Package ‘NormqPCR’
March 23, 2017

Version 1.20.0
Date 2016-10-10
Title Functions for normalisation of RT-qPCR data
Description Functions for the selection of optimal reference genes and the normalisation of real-time quantitative PCR data.
Author Matthias Kohl, James Perkins, Nor Izayu Abdul Rahman
Maintainer James Perkins <j.perkins@ucl.ac.uk>
Depends R(>= 2.14.0), stats, RColorBrewer, Biobase, methods, ReadqPCR, qpcR
biocViews MicrotitrePlateAssay, GeneExpression, qPCR
License LGPL-3
LazyLoad yes
LazyData yes
URL www.bioconductor.org/packages/release/bioc/html/NormqPCR.html
Collate allGenerics.R deltaCt.R selectHKs.R combineTechReps.R
   combineTechRepsWithSD.R CqValues.R deltaDeltaCt.R
   stabMeasureM.R dealWithNA.R geomMean.R stabMeasureRho.R
   CtToCqWrappers.R ComputeNRQs.R
NeedsCompilation no

R topics documented:

- NormqPCR-package ............................................................... 2
- Bladder ................................................................. 3
- BladderRepro .......................................................... 4
- Colon ................................................................. 5
- combineTechReps ..................................................... 7
- combineTechRepsWithSD ............................................. 8
- ComputeNRQs .......................................................... 9
- CqValues ............................................................. 10
- deltaCt ............................................................... 12
- deltaDeltaCt ......................................................... 14
- geNorm .............................................................. 15
- geomMean ............................................................ 17
- makeAllNAs .......................................................... 18
Functions for normalisation of RT-qPCR data.

Details

Package: NormqPCR
Type: Package
Version: 1.7.1
Date: 2014-08-13
Depends: R(>= 2.14.0), stats, RColorBrewer, Biobase, methods, ReadqPCR, qpcR
Imports: ReadqPCR
biocViews: MicrotitrePlateAssay, GeneExpression, qPCR
License: LGPL-3
LazyLoad: yes
LazyData: yes

require(NormqPCR)

Author(s)

Matthias Kohl, James Perkins, Nor Izayu Abdul Rahman
Maintainer: James Perkins <j.perkins@ucl.ac.uk>

References


Examples

```r
## some examples are given in the vignette
## Not run:
library(NormqPCR)
vignette("NormqPCR")

## End(Not run)
```

Bladder


Description

This dataset was used in Andersen et al (2004) to demonstrate normalization of real-time quantitative RT-PCR data by geometric averaging of housekeeping genes.

Usage

data(Bladder)

Format

A qPCRBatch object which contains an expression matrix with the expression of 14 genes measured in 28 samples. The sample information is saved in the phenoData slot with variables

- **Sample.no.** sample number.
- **Grade** Grade of bladder cancer.

The following information on the measured genes is saved in the variables **Symbol** and **Gene.name** of the featureData slot.

- **ATP5B** ATP synthase, H+ transporting, mitochondrial F1 complex, beta polypeptide.
- **HSPCB** Heat shock 90-kDa protein 1, beta.
- **S100A6** S100 calcium-binding protein A6 (calcylin).
- **FLOT2** Flotillin 2.
- **TEGT** Testis enhanced gene transcript (BAX inhibitor 1).
- **UBB** Ubiquitin B.
- **TPT1** Tumor protein, translationally controlled 1.
- **CFL1** Cofilin 1 (non-muscle).
- **ACTB** Actin, beta.
- **RPS23** Ribosomal protein S23.
- **GAPD** Glyceraldehyde-3-phosphate dehydrogenase.
- **UBC** Ubiquitin C.
- **FLJ20030** Hypothetical protein FLJ20030.

For a detailed annotation see Table 1 in Anderson et al. (2004).
Details

The genes included in this data set were selected by screening 99 bladder sample expression profiles.

Source

The data set was obtained from http://www.mdl.dk/Publications_sup1.htm

References


Examples

data(Bladder)
Bladder
head(exprs(Bladder))
pData(Bladder)
fData(Bladder)


Description

This dataset was used in Andersen et al (2004) to demonstrate normalization of real-time quantitative RT-PCR data by geometric averaging of housekeeping genes.

Usage

data(BladderRepro)

Format

A qPCRBatch object which contains an expression matrix with the expression of 8 genes measured in 26 samples. The sample information is saved in the phenoData slot with variables

Sample.no. sample number.
Grade Grade of bladder cancer.

The following information on the measured genes is saved in the variables Symbol and Gene.name of the featureData slot.

CD14 CD14 antigen.
FCN1 Ficolin (collagen/fibrinogen domain containing) 1.
CCNG2 Cyclin G2.
NPAS2 Neuronal PAS domain protein 2.
UBC Ubiquitin C.


CFL1 Cofilin 1 (non-muscle).
ACTB Actin, beta.
GAPD Glyceraldehyde-3-phosphate dehydrogenase.

For a detailed annotation see Table 1 and Supplementary table 1 in Anderson et al. (2004).

Details
This data set was used to check the reproducibility of the results obtained in Andersen et al (2004).

Source
The data set was obtained from http://www.mdl.dk/Publications_sup1.htm

References

Examples
data(BladderRepro)
BladderRepro
head(exprs(BladderRepro))
pData(BladderRepro)
fData(BladderRepro)

Colon

Description
This dataset was used in Andersen et al (2004) to demonstrate normalization of real-time quantitative RT-PCR data by geometric averaging of housekeeping genes.

Usage
data(Colon)

Format
A qPCRBatch object which contains an expression matrix with the expression of 13 genes measured in 40 samples. The sample information is saved in the phenoData slot with variables
Sample.no. sample number.
Classification Classification of colon cancer.

The following information on the measured genes is saved in the variables Symbol and Gene.name of the featureData slot.
UBC  Ubiquitin C.
UBB  Ubiquitin B.
SUI1  Putative translation initiation factor.
NACA Nascent-polypeptide-associated complex alpha polypeptide.
FLJ20030  Hypothetical protein FLJ20030.
CFL1  Cofilin 1 (non-muscle).
ACTB  Actin, beta.
CLTC  Clathrin, heavy polypeptide (Hc).
RPS13  Ribosomal protein S13.
RPS23  Ribosomal protein S23.
GAPD  Glyceraldehyde-3-phosphate dehydrogenase.
TPT1  Tumor protein, translationally controlled 1.
TUBA6  Tubulin alpha 6.

For a detailed annotation see Table 1 in Anderson et al. (2004).

Details
The genes included in this data set were selected by screening 161 colon sample expression profiles.

Source
The data set was obtained from http://www.mdl.dk/Publications_sup1.htm

References

Examples
data(Colon)
Colon
head(exprs(Colon))
pData(Colon)
fData(Colon)
combineTechReps

Combines Technical Replicates

Description

Takes expression set of qPCR values containing technical replicates and combines them.

Usage

combineTechReps(qPCRBatch, ...)

## S4 method for signature 'qPCRBatch'
combineTechReps(qPCRBatch, calc="arith")

Arguments

qPCRBatch Expression set containing qPCR data, read in by a ReadqPCR function and containing technical reps, denoted by _TechRep.n suffix.

... Extra arguments, detailed below

calc use median, arithmetic or geometric mean for combining the values

Details

Takes exprs of qPCR values containing technical replicates and combines them using a specified centrality measure.

Value

qPCRBatch with same number of samples, but with less features, since all technical replicates are replaced with a single value of their means.

Author(s)

James Perkins <j.perkins@ucl.ac.uk>

References


Examples

```r
path <- system.file("exData", package = "NormqPCR")
qPCR.example.techReps <- file.path(path, "qPCR.techReps.txt")
qPcrBatch.qPCR.techReps <- read.qPCR(qPCR.example.techReps)
rownames(exprs(qPcrBatch.qPCR.techReps))
combinedTechReps <- combineTechReps(qPcrBatch.qPCR.techReps)
rownames(exprs(combinedTechReps))
```
combineTechRepsWithSD  Combines Technical Replicates

Description
Takes expression set of qPCR values containing technical replicates and combines them. In addition the appropriate standard deviation (SD) is computed.

Usage
```
combineTechRepsWithSD(qPCRBatch, ...)  
```

## S4 method for signature 'qPCRBatch'
```
combineTechRepsWithSD(qPCRBatch, calc="arith")
```

Arguments
- `qPCRBatch`: Expression set containing qPCR data, read in by a ReadqPCR function and containing technical reps, denoted by _TechRep.n suffix.
- `...`: Extra arguments, detailed below
- `calc`: use median, arithmetic or geometric mean for combining the values

Details
Takes `exprs` of qPCR values containing technical replicates and combines them using a specified centrality measure.

The arithmetic mean (`calc="arith"`) is combined with the classical standard deviation. In case of the geometric mean (`calc="geom"`) the classical standard deviation of the log-values is exponentiated. The median (`calc="median"`) is calculated in connection with the MAD.

Value
`qPCRBatch` with same number of samples, but with less features, since all technical replicates are replaced with a single value of their means. In addition the slot `assayData` includes a matrix with SD values which can be accessed via `se.exprs`.

Author(s)
Matthias Kohl <Matthias.Kohl@stamats.de>

References

See Also
- `combineTechReps`
Examples

```r
path <- system.file("exData", package = "NormqPCR")
qPCR.example.techReps <- file.path(path, "qPCR.techReps.txt")
qPCRBatch.qPCR.techReps <- read.qPCR(qPCR.example.techReps)
rownames(exprs(qPCRBatch.qPCR.techReps))
combinedTechReps <- combineTechRepsWithSD(qPCRBatch.qPCR.techReps)
rownames(exprs(combinedTechReps))
exprs(combinedTechReps)
se.exprs(combinedTechReps)
```

ComputeNRQs

**Compute Normalized Relative Quantities**

Description

This function computes normalized relative quantities (NRQs) for a qPCRBatch.

Usage

```r
ComputeNRQs(qPCRBatch, ...)
## S4 method for signature 'qPCRBatch'
ComputeNRQs(qPCRBatch, hkgs)
```

Arguments

- `qPCRBatch`: an object of class `qPCRBatch`.
- `hkgs`: Names of reference/housekeeping genes.
- `...`: other parameters to be passed to downstream methods.

Details

Allows the user to normalized relative quantities as defined in Hellemanns et al. (2007).

Value

Object of class "qPCRBatch".

Author(s)

Nor Izayu Abdul Rahman, Matthias Kohl <Matthias.Kohl@stamats.de>

References

Hellemans, Jan, Geert Mortier, Anne De Paepe, Frank Speleman and Jo Vandesompele (2007). qBase relative quantification framework and software for management and automated analysis of real-time quantitative PCR data. *Genome Biology*, 8:R19

CqValues  

Compute Cq value and amplification efficiency

Description

This function calculates Cq value and amplification efficiency for a CyclesSet. It is based on function `pcrbatch` of package `qpcR`.

Usage

CqValues(object, ...)  
## S4 method for signature 'CyclesSet'
CqValues(object, Effmethod = "expfit", group = NULL,  
model = 15, check = "uni2", checkPAR = parKOD(),  
remove = "none", exclude = NULL, type = "cpD2",  
labels = NULL, norm = FALSE, baseline = NULL, basefac = 1, smooth = NULL,  
smoothPAR = list(span = 0.1), factor = 1, opt = FALSE,  
optPAR = list(sig.level = 0.05, crit = "ftest"),  
plot = FALSE, verbose = FALSE, ...)

See Also

qPCRBatch-class

Examples

## Example data
path <- system.file("exData", package = "ReadqPCR")
qPCR.example <- file.path(path, "qPCR.example.txt")
Cq.data <- read.qPCR(qPCR.example)

## combine technical replicates
Cq.data1 <- combineTechRepsWithSD(Cq.data)

## add efficiencies
Effs <- file.path(path, "Efficiencies.txt")
Cq.effs <- read.table(file = Effs, row.names = 1, header = TRUE)
rownames(Cq.effs) <- featureNames(Cq.data1)
effs(Cq.data1) <- as.matrix(Cq.effs[, "efficiency", drop = FALSE])
se.effs(Cq.data1) <- as.matrix(Cq.effs[, "SD.efficiency", drop = FALSE])

##
res <- ComputeNRQs(Cq.data1, hkgs = c("gene_az", "gene_gx"))
## NRQs
exprs(res)
## SD of NRQs
se.exprs(res)
CqValues

Arguments

- **object**: an object of class `CyclesSet`.
- **Effmethod**: a character vector defining the methods for computing amplification efficiency.
- **group**: a vector containing the grouping for possible replicates.
- **model**: the model to be used for all runs. Default model is 15.
- **check**: the method for kinetic outlier detection in `KOD`. Method "uni2" is set as default which is a test on sigmoidal structure.
- **checkPAR**: parameters to be supplied to the check method. See `parKOD`.
- **remove**: indicates which runs to be removed. Either none of them, those which failed to fit or from the outlier methods.
- **exclude**: indicates samples to be excluded from calculation, either "" for samples with missing column names or a regular expression defining columns (samples); see 'Details' and 'Examples' in `modlist`.
- **type**: the point on the amplification curve which is used for efficiency estimation; see `efficiency`.
- **labels**: a vector containing labels which define replicate groups. See more details in `pcrbatch` and `ratiobatch`.
- **norm**: a logical value which determines whether the raw data should be normalized within [0, 1] before model fitting or not.
- **baseline**: type of baseline subtraction. More details in `efficiency`.
- **basefac**: a factor when using averaged baseline cycles, such as 0.95.
- **smooth**: the curve smoothing method. See more details in `pcrbatch`.
- **smoothPAR**: parameters to be supplied to smoothing method in `smooth`.
- **factor**: a multiplication factor for the fluorescence response values.
- **opt**: a logical value which determines whether model selection should be applied to each model or not.
- **optPAR**: parameters to be supplied for model selection in `mselect`.
- **plot**: a logical value. If TRUE, the single runs are plotted from the internal modlist for diagnostics.
- **verbose**: a logical value. If TRUE, fitting and tagging results will be displayed in the console.
- **...**: other parameters to be passed to downstream methods.

Details

Allows the user to compute Cq value and amplification efficiency. In addition, all values generated during the computations are saved. This function has four choices of methods for computing amplification efficiency values which are the methods provided by package `qpcR`.

More details on technical replication and normalization is given in the vignette `NormqPCR`.

Value

Object of class "qPCRBatch".

Author(s)

Nor Izayu Abdul Rahman, Matthias Kohl <Matthias.Kohl@stamats.de>
References


See Also

`pcrbatch`, `CyclesSet-class`, `qPCRBatch-class`

Examples

```r
## Read in the raw qPCR data from file "LC480_Example.txt"
path <- system.file("exData", package = "ReadqPCR")
LC480.example <- file.path(path, "LC480_Example.txt")
cycData <- read.LC480(file = LC480.example)

## Read in the sample information data from file "LC480_Example_SampleInfo.txt".
LC480.SamInfo <- file.path(path, "LC480_Example_SampleInfo.txt")
samInfo <- read.LC480SampleInfo(LC480.SamInfo)

## Merge information
cycData1 <- merge(cycData, samInfo)

## Compute Cq values
## 1) use sigmoidal model
res1 <- CqValues(cycData1, Effmethod = "sigfit")
res1
effs(res1)
se.effs(res1)

## 2) fit exponential model (default)
res2 <- CqValues(cycData1, Effmethod = "expfit")
res2
effs(res2)
se.effs(res2)

## 3) use window of linearity
res3 <- CqValues(cycData1, Effmethod = "sliwin")
res3
effs(res3)
se.effs(res3)

## 4) linear regression of efficiency
res4 <- CqValues(cycData1, Effmethod = "LRE")
res4
effs(res4)
se.effs(res4)
```

deltaCt

**Perform normalization with a given housekeeping gene**

Description

Normalise qPCR eset using a given housekeeping gene as control, then perform differential expression analysis using the delta delta Ct method.
Usage

deltaCt(qPCRBatch, ...)
## S4 method for signature 'qPCRBatch'
deltaCt(qPCRBatch, hkgs, combineHkgs=FALSE, calc="arith")
deltaCq(qPCRBatch, hkgs, combineHkgs=FALSE, calc="arith")

Arguments

  qPCRBatch  qPCR-specific expression set, containing qPCR data.
  ...        Extra arguments, detailed below
  hkgs       String containing the name of the name of the housekeeping gene which will be
             used to normalise the rest of the genes.
  combineHkgs Logical - if TRUE, then as long as more than one housekeeper given for argu-
                 ment hkgs, it will combine the housekeepers by finding the geometric mean.
                 Housekeepers can be found using geNorm or NormFinder algorithms.
  calc       use arithmetic or geometric mean.

Details

Takes expression set of qPCR values and normalises them using a housekeeping gene. Returns a
qPCRBatch with exprs set of the same dimensions but with the given hkg value subtracted.

Value

qPCRBatch with exprs set of the same dimensions but with the given hkg value subtracted.

Author(s)

James Perkins <j.perkins@ucl.ac.uk>

References

cncbi.nlm.nih.gov/pubmed/11846609

Perkins, JR, Dawes, JM, McMahon, SB, Bennett, DL, Orenco, C, Kohl, M (2012). ReadqPCR and
NormqPCR: R packages for the reading, quality checking and normalisation of RT-qPCR quantifi-
cation cycle (Cq) data. BMC Genomics, 13, 1:296.

See Also

selectHKs, deltaDeltaCq

Examples

path <- system.file("exData", package = "NormqPCR")
taqman.example <- file.path(path, "example.txt")
qPCRBatch.taqman <- read.taqman(taqman.example)
hkgs<="Actb-Rn00667869_m1"
qPCRBatch.norm <- deltaCq(qPCRBatch = qPCRBatch.taqman, hkgs = hkgs, calc="arith")
head(exprs(qPCRBatch.norm))
deltaDeltaCt

Perform normalization and differential expression with given housekeeping gene

Description

Normalise qPCRBatch RT-qPCR data using housekeeping genes as control, then perform differential expression analysis using the delta delta Cq method.

Usage

deltaDeltaCt(qPCRBatch,...)
## S4 method for signature 'qPCRBatch'
deltaDeltaCt(qPCRBatch, maxNACase=0, maxNAControl=0, hkgs, contrastM, case, control, paired=TRUE, hkgCalc="arith", statCalc="arith")
deltaDeltaCq(qPCRBatch, maxNACase=0, maxNAControl=0, hkgs, contrastM, case, control, paired=TRUE, hkgCalc="arith", statCalc="arith")

Arguments

- qPCRBatch: qPCR-specific expression set, containing qPCR data.
- ...: Extra arguments, detailed below
- maxNACase: Maximum number of NA values allowed before a detector’s reading is discarded for samples designated as case.
- maxNAControl: Maximum number of NA values allowed before a detector’s reading is discarded for samples designated as control.
- hkgs: String containing the name of the name of the housekeeping gene which will be used to normalise the rest of the genes.
- contrastM: A binary matrix which designates case and control samples.
- case: The name of the column in contrastM that corresponds to the case samples.
- control: The name of the column in contrastM that corresponds to the control samples.
- paired: Logical - if TRUE the detectors and housekeepers in the same sample will be paired for calculating standard deviation, effectively meaning we will be calculating standard deviation of the differences. If FALSE, there will be no pairing, and standard deviation will be pooled between the detector and housekeepers.
- hkgCalc: String - either "arith" or "geom", details how the different housekeeper genes should be combined - either by using the arithmetic or geometric mean.
- statCalc: String - either "arith" or "geom", details how genes should be combined - either by using the arithmetic or geometric mean.

Details

Takes expression set of qPCR values and normalises them using different housekeeping genes. Returns separate sets of values for each housekeeping gene given.

Value

matrix with columns containing the detector ids, 2^delta Cq values for the sample of interest and the callibrator sample, alongside their respective standard deviations, the 2^delta delta Cq values and the minimum and maximum values (ie the values that are 1 sd away)


**geNorm**

**Author(s)**

James Perkins <j.perkins@ucl.ac.uk>

**References**


**See Also**

selectHKs, deltaCq

**Examples**

```r
path <- system.file("exData", package = "NormqPCR")
taqman.example <- file.path(path, "example.txt")
quPCRBatch.taqman <- read.taqman(taqman.example)
hkg <- "Actb-Rn00667869_m1"

colM <- cbind(c(0,0,1,1,0,0,1,1),c(1,1,0,0,1,1,0,0))
rownames(colM) <- sampleNames(qPCRBatch.taqman)

ddCq.taqman <- deltaDeltaCq(qPCRBatch = qPCRBatch.taqman, maxNACase=1, maxNAControl=1, hkg=hkg, contrastM=colM, case="interestingPhenotype", control="wildTypePhenotype", statCalc="geom", hkgCalc="arith")

head(ddCq.taqman)
```

---

**geNorm**

**Data set of Vandesompele et al (2002)**

**Description**

This data set was used in Vandesompele et al (2002) to demonstrate normalization of real-time quantitative RT-PCR data by geometric averaging of housekeeping genes.

**Usage**

data(geNorm)

**Format**

A qPCRBatch object which contains an expression matrix with 85 observations on the following 10 variables which stand for expression data of ten potential reference/housekeeping genes

ACTB actin, beta
B2M  beta-2-microglobulin
GAPD  glyceraldehyde-3-phosphate dehydrogenase
HMBS  hydroxymethylbilane synthase
HPRT1  hypoxanthine phosphoribosyltransferase 1
RPL13A  ribosomal protein L13a
SDHA  succinate dehydrogenase complex subunit A
TBP  TATA box binding protein
UBC  ubiquitin C
YWHAZ  tyrosine 3-monooxygenase/tryptophan 5-monooxygenase activation protein, zeta polypeptide

Details
The row names of this data set indicate the various human tissues which were investigated.

**BM**  9 normal bone-marrow samples
**POOL**  9 normal human tissues from pooled organs (heart, brain, fetal brain, lung, trachea, kidney, mammary gland, small intestine and uterus)
**FIB**  20 short-term cultured normal fibroblast samples from different individuals
**LEU**  13 normal leukocyte samples
**NB**  34 neuroblastoma cell lines (independently prepared in different labs from different patients)

Source
The data set was obtained from [http://genomebiology.com/content/supplementary/gb-2002-3-7-research0034-s1.txt](http://genomebiology.com/content/supplementary/gb-2002-3-7-research0034-s1.txt)

References


Examples
```r
data(geNorm)
str(exprs(geNorm$qPCRBatch))
sampleNames(geNorm$qPCRBatch)
```
**Description**

Computation of the geometric mean.

**Usage**

```r
geomMean(x, na.rm = TRUE)
```

**Arguments**

- `x` numeric vector of non-negative Reals
- `na.rm` a logical value indicating whether NA values should be stripped before the computation proceeds.

**Details**

The computation of the geometric mean is done via `\prod(x)^(1/\text{length}(x))`.

**Value**

geometric mean

**Note**

A first version of this function appeared in package SLqPCR.

**Author(s)**

Matthias Kohl <Matthias.Kohl@stamats.de>

**References**


**Examples**

```r
x <- rlnorm(100)
geomMean(x)
```
makeAllNAs

Make all Cq values NA

Description
Make all Cq values for a given detector NA when the number of NAs for that detector is above a given threshold.

Usage
makeAllNAs(qPCRBatch, ...)

## S4 method for signature 'qPCRBatch'
makeAllNAs(qPCRBatch, contrastM, sampleMaxM)

Arguments
qPCRBatch
Expression set containing qPCR data.

... Extra arguments, detailed below

contrastM Contrast Matrix like that used in limma. Columns represent the different sample types, rows are the different samples, with a 1 or 0 in the matrix indicating which sample types the different samples belong to.

sampleMaxM Sample Max Matrix. Columns represent the different sample types. There is one value per column, which represents the max number of NAs allowed for that sample type.

Details
Make all NAs when number of NAs above a given threshold.

Value
qPCRBatch object with a new exprs slot, everything else equal.

Author(s)
James Perkins <j.perkins@ucl.ac.uk>

References

Examples
# read in the data
path <- system.file("exData", package = "NormqPCR")
taqlman.example <- file.path(path, "example.txt")
qPCRBatch.taqlman <- read.taqlman(taqlman.example)
exprs(qPCRBatch.taqlman)["Ccl20.Rn00570287_m1",] # values before
# make contrastM
a <- c(0,0,1,1,0,0,1,1) # one for each sample type, with 1 representing
b <- c(1,1,0,0,1,1,0,0) # position of sample type in the samplenames vector
contM <- cbind(a, b)
colnames(contM) <- c("case", "control") # then give the names of each sample type
rownames(contM) <- sampleNames(qPCRBatch.taqman) # and the rows of the matrix
contM

# make sampleMaxM
sMaxM <- t(as.matrix(c(3,3))) # now make the sample max matrix
colnames(sMaxM) <- c("case", "control") # make sure these line up with samples
sMaxM

# function
goToBatch.taqman.replaced <- makeAllNAs(qPCRBatch.taqman, contrastM, sMaxM)
exprs(qPCRBatch.taqman.replaced)["Cc120_Rn00570287_m1",]

## makeAllNewVal

**Make all Cq values NA**

**Description**

Make all Cq values for a given detector NA when the number of NAs for that detector is above a
given threshold

**Usage**

```r
makeAllNewVal(qPCRBatch, ...)
```

**Arguments**

- `qPCRBatch`: Expression set containing qPCR data.
- `...`: Extra arguments, detailed below
- `contrastM`: Contrast Matrix like that used in limma. Columns represent the different samples
types, rows are the different samples, with a 1 or 0 in the matrix indicating which
sample types the different samples belong to.
- `sampleMaxM`: Sample Max Matrix. Columns represent the different sample types. There is
one value per column, which represents the max number of NAs allowed for
that sample type.
- `newVal`: New value to give the values in the group where the NAs are above the threshold.

**Details**

Make all a given value when number of NAs above a given threshold, with different thresholds
for the different sample classes, using sMaxM and contM to provide this information, as detailed
below.
Value

qPCRBatch object with a new exprs slot, everything else equal

Author(s)

James Perkins <j.perkins@ucl.ac.uk>

References


Examples

```r
# read in the data
path <- system.file("exData", package = "NormqPCR")
taqman.example <- file.path(path, "example.txt")
qPCRBatch.taqman <- read.taqman(taqman.example)
exprs(qPCRBatch.taqman)["Ccl20.Rn00570287_m1",] # values before

# make contrastM
a <- c(0,0,1,1,0,0,1,1) # one for each sample type, with 1 representing
b <- c(1,1,0,0,1,1,0,0) # position of sample type in the samplenames vector
contM <- cbind(a,b)
colnames(contM) <- c("case","control") # then give the names of each sample type
rownames(contM) <- sampleNames(qPCRBatch.taqman) # and the rows of the matrix
contM

# make sampleMaxM
sMaxM <- t(as.matrix(c(3,3))) # now make the sample max matrix
colnames(sMaxM) <- c("case","control") # make sure these line up with samples
sMaxM

# function
qPCRBatch.taqman.replaced <- makeAllNewVal(qPCRBatch.taqman, contM, sMaxM)
exprs(qPCRBatch.taqman.replaced)["Ccl20.Rn00570287_m1",]
```

---

**replaceAboveCutOff**

*Replace Cq values with new value*

Description

Replace Cq values above a given threshold with a new value

Usage

```r
replaceAboveCutOff(qPCRBatch, ...)  
## S4 method for signature 'qPCRBatch'
```
replaceNAs

Arguments

- `qPCRBatch` Expression set containing qPCR data.
- `...` Extra arguments, detailed below
- `newVal` The new value with which to replace the values above the cutoff
- `cutOff` the minimal threshold above which the values will be replaced

Details

Replaces values in the exprs slot of the `qPCRBatch` object that are above a threshold value with a new number

Value

- `qPCRBatch` object with a new exprs slot

Author(s)

James Perkins <j.perkins@ucl.ac.uk>

References


Examples

```r
path <- system.file("exData", package = "NormqPCR")
taqman.example <- file.path(path, "example.txt")
qPCRBatch.taqman <- read.taqman(taqman.example)
exprs(qPCRBatch.taqman)["Ccl20_Rn00570287_m1",]
qPCRBatch.taqman.replaced <- replaceAboveCutOff(qPCRBatch.taqman, newVal = NA, cutOff = 35)
exprs(qPCRBatch.taqman.replaced)["Ccl20_Rn00570287_m1",]
```

---

**replaceNAs**  
*Replace NAs with a given value*

Description

Replace NAs with a given value

Usage

```r
replaceNAs(qPCRBatch, ...)
```

## S4 method for signature 'qPCRBatch'
```
replaceNAs(qPCRBatch, newNA)
```
Arguments

qPCRBatch     Expression set containing qPCR data.
...
newNA        The new value to replace the NAs with

Details

Replaces NA values in the exprs slot of the qPCRBatch object with a given number

Value

qPCRBatch object with a new exprs slot

Author(s)

James Perkins <j.perkins@ucl.ac.uk>

References


Examples

path <- system.file("exData", package = "NormqPCR")
taqman.example <- file.path(path, "example.txt")
qPCRBatch.taqman <- read.taqman(taqman.example)
qPCRBatch.taqman.replaced <- replaceNAs(qPCRBatch.taqman, newNA = 40)
exprs(qPCRBatch.taqman.replaced)["Ccl20.Rn00570287_m1"]

selectHKs

Selection of reference/housekeeping genes

Description

This function can be used to determine a set of reference/housekeeping (HK) genes for gene expression experiments

Usage

selectHKs(qPCRBatch, ...)

## S4 method for signature 'matrix'
selectHKs(qPCRBatch, group, method = "geNorm", minNrHKs = 2, log = TRUE, Symbols, trace = TRUE, na.rm = TRUE)

## S4 method for signature 'qPCRBatch'
selectHKs(qPCRBatch, group, method = "geNorm", minNrHKs = 2, log = TRUE, Symbols, trace = TRUE, na.rm = TRUE)
Arguments

qPCRBatch matrix or qPCRBatch, containing the data (expression matrix) in the exprs slot.

Extra arguments, detailed below:

- group: optional factor not used by all methods, hence may be missing
- method: method to compute most stable genes
- minNrHKs: minimum number of HK genes that should be considered
- log: logical: is data on log-scale
- Symbols: gene symbols
- trace: logical, print additional information
- na.rm: a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

This function can be used to determine a set of reference/housekeeping (HK) genes for gene expression experiments. The default method "geNorm" was proposed by Vandesompele et al. (2002). Currently, the geNorm method by Vandesompele et al. (2002) and the NormFinder method of Andersen et al. (2004) are implemented.

Vandesompele et al. (2002) propose a cut-off value of 0.15 for the pairwise variation. Below this value the inclusion of an additional housekeeping gene is not required.

Value

If method = "geNorm" a list with the following components is returned:

- ranking: ranking of genes from best to worst where the two most stable genes cannot be ranked
- variation: pairwise variation during stepwise selection
- meanM: average expression stability M

If method = "NormFinder" a list with the following components is returned:

- ranking: ranking of genes from best to worst where the two most stable genes cannot be ranked

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

References


Examples

data(geNorm)
tissue <- as.factor(c(rep("BM", 9), rep("FIB", 20), rep("LEU", 13), rep("NB", 34), rep("POOL", 9)))
res.BM <- selectHKs(geNorm.qPCRBatch[, tissue == "BM"], method = "geNorm", Symbols = featureNames(geNorm.qPCRBatch), minNrHK = 2, log = FALSE)

stabMeasureM

Description

Computation of the gene expression stability value M for real-time quantitative RT-PCR data. For more details we refer to Vandesompele et al. (2002).

Usage

stabMeasureM(x, log = TRUE, na.rm = TRUE)

Arguments

x matrix or data.frame containing real-time quantitative RT-PCR data
log logical: is data on log-scale
na.rm a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

The gene expression stability value M is defined as the average pairwise normalization factor; i.e., one needs to specify data from at least two genes. For more details see Vandesompele et al. (2002). Note this dispatches on a transposed expression matrix, not a qPCRBatch object since it is only called from within the selectHKs method.

Value

numeric vector with gene expression stability values

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>
stabMeasureRho

References


See Also
selectHKs

Examples
data(geNorm)
tissue <- as.factor(c(rep("BM", 9), rep("FIB", 20), rep("LEU", 13), rep("NB", 34), rep("POOL", 9)))
res.BM <- selectHKs(geNorm.qPCRBatch[,tissue == "BM"], method = "geNorm", Symbols = featureNames(geNorm.qPCRBatch), minNrHK = 2, log = FALSE)

stabMeasureRho

Gene expression stability value rho

Description
Computation of the gene expression stability value rho for real-time quantitative RT-PCR data. For more details we refer to Andersen et al. (2004).

Usage

stabMeasureRho(x,...)

## S4 method for signature 'x'

stabMeasureRho(x, group, log = TRUE, na.rm = TRUE, returnAll = FALSE)

Arguments

x

matrix containing real-time quantitative RT-PCR data, or qPCRBatch object

... 

Extra arguments, detailed below

group

grouping factor, either a factor vector or a phenoData column called "Group"

log

logical: is data on log-scale

na.rm

a logical value indicating whether NA values should be stripped before the computation proceeds.

returnAll

logical, return additional information.

Details
The gene expression stability value rho is computed. For more details see Andersen et al. (2004).
**Value**

numeric vector with gene expression stability values

If `returnAll == TRUE` a list with the following components is returned

- `rho` stability measure rho of Andersen et al. (2004)
- `d` used by `selectHKs`
- `v` used by `selectHKs`

**Author(s)**

Matthias Kohl <Matthias.Kohl@stamats.de>

**References**

Claus Lindbjerg Andersen, Jens Ledet Jensen and Torben Falck Orntoft (2004). Normalization of Real-Time Quantitative Reverse Transcription-PCR Data: A Model-Based Variance Estimation Approach to Identify Genes Suited for Normalization, Applied to Bladder and Colon Cancer Data Sets. **CANCER RESEARCH** 64, 5245-5250, August 1, 2004. [http://cancerres.aacrjournals.org/cgi/content/full/64/15/5245](http://cancerres.aacrjournals.org/cgi/content/full/64/15/5245)


**See Also**

`selectHKs`

**Examples**

```r
data(Colon)
Class <- pData(Colon)[, "Classification"]
res.Colon <- stabMeasureRho(Colon, group = Class, log = FALSE)

data(Bladder)
Grade <- pData(Bladder)[, "Grade"]
res.Bladder <- stabMeasureRho(Bladder, group = Grade, log = FALSE)
```
Index

*Topic classes
  ComputeNRQs, 9
  CqValues, 10

*Topic datasets
  Bladder, 3
  BladderRepro, 4
  Colon, 5
  geNorm, 15

*Topic data
  combineTechReps, 7
  combineTechRepsWithSD, 8
  deltaCt, 12
  deltaDeltaCt, 14
  geomMean, 17
  makeAllNAs, 18
  makeAllNewVal, 19
  replaceAboveCutOff, 20
  replaceNAs, 21
  selectHKs, 22
  stabMeasureM, 24
  stabMeasureRho, 25

*Topic package
  NormqPCR-package, 2

Bladder, 3
BladderRepro, 4

Colon, 5
combineTechReps, 7, 8
combineTechReps, qPCRBatch-method
  (combineTechReps), 7
combineTechRepsWithSD, 8
combineTechRepsWithSD, qPCRBatch-method
  (combineTechRepsWithSD), 8
ComputeNRQs, 9
ComputeNRQs, qPCRBatch-method
  (ComputeNRQs), 9
ComputeNRQs-methods (ComputeNRQs), 9
CqValues, 10
CqValues, CyclesSet-method (CqValues), 10
CqValues-methods (CqValues), 10
CyclesSet, 11

deltaCq (deltaCt), 12

deltaCt, 12
deltaCt, qPCRBatch-method (deltaCt), 12
deltaCt-methods (deltaCt), 12
deltaDeltaCq (deltaDeltaCt), 14
deltaDeltaCq, qPCRBatch-method
  (deltaDeltaCt), 14
deltaDeltaCq-methods (deltaDeltaCt), 14
deltaDeltaCt, 14
deltaDeltaCt, qPCRBatch-method
  (deltaDeltaCt), 14
deltaDeltaCt-methods (deltaDeltaCt), 14
efficiency, 11
geNorm, 15
geomMean, 17

KOD, 11

makeAllNAs, 18
makeAllNAs, qPCRBatch-method
  (makeAllNAs), 18
makeAllNewVal, 19
makeAllNewVal, qPCRBatch-method
  (makeAllNewVal), 19
modlist, 11
mselect, 11

NormqPCR (NormqPCR-package), 2
NormqPCR-package, 2

parkKOD, 11
pcrbatch, 10–12

qPCRBatch, 9

ratiobatch, 11
replaceAboveCutOff, 20
replaceAboveCutOff, qPCRBatch-method
  (replaceAboveCutOff), 20
replaceNAs, 21
replaceNAs, qPCRBatch-method
  (replaceNAs), 21

selectHKs, 22
INDEX

selectHKs, matrix-method (selectHKs), 22
selectHKs, qPCRBatch-method (selectHKs), 22
selectHKs-methods (selectHKs), 22
stabMeasureM, 24
stabMeasureRho, 25
stabMeasureRho, matrix-method (stabMeasureRho), 25
stabMeasureRho, qPCRBatch-method (stabMeasureRho), 25
stabMeasureRho, x-method (stabMeasureRho), 25
stabMeasureRho-methods (stabMeasureRho), 25