

# Package ‘NormExpression’

March 20, 2024

**Type** Package

**Title** Normalize Gene Expression Data using Evaluated Methods

**Version** 0.1.1

**Author** Zhenfeng Wu , Shan Gao

**Maintainer** Shan Gao <gao\_shan@mail.nankai.edu.cn>

**Description** It provides a framework and a fast and simple way for researchers to evaluate methods (particularly some data-driven methods or their own methods) and then select a best one for data normalization in the gene expression analysis, based on the consistency of metrics and the consistency of datasets.

Zhenfeng Wu, Weixiang Liu, Xiufeng Jin, Deshui Yu, Hua Wang, Gustavo Glusman, Max Robinson, Lin Liu, Jishou Ruan and Shan Gao (2018) <[doi:10.1101/251140](https://doi.org/10.1101/251140)>.

**License** Artistic-2.0

**Encoding** UTF-8

**LazyData** true

**NeedsCompilation** no

**Depends** R (>= 2.10)

**Repository** CRAN

**Date/Publication** 2024-03-20 04:40:02 UTC

## R topics documented:

bkRNA18 . . . . .	2
bkRNA18_factors . . . . .	3
calcFactorRLE . . . . .	4
calcFactorUpperquartile . . . . .	4
calcFactorWeighted . . . . .	5
change_colours . . . . .	6
CV2AUCVC . . . . .	7
estimateSizeFactorsForMatrix . . . . .	8
filteredZero . . . . .	9
findGenes . . . . .	9
gatherCors . . . . .	10

gatherCors4Matrices . . . . .	11
gatherCVs . . . . .	13
gatherCVs4Matrices . . . . .	14
gatherFactors . . . . .	15
getArea . . . . .	17
getAUCVC . . . . .	18
getAUCVCs . . . . .	19
getCor . . . . .	20
getCorMedians . . . . .	21
getCV . . . . .	21
getFactors . . . . .	22
getNormMatrix . . . . .	24
gridAUCVC . . . . .	24
gridAUCVC4Matrices . . . . .	26
identifyUbq . . . . .	27
identifyUbqRepeat . . . . .	28
nonzeroRatio2AUCVC . . . . .	29
optTU . . . . .	31
plotCors . . . . .	32
plotCVs . . . . .	33
plotHC . . . . .	35
scRNA663 . . . . .	35
scRNA663_factors . . . . .	54

<b>Index</b>	<b>55</b>
--------------	-----------

---

bkRNA18

*bkRNA18*


---

## Description

Please refer to the file `/inst/doc/readme.pdf`.

## Usage

```
data("bkRNA18")
```

## Format

A data frame with 57955 observations on the following 18 variables.

```
col13616_1 a numeric vector
col13816_3 a numeric vector
col13916_5 a numeric vector
col14016_7 a numeric vector
col14416_9 a numeric vector
col14516_11 a numeric vector
```

col4716\_13 a numeric vector  
col4816\_97 a numeric vector  
col5216\_17 a numeric vector  
col3616\_2 a numeric vector  
col3816\_4 a numeric vector  
col3916\_6 a numeric vector  
col4016\_8 a numeric vector  
col4416\_10 a numeric vector  
col4516\_12 a numeric vector  
col4716\_14 a numeric vector  
col4816\_98 a numeric vector  
col5216\_18 a numeric vector

**Examples**

```
data(bkRNA18)  
## maybe str(bkRNA18) ; plot(bkRNA18) ...
```

---

<code>bkRNA18_factors</code>	<i>bkRNA18_factors</i>
------------------------------	------------------------

---

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
data("bkRNA18_factors")
```

**Format**

A data frame with 18 observations on the following 13 variables.

HG7 a numeric vector  
ERCC a numeric vector  
TN a numeric vector  
TC a numeric vector  
CR a numeric vector  
NR a numeric vector  
DESeq a numeric vector  
UQ a numeric vector  
TMM a numeric vector  
TU a numeric vector  
NCS a numeric vector  
ES a numeric vector  
GAPDH a numeric vector

**Examples**

```
data(bkRNA18_factors)
## maybe str(bkRNA18_factors) ; plot(bkRNA18_factors) ...
```

---

calcFactorRLE	<i>calcFactorRLE</i>
---------------	----------------------

---

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
calcFactorRLE(data, p = p)
```

**Arguments**

<code>data</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>p</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, p = p)
{
  gm <- exp(rowMeans(.log(data), na.rm = TRUE))
  apply(data, 2, function(u) quantile((u/gm)[u != 0], na.rm = TRUE,
    p = p))
}
```

---

calcFactorUpperquartile	<i>calcFactorUpperquartile</i>
-------------------------	--------------------------------

---

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
calcFactorUpperquartile(data, lib.size, p = p)
```

**Arguments**

data	Please refer to the file /inst/doc/readme.pdf.
lib.size	Please refer to the file /inst/doc/readme.pdf.
p	Please refer to the file /inst/doc/readme.pdf.

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, lib.size, p = p)
{
  y <- t(t(data)/lib.size)
  f <- apply(y, 2, function(x) quantile(x[x != 0], p = p))
}
```

---

calcFactorWeighted     *calcFactorWeighted*

---

**Description**

Please refer to the file /inst/doc/readme.pdf.

**Usage**

```
calcFactorWeighted(obs, ref, libsize.obs, libsize.ref, logratioTrim,
sumTrim, doWeighting, Acutoff)
```

**Arguments**

obs	Please refer to the file /inst/doc/readme.pdf.
ref	Please refer to the file /inst/doc/readme.pdf.
libsize.obs	Please refer to the file /inst/doc/readme.pdf.
libsize.ref	Please refer to the file /inst/doc/readme.pdf.
logratioTrim	Please refer to the file /inst/doc/readme.pdf.
sumTrim	Please refer to the file /inst/doc/readme.pdf.
doWeighting	Please refer to the file /inst/doc/readme.pdf.
Acutoff	Please refer to the file /inst/doc/readme.pdf.

**Examples**

```

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (obs, ref, libsize.obs = NULL, libsize.ref = NULL, logratioTrim = 0.3,
        sumTrim = 0.05, doWeighting = TRUE, Acutoff = -1e+10)
{
  if (all(obs == ref))
    return(1)
  obs <- as.numeric(obs)
  ref <- as.numeric(ref)
  if (is.null(libsize.obs))
    n0 <- sum(obs)
  else n0 <- libsize.obs
  if (is.null(libsize.ref))
    nR <- sum(ref)
  else nR <- libsize.ref
  logR <- log2((obs/n0)/(ref/nR))
  absE <- (log2(obs/n0) + log2(ref/nR))/2
  v <- (n0 - obs)/n0/obs + (nR - ref)/nR/ref
  fin <- is.finite(logR) & is.finite(absE) & (absE > Acutoff)
  logR <- logR[fin]
  absE <- absE[fin]
  v <- v[fin]
  n <- length(logR)
  loL <- floor(n * logratioTrim) + 1
  hiL <- n + 1 - loL
  loS <- floor(n * sumTrim) + 1
  hiS <- n + 1 - loS
  keep <- (rank(logR) >= loL & rank(logR) <= hiL) & (rank(absE) >=
    loS & rank(absE) <= hiS)
  if (doWeighting) {
    2^(sum(logR[keep])/v[keep], na.rm = TRUE)/sum(1/v[keep],
      na.rm = TRUE))
  }
  else {
    2^(mean(logR[keep], na.rm = TRUE))
  }
}

```

---

change\_colours

*change\_colours*


---

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
change_colours(p, palette, type)
```

**Arguments**

p	Please refer to the file /inst/doc/readme.pdf.
palette	Please refer to the file /inst/doc/readme.pdf.
type	Please refer to the file /inst/doc/readme.pdf.

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (p, palette, type)
{
  n <- nlevels(p$data[[deparse(p$mapping$group)]])
  tryCatch(as.character(palette), error = function(e) stop("be vector", call. = FALSE))
  if (n > length(palette))
    stop("Not enough colours in palette.")
  if (missing(type))
    type <- grep("colour|fill", names(p$layers[[1]]$mapping),
                value = TRUE)[1]
  pal <- function(n) palette[seq_len(n)]
  p + discrete_scale(type, "foo", pal)
}
```

---

CV2AUCVC

---

CV2AUCVC

---

**Description**

Please refer to the file /inst/doc/readme.pdf.

**Usage**

```
CV2AUCVC(data, cvResolution = 0.005)
```

**Arguments**

data	Please refer to the file /inst/doc/readme.pdf.
cvResolution	Please refer to the file /inst/doc/readme.pdf.

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, cvResolution = 0.005)
{
  cv_cutoff <- NULL
  uniform_genes_counts <- NULL
  for (i in seq(0, 1, cvResolution)) {
    cv_cutoff <- c(cv_cutoff, i)
    gene_number <- length(which(data <= i))
    uniform_genes_counts <- c(uniform_genes_counts, gene_number)
  }
  getArea(cv_cutoff, uniform_genes_counts)
}
```

---

```
estimateSizeFactorsForMatrix
      estimateSizeFactorsForMatrix
```

---

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
estimateSizeFactorsForMatrix(data, p = p)
```

**Arguments**

<code>data</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>p</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, p = p)
{
  loggeomeans <- rowMeans(.log(data), na.rm = TRUE)
  apply(data, 2, function(cnts) exp(quantile(.log(cnts) - loggeomeans,
    na.rm = TRUE, p = p)))
}
```



---

filteredZero	<i>filteredZero</i>
--------------	---------------------

---

**Description**

Please refer to the file /inst/doc/readme.pdf.

**Usage**

```
filteredZero(data, nonzeroRatio)
```

**Arguments**

data	Please refer to the file /inst/doc/readme.pdf.
nonzeroRatio	Please refer to the file /inst/doc/readme.pdf.

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, nonzeroRatio)
{
  nonzeroCount <- apply(data, 1, function(x) length(which(x !=
    0)))
  geneIndex <- which(nonzeroCount >= ncol(data) * nonzeroRatio)
  return(geneIndex)
}
```

---

findGenes	<i>findGenes</i>
-----------	------------------

---

**Description**

Please refer to the file /inst/doc/readme.pdf.

**Usage**

```
findGenes(g, qlower = NULL, qupper = NULL, pre_ratio = NULL)
```

**Arguments**

g	Please refer to the file /inst/doc/readme.pdf.
qlower	Please refer to the file /inst/doc/readme.pdf.
qupper	Please refer to the file /inst/doc/readme.pdf.
pre_ratio	Please refer to the file /inst/doc/readme.pdf.

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function(g, qlower = NULL, qupper = NULL, pre_ratio = NULL)
{
  gene_name <- rownames(g)
  g <- unlist(g)
  seen <- which(g >= qlower & g <= qupper)
  counts <- length(seen)
  if (counts >= pre_ratio * length(g)) {
    gene_name
  }
}
```

gatherCors

*gatherCors***Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
gatherCors(data, cor_method = c("spearman", "pearson", "kendall"),
HG7 = NULL, ERCC = NULL, TN = NULL, TC = NULL, CR = NULL, NR = NULL,
DESeq = NULL, UQ = NULL, TMM = NULL, TU = NULL, GAPDH = NULL,
pre_ratio = 0.5, lower_trim = 0.05, upper_trim = 0.65, rounds = 1e+06)
```

**Arguments**

data	Please refer to the file <code>/inst/doc/readme.pdf</code> .
cor_method	Please refer to the file <code>/inst/doc/readme.pdf</code> .
HG7	Please refer to the file <code>/inst/doc/readme.pdf</code> .
ERCC	Please refer to the file <code>/inst/doc/readme.pdf</code> .
TN	Please refer to the file <code>/inst/doc/readme.pdf</code> .
TC	Please refer to the file <code>/inst/doc/readme.pdf</code> .
CR	Please refer to the file <code>/inst/doc/readme.pdf</code> .
NR	Please refer to the file <code>/inst/doc/readme.pdf</code> .
DESeq	Please refer to the file <code>/inst/doc/readme.pdf</code> .
UQ	Please refer to the file <code>/inst/doc/readme.pdf</code> .
TMM	Please refer to the file <code>/inst/doc/readme.pdf</code> .

TU	Please refer to the file /inst/doc/readme.pdf.
GAPDH	Please refer to the file /inst/doc/readme.pdf.
pre_ratio	Please refer to the file /inst/doc/readme.pdf.
lower_trim	Please refer to the file /inst/doc/readme.pdf.
upper_trim	Please refer to the file /inst/doc/readme.pdf.
rounds	Please refer to the file /inst/doc/readme.pdf.

### Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, cor_method = c("spearman", "pearson", "kendall"),
  HG7 = NULL, ERCC = NULL, TN = NULL, TC = NULL, CR = NULL,
  NR = NULL, DESeq = NULL, UQ = NULL, TMM = NULL, TU = NULL,
  GAPDH = NULL, pre_ratio = 0.5, lower_trim = 0.05, upper_trim = 0.65,
  rounds = 1e+06)
{
  methodsList <- list(HG7 = HG7, ERCC = ERCC, TN = TN, TC = TC,
    CR = CR, NR = NR, DESeq = DESeq, UQ = UQ, TMM = TMM,
    TU = TU, GAPDH = GAPDH)
  specifiedMethods <- methodsList[!unlist(lapply(methodsList,
    is.null))]
  numMethod <- length(specifiedMethods)
  method_range <- seq(1, numMethod, 1)
  ubq_genes <- identifyUbq(data, pre_ratio = pre_ratio, lower_trim = lower_trim,
    upper_trim = upper_trim, min_ubq = 100)
  cor_value_method <- NULL
  for (j in method_range) {
    norm.matrix <- getNormMatrix(data, specifiedMethods[[j]])
    dataUse2Cor <- norm.matrix[ubq_genes, ]
    cor.result <- getCor(dataUse2Cor, method = cor_method,
      rounds = rounds)
    cor_vm <- cbind(cor.result, rep(names(specifiedMethods)[j],
      times = round(rounds)))
    cor_value_method <- rbind(cor_value_method, cor_vm)
  }
  colnames(cor_value_method) <- c("Value", "Methods")
  return(cor_value_method)
}
```

---

gatherCors4Matrices    *gatherCors4Matrices*

---

### Description

Please refer to the file /inst/doc/readme.pdf.

**Usage**

```
gatherCors4Matrices(..., raw_matrix, cor_method = c("spearman", "pearson", "kendall"),
pre_ratio = 0.5, lower_trim = 0.05, upper_trim = 0.65, rounds = 1e+06)
```

**Arguments**

...	Please refer to the file /inst/doc/readme.pdf.
raw_matrix	Please refer to the file /inst/doc/readme.pdf.
cor_method	Please refer to the file /inst/doc/readme.pdf.
pre_ratio	Please refer to the file /inst/doc/readme.pdf.
lower_trim	Please refer to the file /inst/doc/readme.pdf.
upper_trim	Please refer to the file /inst/doc/readme.pdf.
rounds	Please refer to the file /inst/doc/readme.pdf.

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (... , raw_matrix, cor_method = c("spearman", "pearson",
      "kendall"), pre_ratio = 0.5, lower_trim = 0.05, upper_trim = 0.65,
      rounds = 1e+06)
{
  matrices <- list(...)
  numMethod <- length(matrices)
  method_range <- seq(1, numMethod, 1)
  ubq_genes <- identifyUbq(raw_matrix, pre_ratio = pre_ratio,
    lower_trim = lower_trim, upper_trim = upper_trim, min_ubq = 100)
  cor_value_method <- NULL
  for (j in method_range) {
    dataUse2Cor <- matrices[[j]][ubq_genes, ]
    cor.result <- getCor(dataUse2Cor, method = cor_method,
      rounds = rounds)
    cor_vm <- cbind(cor.result, rep(names(matrices)[j], times = round(rounds)))
    cor_value_method <- rbind(cor_value_method, cor_vm)
  }
  colnames(cor_value_method) <- c("Value", "Methods")
  return(cor_value_method)
}
```

gatherCVs

*gatherCVs***Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
gatherCVs(data, nonzeroRatio, HG7, ERCC, TN, TC, CR, NR,
           DESeq, UQ, TMM, TU, GAPDH, cvNorm, cvResolution)
```

**Arguments**

<code>data</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>nonzeroRatio</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>HG7</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>ERCC</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>TN</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>TC</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>CR</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>NR</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>DESeq</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>UQ</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>TMM</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>TU</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>GAPDH</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>cvNorm</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>cvResolution</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, nonzeroRatio = NULL, HG7 = NULL, ERCC = NULL,
         TN = NULL, TC = NULL, CR = NULL, NR = NULL, DESeq = NULL,
         UQ = NULL, TMM = NULL, TU = NULL, GAPDH = NULL, cvNorm = TRUE,
         cvResolution = 0.005)
{
  if (is.null(nonzeroRatio)) {
    stop("Please provide nonzeroRatio!")
  }
}
```

```

}
methodsList <- list(HG7 = HG7, ERCC = ERCC, TN = TN, TC = TC,
  CR = CR, NR = NR, DESeq = DESeq, UQ = UQ, TMM = TMM,
  TU = TU, GAPDH = GAPDH)
specifiedMethods <- methodsList[!unlist(lapply(methodsList,
  is.null))]
numMethod <- length(specifiedMethods)
method_range_tmp <- seq(1, numMethod, 1)
cv_range_tmp <- seq(0, 1, cvResolution)
method_range_times <- length(cv_range_tmp)
cv_range_times <- length(method_range_tmp)
method_range <- rep(method_range_tmp, each = round(method_range_times))
cv_range <- rep(cv_range_tmp, times = round(cv_range_times))
nozeroIndex <- filteredZero(data, nonzeroRatio = nonzeroRatio)
for (j in method_range_tmp) {
  norm.matrix <- getNormMatrix(data, specifiedMethods[[j]])
  dataUse2CV <- norm.matrix[nozeroIndex, ]
  cv.result <- getCV(dataUse2CV, cvNorm = cvNorm)
  assign(paste(names(specifiedMethods)[j], ".cv", sep = ""),
    cv.result)
}
cv_uniform <- NULL
cv_uniform_all <- mapply(function(i, j) {
  cv.result <- paste(names(specifiedMethods)[j], ".cv",
    sep = "")
  gene_number <- length(which(get(cv.result) <= i))
  cv_uniform_row <- c(i, gene_number, names(specifiedMethods)[j])
  rbind(cv_uniform, cv_uniform_row)
}, cv_range, method_range)
cv_uniform_all <- t(cv_uniform_all)
colnames(cv_uniform_all) <- c("Cutoff", "Counts", "Methods")
return(cv_uniform_all)
}

```

---

gatherCVs4Matrices      *gatherCVs4Matrices*

---

### Description

Please refer to the file `/inst/doc/readme.pdf`.

### Usage

```
gatherCVs4Matrices(..., raw_matrix, nonzeroRatio, cvNorm, cvResolution = 0.005)
```

### Arguments

...                      Please refer to the file `/inst/doc/readme.pdf`.

raw\_matrix              Please refer to the file `/inst/doc/readme.pdf`.

nonzeroRatio    Please refer to the file /inst/doc/readme.pdf.  
 cvNorm            Please refer to the file /inst/doc/readme.pdf.  
 cvResolution    Please refer to the file /inst/doc/readme.pdf.

## Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (... , raw_matrix, nonzeroRatio = NULL, cvNorm = TRUE,
          cvResolution = 0.005)
{
  if (is.null(nonzeroRatio)) {
    stop("Please provide nonzeroRatio!")
  }
  matrices <- list(...)
  matrices_name <- names(matrices)
  numMethod <- length(matrices)
  method_range_tmp <- seq(1, numMethod, 1)
  cv_range_tmp <- seq(0, 1, cvResolution)
  method_range_times <- length(cv_range_tmp)
  cv_range_times <- length(method_range_tmp)
  method_range <- rep(method_range_tmp, each = round(method_range_times))
  cv_range <- rep(cv_range_tmp, times = round(cv_range_times))
  nonzeroIndex <- filteredZero(raw_matrix, nonzeroRatio = nonzeroRatio)
  for (j in method_range_tmp) {
    dataUse2CV <- matrices[[j]][nonzeroIndex, ]
    cv.result <- getCV(dataUse2CV, cvNorm = cvNorm)
    assign(paste(matrices_name[j], ".cv", sep = ""), cv.result)
  }
  cv_uniform <- NULL
  cv_uniform_all <- mapply(function(i, j) {
    cv.result <- paste(matrices_name[j], ".cv", sep = "")
    gene_number <- length(which(get(cv.result) <= i))
    cv_uniform_row <- c(i, gene_number, matrices_name[j])
    rbind(cv_uniform, cv_uniform_row)
  }, cv_range, method_range)
  cv_uniform_all <- t(cv_uniform_all)
  colnames(cv_uniform_all) <- c("Cutoff", "Counts", "Methods")
  return(cv_uniform_all)
}
```

---

gatherFactors

gatherFactors

---

## Description

Please refer to the file /inst/doc/readme.pdf.

**Usage**

```
gatherFactors(data,
  methods = c("HG7", "ERCC", "TN", "TC", "CR", "NR", "DESeq", "UQ", "TMM", "TU"),
  HG7.size = NULL, ERCC.size = NULL, TN.size = NULL, TC.size = NULL,
  CR.size = NULL, NR.size = NULL, pre_ratio = 0.5,
  lower_trim = 0.05, upper_trim = 0.65, min_ubq = 100)
```

**Arguments**

data	Please refer to the file /inst/doc/readme.pdf.
methods	Please refer to the file /inst/doc/readme.pdf.
HG7.size	Please refer to the file /inst/doc/readme.pdf.
ERCC.size	Please refer to the file /inst/doc/readme.pdf.
TN.size	Please refer to the file /inst/doc/readme.pdf.
TC.size	Please refer to the file /inst/doc/readme.pdf.
CR.size	Please refer to the file /inst/doc/readme.pdf.
NR.size	Please refer to the file /inst/doc/readme.pdf.
pre_ratio	Please refer to the file /inst/doc/readme.pdf.
lower_trim	Please refer to the file /inst/doc/readme.pdf.
upper_trim	Please refer to the file /inst/doc/readme.pdf.
min_ubq	Please refer to the file /inst/doc/readme.pdf.

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, methods = c("HG7", "ERCC", "TN", "TC", "CR",
  "NR", "DESeq", "UQ", "TMM", "TU"), HG7.size = NULL, ERCC.size = NULL,
  TN.size = NULL, TC.size = NULL, CR.size = NULL, NR.size = NULL,
  pre_ratio = 0.5, lower_trim = 0.05, upper_trim = 0.65, min_ubq = 100)
{
  method1 <- as.list(methods)
  numMethod <- length(method1)
  method_range <- seq(1, numMethod, 1)
  for (i in method_range) {
    if (method1[[i]] == "HG7" || method1[[i]] == "ERCC" ||
      method1[[i]] == "TN" || method1[[i]] == "TC" || method1[[i]] ==
      "CR" || method1[[i]] == "NR") {
      size.name <- paste(method1[[i]], ".size", sep = "")
      out.name1 <- paste(method1[[i]], ".factors", sep = "")
      if (is.null(size.name)) {
        stop("Please provide", size.name, "!")
      }
    }
    else {
```



```

        assign(out.name1, getFactors(data, method = "sizefactor",
        lib.size = get(size.name)))
    }
}
if (method1[[i]] == "DESeq" || method1[[i]] == "RLE" ||
    method1[[i]] == "UQ" || method1[[i]] == "TMM") {
    out.name2 <- paste(method1[[i]], ".factors", sep = "")
    assign(out.name2, getFactors(data, method = method1[[i]]))
}
if (method1[[i]] == "TU") {
    TU.factors <- getFactors(data, method = "TU", pre_ratio = pre_ratio,
        lower_trim = lower_trim, upper_trim = upper_trim,
        min_ubq = min_ubq)
}
}
factors.list <- NULL
for (m in methods) {
    m.factors <- paste(m, ".factors", sep = "")
    factors.list <- c(factors.list, m.factors)
}
factors.result <- NULL
for (i in method_range) {
    factors.result <- cbind(factors.result, get(factors.list[i]))
}
colnames(factors.result) <- methods
return(factors.result)
}

```

---

getArea

*getArea*


---

## Description

Please refer to the file `/inst/doc/readme.pdf`.

## Usage

```
getArea(x, y)
```

## Arguments

x	Please refer to the file <code>/inst/doc/readme.pdf</code> .
y	Please refer to the file <code>/inst/doc/readme.pdf</code> .

## Examples

```

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

```

```

## The function is currently defined as
function (x, y)
{
  x <- x/max(x)
  y <- y/max(y)
  if (!(is.numeric(x) || is.complex(x)) || !(is.numeric(y) ||
    is.complex(y))) {
    stop("Arguments 'x' and 'y' must be real or complex vectors.")
  }
  if (length(x) != length(y)) {
    stop("The length of two input vectors should be equal!")
  }
  m <- length(x)
  n <- 2 * m
  xp <- c(x, x[m:1])
  yp <- c(numeric(m), y[m:1])
  p1 <- sum(xp[1:(n - 1)] * yp[2:n]) + xp[n] * yp[1]
  p2 <- sum(xp[2:n] * yp[1:(n - 1)]) + xp[1] * yp[n]
  return(0.5 * (p1 - p2))
}

```

---

getAUCVC

*getAUCVC*


---

## Description

Please refer to the file `/inst/doc/readme.pdf`.

## Usage

```
getAUCVC(data, nonzeroRatio = NULL, cvNorm = TRUE, cvResolution = 0.005)
```

## Arguments

<code>data</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>nonzeroRatio</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>cvNorm</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>cvResolution</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

## Examples

```

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, nonzeroRatio = NULL, cvNorm = TRUE, cvResolution = 0.005)
{
  nonzeroIndex <- filteredZero(data, nonzeroRatio = nonzeroRatio)

```

```

dataUse2CV <- data[nozeroIndex, ]
cv.result <- getCV(dataUse2CV, cvNorm = cvNorm)
CV2AUCVC(cv.result, cvResolution = cvResolution)
}

```

---

getAUCVCs

*getAUCVCs*


---

### Description

Please refer to the file `/inst/doc/readme.pdf`.

### Usage

```
getAUCVCs(..., nonzeroRatio = NULL, cvNorm = TRUE, cvResolution = 0.005)
```

### Arguments

<code>...</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>nonzeroRatio</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>cvNorm</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>cvResolution</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

### Examples

```

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (... , nonzeroRatio = NULL, cvNorm = TRUE, cvResolution = 0.005)
{
  matrices <- list(...)
  numMethod <- length(matrices)
  method_range <- seq(1, numMethod, 1)
  result <- NULL
  for (i in method_range) {
    AUCVC.result <- getAUCVC(matrices[[i]], nonzeroRatio = nonzeroRatio,
      cvNorm = cvNorm, cvResolution = cvResolution)
    result <- c(result, AUCVC.result)
    names(result)[i] <- names(matrices)[i]
  }
  sorted_AUCVCs <- sort(result, decreasing = TRUE)
  return(sorted_AUCVCs)
}

```

getCor

*getCor*

---

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
getCor(data, method = c("spearman", "pearson", "kendall"), rounds = 1e+06)
```

**Arguments**

<code>data</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>method</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>rounds</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, method = c("spearman", "pearson", "kendall"),
         rounds = 1e+06)
{
  sp_result <- NULL
  method <- match.arg(method)
  for (i in 1:rounds) {
    rg1 <- sample(1:nrow(data), size = 1)
    rg2 <- sample(1:nrow(data), size = 1)
    while (rg1 == rg2) {
      rg2 <- sample(1:nrow(data), size = 1)
    }
    gene1 <- unlist(data[rg1, ])
    gene2 <- unlist(data[rg2, ])
    sp_value <- cor(gene1, gene2, method = method)
    sp_result <- c(sp_result, sp_value)
  }
  return(sp_result)
}
```

---

getCorMedians	<i>getCorMedians</i>
---------------	----------------------

---

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
getCorMedians(data)
```

**Arguments**

data	Please refer to the file <code>/inst/doc/readme.pdf</code> .
------	--

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data)
{
  if (!is.data.frame(data))
    data <- data.frame(data)
  if (is.factor(data$Value))
    data$Value <- as.numeric(as.character(data$Value))
  sorted_result <- sort(tapply(data$Value, data$Methods, median),
    decreasing = FALSE)
  return(sorted_result)
}
```

---

getCV	<i>getCV</i>
-------	--------------

---

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
getCV(data, cvNorm = TRUE)
```

**Arguments**

data	Please refer to the file <code>/inst/doc/readme.pdf</code> .
cvNorm	Please refer to the file <code>/inst/doc/readme.pdf</code> .

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, cvNorm = TRUE)
{
  if (!is.matrix(data))
    data <- as.matrix(data)
  if (cvNorm) {
    rawCV <- apply(data, 1, function(x) {
      sd(log2(x[x != 0]))/mean(log2(x[x != 0]))
    })
    (rawCV - min(rawCV))/(max(rawCV) - min(rawCV))
  }
  else {
    apply(data, 1, function(x) {
      sd(x)/mean(x)
    })
  }
}
```

---

getFactors

*getFactors*


---

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
getFactors(data, method = c("sizefactor", "DESeq", "RLE", "UQ", "TMM", "TU"),
  lib.size = NULL, pre_ratio = 0.5, lower_trim = 0.05, upper_trim = 0.65, min_ubq = 100)
```

**Arguments**

<code>data</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>method</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>lib.size</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>pre_ratio</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>lower_trim</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>upper_trim</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>min_ubq</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

## Examples

```

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, method = c("sizefactor", "DESeq", "RLE", "UQ",
  "TMM", "TU"), lib.size = NULL, pre_ratio = 0.5, lower_trim = 0.05,
  upper_trim = 0.65, min_ubq = 100)
{
  if (!is.matrix(data))
    data <- as.matrix(data)
  if (any(is.na(data)))
    stop("NA counts not permitted")
  if (is.null(lib.size))
    libsize <- colSums(data)
  else libsize <- lib.size
  if (any(is.na(libsize)))
    stop("NA libsizes not permitted")
  method <- match.arg(method)
  i <- apply(data <= 0, 1, all)
  if (any(i))
    data <- data[!i, , drop = FALSE]
  f <- switch(method, sizefactor = 1e+06/libsize, DESeq = 1/estimateSizeFactorsForMatrix(data,
    p = 0.5), RLE = calcFactorRLE(data, p = 0.5)/libsize,
    UQ = calcFactorUpperquartile(data, lib.size = libsize,
    p = 0.75), TMM = {
    fq <- calcFactorUpperquartile(data = data, lib.size = libsize,
    p = 0.75)
    refColumn <- which.min(abs(fq - mean(fq)))
    if (length(refColumn) == 0 | refColumn < 1 | refColumn >
    ncol(data)) refColumn <- 1
    f <- rep(NA, ncol(data))
    for (i in 1:ncol(data)) {
      f[i] <- calcFactorWeighted(obs = data[, i], ref = data[,
      refColumn], libsize.obs = libsize[i], libsize.ref = libsize[refColumn],
      logratioTrim = 0.3, sumTrim = 0.05, doWeighting = TRUE,
      Acutoff = -1e+10)
    }
    f
  }, TU = {
    if (!is.data.frame(data)) data <- data.frame(data)
    ubq_genes <- identifyUbq(data, lower_trim = lower_trim,
    upper_trim = upper_trim, pre_ratio = pre_ratio,
    min_ubq = min_ubq)
    ubq_sums <- colSums(data[ubq_genes, ])
    mean(ubq_sums)/ubq_sums
  }, )
  if (method == "RLE" || method == "UQ" || method == "TMM") {
    f <- 1e+06/libsize/f
  }
  norm.factors <- f/exp(mean(base::log(f)))
}

```

```

    round(norm.factors, digits = 5)
  }

```

---

getNormMatrix	<i>getNormMatrix</i>
---------------	----------------------

---

### Description

Please refer to the file /inst/doc/readme.pdf.

### Usage

```
getNormMatrix(data, norm.factors)
```

### Arguments

data	Please refer to the file /inst/doc/readme.pdf.
norm.factors	Please refer to the file /inst/doc/readme.pdf.

### Examples

```

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, norm.factors)
{
  data * matrix(rep(norm.factors, dim(data)[1]), nrow = dim(data)[1],
               ncol = length(norm.factors), byrow = T)
}

```

---

gridAUCVC	<i>gridAUCVC</i>
-----------	------------------

---

### Description

Please refer to the file /inst/doc/readme.pdf.

### Usage

```

gridAUCVC(data, dataType = c("bk", "sc"), HG7 = NULL, ERCC = NULL, TN = NULL,
TC = NULL, CR = NULL, NR = NULL, DESeq = NULL, UQ = NULL, TMM = NULL, TU = 0,
GAPDH = NULL, nonzeroRatios = c(0.7, 0.8, 0.9, 1), cvNorm = TRUE, cvResolution = 0.005)

```



**Arguments**

data	Please refer to the file /inst/doc/readme.pdf.
dataType	Please refer to the file /inst/doc/readme.pdf.
HG7	Please refer to the file /inst/doc/readme.pdf.
ERCC	Please refer to the file /inst/doc/readme.pdf.
TN	Please refer to the file /inst/doc/readme.pdf.
TC	Please refer to the file /inst/doc/readme.pdf.
CR	Please refer to the file /inst/doc/readme.pdf.
NR	Please refer to the file /inst/doc/readme.pdf.
DESeq	Please refer to the file /inst/doc/readme.pdf.
UQ	Please refer to the file /inst/doc/readme.pdf.
TMM	Please refer to the file /inst/doc/readme.pdf.
TU	Please refer to the file /inst/doc/readme.pdf.
GAPDH	Please refer to the file /inst/doc/readme.pdf.
nonzeroRatios	Please refer to the file /inst/doc/readme.pdf.
cvNorm	Please refer to the file /inst/doc/readme.pdf.
cvResolution	Please refer to the file /inst/doc/readme.pdf.

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, dataType = c("bk", "sc"), HG7 = NULL, ERCC = NULL,
        TN = NULL, TC = NULL, CR = NULL, NR = NULL, DESeq = NULL,
        UQ = NULL, TMM = NULL, TU = 0, GAPDH = NULL, nonzeroRatios = c(0.7,
        0.8, 0.9, 1), cvNorm = TRUE, cvResolution = 0.005)
{
  grid_result <- NULL
  if (length(TU) == 1 && TU == 1) {
    colnames_paraMatrix <- c("nonzeroRatio", "pre_ratio",
        "lower_trim", "upper_trim")
    write.table(t(as.matrix(colnames_paraMatrix)), file = "bestPara.txt",
        sep = "\t", row.names = FALSE, col.names = FALSE)
  }
  for (i in nonzeroRatios) {
    if (dataType == "sc") {
      if ((ncol(data) * i) <= 100) {
        cat("nonzeroRatio:", i, " is too small!\n")
        stop("We suggest that the minimal counts of
            nonzero samples should be greater than 100!")
      }
    }
  }
  result <- nonzeroRatio2AUCVC(data = data, dataType = dataType,
```

```

HG7 = HG7, ERCC = ERCC, TN = TN, TC = TC, CR = CR,
NR = NR, DESeq = DESeq, UQ = UQ, TMM = TMM, TU = TU,
GAPDH = GAPDH, nonzeroRatio = i, cvNorm = cvNorm,
cvResolution = cvResolution)
nonzeroM <- matrix(i, 1, 1, TRUE)
colnames(nonzeroM) <- "NonzeroRatio"
grid_record <- cbind(nonzeroM, result)
grid_result <- rbind(grid_result, grid_record)
}
return(grid_result)
}

```

---

gridAUCVC4Matrices      *gridAUCVC4Matrices*

---

### Description

Please refer to the file `/inst/doc/readme.pdf`.

### Usage

```
gridAUCVC4Matrices(..., nonzeroRatios = NULL, cvNorm = TRUE, cvResolution = 0.005)
```

### Arguments

<code>...</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>nonzeroRatios</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>cvNorm</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>cvResolution</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

### Examples

```

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (... , nonzeroRatios = NULL, cvNorm = TRUE, cvResolution = 0.005)
{
  if (is.null(nonzeroRatios)) {
    stop("Please provide nonzeroRatios!")
  }
  matrices <- list(...)
  numMethod <- length(matrices)
  grid_result <- NULL
  for (i in nonzeroRatios) {
    result.sorted <- getAUCVCs(..., nonzeroRatio = i, cvNorm = cvNorm,
      cvResolution = cvResolution)
    grid_record <- c(i, result.sorted)
  }
}

```

```

        names(grid_record)[1] <- "NonzeroRatio"
        grid_result <- c(grid_result, names(grid_record), grid_record)
    }
    grid_result2 <- matrix(grid_result, ncol = numMethod + 1,
        byrow = TRUE)
    return(grid_result2)
}

```

---

identifyUbq

*identifyUbq*


---

### Description

Please refer to the file `/inst/doc/readme.pdf`.

### Usage

```
identifyUbq(data, pre_ratio = 0.5, lower_trim = 0.05, upper_trim = 0.65, min_ubq = 100)
```

### Arguments

<code>data</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>pre_ratio</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>lower_trim</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>upper_trim</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>min_ubq</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

### Examples

```

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, pre_ratio = 0.5, lower_trim = 0.05, upper_trim = 0.65,
    min_ubq = 100)
{
  qlower <- apply(data, 2, function(x) quantile(x[x != 0],
    p = lower_trim))
  qupper <- apply(data, 2, function(x) quantile(x[x != 0],
    p = upper_trim))
  ubq_genes <- NULL
  for (i in 1:nrow(data)) {
    genes_finded <- findGenes(data[i, ], qlower = qlower,
        qupper = qupper, pre_ratio = pre_ratio)
    ubq_genes <- c(ubq_genes, genes_finded)
  }
  if (length(ubq_genes) < min_ubq) {

```

```

    cat("Parameters range", lower_trim, "-", upper_trim,
        "...identified too few ubiquitous genes (", length(ubq_genes),
        "), trying range 5-95 instead", "\n")
    ubq_genes <- identifyUbqRepeat(data, pre_ratioC = pre_ratio,
        lower_trimC = 0.05, upper_trimC = 0.95)
  }
  return(ubq_genes)
}

```

---

identifyUbqRepeat      *identifyUbqRepeat*

---

## Description

Please refer to the file `/inst/doc/readme.pdf`.

## Usage

```
identifyUbqRepeat(data, pre_ratioC = NULL, lower_trimC = NULL, upper_trimC = NULL)
```

## Arguments

<code>data</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>pre_ratioC</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>lower_trimC</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>upper_trimC</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

## Examples

```

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, pre_ratioC = NULL, lower_trimC = NULL, upper_trimC = NULL)
{
  qlower <- apply(data, 2, function(x) quantile(x[x != 0],
    p = lower_trimC))
  qupper <- apply(data, 2, function(x) quantile(x[x != 0],
    p = upper_trimC))
  ubq_genes <- NULL
  for (i in 1:nrow(data)) {
    genes_finded <- findGenes(data[i, ], qlower = qlower,
      qupper = qupper, pre_ratio = pre_ratioC)
    ubq_genes <- c(ubq_genes, genes_finded)
  }
  return(ubq_genes)
}

```

---

nonzeroRatio2AUCVC      *nonzeroRatio2AUCVC*

---

### Description

Please refer to the file `/inst/doc/readme.pdf`.

### Usage

```
nonzeroRatio2AUCVC(data, dataType = c("bk", "sc"),
  HG7 = NULL, ERCC = NULL, TN = NULL, TC = NULL, CR = NULL, NR = NULL, DESeq = NULL,
  UQ = NULL, TMM = NULL, TU = 0, GAPDH = NULL, nonzeroRatio = NULL, cvNorm = TRUE,
  cvResolution = 0.005)
```

### Arguments

<code>data</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>dataType</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>HG7</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>ERCC</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>TN</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>TC</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>CR</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>NR</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>DESeq</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>UQ</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>TMM</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>TU</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>GAPDH</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>nonzeroRatio</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>cvNorm</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>cvResolution</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

### Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, dataType = c("bk", "sc"), HG7 = NULL, ERCC = NULL,
  TN = NULL, TC = NULL, CR = NULL, NR = NULL, DESeq = NULL,
  UQ = NULL, TMM = NULL, TU = 0, GAPDH = NULL, nonzeroRatio = NULL,
```

```

cvNorm = TRUE, cvResolution = 0.005)
{
  nonzeroIndex <- filteredZero(data, nonzeroRatio = nonzeroRatio)
  methodsList <- list(HG7 = HG7, ERCC = ERCC, TN = TN, TC = TC,
    CR = CR, NR = NR, DESeq = DESeq, UQ = UQ, TMM = TMM,
    TU = TU, GAPDH = GAPDH)
  specifiedMethods <- methodsList[!unlist(lapply(methodsList,
    is.null))]
  if (length(TU) == 1 && TU == 0) {
    specifiedMethods$TU <- NULL
  }
  if (length(TU) == 1 && TU == 1) {
    if (dataType == "bk") {
      optimalPara <- optTU(data, nonzeroRatio = nonzeroRatio,
        pre_ratio_range = c(1, 1), prResolution = 0.1,
        lower_range = c(0.05, 0.4), upper_range = c(0.6,
          0.95), qResolution = 0.05, min_ubq = 1000,
        cvNorm = cvNorm, cvResolution = cvResolution)
    }
    else {
      optimalPara <- optTU(data, nonzeroRatio = nonzeroRatio,
        pre_ratio_range = c(0.2, 0.6), prResolution = 0.1,
        lower_range = c(0.05, 0.4), upper_range = c(0.6,
          0.95), qResolution = 0.05, min_ubq = 100, cvNorm = cvNorm,
        cvResolution = cvResolution)
    }
    optimalPara <- as.matrix(optimalPara)
    lower_trim <- optimalPara["lower", 1]
    upper_trim <- optimalPara["upper", 1]
    pre_ratio <- optimalPara["ratio", 1]
    para <- c(nonzeroRatio, pre_ratio, lower_trim, upper_trim)
    names(para)[1] <- "nonzeroRatio"
    paraMatrix <- t(as.matrix(para))
    write.table(paraMatrix, file = "bestPara.txt", sep = "\t",
      row.names = FALSE, col.names = FALSE, append = TRUE)
    TU.factors <- getFactors(data, method = "TU", lower_trim = lower_trim,
      upper_trim = upper_trim, pre_ratio = pre_ratio, min_ubq = 100)
    norm.matrix <- getNormMatrix(data, TU.factors)
    dataUse2CV <- norm.matrix[nonzeroIndex, ]
    cv.result <- getCV(dataUse2CV, cvNorm = cvNorm)
    TU.AUCVC <- CV2AUCVC(cv.result, cvResolution = cvResolution)
    specifiedMethods$TU <- NULL
  }
  numMethod <- length(specifiedMethods)
  if (numMethod >= 1) {
    method_range <- seq(1, numMethod, 1)
    for (i in method_range) {
      norm.matrix <- getNormMatrix(data, specifiedMethods[[i]])
      dataUse2CV <- norm.matrix[nonzeroIndex, ]
      cv.result <- getCV(dataUse2CV, cvNorm = cvNorm)
      assign(names(specifiedMethods)[i], CV2AUCVC(cv.result,
        cvResolution = cvResolution))
    }
  }
}

```

```

AUCVC.result <- NULL
for (i in method_range) {
  AUCVC.result <- cbind(AUCVC.result, get(names(specifiedMethods)[i]))
}
colnames(AUCVC.result) <- names(specifiedMethods)
if (length(TU) == 1 && TU == 1) {
  AUCVC.result <- cbind(AUCVC.result, TU.AUCVC)
  colnames(AUCVC.result) <- c(names(specifiedMethods),
    "TU")
}
}
if (numMethod == 0 && TU == 0)
  stop("Please specify at least one method!")
if (numMethod == 0 && TU == 1) {
  AUCVC.result <- as.matrix(TU.AUCVC)
  colnames(AUCVC.result) <- "TU"
}
return(AUCVC.result)
}

```

---

optTU

*optTU*


---

## Description

Please refer to the file `/inst/doc/readme.pdf`.

## Usage

```

optTU(data, nonzeroRatio = NULL, pre_ratio_range = c(0.2, 0.6), prResolution = 0.1,
lower_range = c(0.05, 0.4), upper_range = c(0.6, 0.95),
qResolution = 0.05, min_ubq = 100, cvNorm = TRUE, cvResolution = 0.005)

```

## Arguments

<code>data</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>nonzeroRatio</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>pre_ratio_range</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>prResolution</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>lower_range</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>upper_range</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>qResolution</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>min_ubq</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>cvNorm</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>cvResolution</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

## Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, nonzeroRatio = NULL, pre_ratio_range = c(0.2,
  0.6), prResolution = 0.1, lower_range = c(0.05, 0.4), upper_range = c(0.6,
  0.95), qResolution = 0.05, min_ubq = 100, cvNorm = TRUE,
  cvResolution = 0.005)
{
  if (is.null(nonzeroRatio)) {
    stop("Please provide nonzeroRatios!")
  }
  pre_ratio_times <- (pre_ratio_range[2] - pre_ratio_range[1] +
    prResolution) * 10
  lower_times <- (upper_range[2] - upper_range[1] + qResolution)/qResolution
  lower_range_tmp <- rep(seq(lower_range[1], lower_range[2],
    qResolution), each = round(lower_times))
  lower_range2 <- rep(lower_range_tmp, times = round(pre_ratio_times))
  upper_times <- (lower_range[2] - lower_range[1] + qResolution)/qResolution
  upper_range_tmp <- rep(seq(upper_range[1], upper_range[2],
    qResolution), times = round(upper_times))
  upper_range2 <- rep(upper_range_tmp, times = round(pre_ratio_times))
  lower_upper_tmp_len <- length(lower_range_tmp)
  pre_ratio_range2 <- rep(seq(pre_ratio_range[1], pre_ratio_range[2],
    0.1), each = round(lower_upper_tmp_len))
  nonzeroIndex <- filteredZero(data, nonzeroRatio = nonzeroRatio)
  all_aucvc <- mapply(function(lower_trim, upper_trim, pre_ratio) {
    factors.TU <- getFactors(data, method = "TU", lower_trim = lower_trim,
      upper_trim = upper_trim, pre_ratio = pre_ratio, min_ubq = min_ubq)
    norm.TU <- getNormMatrix(data, factors.TU)
    dataUse2CV <- norm.TU[nonzeroIndex, ]
    cv.TU <- getCV(dataUse2CV, cvNorm = cvNorm)
    TU.AUCVC <- CV2AUCVC(cv.TU, cvResolution = cvResolution)
    return(c(TU.AUCVC = TU.AUCVC, lower = lower_trim, upper = upper_trim,
      ratio = pre_ratio))
  }, lower_range2, upper_range2, pre_ratio_range2)
  all_aucvc2 <- t(all_aucvc)
  max_index <- which(max(all_aucvc2[, "TU.AUCVC"]) == all_aucvc2[,
    "TU.AUCVC"])
  return(all_aucvc2[max_index, ])
}
```

---

plotCors

*plotCors*

---

## Description

Please refer to the file `/inst/doc/readme.pdf`.



**Usage**

```
plotCors(data, methods = c("None", "HG7", "ERCC", "TN", "TC", "CR", "NR", "DESeq",
"UQ", "TMM", "TU"), legend.position = c(0.15, 0.56))
```

**Arguments**

```
data           Please refer to the file /inst/doc/readme.pdf.
methods        Please refer to the file /inst/doc/readme.pdf.
legend.position Please refer to the file /inst/doc/readme.pdf.
```

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, methods = c("None", "HG7", "ERCC", "TN", "TC",
"CR", "NR", "DESeq", "UQ", "TMM", "TU"), legend.position = c(0.15,
0.56))
{
  if (!is.data.frame(data))
    data <- data.frame(data)
  if (is.factor(data$Value))
    data$Value <- as.numeric(as.character(data$Value))
  data$Methods <- factor(data$Methods, levels = methods, labels = methods)
  change_colours(ggplot(data = data, aes(x = Value, y = ..count../sum(..count..)) +
    geom_freqpoly(aes(group = Methods, color = Methods),
    size = 3, bins = 50) + xlab("Spearman correlation") +
    ylab("Fraction of gene pairs") + theme_bw() + theme(panel.grid.minor = element_blank(),
    axis.title.x = element_text(size = 48), axis.title.y = element_text(size = 48),
    axis.text.x = element_text(size = 38), axis.text.y = element_text(size = 38),
    legend.text = element_text(size = 39), legend.title = element_text(size = 43),
    legend.position = legend.position, legend.background = element_blank(),
    legend.key = element_blank(), legend.key.height = unit(1.8,
    "cm"), plot.margin = unit(c(0.5, 1, 0.5, 0.5), "cm")) +
    scale_x_continuous(expand = c(0.01, 0.01), breaks = round(seq(-1,
    1, 0.25), 2)) + scale_y_continuous(expand = c(0.01,
    0)) + guides(color = guide_legend(title = NULL), c("olivedrab",
    "blue", "red", "violet", "orange", "yellow", "magenta",
    "peru", "black", "maroon", "lightblue", "darkslateblue",
    "seashell4", "tan2", "darkgreen", "springgreen")))
}
```

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
plotCVs(data, methods = c("None", "HG7", "ERCC", "TN", "TC", "CR", "NR",
"DESeq", "UQ", "TMM", "TU"), legend.position = c(0.85, 0.48))
```

**Arguments**

<code>data</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>methods</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>legend.position</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, methods = c("None", "HG7", "ERCC", "TN", "TC",
"CR", "NR", "DESeq", "UQ", "TMM", "TU"), legend.position = c(0.85,
0.48))
{
  if (!is.data.frame(data))
    data <- data.frame(data)
  if (is.factor(data$Cutoff))
    data$Cutoff <- as.numeric(as.character(data$Cutoff))
  if (is.factor(data$Counts))
    data$Counts <- as.numeric(as.character(data$Counts))
  data$Methods <- factor(data$Methods, levels = methods, labels = methods)
  change_colours(ggplot(data = data, aes(x = Cutoff, y = Counts)) +
    geom_line(aes(group = Methods, color = Methods), size = 3) +
    xlab("Normalized CV cutoff") + ylab("Number of uniform genes") +
    theme_bw() + theme(panel.grid.minor = element_blank(),
axis.title.x = element_text(size = 48), axis.title.y = element_text(size = 48),
axis.text.x = element_text(size = 38), axis.text.y = element_text(size = 38),
legend.text = element_text(size = 39), legend.title = element_text(size = 43),
legend.position = legend.position, legend.background = element_blank(),
legend.key = element_blank(), legend.key.height = unit(1.8,
"cm"), plot.margin = unit(c(0.5, 0.5, 0.5, 0.5),
"cm")) + scale_x_continuous(breaks = seq(0, 1, 0.2)) +
scale_y_continuous() + guides(color = guide_legend(title = NULL)),
c("olivedrab", "blue", "red", "violet", "orange", "yellow",
"magenta", "peru", "black", "maroon", "lightblue",
"darkslateblue", "seashell4", "tan2", "darkgreen",
"springgreen"))
}
```

---

plotHC

*plotHC*


---

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
plotHC(data, method = c("spearman", "pearson", "kendall"), mar = c(9, 1, 0, 20))
```

**Arguments**

<code>data</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>method</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .
<code>mar</code>	Please refer to the file <code>/inst/doc/readme.pdf</code> .

**Examples**

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (data, method = c("spearman", "pearson", "kendall"),
         mar = c(9, 1, 0, 20))
{
  if (!is.data.frame(data))
    data <- data.frame(data)
  method <- match.arg(method)
  hc <- hclust(as.dist(1 - cor(data, method = method)))
  dend <- as.dendrogram(hc)
  dend <- dend %>% set("labels_cex", 6.5) %>% set("branches_lwd",
        6.5)
  par(mar = mar, mgp = c(10, 5, 0), cex.axis = 6)
  plot(dend, horiz = TRUE)
  axis(side = 1, lwd = 8)
}
```

---

scRNA663

*scRNA663*


---

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
data("scRNA663")
```

**Format**

A data frame with 57955 observations on the following 663 variables.

col361\_1 a numeric vector  
col361\_2 a numeric vector  
col361\_3 a numeric vector  
col361\_4 a numeric vector  
col361\_5 a numeric vector  
col361\_6 a numeric vector  
col361\_7 a numeric vector  
col361\_8 a numeric vector  
col361\_9 a numeric vector  
col361\_10 a numeric vector  
col361\_11 a numeric vector  
col361\_12 a numeric vector  
col361\_13 a numeric vector  
col361\_14 a numeric vector  
col361\_15 a numeric vector  
col361\_16 a numeric vector  
col361\_17 a numeric vector  
col361\_18 a numeric vector  
col361\_19 a numeric vector  
col361\_20 a numeric vector  
col361\_21 a numeric vector  
col361\_22 a numeric vector  
col361\_23 a numeric vector  
col361\_24 a numeric vector  
col361\_25 a numeric vector  
col361\_26 a numeric vector  
col361\_27 a numeric vector  
col361\_28 a numeric vector  
col361\_29 a numeric vector  
col361\_30 a numeric vector  
col361\_31 a numeric vector  
col361\_32 a numeric vector

col361\_33 a numeric vector  
col361\_34 a numeric vector  
col361\_35 a numeric vector  
col361\_36 a numeric vector  
col361\_37 a numeric vector  
col361\_38 a numeric vector  
col361\_39 a numeric vector  
col361\_40 a numeric vector  
col361\_41 a numeric vector  
col361\_42 a numeric vector  
col361\_43 a numeric vector  
col361\_44 a numeric vector  
col361\_45 a numeric vector  
col361\_46 a numeric vector  
col361\_47 a numeric vector  
col361\_48 a numeric vector  
col361\_49 a numeric vector  
col361\_50 a numeric vector  
col361\_51 a numeric vector  
col361\_52 a numeric vector  
col361\_53 a numeric vector  
col361\_54 a numeric vector  
col361\_55 a numeric vector  
col361\_56 a numeric vector  
col361\_57 a numeric vector  
col361\_58 a numeric vector  
col361\_59 a numeric vector  
col361\_60 a numeric vector  
col361\_61 a numeric vector  
col361\_62 a numeric vector  
col361\_63 a numeric vector  
col361\_64 a numeric vector  
col361\_65 a numeric vector  
col361\_66 a numeric vector  
col361\_67 a numeric vector  
col361\_68 a numeric vector  
col361\_69 a numeric vector

col361\_70 a numeric vector  
col361\_71 a numeric vector  
col381\_1 a numeric vector  
col381\_2 a numeric vector  
col381\_6 a numeric vector  
col381\_7 a numeric vector  
col381\_8 a numeric vector  
col381\_10 a numeric vector  
col381\_11 a numeric vector  
col381\_12 a numeric vector  
col381\_13 a numeric vector  
col381\_14 a numeric vector  
col381\_15 a numeric vector  
col381\_16 a numeric vector  
col381\_17 a numeric vector  
col381\_19 a numeric vector  
col381\_20 a numeric vector  
col381\_21 a numeric vector  
col381\_22 a numeric vector  
col381\_23 a numeric vector  
col381\_24 a numeric vector  
col381\_25 a numeric vector  
col381\_26 a numeric vector  
col381\_27 a numeric vector  
col381\_28 a numeric vector  
col381\_29 a numeric vector  
col381\_30 a numeric vector  
col381\_31 a numeric vector  
col381\_33 a numeric vector  
col381\_34 a numeric vector  
col381\_35 a numeric vector  
col381\_36 a numeric vector  
col381\_37 a numeric vector  
col381\_39 a numeric vector  
col381\_40 a numeric vector  
col381\_41 a numeric vector  
col381\_42 a numeric vector

col381\_43 a numeric vector  
col381\_46 a numeric vector  
col381\_47 a numeric vector  
col381\_48 a numeric vector  
col381\_49 a numeric vector  
col381\_52 a numeric vector  
col381\_55 a numeric vector  
col381\_56 a numeric vector  
col381\_57 a numeric vector  
col381\_59 a numeric vector  
col381\_60 a numeric vector  
col381\_61 a numeric vector  
col381\_62 a numeric vector  
col381\_64 a numeric vector  
col381\_65 a numeric vector  
col381\_66 a numeric vector  
col381\_67 a numeric vector  
col381\_68 a numeric vector  
col381\_69 a numeric vector  
col381\_70 a numeric vector  
col381\_72 a numeric vector  
col3911\_47 a numeric vector  
col3911\_48 a numeric vector  
col3911\_49 a numeric vector  
col3911\_50 a numeric vector  
col3911\_51 a numeric vector  
col3911\_52 a numeric vector  
col3911\_53 a numeric vector  
col3911\_54 a numeric vector  
col3911\_55 a numeric vector  
col3911\_56 a numeric vector  
col3911\_57 a numeric vector  
col3911\_58 a numeric vector  
col3911\_59 a numeric vector  
col3911\_60 a numeric vector  
col3911\_61 a numeric vector  
col3911\_62 a numeric vector

col3911\_63 a numeric vector  
col3911\_64 a numeric vector  
col3911\_69 a numeric vector  
col3911\_70 a numeric vector  
col3911\_71 a numeric vector  
col3911\_72 a numeric vector  
col3911\_73 a numeric vector  
col3911\_74 a numeric vector  
col3911\_75 a numeric vector  
col3911\_76 a numeric vector  
col3911\_77 a numeric vector  
col3911\_78 a numeric vector  
col3911\_79 a numeric vector  
col3911\_80 a numeric vector  
col3911\_81 a numeric vector  
col3911\_82 a numeric vector  
col3911\_83 a numeric vector  
col3911\_84 a numeric vector  
col3911\_85 a numeric vector  
col3911\_86 a numeric vector  
col3911\_87 a numeric vector  
col3911\_88 a numeric vector  
col3911\_89 a numeric vector  
col3911\_90 a numeric vector  
col3911\_91 a numeric vector  
col3911\_92 a numeric vector  
col3911\_93 a numeric vector  
col3911\_94 a numeric vector  
col3911\_95 a numeric vector  
col3911\_96 a numeric vector  
col3912\_20 a numeric vector  
col3912\_21 a numeric vector  
col3912\_22 a numeric vector  
col3912\_23 a numeric vector  
col3912\_24 a numeric vector  
col3912\_25 a numeric vector  
col3912\_26 a numeric vector



col3912\_27 a numeric vector  
col3912\_28 a numeric vector  
col3912\_29 a numeric vector  
col3912\_30 a numeric vector  
col3912\_31 a numeric vector  
col3912\_32 a numeric vector  
col3912\_33 a numeric vector  
col3912\_34 a numeric vector  
col3912\_35 a numeric vector  
col3912\_36 a numeric vector  
col3912\_37 a numeric vector  
col3912\_38 a numeric vector  
col3912\_39 a numeric vector  
col3912\_40 a numeric vector  
col3912\_41 a numeric vector  
col3912\_42 a numeric vector  
col3912\_43 a numeric vector  
col3912\_44 a numeric vector  
col3912\_45 a numeric vector  
col3912\_46 a numeric vector  
col3912\_47 a numeric vector  
col3912\_48 a numeric vector  
col3912\_49 a numeric vector  
col3912\_50 a numeric vector  
col3912\_52 a numeric vector  
col3912\_53 a numeric vector  
col3912\_54 a numeric vector  
col3912\_55 a numeric vector  
col3912\_56 a numeric vector  
col3912\_57 a numeric vector  
col3912\_58 a numeric vector  
col3912\_59 a numeric vector  
col3912\_60 a numeric vector  
col3912\_61 a numeric vector  
col3912\_62 a numeric vector  
col3912\_63 a numeric vector  
col3912\_64 a numeric vector

col3912\_65 a numeric vector  
col3912\_66 a numeric vector  
col3912\_67 a numeric vector  
col3912\_68 a numeric vector  
col3912\_69 a numeric vector  
col3912\_70 a numeric vector  
col3912\_72 a numeric vector  
col3912\_73 a numeric vector  
col3912\_74 a numeric vector  
col3913\_31 a numeric vector  
col3913\_32 a numeric vector  
col3913\_33 a numeric vector  
col3913\_34 a numeric vector  
col3913\_35 a numeric vector  
col3913\_36 a numeric vector  
col3913\_38 a numeric vector  
col3913\_39 a numeric vector  
col3913\_40 a numeric vector  
col3913\_41 a numeric vector  
col3913\_42 a numeric vector  
col3913\_43 a numeric vector  
col3913\_44 a numeric vector  
col3913\_45 a numeric vector  
col3913\_46 a numeric vector  
col3913\_47 a numeric vector  
col3913\_48 a numeric vector  
col3913\_49 a numeric vector  
col3913\_50 a numeric vector  
col3913\_51 a numeric vector  
col3913\_52 a numeric vector  
col3913\_53 a numeric vector  
col3913\_54 a numeric vector  
col3913\_55 a numeric vector  
col3913\_56 a numeric vector  
col3913\_57 a numeric vector  
col3913\_58 a numeric vector  
col3913\_59 a numeric vector

col3913\_60 a numeric vector  
col3913\_61 a numeric vector  
col3913\_62 a numeric vector  
col3913\_63 a numeric vector  
col3913\_64 a numeric vector  
col3913\_65 a numeric vector  
col3913\_66 a numeric vector  
col3913\_67 a numeric vector  
col3913\_68 a numeric vector  
col3913\_69 a numeric vector  
col3913\_70 a numeric vector  
col3913\_71 a numeric vector  
col3913\_72 a numeric vector  
col3913\_73 a numeric vector  
col3913\_74 a numeric vector  
col3913\_75 a numeric vector  
col3913\_76 a numeric vector  
col3913\_77 a numeric vector  
col3913\_78 a numeric vector  
col3913\_79 a numeric vector  
col3913\_80 a numeric vector  
col3913\_81 a numeric vector  
col3913\_82 a numeric vector  
col3913\_83 a numeric vector  
col3913\_85 a numeric vector  
col401\_1 a numeric vector  
col401\_2 a numeric vector  
col401\_3 a numeric vector  
col401\_4 a numeric vector  
col401\_5 a numeric vector  
col401\_6 a numeric vector  
col401\_7 a numeric vector  
col401\_8 a numeric vector  
col401\_9 a numeric vector  
col401\_10 a numeric vector  
col401\_11 a numeric vector  
col401\_12 a numeric vector

col401\_13 a numeric vector  
col401\_14 a numeric vector  
col401\_15 a numeric vector  
col401\_16 a numeric vector  
col401\_17 a numeric vector  
col401\_18 a numeric vector  
col401\_19 a numeric vector  
col401\_20 a numeric vector  
col401\_21 a numeric vector  
col401\_22 a numeric vector  
col401\_23 a numeric vector  
col401\_24 a numeric vector  
col401\_25 a numeric vector  
col401\_26 a numeric vector  
col401\_27 a numeric vector  
col401\_28 a numeric vector  
col401\_29 a numeric vector  
col401\_30 a numeric vector  
col401\_31 a numeric vector  
col401\_32 a numeric vector  
col401\_33 a numeric vector  
col401\_34 a numeric vector  
col401\_35 a numeric vector  
col401\_37 a numeric vector  
col401\_38 a numeric vector  
col401\_39 a numeric vector  
col401\_40 a numeric vector  
col401\_41 a numeric vector  
col401\_42 a numeric vector  
col401\_44 a numeric vector  
col401\_45 a numeric vector  
col401\_46 a numeric vector  
col401\_47 a numeric vector  
col401\_48 a numeric vector  
col401\_49 a numeric vector  
col401\_50 a numeric vector  
col4411\_1 a numeric vector

col4411\_2 a numeric vector  
col4411\_3 a numeric vector  
col4411\_4 a numeric vector  
col4411\_8 a numeric vector  
col4411\_11 a numeric vector  
col4411\_12 a numeric vector  
col4411\_13 a numeric vector  
col4411\_14 a numeric vector  
col4411\_15 a numeric vector  
col4411\_16 a numeric vector  
col4411\_17 a numeric vector  
col4411\_18 a numeric vector  
col4411\_19 a numeric vector  
col4411\_20 a numeric vector  
col4411\_24 a numeric vector  
col4411\_28 a numeric vector  
col4411\_29 a numeric vector  
col4411\_30 a numeric vector  
col4411\_32 a numeric vector  
col4411\_33 a numeric vector  
col4411\_36 a numeric vector  
col4411\_38 a numeric vector  
col4411\_40 a numeric vector  
col4411\_41 a numeric vector  
col4411\_42 a numeric vector  
col4411\_43 a numeric vector  
col4411\_47 a numeric vector  
col4411\_48 a numeric vector  
col4411\_50 a numeric vector  
col4411\_53 a numeric vector  
col4411\_58 a numeric vector  
col4411\_59 a numeric vector  
col4411\_60 a numeric vector  
col4411\_64 a numeric vector  
col4411\_66 a numeric vector  
col4411\_67 a numeric vector  
col4411\_68 a numeric vector

col4411\_69 a numeric vector  
col4411\_70 a numeric vector  
col4411\_71 a numeric vector  
col4411\_73 a numeric vector  
col4411\_74 a numeric vector  
col4411\_75 a numeric vector  
col4411\_76 a numeric vector  
col4411\_77 a numeric vector  
col4411\_79 a numeric vector  
col4411\_80 a numeric vector  
col4411\_82 a numeric vector  
col4411\_83 a numeric vector  
col4411\_84 a numeric vector  
col4411\_85 a numeric vector  
col4411\_86 a numeric vector  
col4411\_87 a numeric vector  
col4411\_89 a numeric vector  
col4411\_90 a numeric vector  
col4411\_91 a numeric vector  
col4411\_92 a numeric vector  
col4411\_93 a numeric vector  
col4411\_94 a numeric vector  
col4412\_1 a numeric vector  
col4412\_2 a numeric vector  
col4412\_3 a numeric vector  
col4412\_4 a numeric vector  
col4412\_5 a numeric vector  
col4412\_6 a numeric vector  
col4412\_9 a numeric vector  
col4412\_10 a numeric vector  
col4412\_12 a numeric vector  
col4412\_14 a numeric vector  
col4412\_17 a numeric vector  
col4412\_18 a numeric vector  
col4412\_19 a numeric vector  
col4412\_20 a numeric vector  
col4412\_23 a numeric vector

col4412\_26 a numeric vector  
col4412\_27 a numeric vector  
col4412\_28 a numeric vector  
col4412\_30 a numeric vector  
col4412\_31 a numeric vector  
col4412\_32 a numeric vector  
col4412\_35 a numeric vector  
col4412\_36 a numeric vector  
col4412\_37 a numeric vector  
col4412\_38 a numeric vector  
col4412\_40 a numeric vector  
col4417\_67 a numeric vector  
col4417\_68 a numeric vector  
col4417\_69 a numeric vector  
col4417\_70 a numeric vector  
col4417\_71 a numeric vector  
col4417\_75 a numeric vector  
col4417\_76 a numeric vector  
col4417\_77 a numeric vector  
col4417\_78 a numeric vector  
col4417\_79 a numeric vector  
col4417\_80 a numeric vector  
col4417\_82 a numeric vector  
col4417\_83 a numeric vector  
col4417\_84 a numeric vector  
col4417\_85 a numeric vector  
col4417\_86 a numeric vector  
col4417\_87 a numeric vector  
col4417\_88 a numeric vector  
col4417\_89 a numeric vector  
col4417\_90 a numeric vector  
col4417\_91 a numeric vector  
col4417\_92 a numeric vector  
col4417\_93 a numeric vector  
col4417\_94 a numeric vector  
col4417\_95 a numeric vector  
col4417\_96 a numeric vector

col4418\_17 a numeric vector  
col4418\_19 a numeric vector  
col4418\_21 a numeric vector  
col4418\_22 a numeric vector  
col4418\_23 a numeric vector  
col4418\_24 a numeric vector  
col4418\_25 a numeric vector  
col4418\_26 a numeric vector  
col4418\_28 a numeric vector  
col4418\_29 a numeric vector  
col4418\_31 a numeric vector  
col4418\_32 a numeric vector  
col4418\_33 a numeric vector  
col4418\_34 a numeric vector  
col4418\_43 a numeric vector  
col4418\_45 a numeric vector  
col4418\_46 a numeric vector  
col4418\_47 a numeric vector  
col4418\_48 a numeric vector  
col4418\_50 a numeric vector  
col4418\_51 a numeric vector  
col4418\_52 a numeric vector  
col4418\_53 a numeric vector  
col4418\_54 a numeric vector  
col4418\_55 a numeric vector  
col4418\_56 a numeric vector  
col4418\_58 a numeric vector  
col4418\_59 a numeric vector  
col4418\_60 a numeric vector  
col4418\_61 a numeric vector  
col4418\_62 a numeric vector  
col4418\_63 a numeric vector  
col4418\_67 a numeric vector  
col4418\_68 a numeric vector  
col4418\_70 a numeric vector  
col4418\_71 a numeric vector  
col4418\_72 a numeric vector



col4418\_73 a numeric vector  
col4418\_74 a numeric vector  
col4418\_76 a numeric vector  
col4418\_78 a numeric vector  
col4418\_81 a numeric vector  
col4418\_85 a numeric vector  
col4418\_86 a numeric vector  
col4418\_88 a numeric vector  
col4418\_94 a numeric vector  
col4512\_42 a numeric vector  
col4512\_43 a numeric vector  
col4512\_45 a numeric vector  
col4512\_47 a numeric vector  
col4512\_48 a numeric vector  
col4512\_50 a numeric vector  
col4512\_51 a numeric vector  
col4512\_52 a numeric vector  
col4512\_54 a numeric vector  
col4512\_55 a numeric vector  
col4512\_56 a numeric vector  
col4512\_57 a numeric vector  
col4512\_58 a numeric vector  
col4512\_59 a numeric vector  
col4512\_60 a numeric vector  
col4512\_61 a numeric vector  
col4512\_62 a numeric vector  
col4512\_63 a numeric vector  
col4512\_64 a numeric vector  
col4512\_65 a numeric vector  
col4512\_66 a numeric vector  
col4512\_67 a numeric vector  
col4512\_68 a numeric vector  
col4512\_69 a numeric vector  
col4512\_70 a numeric vector  
col4512\_71 a numeric vector  
col4512\_76 a numeric vector  
col4512\_77 a numeric vector

col4512\_78 a numeric vector  
col4512\_81 a numeric vector  
col4512\_83 a numeric vector  
col4512\_84 a numeric vector  
col4512\_85 a numeric vector  
col4512\_88 a numeric vector  
col4512\_89 a numeric vector  
col4512\_90 a numeric vector  
col4512\_91 a numeric vector  
col4512\_92 a numeric vector  
col4512\_93 a numeric vector  
col4512\_94 a numeric vector  
col4512\_95 a numeric vector  
col4517\_24 a numeric vector  
col4517\_25 a numeric vector  
col4518\_2 a numeric vector  
col4518\_3 a numeric vector  
col4518\_4 a numeric vector  
col4518\_5 a numeric vector  
col4518\_6 a numeric vector  
col4518\_7 a numeric vector  
col4518\_8 a numeric vector  
col4518\_9 a numeric vector  
col4518\_11 a numeric vector  
col4518\_12 a numeric vector  
col4518\_13 a numeric vector  
col4518\_14 a numeric vector  
col4518\_35 a numeric vector  
col4717\_26 a numeric vector  
col4717\_27 a numeric vector  
col4717\_28 a numeric vector  
col4717\_29 a numeric vector  
col4717\_30 a numeric vector  
col4717\_31 a numeric vector  
col4717\_32 a numeric vector  
col4717\_33 a numeric vector  
col4717\_34 a numeric vector

col4717\_35 a numeric vector  
col4717\_36 a numeric vector  
col4717\_37 a numeric vector  
col4717\_38 a numeric vector  
col4717\_41 a numeric vector  
col4717\_42 a numeric vector  
col4717\_44 a numeric vector  
col4717\_45 a numeric vector  
col4717\_47 a numeric vector  
col4717\_48 a numeric vector  
col4717\_49 a numeric vector  
col4717\_50 a numeric vector  
col4717\_51 a numeric vector  
col4717\_54 a numeric vector  
col4717\_57 a numeric vector  
col4717\_58 a numeric vector  
col4717\_59 a numeric vector  
col4717\_60 a numeric vector  
col4717\_63 a numeric vector  
col4717\_64 a numeric vector  
col4717\_65 a numeric vector  
col4717\_66 a numeric vector  
col4816\_2 a numeric vector  
col4816\_4 a numeric vector  
col4816\_5 a numeric vector  
col4816\_6 a numeric vector  
col4816\_7 a numeric vector  
col4816\_8 a numeric vector  
col4816\_9 a numeric vector  
col4816\_10 a numeric vector  
col4816\_11 a numeric vector  
col4816\_12 a numeric vector  
col4816\_13 a numeric vector  
col4816\_14 a numeric vector  
col4816\_16 a numeric vector  
col4816\_17 a numeric vector  
col4816\_18 a numeric vector

col4816\_19 a numeric vector  
col4816\_20 a numeric vector  
col4816\_21 a numeric vector  
col4816\_22 a numeric vector  
col4816\_24 a numeric vector  
col4816\_26 a numeric vector  
col4816\_27 a numeric vector  
col4816\_28 a numeric vector  
col4816\_30 a numeric vector  
col4816\_31 a numeric vector  
col4816\_34 a numeric vector  
col4816\_35 a numeric vector  
col4816\_36 a numeric vector  
col4816\_39 a numeric vector  
col4816\_40 a numeric vector  
col4816\_42 a numeric vector  
col4816\_43 a numeric vector  
col4816\_44 a numeric vector  
col4816\_46 a numeric vector  
col4816\_47 a numeric vector  
col4816\_48 a numeric vector  
col4816\_50 a numeric vector  
col4816\_51 a numeric vector  
col4816\_52 a numeric vector  
col4816\_53 a numeric vector  
col4816\_54 a numeric vector  
col4816\_55 a numeric vector  
col4816\_56 a numeric vector  
col4816\_58 a numeric vector  
col4816\_60 a numeric vector  
col4816\_61 a numeric vector  
col4816\_62 a numeric vector  
col4816\_63 a numeric vector  
col4816\_64 a numeric vector  
col4816\_65 a numeric vector  
col4816\_66 a numeric vector  
col4816\_67 a numeric vector

col4816\_68 a numeric vector  
col4816\_69 a numeric vector  
col4816\_70 a numeric vector  
col4816\_71 a numeric vector  
col4816\_72 a numeric vector  
col4816\_73 a numeric vector  
col4816\_74 a numeric vector  
col4816\_75 a numeric vector  
col4816\_76 a numeric vector  
col4816\_77 a numeric vector  
col4816\_78 a numeric vector  
col4816\_79 a numeric vector  
col4816\_80 a numeric vector  
col4816\_81 a numeric vector  
col4816\_82 a numeric vector  
col4816\_85 a numeric vector  
col4816\_86 a numeric vector  
col4816\_88 a numeric vector  
col4816\_90 a numeric vector  
col4816\_92 a numeric vector  
col4816\_93 a numeric vector  
col4816\_94 a numeric vector  
col4816\_95 a numeric vector  
col4816\_96 a numeric vector  
col4817\_1 a numeric vector  
col4817\_2 a numeric vector  
col4817\_3 a numeric vector  
col4817\_4 a numeric vector  
col4817\_5 a numeric vector  
col4817\_7 a numeric vector  
col4817\_9 a numeric vector  
col4817\_11 a numeric vector  
col4817\_12 a numeric vector  
col4817\_13 a numeric vector  
col4817\_15 a numeric vector  
col4817\_16 a numeric vector  
col4817\_18 a numeric vector  
col4817\_20 a numeric vector  
col4817\_21 a numeric vector

**Examples**

```
data(scRNA663)
## maybe str(scRNA663) ; plot(scRNA663) ...
```

---

scRNA663_factors	<i>scRNA663_factors</i>
------------------	-------------------------

---

**Description**

Please refer to the file `/inst/doc/readme.pdf`.

**Usage**

```
data("scRNA663_factors")
```

**Format**

A data frame with 663 observations on the following 12 variables.

HG7 a numeric vector

ERCC a numeric vector

TN a numeric vector

TC a numeric vector

CR a numeric vector

NR a numeric vector

DESeq a numeric vector

UQ a numeric vector

TMM a numeric vector

TU a numeric vector

NCS a numeric vector

ES a numeric vector

**Examples**

```
data(scRNA663_factors)
## maybe str(scRNA663_factors) ; plot(scRNA663_factors) ...
```

# Index

## \* datasets

- [bkRNA18](#), [2](#)
- [bkRNA18\\_factors](#), [3](#)
- [scRNA663](#), [35](#)
- [scRNA663\\_factors](#), [54](#)

- [bkRNA18](#), [2](#)
- [bkRNA18\\_factors](#), [3](#)

- [calcFactorRLE](#), [4](#)
- [calcFactorUpperquartile](#), [4](#)
- [calcFactorWeighted](#), [5](#)
- [change\\_colours](#), [6](#)
- [CV2AUCVC](#), [7](#)

- [estimateSizeFactorsForMatrix](#), [8](#)

- [filteredZero](#), [9](#)
- [findGenes](#), [9](#)

- [gatherCors](#), [10](#)
- [gatherCors4Matrices](#), [11](#)
- [gatherCVs](#), [13](#)
- [gatherCVs4Matrices](#), [14](#)
- [gatherFactors](#), [15](#)
- [getArea](#), [17](#)
- [getAUCVC](#), [18](#)
- [getAUCVCs](#), [19](#)
- [getCor](#), [20](#)
- [getCorMedians](#), [21](#)
- [getCV](#), [21](#)
- [getFactors](#), [22](#)
- [getNormMatrix](#), [24](#)
- [gridAUCVC](#), [24](#)
- [gridAUCVC4Matrices](#), [26](#)

- [identifyUbq](#), [27](#)
- [identifyUbqRepeat](#), [28](#)

- [nonzeroRatio2AUCVC](#), [29](#)

- [optTU](#), [31](#)

- [plotCors](#), [32](#)
- [plotCVs](#), [33](#)
- [plotHC](#), [35](#)

- [scRNA663](#), [35](#)
- [scRNA663\\_factors](#), [54](#)