Package 'RNAsense'

November 12, 2026

Type Package

Title Analysis of Time-Resolved RNA-Seq Data

Version 1.26.0

Description RNA-sense tool compares RNA-seq time curves in two experimental conditions, i.e. wild-type and mutant, and works in three steps. At Step 1, it builds expression profile for each transcript in one condition (i.e. wild-type) and tests if the transcript abundance grows or decays significantly. Dynamic transcripts are then sorted to non-overlapping groups (time profiles) by the time point of switch up or down. At Step 2, RNA-sense outputs the groups of differentially expressed transcripts, which are up- or downregulated in the mutant compared to the wild-type at each time point. At Step 3, Correlations (Fisher's exact test) between the outputs of Step 1 (switch up- and switch down- time profile groups) and the outputs of Step2 (differentially expressed transcript groups) are calculated. The results of the correlation analysis are printed as two-dimensional color plot, with time profiles and differential expression groups at y- and x-axis, respectively, and facilitates the biological interpretation of the data.

License GPL-3

Encoding UTF-8

LazyData false

Depends R (>= 3.6)

RoxygenNote 6.1.1

BugReports https://github.com/marcusrosenblatt/RNAsense

biocViews RNASeq, GeneExpression, DifferentialExpression

Imports ggplot2, parallel, NBPSeq, qvalue, SummarizedExperiment, stats, utils, methods

Suggests knitr, rmarkdown

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/RNAsense

git_branch RELEASE_3_23

git_last_commit c8168ce

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Marcus Rosenblatt [cre],
Gao Meijang [aut],
Helge Hass [aut],
Daria Onichtchouk [aut]

Maintainer Marcus Rosenblatt <marcus.rosenblatt@gmail.com>

Contents

combineResults 2
getFC 3
getSwitch 4
MZsox 5
outputGeneTables 5
plotSSGS 7

Index **8**

`combineResults` *Combine results*

Description

Results of switch and fold change analysis are collected in one data.frame

Usage

```
combineResults(myresultSwitch = resultSwitch, myresultFC = resultFC,
  nrcores = 1)
```

Arguments

`myresultSwitch`	data.frame, output of getSwitch
`myresultFC`	data.frame, output of getFC
`nrcores`	Numeric, Number of cores for parallelization, default 1 for no parallelization

Value

Data.frame containing information on switch and fold change detection for each gene

Author(s)

Marcus Rosenblatt, <marcus.rosenblatt@fdm.uni-freiburg.de>

Examples

```
data(MZsox)
mydata <- MZsox[seq(1,nrow(MZsox), by=10),]
resultFC <- getFC(dataset = mydata,
myanalyzeConditions = c("WT", "MZsox"),
cores = 1,
mytimes = c(2.5,3,3.5,4,4.5,5,5.5,6))
resultSwitch <- getSwitch(dataset = mydata,
experimentStepDetection = "WT",
cores = 1,
mytimes = c(2.5,3,3.5,4,4.5,5,5.5,6))
combineResults(resultSwitch, resultFC)
```

`getFC` *Detect fold changes*

Description

For each gene and for each time point, RNA-seq count data is analyzed for fold changes between two experimental conditions. This functions bases on functions from the R package NBPSeq package for fold change analysis

Usage

```
getFC(dataset = mydata, myanalyzeConditions = analyzeConditions,
  cores = 1, mytimes = times)
```

Arguments

`dataset`	Object of class SummarizedExperiment, output of SummarizedExperiment, as assays use a numeric matrix with your RNAseq count data, rows correspond to different genes, columns correspond to different experiments, as rowData provide a DataFrame with columns name (geneID) and genename (the gene names), as colData provide a DataFrame with columns condition, time and replicate
`myanalyzeConditions`	Character vector, Name of experimental conditions
`cores`	Numeric, Number of cores for parallelization, default 1 for no parallelization
`mytimes`	Numeric vector, Time points of the time-resolved RNA-seq data

Value

Data.frame containing gene names, log fold change and p-values calculated from NBPSeq, each gene appears as often as available time points

Author(s)

Marcus Rosenblatt, <marcus.rosenblatt@fdm.uni-freiburg.de>

Examples

```
data(MZsox)
mydata <- MZsox[seq(1,nrow(MZsox), by=10),]
resultFC <- getFC(dataset = mydata,
myanalyzeConditions = c("WT", "MZsox"),
cores = 1,
mytimes = c(2.5,3,3.5,4,4.5,5,5.5,6))
```

`getSwitch` *Detect switching genes*

Description

For each gene, time-resolved RNA-seq measurements are analyzed for occurence of switches (up or down)

Usage

```
getSwitch(dataset = mydata, experimentStepDetection = "WT",
  pValueSwitch = 0.05, cores = 1, mytimes = times)
```

Arguments

`dataset`	Object of class SummarizedExperiment, output of SummarizedExperiment, as assays use a numeric matrix with your RNAseq count data, rows correspond to different genes, columns correspond to different experiments, as rowData provide a DataFrame with columns name (geneID) and genename (the gene names), as colData provide a DataFrame with columns condition, time and replicate
`experimentStepDetection`	Character, Name of condition for which switch detection is performed
`pValueSwitch`	Numeric, A threshold for counting cells as being invaded or not. When cells move towards negative z-direction, threshold should be negative.
`cores`	Numeric, Number of cores for parallelization, default 1 for no parallelization
`mytimes`	Numeric vector, Time points of the time-resolved RNA-seq data

Value

Data.frame containing gene names and results of switch detection, information about switch time point and direction

Author(s)

Marcus Rosenblatt, <marcus.rosenblatt@fdm.uni-freiburg.de>

Examples

```
data(MZsox)
mydata <- MZsox[seq(1,nrow(MZsox), by=10),]
resultSwitch <- getSwitch(dataset = mydata,
experimentStepDetection = "WT",
cores = 1,
mytimes = c(2.5,3,3.5,4,4.5,5,5.5,6))
```

`MZsox` *Time resolved RNA seq data for early zygotic development of zebra fish.*

Description

A dataset containing RNA seq count data for two experimental conditions at 8 different time points including replicates.

Usage

```
MZsox
```

Format

A SummarizedExperiment object with 15775 rows and 40 columns. Arguments rowData and colData give covariate information on the count data as follows:

- rowData: name, name of the gene
- rowData: genename, identifier of gene
- colData: condition, name of condition
- colData: time, measurement time in hours post fertilization
- colData: replicate, identifier of replicate

`outputGeneTables` *Output gene tables*

Description

Output information on switching genes (up/down) in tabular format (gene identifier/gene name) are created as .txt file and written to the specified working directory. Two of the generated files (geneNamelist) contain gene lists with gene name for genes that switch up and down respectively. The other two (genelist) contain exactly the same output but with gene identifiers instead of gene names depending on what you prefer for further analysis. Each column corresponds to a combination of switch time point, fold change direction and time point of fold change. All genes for which fold change was detected at the indicated time point and switch was detected at the indicated time point

are listed in the corresponding column. Note that a single gene may appear multiple times. The fifth .txt file (switchList) contains information on detected switches in a different format. The output consists of table with six columns with each row corresponding to one gene. Detected switches are indicated by 1, -1 and 0 for switch up, switch down and no switch, respectively. If a switch was detected, the column timepoint indicated the corresponding time point of switch detection.

Usage

```
outputGeneTables(myresultCombined = resultCombined, mytimes = times,
  myanalyzeConditions = analyzeConditions, mywd = NULL)
```

Arguments

myresultCombined	data.frame, output of combineResults
mytimes	Numeric vector, Time points of the time-resolved RNA-seq data
myanalyzeConditions	character vector, the conditions that were analyzed
mywd	character, working directory to which results will be written, if NULL the current working directory is used

Value

Working directory where results have been written to

Author(s)

Marcus Rosenblatt, <marcus.rosenblatt@fdm.uni-freiburg.de>

Examples

```
library(ggplot2)
data(MZsox)
mydata <- MZsox[seq(1,nrow(MZsox), by=10),]
resultFC <- getFC(dataset = mydata,
myanalyzeConditions = c("WT", "MZsox"),
cores = 1,
mytimes = c(2.5,3,3.5,4,4.5,5,5.5,6))
resultSwitch <- getSwitch(dataset = mydata,
experimentStepDetection = "WT",
cores = 1,
mytimes = c(2.5,3,3.5,4,4.5,5,5.5,6))
resultCombined <- combineResults(resultSwitch, resultFC)
outputGeneTables(resultCombined,
mytimes = c(2.5,3,3.5,4,4.5,5,5.5,6),
myanalyzeConditions = c("WT", "MZsox"))
```

plotSSGS *plot SSGS gene classes*

Description

Genes are sorted into groups with respect to switch time and time point of fold change detection. For each group, results of wild type and knockdown-condition are compared by means of fisher's exact test to show whether the knocked down gene enhances or suppresses the respective gene group.

Usage

```
plotSSGS(myresultCombined = resultCombined, mytimes = times,
  myanalyzeConditions = analyzeConditions)
```

Arguments

myresultCombined	data.frame, output of combineResults
mytimes	Numeric vector, Time points of the time-resolved RNA-seq data
myanalyzeConditions	character vector, the conditions that were analyzed

Value

SSGS color plot in ggplot format

Author(s)

Marcus Rosenblatt, <marcus.rosenblatt@fdm.uni-freiburg.de>

Examples

```
library(ggplot2)
data(MZsox)
mydata <- MZsox[seq(1,nrow(MZsox), by=10),]
resultFC <- getFC(dataset = mydata,
myanalyzeConditions = c("WT", "MZsox"),
cores = 1,
mytimes = c(2.5,3,3.5,4,4.5,5,5.5,6))
resultSwitch <- getSwitch(dataset = mydata,
experimentStepDetection = "WT",
cores = 1,
mytimes = c(2.5,3,3.5,4,4.5,5,5.5,6))
resultCombined <- combineResults(resultSwitch, resultFC)
plotSSGS(myresultCombined = resultCombined,
mytimes = c(2.5,3,3.5,4,4.5,5,5.5,6),
myanalyzeConditions = c("WT", "MZsox"))
```

Index

∗ **datasets**
MZsox, 5

combineResults, 2, *6, 7*

DataFrame, *3, 4*

getFC, *2*, 3
getSwitch, *2*, 4

MZsox, 5

outputGeneTables, 5

plotSSGS, 7

SummarizedExperiment, *3–5*