Package ‘RnaSeqSampleSize’

January 31, 2017

Type Package
Title RnaSeqSampleSize
Version 1.6.0
Date 2015-04-12
Author Shilin Zhao, Chung-I Li, Yan Guo, Quanhu Sheng, Yu Shyr
Maintainer Shilin Zhao <zhaoshilin@gmail.com>
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
License GPL (>= 2)
LazyLoad yes
LazyData yes
Depends R (>= 2.10), RnaSeqSampleSizeData
Imports biomaRt,edgeR,heatmap3,matlab,KEGGREST,Rcpp (>= 0.11.2)
LinkingTo Rcpp
VignetteBuilder knitr
Suggests BiocStyle,knitr
biocViews ExperimentalDesign, Sequencing, RNAses, GeneExpression,
DifferentialExpression
NeedsCompilation yes

R topics documented:

convertIdOneToOne .............................................. 2
est_count_dispersion ........................................ 2
est_power ......................................................... 3
est_power_curve ............................................... 4
est_power_distribution ....................................... 5
optimize_parameter ........................................... 6
plot_power_curve ............................................. 7
sample_size ..................................................... 8
sample_size_distribution .................................... 9

Index .............................................................. 11
convertIdOneToOne  convertId

Description
A function to convert ID based on the biomaRt package.

Usage
convertIdOneToOne(x, dataset = "hsapiens_gene_ensembl",
    filters = "uniprot_swissprot", attributes = c(filters, "entrezgene"),
    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", "O75379", "Q13068", "A2MYD1", "P60709", "P30462", "P30475", "P30479")
convertIdOneToOne(x, filters="uniprot_swissprot", verbose=TRUE)

est_count_dispersion  est_count_dispersion

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")
\textbf{\textit{est_power}}

\textbf{Arguments}

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

\textbf{Details}

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

\textbf{Value}

A \texttt{DEGlist} from \texttt{edgeR} package.

\textbf{Examples}

\begin{verbatim}
counts <- matrix(sample(1:1000, 6000, replace=TRUE), ncol=6)
est_count_dispersion(counts=counts, group=rep(0,6))
\end{verbatim}

\textbf{Description}

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

\textbf{Usage}

\begin{verbatim}
est_power(n, w = 1, rho = 2, lambda0 = 5, phi0 = 1, alpha = 0.05, f, m = 20000, m1 = 200)
\end{verbatim}

\textbf{Arguments}

- \texttt{n}: Number of samples.
- \texttt{w}: Ratio of normalization factors between two groups.
- \texttt{rho}: minimum fold changes for prognostic genes between two groups.
- \texttt{lambda0}: Average read counts for prognostic genes.
- \texttt{phi0}: Dispersion for prognostic genes.
- \texttt{alpha}: alpha level.
- \texttt{f}: FDR level.
- \texttt{m}: Total number of genes for testing.
- \texttt{m1}: Expected number of prognostic genes.
est_power_curve

Value

Estimate power

Examples

```r
e<-63; rho<2; lambda0<5; phi0<0.5; f<0.01
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

```r
est_power_curve(n, w = 1, rho = 2, lambda0 = 5, phi0 = 1,
alpha = 0.05, f = 0.05, ...)
```

Arguments

- `n`: Numer of samples.
- `w`: Ratio of normalization factors between two groups.
- `rho`: Minimum fold changes for prognostic genes between two groups.
- `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

```r
## Not run:
result1<-est_power_curve(n=63, f=0.01, rho=2, lambda0=5, phi0=0.5)
result2<-est_power_curve(n=63, f=0.05, rho=2, lambda0=5, phi0=0.5)
plot_power_curve(list(result1,result2))
## End(Not run)
```
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,
rho = 2, repNumber = 100, dispersionDigits = 1, distributionObject,
libSize, minAveCount = 5, maxAveCount = 2000, seed = 123, selectedGenes,
pathway, species = "hsa", storeProcess = FALSE,
countFilterInRawDistribution = TRUE, selectedGeneFilterByCount = FALSE,
removedGene0Power = TRUE)
```

Arguments

- `n` Numer 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.
- `rho` Minimum fold changes for prognostic genes between two groups.
- `repNumber` Number of genes used in estimation of read counts and dispersion distribution.
- `dispersionDigits` Digits of dispersion.
- `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`.
- `seed` Optional. A integer, seed for randomly selecting genes.
- `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.
optimize_parameter

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 estitamate 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
## Not run:
#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.
est_power_distribution(n=65,f=0.01,rho=2,distributionObject="TCGA_READ",repNumber=5)

#Power estimation based on some interested genes. We use storeProcess=TRUE to return the details for all selectedGenes<names(TCGA_READ$pseudo.counts.mean)[c(1,3,5,7,9,12:30)]>
powerDistribution<est_power_distribution(n=65,f=0.01,rho=2,distributionObject="TCGA_READ",selectedGenes=selectedGenes,minAveCount=1,storeProcess=TRUE)
str(powerDistribution)
mean(powerDistribution$power)

#Power estimation based on genes in interested pathway
dataPathway<-pathwaysReader(pathway="00010")
powerDistribution<est_power_distribution(n=65,f=0.01,rho=2,distributionObject="TCGA_READ",pathway=00010,minAveCount=1,storeProcess=TRUE)
mean(powerDistribution$power)

## End(Not run)

---

optimize_parameter

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

Usage
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**

```r
# 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
## Not run:
result <- optimize_parameter(fun = sample_size, opt1 = "lambda0", opt2 = "phi0", opt1Value = c(1, 3, 5), opt2Value = c(1.5, 2, 3), power = 0.8)
## End(Not run)
```

**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, cexLegend = 1, type = "b", xlab = "Sample Size", ylab = "Power", pch = 16, lwd = 3, las = 1, cex = 1.5,
main = "Power Curve", col = "red")
```

**Arguments**

- `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` the cex for legend.
sample_size

type 1-character string giving the type of plot desired. The following values are possible, for details, see plot: "p" for points, "l" for lines, "b" for both points and lines, "c" for empty points joined by lines, "o" for overplotted points and lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.

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, see also title.

col The line color.

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))

sample_size sample_size

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 = 500, f = 0.1, k = 1, w = 1,
rho = 2, lambda0 = 5, phi0 = 1, showMessage = FALSE, storeProcess = FALSE)

Arguments

power Power to detecte 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.
w Ratio of normalization factors between two groups.
rho minimum fold changes for prognostic genes between two groups.
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.
sample_size_distribution

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, seed = 123, selectedGenes, pathway, species = "hsa", countFilterInRawDistribution = TRUE, selectedGeneFilterByCount = FALSE)
```

Arguments

- `power`: Power to detecte 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.
- `w`: Ratio of normalization factors between two groups.
- `rho`: minimum fold changes for prognostic genes between two groups.
- `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.
sample_size_distribution

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.
seed         Optional. A integer, seed for randomly selecting genes.
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.
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
## Not run:
## 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)

## End(Not run)
Index

convertIdOneToOne, 2
est_count_dispersion, 2
est_power, 3
est_power_curve, 4, 7
est_power_distribution, 5
optimize_parameter, 6
plot, 8
plot_power_curve, 7
sample_size, 7, 8
sample_size_distribution, 9
title, 8