Package ‘RnaSeqSampleSize’

May 27, 2024

Type Package
Title RnaSeqSampleSize
Version 2.14.0
Date 2021-12-02
Description RnaSeqSampleSize package provides a sample size calculation method based on negative binomial model and the exact test for assessing differential expression analysis of RNA-seq data. It controls FDR for multiple testing and utilizes the average read count and dispersion distributions from real data to estimate a more reliable sample size. It is also equipped with several unique features, including estimation for interested genes or pathway, power curve visualization, and parameter optimization.
License GPL (>= 2)
LazyLoad yes
Depends R (>= 4.0.0), ggplot2, RnaSeqSampleSizeData
Imports biomaRt, edgeR, heatmap3, matlab, KEGGREST, methods, grDevices, graphics, stats, Rcpp (>= 0.11.2), recount, ggpubr, SummarizedExperiment, tidyr, dplyr, tidyselect, utils
LinkingTo Rcpp
VignetteBuilder knitr
Suggests BiocStyle, knitr, testthat
biocViews ImmunoOncology, ExperimentalDesign, Sequencing, RNASeq, GeneExpression, DifferentialExpression
RoxygenNote 7.1.2
Encoding UTF-8

git_url https://git.bioconductor.org/packages/RnaSeqSampleSize
git_branch RELEASE_3_19
git_last_commit 9238b7e
git_last_commit_date 2024-04-30
Repository Bioconductor 3.19
Description

A function analyze data set

Usage

```r
analyze_dataset(
  expObj,
  expObjGroups = NULL,
  fdrCut = 0.05,
  subset = 0,
  repN = 2,
  useAllSamplesAsNegativeControl = FALSE
)
```
convertIdOneToOne

Arguments

expObj RangedSummarizedExperiment object.
expObjGroups sample groups. Should be a vector of 0 and 1. 0 as control samples.
fdrCut FDR cutoff to select differential genes.
subset RangedSummarizedExperiment object.
repN Number of replications.
useAllSamplesAsNegativeControl Logic. If true, will Use all samples in the obj as negative control.

Value

Figures and a list of result data.

Examples

1

----------
convertIdOneToOne  convertId
----------

Description

A function to convert ID based on the biomaRt package.

Usage

convertIdOneToOne(
  x,
  dataset = "hsapiens_gene_ensembl",
  filters = "uniprotswissprot",
  attributes = c(filters, "entrezgene_id"),
  verbose = FALSE
)

Arguments

x the Ids need to be converted.
dataset Dataset you want to use. To see the different datasets available within a biomaRt you can e.g. do: mart = useMart('ensembl'), followed by listDatasets(mart).
filters Filters (one or more) that should be used in the query. A possible list of filters can be retrieved using the function listFilters.
attributes Attributes you want to retrieve. A possible list of attributes can be retrieved using the function listAttributes.
verbose Logical. Indicate report extra information on progress or not.
Details
A function to convert ID based on the biomaRt package.

Value
A converted ID character with the same order of parameter x.

Examples
x<-c("Q04837","P0C0L4","P0C0L5","075379","Q13068","A2MYD1")
convertIdOneToOne(x, filters="uniprotswissprot", verbose=TRUE)

Description
A function to estimate the gene read count and dispersion distribution of RNA-seq data.

Usage
est_count_dispersion(
  counts,
  group = rep(1, NCOL(counts)),
  subSampleNum = 20,
  minAveCount = 1,
  convertId = FALSE,
  dataset = "hsapiens_gene_ensembl",
  filters = "hgnc_symbol"
)

Arguments
- **counts**: numeric matrix of read counts.
- **group**: vector or factor giving the experimental group/condition for each sample/library.
- **subSampleNum**: number of samples used to estimate distribution.
- **minAveCount**: Only genes with average read counts above this value are used in the estimation of distribution.
- **convertId**: logical, whether to convert th gene Id into entrez gene Id. If set as True, then dataset and filters parameter should also be set.
- **dataset**: Dataset you want to use. To see the different datasets available within a biomaRt you can e.g. do: mart = useMart('ensembl'), followed by listDatasets(mart).
- **filters**: Filters (one or more) that should be used in the query. A possible list of filters can be retrieved using the function listFilters.
Details

A function to estimate the gene read count and dispersion distribution of RNA-seq data.

Value

A DEGlist from edgeR package.

Examples

counts<-matrix(sample(1:1000,6000,replace=TRUE),ncol=6)
est_count_dispersion(counts=counts,group=rep(0,6))

Description

A function to estimate the power for differential expression analysis of RNA-seq data.

Usage

est_power(
  n,
  w = 1,
  k = 1,
  rho = 2,
  lambda0 = 5,
  phi0 = 1,
  alpha = 0.05,
  f,
  m = 20000,
  m1 = 200
)

Arguments

n Numer of samples.
w Ratio of normalization factors between two groups.k Ratio of sample size between two groups (Treatment/Control).
rho minimum fold changes for prognostic genes between two groups (Treatment/Control).
lambda0 Average read counts for prognostic genes.phi0 Dispersion for prognostic genes.alpha alpha level.f FDR level.m Total number of genes for testing.m1 Expected number of prognostic genes.
Value

Estimate power

Examples

\[
\text{est_power}(n=63, \text{rho}=2, \lambda_0=5, \phi_0=0.5, f=0.01)
\]

\[
\text{est_power(n=n, rho=rho, lambda0=lambda0, phi0=phi0, f=f)}
\]

Description

A function to estimate the power curve for differential expression analysis of RNA-seq data.

Usage

\[
\text{est_power_curve}(n, \text{w} = 1, \text{rho} = 2, \lambda_0 = 5, \phi_0 = 1, \alpha = 0.05, f = 0.05, \ldots)
\]

Arguments

- `n`: Numer of samples.
- `w`: Ratio of normalization factors between two groups.
- `rho`: minimum fold changes for prognostic genes between two groups (Treatment/Control).
- `lambda0`: Average read counts for prognostic genes.
- `phi0`: Dispersion for prognostic genes.
- `alpha`: alpha level.
- `f`: FDR level
- `...`: other parameters for `est_power` function.

Value

A list including parameters, sample size and power.

Examples

\[
\text{result1}<-\text{est_power_curve(n=63, f=0.01, rho=2, lambda0=5, phi0=0.5)}
\]

\[
\text{result2}<-\text{est_power_curve(n=63, f=0.05, rho=2, lambda0=5, phi0=0.5)}
\]

\[
\text{plot_power_curve(list(result1,result2))}
\]
Description

A function to estimate the power for differential expression analysis of RNA-seq data.

Usage

```r
est_power_distribution(
  n,
  f = 0.1,
  m = 10000,
  m1 = 100,
  w = 1,
  k = 1,
  rho = 2,
  repNumber = 100,
  dispersionDigits = 1,
  distributionObject,
  libSize,
  minAveCount = 5,
  maxAveCount = 2000,
  selectedGenes,
  pathway,
  species = "hsa",
  storeProcess = FALSE,
  countFilterInRawDistribution = TRUE,
  selectedGeneFilterByCount = FALSE,
  removedGene0Power = TRUE
)
```

Arguments

- **n**: Number of samples.
- **f**: FDR level.
- **m**: Total number of genes for testing.
- **m1**: Expected number of prognostic genes.
- **w**: Ratio of normalization factors between two groups.
- **k**: Ratio of sample size between two groups (Treatment/Control).
- **rho**: Minimum fold changes for prognostic genes between two groups (Treatment/Control).
- **repNumber**: Number of genes used in estimation of read counts and dispersion distribution.
- **dispersionDigits**: Digits of dispersion.
est_power_distribution

distributionObject
A DGEList object generated by est_count_dispersion function. RnaSeqSampleSizeData package contains 13 datasets from TCGA, you can set distributionObject as any one of "TCGA_BLCA","TCGA_BRCA","TCGA_CESC","TCGA_COAD","TCGA_HNSC","TCGA_KIRC","TCGA_LGG","TCGA_LUAD","TCGA_LUSC","TCGA_PRAD","TCGA_READ","TCGA_THCA","TCGA_UCEC" to use them.

libSize numeric vector giving the total count for each sample. If not specified, the libsize in distributionObject will be used.

minAveCount Minimal average read count for each gene. Genes with smaller read counts will not be used.

maxAveCount Maximal average read count for each gene. Genes with larger read counts will be taken as maxAveCount.

selectedGenes Optional. Name of interested genes. Only the read counts and dispersion distribution for these genes will be used in power estimation.

pathway Optional. ID of interested KEGG pathway. Only the read counts and dispersion distribution for genes in this pathway will be used in power estimation.

species Optional. Species of interested KEGG pathway.

storeProcess Logical. Store the power and n in sample size or power estimation process.

countFilterInRawDistribution Logical. If the count filter will be applied on raw count distribution. If not, count filter will be applied on libSize scaled count distribution.

selectedGeneFilterByCount Logical. If the count filter will be applied to selected genes when selectedGenes parameter was used.

removedGene0Power Logical. When selectedGenes or pathway are used, some genes may have read count less than minAveCount and will be removed by count filter. This parameter indicates if they will be used as 0 power in power estimation. If not, they will not be used in power estimation.

Details
A function to estimate the power for differential expression analysis of RNA-seq data.

Value
Average power or a list including count, distribution and power for each gene.

Examples
#Please note here the parameter repNumber was very small (2) to make the example code faster. #We suggest repNumber should be at least set as 100 in real analysis.
est_power_distribution(n=65,f=0.01,rho=2,distributionObject="TCGA_READ",repNumber=2)
#Power estimation based on some interested genes. We use storeProcess=TRUE to return the details for all selected genes.
selectedGenes<-c("A1BG","A2BP1","A2M","A4GALT","AAAS")
powerDistribution<-est_power_distribution(n=65,f=0.01,rho=2,distributionObject="TCGA_READ", selectedGenes=selectedGenes,minAveCount=1,storeProcess=TRUE,repNumber=2)
**optimize_parameter**

```r
str(powerDistribution)
mean(powerDistribution$power)
# Power estimation based on genes in interested pathway
## Not run:
powerDistribution<-est_power_distribution(n=65,f=0.01,rho=2,distributionObject="TCGA_READ",pathway="00010",minAveCount=1,storeProcess=TRUE,repNumber=2)
mean(powerDistribution$power)
## End(Not run)
```

---

### Description

A function to optimize the parameters in power or sample size estimation.

### Usage

```r
optimize_parameter(
  fun = est_power,
  opt1, opt2,
  opt1Value, opt2Value,
  main,
  ...
)
```

### Arguments

- **fun**: function to be optimized, can be `est_power`, `sample_size`.
- **opt1**: parameter1 to be optimized.
- **opt2**: parameter2 to be optimized.
- **opt1Value**: values of parameter1 to be optimized.
- **opt2Value**: values of parameter2 to be optimized.
- **main**: Title of optimization result figure.
- **...**: Other parameters for optimized function.

### Details

A function to optimize the parameters in power or sample size estimation.

### Value

A power or sample size matrix, generated by different pair of two parameters.
Examples

# Optimization for power estimation
result<-optimize_parameter(fun=est_power,opt1="n",opt2="lambda0",opt1Value=c(3,5,10,15,20),
                         opt2Value=c(1:5,10,20))
# Optimization for sample size estimation
result<-optimize_parameter(fun=sample_size,opt1="lambda0",opt2="phi0",opt1Value=c(1,3),
                         opt2Value=c(1.5,2),power=0.8)

plot_gene_counts_range

Description

A function to plot proportion of genes in different count range.

Usage

plot_gene_counts_range(expObj, targetSize = NULL)

Arguments

expObj RangedSummarizedExperiment object or an expression matrix.
targetSize The target library size to scale to. Will not do scale if set as NULL.

Value

A barplot.

Examples

1

plot(mappedReads_percent)

Description

A function to plot percent of mapped reads in total reads. Only RangedSummarizedExperiment object generated by recount package have total reads information to this.

Usage

plot(mappedReads_percent(expObj, groupVar = NULL)
### Arguments
- `expObj` RangedSummarizedExperiment object generated by recount package.
- `groupVar` variable name in colData(expObj) to be used to group the samples to make box-plot.

### Value
A barplot or boxplot.

### Examples
```r
plot_power_curve()
```

### Description
A function to plot power curves based on the result of `sample_size` or `est_power_curve` function.

### Usage
```r
plot_power_curve(
    result,  # the result of `sample_size` or `est_power_curve` function. The storeProcess parameter should be set as True when performing `sample_size` function. If you want to plot more than one curves in the same figure, the results from `sample_size` function should first be combined into a new list. At most five curves were allowed in one figure.
    cexLegend = 1,  # the cex for legend.
    type = "b",  # 1-character string giving the type of plot desired. The following values are possible, for details, see plot.
    xlab = "Sample Size",  # x-axis label.
    ylab = "Power",  # y-axis label.
    pch = 16,  # point character.
    lwd = 3,  # line width.
    las = 1,  # line angle.
    cex = 1.5,  # character expansion.
    main = "Power Curve",  # main title.
    col = "red"  # color.
)
```
sample_size

xlab a label for the x axis, defaults to a description of x.
ylab a label for the y axis, defaults to a description of y.
pch Either an integer specifying a symbol or a single character to be used as the default in plotting points.
lwd The line width.
las Numeric in 0,1,2,3; the style of axis labels.
cex A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default.
main a main title for the plot
col The line color.

Value
A power curve plot.

Examples

result1<-sample_size(rho=2,phi0=1,lambda0=1,f=0.01,power=0.8,m=20000,m1=500,
showMessage=TRUE,storeProcess=TRUE)
result2<-sample_size(rho=4,phi0=1,lambda0=1,f=0.01,power=0.8,m=20000,m1=500,
showMessage=TRUE,storeProcess=TRUE)
plot_power_curve(list(result1,result2))

Description
A function to estimate the sample size for differential expression analysis of RNA-seq data.

Usage

sample_size(
  power = 0.8,
  m = 20000,
  m1 = 200,
  f = 0.1,
  k = 1,
  w = 1,
  rho = 2,
  lambda0 = 5,
  phi0 = 1,
  showMessage = FALSE,
  storeProcess = FALSE
)
sample_size_distribution

Arguments

- **power**: Power to detect prognostic genes.
- **m**: Total number of genes for testing.
- **m1**: Expected number of prognostic genes.
- **f**: FDR level
- **k**: Ratio of sample size between two groups (Treatment/Control).
- **w**: Ratio of normalization factors between two groups.
- **rho**: Minimum fold changes for prognostic genes between two groups (Treatment/Control).
- **lambda0**: Average read counts for prognostic genes.
- **phi0**: Dispersion for prognostic genes.
- **showMessage**: Logical. Display the message in the estimation process.
- **storeProcess**: Logical. Store the power and n in sample size or power estimation process.

Details

A function to estimate the sample size for differential expression analysis of RNA-seq data.

Value

Estimate sample size or a list including parameters and sample size in the process.

Examples

```r
power <- 0.8; rho <- 2; lambda0 <- 5; phi0 <- 0.5; f <- 0.01
sample_size(power = power, f = f, rho = rho, lambda0 = lambda0, phi0 = phi0)
```

Description

A function to estimate the sample size based on read counts and dispersion distribution in real data.

Usage

```r
sample_size_distribution(
    power = 0.8,
    m = 10000,
    m1 = 100,
    f = 0.1,
    k = 1,
    w = 1,
)```
rho = 2,
showMessage = FALSE,
storeProcess = FALSE,
distributionObject,
libSize,
minAveCount = 5,
maxAveCount = 2000,
repNumber = 100,
dispersionDigits = 1,
selectedGenes,
pathway,
species = "hsa",
countFilterInRawDistribution = TRUE,
selectedGeneFilterByCount = FALSE
)

Arguments

power Power to detect prognostic genes.
m Total number of genes for testing.
m1 Expected number of prognostic genes.
f FDR level
k Ratio of sample size between two groups (Treatment/Control).
w Ratio of normalization factors between two groups.
rho minimum fold changes for prognostic genes between two groups (Treatment/Control).
showMessage Logical. Display the message in the estimation process.
storeProcess Logical. Store the power and n in sample size or power estimation process.
libSize numeric vector giving the total count for each sample. If not specified, the libsize in distributionObject will be used.
minAveCount Minimal average read count for each gene. Genes with smaller read counts will not be used.
maxAveCount Maximal average read count for each gene. Genes with larger read counts will be taken as maxAveCount.
repNumber Number of genes used in estimation of read counts and dispersion distribution.
dispersionDigits Digits of dispersion.
selectedGenes Optional. Name of interested genes. Only the read counts and dispersion distribution for these genes will be used in power estimation.
pathway Optional. ID of interested KEGG pathway. Only the read counts and dispersion distribution for genes in this pathway will be used in power estimation.
**sample_size_distribution**

- **species** Optional. Species of interested KEGG pathway.
- **countFilterInRawDistribution**
  Logical. If the count filter will be applied on raw count distribution. If not, count filter will be applied on libSize scaled count distribution.
- **selectedGeneFilterByCount**
  Logical. If the count filter will be applied to selected genes when selectedGenes parameter was used.

**Details**

A function to estimate the sample size based on read counts and dispersion distribution in real data.

**Value**

Estimate sample size or a list including parameters and sample size in the process.

**Examples**

#Please note here the parameter repNumber was very small (5) to make the example code faster.
#We suggest repNumber should be at least set as 100 in real analysis.
sample_size_distribution(power=0.8,f=0.01,distributionObject="TCGA_READ",repNumber=5, showMessage=TRUE)
Index

analyze_dataset, 2
convertIdOneToOne, 3
est_count_dispersion, 4
est_power, 5
est_power_curve, 6, 11
est_power_distribution, 7
optimize_parameter, 9
plot_gene_counts_range, 10
plot_mappedReads_percent, 10
plot_power_curve, 11
sample_size, 11, 12
sample_size_distribution, 13