Package ‘DGEAR’

August 30, 2023

Type Package

Title Differential Gene Expression Analysis with R

Version 0.1.3

Author Koushik Bardhan [aut, cre, ctb]
(<`https://orcid.org/0009-0002-8846-8347`>),
Chiranjib Sarkar [aut, ths] (<`https://orcid.org/0000-0003-1536-7449`>)

Maintainer Koushik Bardhan <`koushikbardhan2000@gmail.com`>

Description Analyses gene expression data derived from experiments to detect differentially expressed genes by employing the concept of majority voting with five different statistical models. It includes functions for differential expression analysis, significance testing, etc. It simplifies the process of uncovering meaningful patterns and trends within gene expression data, aiding researchers in downstream analysis. Boyer, R.S., Moore, J.S. (1991) <doi:10.1007/978-94-011-3488-0_5>.

License GPL-3

Encoding UTF-8

LazyData true

Imports DescTools

RoxygenNote 7.2.3

Depends R (>= 2.10)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2023-08-30 19:30:10 UTC

R topics documented:

`DGEAR` *Differential Gene Expression Analysis with R*

Description

Main function which incorporates results from five statistical models and detects DEGs through majority voting

Usage

```
DGEAR(dataframe, con1, con2, exp1, exp2, alpha, votting_cutoff)
```

Arguments

`dataframe`	A matrix containing the gene expression data
`con1`	Starting column of the control of the expression data
`con2`	Ending column of the control of the expression data
`exp1`	Starting column of the experiment of the expression data
`exp2`	Ending column of the experiment of the expression data
`alpha`	Value of significance level ranging from 0 to 1 (0.05 states 5 % significance)(Default = 0.05).
`votting_cutoff`	A numeric value serves as Majority voting (Default = 2)

Details

To use this tool the necessary parameters are con1 = Control start column, con2 = Control end column, exp1 = Experiment start column, exp2 = Experiment end column, alpha = Value of significance level, voting_cutoff = Majority voting value (not more than 5, since there are 5 statistical methods which take part in the majority voting)

Value

A matrix containing Differentially Expressed Genes(DEGs) detected

Examples

```
library(DGEAR)
data("gene_exp_data")
DGEAR(dataframe = gene_exp_data, con1 = 1, con2 = 10,
  exp1 = 11, exp2 = 20, alpha = 0.05, votting_cutoff = 2)
```

`gene_exp_data` *A dataset containing gene expression data*

Description

This dataset contains simulated gene expression data for analysis.

Usage

```
gene_exp_data
```

Format

A data frame with 20 rows and 20 columns, where columns 1 to 10 being control and 11 to 20 being experiment.

Examples

```
# Data will be loaded with lazy loading and can be accessible when needed.
data("gene_exp_data")
head(gene_exp_data)
```

`perform_anova` *Function for ANOVA One-Way Test*

Description

Function for ANOVA One-Way Test

Usage

```
perform_anova(datafile, con, exp, alpha = 0.05)
```

Arguments

`datafile`	A matrix containing the gene expression data
`con`	A data frame or matrix containing the expression values for the control.
`exp`	A data frame or matrix containing the expression values for the experiment.
`alpha`	Value of significance level ranging from 0 to 1 (default = 0.05 states 5 % significance).

Value

A data frame containing values for statistic score, p-values etc for each gene being tested.

Examples

```
library(DGEAR)
data("gene_exp_data")
data = read_and_preprocess_data(datafile = gene_exp_data, con1=1,con2=10,exp1=11,exp2=20)
perform_anova(datafile = data$datafile, con= data$con, exp= data$exp)
```

`perform_dunnett_test` *Function for Dunnett's Test*

Description

Function for Dunnett's Test

Usage

```
perform_dunnett_test(datafile, con, exp, alpha = 0.05)
```

Arguments

`datafile`	A matrix containing the gene expression data
`con`	A data frame or matrix containing the expression values for the control.
`exp`	A data frame or matrix containing the expression values for the experiment.
`alpha`	Value of significance level ranging from 0 to 1 (default = 0.05 states 5 % significance).

Value

A data frame containing values for statistic score, p-values etc for each gene being tested.

Examples

```
library(DGEAR)
data("gene_exp_data")
data = read_and_preprocess_data(datafile = gene_exp_data, con1=1,con2=10,exp1=11,exp2=20)
perform_dunnett_test(datafile = data$datafile, con= data$con, exp= data$exp)
```

`perform_h_test` *Function for Half's-T-Test Analysis*

Description

Function for Half's-T-Test Analysis

Usage

```
perform_h_test(con, exp, alpha = 0.05, FC)
```

Arguments

`con`	A data frame or matrix containing the expression values for the control.
`exp`	A data frame or matrix containing the expression values for the experiment.
`alpha`	Value of significance level ranging from 0 to 1 (default = 0.05 states 5 % significance).
`FC`	An array or list containing fold change values for each gene, calculated by

Value

A data frame containing values for statistic score, p-values etc for each gene being tested.

Examples

```
library(DGEAR)
data("gene_exp_data")
data = read_and_preprocess_data(datafile = gene_exp_data, con1=1,con2=10,exp1=11,exp2=20)
perform_h_test(con= data$con, exp= data$exp, FC = data$FC)
```

`perform_t_test` *Function for t-Test Analysis*

Description

Function for t-Test Analysis

Usage

```
perform_t_test(con, exp, alpha = 0.05)
```

Arguments

`con`	A data frame or matrix containing the expression values for the control.
`exp`	A data frame or matrix containing the expression values for the experiment.
`alpha`	Value of significance level ranging from 0 to 1 (default = 0.05 states 5 % significance).

Value

A data frame containing values for statistic score, p-values etc for each gene being tested.

Examples

```
library(DGEAR)
data("gene_exp_data")
data = read_and_preprocess_data(datafile = gene_exp_data, con1=1,con2=10,exp1=11,exp2=20)
perform_t_test(con= data$con, exp= data$exp)
```

`perform_wilcox_test` *Function for Wilcoxon-Mann-Whitney U-Test*

Description

Function for Wilcoxon-Mann-Whitney U-Test

Usage

```
perform_wilcox_test(con, exp, alpha = 0.05)
```

Arguments

`con`	A data frame or matrix containing the expression values for the control.
`exp`	A data frame or matrix containing the expression values for the experiment.
`alpha`	Value of significance level ranging from 0 to 1 (default = 0.05 states 5 % significance).

Value

A data frame containing values for statistic score, p-values etc for each gene being tested.

Examples

```
library(DGEAR)
data("gene_exp_data")
data = read_and_preprocess_data(datafile = gene_exp_data, con1=1,con2=10,exp1=11,exp2=20)
perform_wilcox_test(con= data$con, exp= data$exp)
```

`read_and_preprocess_data` *Function to read data and perform initial pre-processing*

Description

Function to read data and perform initial pre-processing

Usage

```
read_and_preprocess_data(
  datafile,
  con1,
  con2,
  exp1,
  exp2,
  alpha = 0.05,
  votting_cutoff = 2
)
```

Arguments

`datafile`	A matrix or data frame containing gene expression data
`con1`	Starting column of the control of the expression data
`con2`	Ending column of the control of the expression data
`exp1`	Starting column of the experiment of the expression data
`exp2`	Ending column of the experiment of the expression data
`alpha`	Value of significance level ranging from 0 to 1 (0.05 states 5 % significance)(Default = 0.05).
`votting_cutoff`	A numeric value serves as Majority voting (Default = 2)

Value

A large list containing the data file and the input values

Examples

```
data("gene_exp_data")
read_and_preprocess_data(datafile = gene_exp_data, con1=1,con2=10,exp1=11,exp2=20)
```

Index