1–48 DOI 10.18637/jss.v067.i01.
- Boulter N, Suarez FG, Schibeci S, Sunderland T, Tolhurst O, Hunter T, Hodge G, Handelsman D, Simanainen U, Hendriks E, Duggan K. 2016. A simple, accurate and universal method for quantification of PCR. *BMC Biotechnology* 16:27 DOI 10.1186/s12896-016-0256-y.
- Derveaux S, Vandesompele J, Hellemans J. 2010. How to do successful gene expression analysis using real-time PCR. *Methods* 50:227–230 DOI 10.1016/j.ymeth.2009.11.001.
- Dvinge H, Bertone P. 2009. HTqPCR: high-throughput analysis and visualization of quantitative real-time PCR data in R. *Bioinformatics* 25:3325–3326 DOI 10.1093/bioinformatics/btp578.
- Ganger MT, Dietz GD, Ewing SJ. 2017. A common base method for analysis of qPCR data and the application of simple blocking in qPCR experiments. *BMC Bioinformatics* 18:534 DOI 10.1186/s12859-017-1949-5.
- Ganger MT, Dietz GD, Headley P, Ewing SJ. 2020. Application of the common base method to regression and analysis of covariance (ANCOVA) in qPCR experiments and subsequent relative expression calculation. *BMC Bioinformatics* 21:423 DOI 10.1186/s12859-020-03696-y.
- Harshitha R, Arunraj DR. 2021. Real-time quantitative PCR: a tool for absolute and relative quantification. *Biochemistry and Molecular Biology Education* 49:800–812 DOI 10.1002/bmb.21552.
- Kuznetsova A, Brockhoff PB, Christensen RHB. 2017. lmerTest package: tests in linear mixed effects models. *Journal of Statistical Software* 82:1–26 DOI 10.18637/jss.v082.i13.
- Li X, Wang Y, Li J, Mei X, Liu Y, Huang H. 2022. qPCRtools: an R package for qPCR data processing and visualization. *Frontiers in Genetics* 13:1002704 DOI 10.3389/fgene.2022.1002704.
- Lievens A, Van Aelst S, Van den Bulcke M, Goetghebeur E. 2012. Enhanced analysis of real-time PCR data by using a variable efficiency model: FPK-PCR. *Nucleic Acids Research* 40:e10–e10 DOI 10.1093/nar/gkr924.
- Livak KJ, Schmittgen TD. 2001. Analysis of relative gene expression data using real-time quantitative PCR and the 2- $\Delta\Delta$ CT method. *Methods* 25:402–408 DOI 10.1006/meth.2001.1262.
- McDavid A, Finak G, Chattopadhyay PK, Dominguez M, Lamoreaux L, Ma SS, Roederer M, Gottardo R. 2012. Data exploration, quality control and testing in single-cell qPCR-based gene expression experiments. *Bioinformatics* 29:461–467 DOI 10.1093/bioinformatics/bts714.
- Mirzaghaderi G. 2024. rtPCR: a package for statistical analysis of real-time PCR data in R. PREPRINT (Version 1) available at Research Square DOI 10.21203/rs3rs-4193266/v1.

- Navarro E, Serrano-Heras G, Castaño MJ, Solera J. 2015. Real-time PCR detection chemistry. *Clinica Chimica Acta* **439**:231–250 DOI [10.1016/j.cca.2014.10.017](https://doi.org/10.1016/j.cca.2014.10.017).
- Pabinger S, Rödiger S, Kriegner A, Vierlinger K, Weinhäusel A. 2014. A survey of tools for the analysis of quantitative PCR (qPCR) data. *Biomolecular Detection and Quantification* **1**:23–33 DOI [10.1016/j.bdq.2014.08.002](https://doi.org/10.1016/j.bdq.2014.08.002).
- Perkins JR, Dawes JM, McMahon SB, Bennett DL, Orengo C, Kohl M. 2012. ReadqPCR and NormqPCR: R packages for the reading, quality checking and normalisation of RT-qPCR quantification cycle (Cq) data. *BMC Genomics* **13**:1–8 DOI [10.1186/1471-2164-13-296](https://doi.org/10.1186/1471-2164-13-296).
- Pfaffl MW, Horgan GW, Dempfle L. 2002. Relative expression software tool (REST©) for group-wise comparison and statistical analysis of relative expression results in real-time PCR. *Nucleic Acids Research* **30**:e36–e36 DOI [10.1093/nar/30.9.e36](https://doi.org/10.1093/nar/30.9.e36).
- Ritz C, Spiess A-N. 2008. qpcR: an R package for sigmoidal model selection in quantitative real-time polymerase chain reaction analysis. *Bioinformatics* **24**:1549–1551 DOI [10.1093/bioinformatics/btn227](https://doi.org/10.1093/bioinformatics/btn227).
- Rödiger S, Burdukiewicz M, Schierack P. 2015. chipPCR: an R package to pre-process raw data of amplification curves. *Bioinformatics* **31**:2900–2902 DOI [10.1093/bioinformatics/btv205](https://doi.org/10.1093/bioinformatics/btv205).
- Ruijter JM, Barnewall RJ, Marsh IB, Szentirmay AN, Quinn JC, Van Houdt R, Gunst QD, Van den Hoff MJB. 2021. Efficiency correction is required for accurate quantitative PCR analysis and reporting. *Clinical Chemistry* **67**:829–842 DOI [10.1093/clinchem/hvab052](https://doi.org/10.1093/clinchem/hvab052).
- Ruiz-Villalba A, Ruijter JM, Van den Hoff MJB. 2021. Use and misuse of Cq in qPCR data analysis and reporting. *Life* **11**:496 DOI [10.3390/life11060496](https://doi.org/10.3390/life11060496).
- Searle SR, Speed FM, Milliken GA. 1980. Population marginal means in the linear model: an alternative to least squares means. *The American Statistician* **34**:216–221 DOI [10.1080/00031305.1980.10483031](https://doi.org/10.1080/00031305.1980.10483031).
- Taylor SC, Nadeau K, Abbasi M, Lachance C, Nguyen M, Fenrich J. 2019. The ultimate qPCR experiment: producing publication quality, reproducible data the first time. *Trends in Biotechnology* **37**:761–774 DOI [10.1016/j.tibtech.2018.12.002](https://doi.org/10.1016/j.tibtech.2018.12.002).
- Wilkinson L. 2016. *ggplot2: elegant graphics for data analysis*. New York: Springer-Verlag.
- Yuan JS, Reed A, Chen F, Stewart CN. 2006. Statistical analysis of real-time PCR data. *BMC Bioinformatics* **7**:85 DOI [10.1186/1471-2105-7-85](https://doi.org/10.1186/1471-2105-7-85).
- Zhang JD, Ruschhaupt M, Biczok R. 2013. ddCt method for qRT-PCR data analysis. *Citeseer* **48**:346–356.