Package 'qpcrNorm'

July 17, 2025

Type Package

Title Data-driven normalization strategies for high-throughput qPCR data.

Version 1.67.0

Date 2009-11-08

Author Jessica Mar

Maintainer Jessica Mar <jess@jimmy.harvard.edu>

Description The package contains functions to perform normalization of high-throughput qPCR data. Basic functions for processing raw Ct data plus functions to generate diagnostic plots are also available.

License LGPL (>= 2)

Depends methods, Biobase, limma, affy

Collate qpcrNormsource.R

biocViews Preprocessing, GeneExpression

git_url https://git.bioconductor.org/packages/qpcrNorm

git_branch devel

git_last_commit 0741149

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-07-16

Contents

---

calcCV *Calculates the Average Gene-Specific Coefficient of Variation*

Description

This function calculates the coefficient of variation for each gene in the qPCR experiment, and returns the average coefficient of variation across all genes.

Usage

```
calcCV(qBatch)
```

Arguments

qBatch	A qpcrBatch object.

Value

A numeric value.

Author(s)

Jess Mar <jess@jimmy.harvard.edu>

Examples

```
data(qpcrBatch.object)
mynormRI.data <- normQpcrRankInvariant(qpcrBatch.object, 1)
mynormQuant.data <- normQpcrQuantile(qpcrBatch.object)
barplot(c(calcCV(mynormRI.data), calcCV(mynormQuant.data)), col=c("red", "blue"))
```

---

ctQc *Quality Control Filter for Replicate Ct Values*

Description

This function applies a quality control filter to triplicate Ct values before combining them into a single summary Ct measure.

Current implementation can only handle three replicates.

Usage

```
ctQc(x)
```

Arguments

x	Matrix with three columns, corresponding to the triplicate Ct values.

Details

Applying ctQc is an alternative to averaging the triplicate Ct values. This filter was originally developed by Yasumasa Kimura 1. For each primer, sort Ct values in ascending order so we have [ct1, ct2, ct3]. 2. Take two differences difference1 := ct2 - ct1 difference2 := ct3 - ct2 If either or both have ct 40, we don't calculate the difference and set it to region 4 in the next step. 3. Take an average according to the differences. Here we have 2 thresholds : 0.2 and 1.0. With the thresholds, we classify the 2 differences into the below regions. region1 : difference <= 0.2 region2 : 0.2 < difference <= 1.0 region3 : 1.0 < difference region4 : either or both ct are 40 If the 2 differences are in same region, we take an average of 3 ct values. If the 2 differences are in different regions, we take an average of 2 ct values which are in smaller number region.

Value

Numeric vector of Ct values combined over the three replicates according to the QC filter.

Author(s)

Yasumasa Kimura
Jess Mar <jess@jimmy.harvard.edu>

See Also

readQpcr, readQpcrBatch

Examples

```
## myQpcrBatch <- readQpcrBatch(qc=T) # reads in data from a batch of qPCR experiments, applies ctQc to raw Ct v
```

`matrixByPlate` *Internal function to reorganize qPCR data into a rectangular structure.*

Description

This function takes a vector of Ct values from a single qPCR experiment and reorganizes it into a matrix structure. Each column in the matrix represents a different plate that was used in the experiment.

Usage

```
matrixByPlate(xvec, plateIndex)
```

Arguments

`xvec`	Numeric vector of Ct values.
`plateIndex`	Character vector, denoting plate index of each gene or primer pair.

Details

On the resulting matrix structure:
The number of rows equals the maximum number of genes or primer pairs that were used on a plate in the experiment. For plates with less genes, NA values are padded at the end of the column vector to complete the rectangular structure. Note: these NA values do not affect downstream calculations.

Value

A matrix object.

Author(s)

Jess Mar <jess@jimmy.harvard.edu>

See Also

normQpcrQuantile

Examples

```
data(qpcrBatch.object)
mynormQuant.data <- normQpcrQuantile(qpcrBatch.object)
```

---

normQpcrHouseKeepingGenes

Function for Housekeeping Gene Normalization of qPCR Data.

Description

Implements housekeeping gene normalization for a qpcrBatch object.

Usage

```
normQpcrHouseKeepingGenes(qBatch, hkeep.genes)
```

Arguments

qBatch	A qpcrBatch object to be normalized.
hkeep.genes	Character vector, specifying which housekeeping genes to be used for normalization.

Details

The names in hkeep.genes must be a subset of the gene or primer pair names slot in the qpcrBatch object.

Value

A qpcrBatch object, the normalized slot is now set at TRUE.

Author(s)

Jess Mar <jess@jimmy.harvard.edu>

See Also

normQpcrQuantile, normQpcrRankInvariant

Examples

```
data(qpcrBatch.object)
mynormHK.data <- normQpcrHouseKeepingGenes(qpcrBatch.object, c("Gpx4"))
```

---

normQpcrQuantile *Function for Quantile Normalization of qPCR Data.*

Description

Implements quantile normalization for a qpcrBatch object. We have adapted this algorithm from the function normalizeBetweenArrays from the **limma** package.

Data in a qpcrBatch object is normalized such that within an experiment, the expression distributions

across plates are more or less identical, and across experiments, the expression distributions

are also now more or less identical.

Usage

```
normQpcrQuantile(qBatch)
```

Arguments

qBatch	A link{qpcrBatch} object.

Value

A link{qpcrBatch} object, the normalized slot is now set at TRUE.

Author(s)

Jess Mar <jess@jimmy.harvard.edu>

See Also

normQpcrRankInvariant, normalizeBetweenArrays

Examples

```
data(qpcrBatch.object)
mynormQuant.data <- normQpcrQuantile(qpcrBatch.object)
```

normQpcrRankInvariant *Function for Rank-Invariant Set Normalization for qPCR Data.*

Description

Implements rank-invariant set normalization for a qpcrBatch object. We have adapted this algorithm from the function normalize.invariantset from the **affy** package.

Usage

```
normQpcrRankInvariant(qBatch, refType, rem.highCt = FALSE, thresh.Ct = 30)
```

Arguments

qBatch	A qpcrBatch object.
refType	Indicates what reference sample should be used, can be an integer or character string. See Details below.
rem.highCt	Logical indicator, TRUE if user wishes to remove genes with high Ct values (very low expression) that may be associated poor data quality.
thresh.Ct	Numerical value indicating the Ct value cutoff threshold, if rem.highCt = FALSE, genes with Ct values > thresh.Ct are removed from the data set.

Details

The algorithm computes all rank-invariant sets of genes between pairwise comparisons where each experimental sample in the qpcrBatch object is paired against a reference. There are several ways to specify what a sensible choice for the reference sample should be.

1. The reference is an experimental sample in the qpcrBatch object.
Specify refType as an integer value, corresponding to the index of which experimental sample is the reference.

2. The reference is the sample which is closest to mean of all the experiments.
Specify refType = "mean".

3. The reference is the sample which is closest to median of all the experiments.
Specify refType = "median".

4. The reference is the mean of all experiments in the qpcrBatch object.
Specify refType = "pseudo.mean".

5. The reference is the median of all experiments in the qpcrBatch object.
Specify refType = "pseudo.median".

Value

A qpcrBatch object, the normalized slot is now set at TRUE. The names of the rank-invariant genes used for normalization are stored as a vector in the normGenes slot of the qpcrBatch object returned. To retrieve the rank-invariant gene names, use qpcrBatch@normGenes.

Author(s)

Jess Mar <jess@jimmy.harvard.edu>

See Also

normQpcrQuantile, normalize.invariantset

Examples

```
data(qpcrBatch.object)
mynormRI.data <- normQpcrRankInvariant(qpcrBatch.object, 1)
mynormRI.data@normGenes  # retrieves names of genes in the rank-invariant set
```

---

plotVarMean *Constructs scatter plot to compare the effects of two normalization algorithms on a qPCR dataset.*

Description

This function makes a scatter plot which serves as a useful exploratory tool in evaluating whether one normalization algorithm has been more effective than another on a given qPCR dataset.

Usage

```
plotVarMean(qpcrBatch1, qpcrBatch2, normTag1 = "Normalization Type1", normTag2 = "Normalization Typ
```

Arguments

qpcrBatch1	A qpcrBatch object.
qpcrBatch2	A qpcrBatch object.
normTag1	Character string denoting what normalization algorithm was used for this data set.
normTag2	Character string denoting what normalization algorithm was used for this data set.
...	Further arguments can be supplied to the plot function.

Details

For each gene, the function plots its log-transformed ratio of its expression variance in one normalized dataset versus another normalized dataset, i.e. let Gij be the variance of the expression values of gene i that have been normalized with method j. We plot the natural log-transformed ratio of Gij to Gik on the y-axis, and the average expression of gene i on the x-axis for all genes. /cr The red curve represents a smoothed lowess curve that has been fitted to reflect the overall trend of the data. When the red curve drops below y = 0 (the blue dotted line) we know that method j effects a greater reduction in the variation of the data over method k. Similarly, when the red curve is above y = 0, method k is more effective in reducing the variation in the data than method j. If the data from both methods have similar variances then the red curve should remain at y = 0. Bolstad et al. (2003) originally used these plots for variance comparisons of different normalization methods for high density oligonucleotide array data.

Value

A plot object.

Author(s)

Jess Mar <jess@jimmy.harvard.edu>

References

Bolstad B et al. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics, 2003.

See Also

plot

Examples

```
 # data(qpcrBatch.object)
 # mynormRI.data <- normQpcrRankInvariant(qpcrBatch.object, 1)
 # mynormQuant.data <- normQpcrQuantile(qpcrBatch.object)
# plotVarMean(mynormRI.data, mynormQuant.data, normTag1="Rank-Invariant", normTag2="Quantile", main="Compar
```

---

qpcrBatch-class *Class qpcrBatch*

Description

This is a class representation for qPCR expression data.

Objects from the Class

Objects can be created using the function readQpcr or readQpcrBatch to read in raw data from a text file(s). Objects can also be created by using new("qpcrBatch", ...).

Slots

geneNames: Character vector denoting gene or primer pair names.

plateIndex: Character vector denoting plate indices.

exprs: Matrix of qPCR expression values, normally these are the Ct values.

normalized: Logical value, TRUE if expression data has been normalized.

normGenes: Character vector of genes used by the normalization algorithm.

Methods

No methods have yet been defined with class "qpcrBatch" in the signature.

Note

This class is better describe in the vignette.

Author(s)

Jess Mar <jess@jimmy.harvard.edu>

Examples

```
## load example data
data(qpcrBatch.object)
class(qpcrBatch.object)
```

---

qpcrBatch.object *qpcrBatch instance qpcrBatch.object*

Description

This is an artifically generated qPCR data set. The data set has been closely simulated from original data for 2396 genes on 13 time points. Each measurement within the one sample was repeated over three replicate wells, across multiple plates.

Usage

```
data(qpcrBatch.object)
```

Format

A data frame with 2396 observations on the following 41 variables.

`Primers` Character vector of gene or primer pair names.

`Plate_Index` Numeric vector denoting plate indices.

`Time1_Rep1` Ct values for first time point, first replicate.

`Time1_Rep2` Ct values for first time point, second replicate.

`Time1_Rep3` Ct values for first time point, third replicate.

Examples

```
data(qpcrBatch.object)
```

---

readQpcr *Data Input Function for a Single qPCR Experiment.*

Description

This function reads in data from a single qPCR experiment. The text file must have the following structure:
1st column = names denoting genes or primer pairs
2nd column = plate index of each gene or primer pair
remaining columns = (replicate) Ct values.

Usage

```
readQpcr(fileName, header = FALSE, qc = FALSE, quote = "\"", dec = ".", fill = TRUE, comment.char = ""
```

Arguments

fileName	Character string.
header	Logical value, TRUE if the file contains the names of the variables as its first line.
qc	Logical value, TRUE if a QC filter ctQc should be applied to the data. If qc = F, the replicate Ct values will be averaged.
quote	Set of quoting characters. To disable quoting, set quote = "". See scan for behaviour on quotes embedded in quotes.
dec	Character used for decimal points.
fill	Logical value, TRUE if in case rows have unequal length, blank fields are implicitly added. See read.table.
comment.char	Character vector of length one containing a single character or an empty string. Use "" to turn off the interpretation of comments altogether.
...	further arguments to be passed to read.table.

Details

Note: the majority of arguments to readQpcr are identical to those supplied to read.table. These have been included to give the user greater control over data input, should the data deviate from a standard tab-delimited file structure. For a standard tab-delimited text file (without column headings), specifying the fileName should be sufficient.

Value

A qpcrBatch object.

Author(s)

Jess Mar <jess@jimmy.harvard.edu>

See Also

readQpcrBatch, ctQc

Examples

```
## onerun.data <- readQpcr("singleQpcrRun.txt")
```

readQpcrBatch *Data Input Function for a Batch of qPCR Experiments.*

Description

This function reads in data from multiple qPCR experiments from the one batch. Each text file in the batch must meet the structure required by readQpcr.
Note: In order to qualify as a batch, it is assumed that the same set of primers are being analyzed in each experiment.

Usage

```
readQpcrBatch(..., filenames = character(), header = FALSE, qc = FALSE)
```

Arguments

...	Filenames separated by a comma.
filenames	Character vector specifying file names.
header	Logical value, TRUE if the file contains the names of the variables as its first line.
qc	Logical value, TRUE if a QC filter ctQc should be applied to the data. If qc = F, the replicate Ct values will be averaged. See ctQc.

Details

If the function is called with no arguments readQpcrBatch() all the files in the working directory are read and put into a qpcrBatch object. All files must conform to the following structure:
1st column = names denoting genes or primer pairs
2nd column = plate index of each gene or primer pair
remaining columns = (replicate) Ct values

Note: the majority of arguments to readQpcr are identical to those supplied to read.table. These have been included to give the user greater control over data input, should the data deviate from a standard tab-delimited file structure. For a set of standard tab-delimited text files (without column headers), specifying the filenames should be sufficient.

Value

A qpcrBatch object.

Author(s)

Jess Mar <jess@jimmy.harvard.edu>

See Also

ctQc, readQpcr, setwd

Examples

```
## myBatch <- readQpcrBatch()
```

writeQpcr *Writes qpcrBatch object out to a File.*

Description

This function writes a qpcrBatch out to a tab-delimited text file. writeQpcr can be used to write out the normalized qPCR data out to an external file.

Usage

```
writeQpcr(qBatch, fileName, ...)
```

Arguments

qBatch	A qpcrBatch object.
fileName	Character string specifying name of the output file.
...	Extra arguments to be passed to write.table.

Details

Function creates a tab-delimited text file with three columns,
1st column = names denoting genes or primer pairs 2nd column = plate index 3rd column = normalized Ct value

Author(s)

Jess Mar <jess@jimmy.harvard.edu>

References

Mar J et al. Data-driven Normalization Strategies for qPCR Data. Technical Report, 2008.

See Also

write.table

Examples

```
## writeQpcr(qpcrBatch.object, "output1.txt")
```

Index