Package ‘CpmERCCutoff’

October 12, 2022

Type Package

Title Calculation of Log2 Counts per Million Cutoff from ERCC Controls

Version 1.0.0

Description Implementation of the empirical method to derive log2 counts per million (CPM) cutoff to filter out lowly expressed genes using ERCC spike-ins as described in Goll and Bosinger et.al (2022)<doi:10.1101/2022.06.23.497396>. This package utilizes the synthetic mRNA control pairs developed by the External RNA Controls Consortium (ERCC) (ERCC 1 / ERCC 2) that are spiked into sample pairs at known ratios at various absolute abundances. The relationship between the observed and expected fold changes is then used to empirically determine an optimal log2 CPM cutoff for filtering out lowly expressed genes.

Depends R (>= 3.6)

License GPL (>= 3)

Encoding UTF-8

LazyData true

Imports stats, graphics

Suggests spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.2.0

NeedsCompilation no

Language en-US

Author Tyler Grimes [aut] (<https://orcid.org/0000-0002-5653-3418>),
Travis L. Jensen [aut] (<https://orcid.org/0000-0002-0322-0469>),
Kristen Steenbergen [ctb, cre],
The Emmes Company LLC [cph] (Copyright holder of CpmERCCutoff package (Copyright (C) 2022)),
Johannes B. Goll [aut] (<https://orcid.org/0000-0002-9968-4080>)

Maintainer Kristen Steenbergen <ksteenbergen@emmes.com>

Repository CRAN

Date/Publication 2022-09-13 11:20:07 UTC

R topics documented:

exp_input *A data frame of expected ERCC1 and ERCC2 ratios*

Description

A data frame that contains the expected spike in ERCC data. This data can be obtained from 'ERCC Controls Analysis' manual located on Thermo Fisher's ERCC RNA Spike-In Mix product [page](https://assets.thermofisher.com/TFS-Assets/LSG/manuals/cms_095046.txt). The 'exp_input' data frame mirrors the fields shown in the ERCC manual. For the LCPM cutoff calculation, the last column of the log2 expected fold change ratios are used. Ensure that this column is titled "expected_lfc_ratio". See the example code below for formatting the data.

Usage

```
exp_input
```

Format

A data frame with 92 rows and 4 columns: Each row represents an ERCC transcript. Columns are described below:

ercc_id ERCC spike-in mRNA Ids (ERCC-00002 – ERCC-00171)

subgroup ERCC subgroups (A – D)

ercc1_conc ERCC1 concentration (0.014 – 30,000)

ercc2_conc ERCC2 concentration (0.007 – 30,000)

expected_fc_ratio Expected fold change ratio (.5 – 4)

expected_lfc_ratio Expected log2 fold change ratio (-1 – 2)

Examples

```
# Order rows by ERCC ID and assign to rownames.
exp_input = exp_input[order(exp_input$ercc_id), ]
rownames(exp_input) = exp_input$ercc_id
```

`getLowLcpmCutoff` *Function to empirically determine a log2 CPM cutoff based on ERCC RNA spike-in*

Description

This function uses spike-in ERCC data, known control RNA probes, and paired samples to fit a 3rd order polynomial to determine an expression cutoff that meets the specified correlation between expected and observed fold changes. The `obs` data frame used as input for the observed expression of the 92 ERCC RNA spike-ins and stores the coverage-normalized read log2 counts per million (LCPM) that mapped to the respective ERCC sequences. Typically, prior to LCPM calculation, the read count data is normalized for any systematic differences in read coverage between samples, for example, by using the TMM normalization method as implemented in the `edgeR` package.

For each bootstrap replicate, the paired samples are subsampled with replacement. The mean LCPM of each ERCC transcript is determined by first calculating the average LCPM value for each paired sample, and then taking the mean of those averages. The ERCC transcripts are sorted based on these means, and are then grouped into `n.bins` ERCC bins. Next, the Spearman correlation metric is used to calculate the association between the empirical and expected log fold change (LFC) of the ERCCs in each bin for each sample. Additionally, the average LCPM for the ERCCs in each bin are calculated for each sample. This leads to a pair of values - the average LCPM and the association value - for each sample and each ERCC bin. Outliers within each ERCC bin are identified and removed based on >1.5 IQR. A 3rd order polynomial is fit with the explanatory variable being the average LCPM and the response variable being the Spearman correlation value between expected and observed log2 fold changes. The fitted curve is used to identify the average LCPM value with a Spearman correlation of `cor.value`. The results are output as an "empLCM" object as described below. The `summary.empLCPM` function can be used to extract a summary of the results, and the `plot.empLCPM` function to plot the results for visualization.

Usage

```
getLowLcpmCutoff(
  obs,
  exp,
  pairs,
  n.bins = 7,
  rep = 1000,
  ci = 0.95,
  cor.value = 0.9,
  remove.outliers = TRUE,
  seed = 20220719
)
```

Arguments

`obs`	A data frame of observed spike-in ERCC data. Each row is an ERCC transcript, and each column is a sample. Data are read coverage-normalized log2 counts per million (LCPM).

`exp` A data frame of expected ERCC Mix 1 and Mix 2 ratios with a column titled ‘expected_lfc_ratio‘ containing the expected log2 fold-change ratios. This data can be obtained from ’ERCC Controls Analysis’ manual located on Thermo Fisher’s ERCC RNA Spike-In Mix product [page](https://assets.thermofisher.com/TFS-Assets/LSG/manuals/cms_095046.txt). The ’exp_input’ data frame mirrors the fields shown in the ERCC manual. For the LCPM cutoff calculation, the last column containing the log2 expected fold change ratios are used. Ensure that this column is titled "expected_lfc_ratio". See the example code below for formatting the data. #

`pairs` A 2-column data frame where each row indicates a sample pair with the first column indicating the sample that received ERCC spike-ins from Mix 1 and the second column indicating the sample receiving Mix 2.

`n.bins` Integer. The number of abundance bins to create. Default is 7.

`rep` Integer. The number of bootstrap replicates. Default is 1000.

`ci` Numeric. The confidence interval. Default is 0.95.

`cor.value` Numeric. The desired Spearman correlation between the empirical log2 fold change across the ERCC transcripts. Default is 0.9.

`remove.outliers`
If TRUE (default) outliers are identified as exceeding 1.5 IQR, and are removed prior to fitting the polynomial. Set to FALSE to keep all points.

`seed` Integer. The reproducibility seed. Default is 20220719.

Value

An "empLCPM" object is returned, which contains the following named elements:

`cutoff` a vector containing 3 values: the threshold value, upper confidence interval, and the lower confidence interval value.
`args` a key: value list of arguments that were provided.
`res` a list containing the main results and other information from the input.
The `summary.empLCPM` function should be used to extract a summary table.

See Also

`summary.empLCPM`, `plot.empLCPM`, `print.empLCPM`

Examples

```
library(CpmERCCutoff)
###############################
# Load and wrangle input data:
###############################
# Load observed read counts
data("obs_input")

# Set ERCC Ids to rownames
rownames(obs_input) = obs_input$X
```

```
# Load expected ERCC data:
data(”exp_input”)

# Order rows by ERCC ID.
exp_input = exp_input[order(exp_input$ercc_id), ]
rownames(exp_input) = exp_input$ercc_id

# Load metadata:
data(”mta_dta”)

# Pair samples that received ERCC Mix 1 with samples that received ERCC Mix 2.
# The resulting 2-column data frame is used for the 'pairs' argument.
# Note: the code here will depend on the details of the given experiment. In
#       this example, the post-vaccination samples (which received Mix 2)
#       for each subject are paired to their pre-vaccination samples (which
#       received Mix 1).
pairs_input = cbind(
  mta_dta[mta_dta$spike == 2, 'samid'],
  mta_dta[match(mta_dta[mta_dta$spike == 2, 'subid'],
                mta_dta[mta_dta$spike == 1,'subid']), 'samid'])
# Put Mix 1 in the first column and Mix 2 in the second.
pairs_input = pairs_input[, c(2, 1)]

###############################
# Run getLowLcpmCutoff Function:
###############################'
# Note: Here we use `rep = 10` for only 10 bootstrap replicates
#       to decrease the run time for this example; a lager number
#       should be used in practice (default = 1000).
res = getLowLcpmCutoff(obs = obs_input,
                       exp = exp_input,
                       pairs = pairs_input,
                       n.bins = 7,
                       rep = 10,
                       cor.value = 0.9,
                  remove.outliers = TRUE,
                       seed = 20220719)

# Print a short summary of the results:
res

# Extract a summary table of the results:
summary(res)

# Create a plot of the results:
plot(x = res,
     main = ”Determination of Empirical Minimum Expression Cutoffs using ERCCs”,
     col.trend = ”blue”,
     col.outlier = c(”black”, ”red”))
```

`mta_dta` *A data frame containing sample-level ERCC meta data*

Description

A data frame containing sample-level ERCC meta data. In this experiment, subjects had repeated measures and the baseline samples were spiked with ERCC1 and four post-vaccination samples were each spiked with ERCC2. This meta data is used to create the 2-column data frame of sample pairs where the first column will contain sample IDs that received ERCC1, and the second column will contain sample IDs that received ERCC2. The pairs data frame is used as input to the ‘pairs‘ argument in the ‘getLowCpmCutOff‘ function. Use ‘?getLowCpmcutOff‘ to review example code regarding the formatting of the ‘pairs‘ data frame.

Usage

```
mta_dta
```

Format

A data frame with 49 rows and 4 columns: Each row is a sample, and each column contains metadata such as subject IDs, spike in control type, and treatment groups. For this study, data was collected at various time points, however, under different experiment conditions, the ’day’ column can be represented as treatment group.

samid Sample IDs (SAM01 – SAM49)

subid Subject/Participant ID (SUB01 – SUB10)

day collection day (0 – 14)

spike ERCC spike in control Mix (1 or 2)

`obs_input` *A data frame of observed spike in ERCC normalized LCPM data*

Description

A data frame of observed gene expression results for ERCC RNA that was spiked into samples. Data are read coverage-normalized log2 counts per million (LCPM).

Usage

```
obs_input
```

Format

A data frame with 92 rows and 50 Columns: Each row is an ERCC transcript, and each column is a sample. Data are read coverage-normalized LCPM.

X This first column is ERCC spike in mRNA Ids (ERCC-00002 – ERCC-00171)

SAM15 Sample 15 containing normalized log2 counts per million

SAM36 Sample 36 containing normalized log2 counts per million

SAM19 Sample 19 containing normalized log2 counts per million

SAM53 Sample 53 containing normalized log2 counts per million

SAM42 Sample 42 containing normalized log2 counts per million

SAM32 Sample 32 containing normalized log2 counts per million

SAM18 Sample 18 containing normalized log2 counts per million

SAM48 Sample 48 containing normalized log2 counts per million

SAM26 Sample 26 containing normalized log2 counts per million

SAM37 Sample 37 containing normalized log2 counts per million

SAM38 Sample 38 containing normalized log2 counts per million

SAM29 Sample 29 containing normalized log2 counts per million

SAM17 Sample 17 containing normalized log2 counts per million

SAM41 Sample 41 containing normalized log2 counts per million

SAM09 Sample 09 containing normalized log2 counts per million

SAM07 Sample 07 containing normalized log2 counts per million

SAM14 Sample 14 containing normalized log2 counts per million

SAM02 Sample 02 containing normalized log2 counts per million

SAM05 Sample 05 containing normalized log2 counts per million

SAM25 Sample 25 containing normalized log2 counts per million

SAM08 Sample 08 containing normalized log2 counts per million

SAM28 Sample 28 containing normalized log2 counts per million

SAM44 Sample 44 containing normalized log2 counts per million

SAM04 Sample 04 containing normalized log2 counts per million

SAM10 Sample 10 containing normalized log2 counts per million

SAM31 Sample 31 containing normalized log2 counts per million

SAM21 Sample 21 containing normalized log2 counts per million

SAM20 Sample 20 containing normalized log2 counts per million

SAM52 Sample 52 containing normalized log2 counts per million

SAM46 Sample 46 containing normalized log2 counts per million

SAM01 Sample 01 containing normalized log2 counts per million

SAM13 Sample 13 containing normalized log2 counts per million

SAM39 Sample 39 containing normalized log2 counts per million

SAM49 Sample 49 containing normalized log2 counts per million

SAM30 Sample 30 containing normalized log2 counts per million

SAM50 Sample 50 containing normalized log2 counts per million

SAM11 Sample 11 containing normalized log2 counts per million

SAM35 Sample 35 containing normalized log2 counts per million

SAM06 Sample 06 containing normalized log2 counts per million

SAM27 Sample 27 containing normalized log2 counts per million

SAM33 Sample 33 containing normalized log2 counts per million

SAM22 Sample 22 containing normalized log2 counts per million

SAM24 Sample 24 containing normalized log2 counts per million

SAM16 Sample 16 containing normalized log2 counts per million

SAM34 Sample 34 containing normalized log2 counts per million

SAM03 Sample 03 containing normalized log2 counts per million

SAM47 Sample 47 containing normalized log2 counts per million

SAM40 Sample 40 containing normalized log2 counts per million

SAM23 Sample 23 containing normalized log2 counts per million

`plot.empLCPM` *Plot the empirically derived LCPM cutoff*

Description

Plot method for class "empLCPM" that plots the 3rd order polynomial fit for the Spearman correlation between the expected and observed ERCC fold changes across log2 CPM abundance bins and highlights the empirically derived LCPM cutoff.

Usage

```
## S3 method for class 'empLCPM'
plot(x, main = "", col.trend = "purple", col.outlier = c("black", "red"), ...)
```

Arguments

`x`	A 'empLCPM' object from `getLowLcpmCutoff`.
`main`	An (optional) title for the plot.
`col.trend`	Color used for the 3rd order polynomial fit.
`col.outlier`	A vector specifying the default color for the points in the scatterplot, and the color for the outlier points. The default is to color all non-outlier points black and the outliers red.
`...`	Additional arguments are ignored.

Value

The function `plot.empLCPM` creates a plot that summarizes 3rd order polynomial fit for the Spearman correlation between the expected and observed ERCC fold changes across log2 CPM abundance bins. Vertical lines are used to indicate the optimal cutoff value, along with the lower and upper 95% bootstrap confidence interval highlighted by dashed vertical lines.

`print.empLCPM` *Print the empirically derived LCPM cutoff results*

Description

Print method for class "empLCPM".

Usage

```
## S3 method for class 'empLCPM'
print(x, ...)
```

Arguments

`x`	A 'empLCPM' object from `getLowLcpmCutoff`.
`...`	Additional arguments are ignored.

Value

The function `print.empLCPM` extracts the key results and returns a string of the summary.

`summary.empLCPM` *Summarizing the empirically derived LCPM cutoff*

Description

Summary method for class "empLCPM".

Usage

```
## S3 method for class 'empLCPM'
summary(object, ...)
```

Arguments

`object`	A 'empLCPM' object from `getLowLcpmCutoff`.
`...`	Additional arguments are ignored.

Value

The function `summary.empLCPM` extracts the key results and returns a string of the summary.

Index