Package 'splineTimeR'

November 12, 2026

Type Package

Title Time-course differential gene expression data analysis using spline regression models followed by gene association network reconstruction

Version 1.40.0

Date 2020-07-16

Author Agata Michna

Maintainer Herbert Braselmann <hbraselmann@online.de>, Martin Selmansberger <martin.selmansberger@helmholtz-muenchen.de>

Depends R (>= 3.3), Biobase, igraph, limma, GSEABase, gtools, splines, GeneNet (>= 1.2.13), longitudinal (>= 1.1.12), FIs

Description This package provides functions for differential gene expression analysis of gene expression time-course data. Natural cubic spline regression models are used. Identified genes may further be used for pathway enrichment analysis and/or the reconstruction of time dependent gene regulatory association networks.

License GPL-3

biocViews GeneExpression, DifferentialExpression, TimeCourse, Regression, GeneSetEnrichment, NetworkEnrichment, NetworkInference, GraphAndNetwork

VignetteBuilder knitr

Suggests knitr

NeedsCompilation no

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

git_branch RELEASE_3_23

git_last_commit e0685c8

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

networkProperties 2
pathEnrich 3
splineDiffExprs 5
splineNetRecon 7
splinePlot 8
TCsimData 10

Index **12**

`networkProperties` *Scale-free properties of a network*

Description

Function plots degree distribution of nodes in a given network.

Usage

```
networkProperties(igr)
```

Arguments

`igr`	an object or list of objects of class `igraph`

Details

Biological networks are thought to be scale-free. Scale-free networks follow a power-law distribution of the degrees of nodes in the network. This distribution is characterised by the degree exponent gamma, which for biological networks ranges between 2 and 3.

The function calculates the degree exponent(s) of given network(s) in comparison with degree exponents of biological networks derived from Reactome and BioGRID repositories. Both of these networks are built using functional interaction pairs extracted from mentioned repositories and provided in `FIs` data package (see example).

For each given `igraph` three types of plots are created: empirical cumulative distribution, degree distribution and power-law degree distribution on log-log scale with fitted trend line.

Value

A summary table containing number of nodes, number of edges and degree exponents for each given network.

Additionally two .pdf files are created. One containing empirical cumulative distribution frequency plots together with degree distributions and second with plots of power-law degree distribution on log-log scale.

Author(s)

Agata Michna, Martin Selmansberger

References

Barabasi, A. L. and Albert, R. (1999). Emergence of scaling in random networks. *Science* 286, 509-512.

Barabasi, A. L. and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.* 5, 101-113.

See Also

FIs

Examples

```
## load "eSetObject" containing simulated time-course data
data(TCsimData)

## reconstruct gene association networks from time-course data
igr <- splineNetRecon(eSet = TCsimData, treatmentType = "T2", cutoff.ggm = c(0.8,0.9))

## check for scale-free properties of reconstructed networks (igraphs)
scaleFreeProp <- networkProperties(igr)
head(scaleFreeProp)

## the functional interaction pairs provided in FIs data package
library(FIs)
data(FIs)
names(FIs)
head(FIs$FIs_Reactome)
head(FIs$FIs_BioGRID)
```

---

pathEnrich *Pathway enrichment analysis*

Description

Function performs a pathway enrichment analysis of a definied set of genes.

Usage

```
pathEnrich(geneList, geneSets, universe=NULL)
```

Arguments

geneList	vector of gene names to be used for pathway enrichment
geneSets	"GeneSetColletion" object with functional pathways gene sets
universe	number of genes that were probed in the initial experiment

Details

geneSets is a "GeneSetColletion" object containing gene sets from various databases. Different sources for gene sets data are allowed and have to be provided in Gene Matrix Transposed file format (*.gmt), where each gene set is described by a pathway name, a description, and the genes in the gene set. Two examples are shown to demonstrate how to define geneSets object. See examples.

The variable universe represents a total number of genes that were probed in the initial experiment, e.g. the number of all genes on a microarray. If universe is not definied, universe is equal to the number of all genes that can be mapped to any pathways in chosen database.

Value

A data.frame with following columns:

pathway	names of enriched pathways
description	gene set description (e.g. a link to the named gene set in MSigDB)
genes_in_pathway	total number of known genes in the pathway
%_match	number of matched genes refered to the total number of known genes in the pathway given in %
pValue	p-value
adj.pValue	Benjamini-Hochberg adjucted p-value
overlap	genes from input genes list that overlap with all known genes in the pathway

Additionally an .txt file containing all above information is created.

Author(s)

Agata Michna

References

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S. and Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *PNAS* 102(43), 15545-15550.

http://www.broadinstitute.org/gsea/msigdb/collections.jsp

http://www.reactome.org/pages/download-data/

Examples

```
## Not run:
  ## Example 1 - using gene sets from the Molecular Signatures Database (MSigDB collections)
   ## Download .gmt file 'c2.all.v5.0.symbols.gmt' (all curated gene sets, gene symbols)
   ## from the Broad, http://www.broad.mit.edu/gsea/downloads.jsp#msigdb, then
   geneSets <- getGmt("/path/to/c2.all.v5.0.symbols.gmt")
   ## load "eSetObject" containing simulated time-course data
```

```
   data(TCsimData)
   ## check for differentially expressed genes
  diffExprs <- splineDiffExprs(eSetObject = TCsimData, df = 3, cutoff.adj.pVal = 0.01, reference = "T1")
   ## use differentially expressed genes for pathway enrichment analysis
  enrichPath <- pathEnrich(geneList = rownames(diffExprs), geneSets = geneSets, universe = 6536)
## End(Not run)

## Not run:
   ## Example 2 - using gene sets from the Reactome Pathway Database
   ## Download and unzip .gmt.zip file 'ReactomePathways.gmt.zip'
  ## ("Reactome Pathways Gene Set" under "Specialized data formats") from the Reactome website
   ## http://www.reactome.org/pages/download-data/, then
   geneSets <- getGmt("/path/to/ReactomePathways.gmt")
   data(TCsimData)
  diffExprs <- splineDiffExprs(eSetObject = TCsimData, df = 3, cutoff.adj.pVal = 0.01, reference = "T1")
  enrichPath <- pathEnrich(geneList = rownames(diffExprs), geneSets = geneSets, universe = 6536)
## End(Not run)

## Small example with gene sets consist of KEGG pathways only
geneSets <- getGmt(system.file("extdata", "c2.cp.kegg.v5.0.symbols.gmt", package="splineTimeR"))
data(TCsimData)
diffExprs <- splineDiffExprs(eSetObject = TCsimData, df = 3, cutoff.adj.pVal = 0.01, reference = "T1")
enrichPath <- pathEnrich(geneList = rownames(diffExprs), geneSets = geneSets, universe = 6536)
```

`splineDiffExprs` *Differential expression analysis based on natural cubic spline regression models for time-course data*

Description

The function compares time dependent behaviour of genes in two different groups. Applying empirical Bayes moderate F-statistic on differences in coefficients of fitted natural cubic spline regression models, differentially expressed in time genes are determined. The function is a wrapper of other R-functions to simplify differential expression analysis of time-course data.

Usage

```
splineDiffExprs(eSetObject, df, cutoff.adj.pVal=1, reference, intercept=TRUE)
```

Arguments

`eSetObject`	`ExpressionSet` object of class `ExpressionSet` containing log-ratios or log-values of expression for a series of microarrays
`df`	number of degrees of freedom
`cutoff.adj.pVal`	Benjamini-Hochberg adjusted p-value cut-off
`reference`	character defining which treatment group should be considered as reference
`intercept`	if `TRUE`, F-test includes all parameters; if `FALSE`, F-test includes shape parameters only; default is `TRUE`

Details

The function fits a temporal trend using a natural cubic spline regression to simulate nonlinear behaviour of genes over time.

The input `eSetObject` must be provided as an object of class `ExpressionSet` which contains `SampleName`, `Time`, `Treatment` and if applicable `Replicates` variables (columns) included in the phenotypic data of the `eSetObject` (`pData(eSetObject)`). Two types of `Treatment` defining two groups to compare have to be definied.

Replicates are not required. The time points for compared treatment groups should be identical.

User has to define number of degrees of freedom (`df`) for the spline regression model. Choosing effective degrees of freedom in range 3-5 is reasonable.

Time dependent differential expression of a gene is determined by the application of empirical Bayes moderate F-statistics on the differences of coefficient values of the fitted natural cubic spline regression models for the same gene in the two compared treatment groups. In other words, comparing the coefficient values of the fitted splines in both groups allows the detection of differences in the shape of the curves, which represent the gene expressions changes over time. Ouptut table containing Benjamini-Hochberg adjusted p-value (`adj.P.Value`) is used to define differentially expressed genes. The default value for `cutoff.adj.pVal` is set to `1`, which means that all genes are included in output table.

Value

A data.frame with rows defining names/IDs of differentially expressed genes and additional columns described below.

The first columns contain all feature data of the `eSetObject` (`fData(eSetObject)`), if any feature data were defined. Otherwise, only one column `row_IDs`, containing the row names is created. The `b_0`, `b_1`,..., `b_m` coefficients correspond to the reference model parameters. The `d_0`, `d_1`,..., `d_m` coefficients represent the differences between the reference model parameters and the model parameters in the compared group. `AveExprs` refers to the average log2-expression for a probe (representing a gene) over all arrays. The `F` column contains moderate F-statistics, `P.Value` raw p-value and `adj.P.Value` Benjamini-Hochberg adjusted p-value.

Author(s)

Agata Michna

See Also

limma

Examples

```
## load "eSetObject" containing simulated time-course data
data(TCsimData)
pData(TCsimData)

## define function parameters
df <- 3
cutoff.adj.pVal <- 0.01
```

```
reference <- "T1"
intercept <- TRUE

diffExprs <- splineDiffExprs(eSetObject = TCsimData, df, cutoff.adj.pVal, reference, intercept)
head(diffExprs,3)
```

---

splineNetRecon *Network reconstruction based on partial correlation method with shrinkage approach*

Description

splineNetRecon reconstructs gene association networks from time-course data. Based on given object of class ExpressionSet, longitudinal data object is created. Subsequantly the function estimates edges using partial correlation method with shrinkage approach applying ggm.estimate.pcor and network.test.edges functions. As a result an object or list of object of class igraph is created.

Usage

```
splineNetRecon(eSetObject, treatmentType, probesForNR="all",
            cutoff.ggm=0.8, method="dynamic", saveEdges=FALSE)
```

Arguments

eSetObject	ExpressionSet object of class ExpressionSet containing log-ratios or log-values of expression for a series of microarrays
treatmentType	a character string containing a type of Treatment defining samples considered for network reconstruction
probesForNR	a vector of character string containing names/IDs used for network reconstruction
cutoff.ggm	number or vector of numbers between 0 and 1 defining cutoff for significant posterior probability; default value is 0.8
method	method used to estimate the partial correlation matrix; available options are "static" and "dynamic" (default) - both are shrinkage methods
saveEdges	if TRUE, .Rdata file with all edges is created; default is FALSE

Details

The input eSetObject must be provided as an object of class ExpressionSet which contains SampleName, Time, Treatment and if applicable Replicates variables (columns) included in the phenotypic data of the eSetObject (pData(eSetObject)). Two types of Treatment defining two groups to compare have to be definied.

Gene association network reconstruction is conducted for a selected type of Treatment. This allows to find regulatory association between genes under a certain condition (treatment). First, a

longitudinal data object of the gene expression data with possible repicates is created. This object is used to estimate partial correlation with selected shrinkage method ("dynamic" or "static") with the ggm.estimate.pcor function (for details see ggm.estimate.pcor function help). Finally, the network.test.edges function estimates the probabilities for all possible edges and lists them in descending order (for details see network.test.edges help).

cutoff.ggm can be a single number or a vector of numbers. If more than one value for cutoff.ggm is definied than function returns a list of objects of class igraph for each definied cutoff.ggm value. Otherwise a single object of class igraph with one selected probability is returned.

Value

An object or list of objects of class igraph.

If saveEdges is TRUE, .Rdata file with all possible edges is created.

Author(s)

Agata Michna

See Also

http://strimmerlab.org/software/genenet/

http://strimmerlab.org/software/longitudinal/

Examples

```
## load "eSetObject" containing simulated time-course data
data(TCsimData)

## define function parameters
treatmentType = "T2"
probesForNR = "all"
cutoff.ggm = 0.8
method = "dynamic"

## reconstruct gene association network from time-course data
igr <- splineNetRecon(eSetObject = TCsimData, treatmentType, probesForNR, cutoff.ggm, method)
plot(igr, vertex.label = NA, vertex.size = 3)
```

---

splinePlot *Plot spline regression curves of time-course data*

Description

Function visualises time dependent behaviour of genes in two compared groups. The natural cubic spline regression curves fitted to discrete, time dependent expression data are plotted. One plot shows two curves - representing the reference group and the compared group, respectively. See also splineDiffExprs function.

Usage

```
splinePlot(eSetObject, df, reference, toPlot="all")
```

Arguments

eSetObject	ExpressionSet object of class ExpressionSet containing log-ratios or log-values of expression for a series of microarrays
df	number of degrees of freedom
reference	character defining which treatment group should be considered as reference
toPlot	vector of genes to plot; defalut is toPlot = "all"

Details

The input eSetObject must be provided as an object of class ExpressionSet which contains SampleName, Time, Treatment and if applicable Replicates variables (columns) included in the phenotypic data of the eSetObject (pData(eSetObject)). Two types of Treatment defining two groups to compare have to be definied.

Replicates are not required. The time points for compared treatment groups should be identical.

User has to define number of degrees of freedom (df) for the spline regression model. Choosing effective degrees of freedom in range 3-5 is reasonable.

Genes to plot, given as a vector of characters, can be selected by the user. Provided names have to be a part of a row name vector of eSetObject (rownames(exprs(eSetObject))). If genes to plot are not definied, all genes are plotted.

Value

A .pdf file containing plots for chosen genes.

Author(s)

Agata Michna

See Also

limma

Examples

```
## load "eSetObject" object containing simulated time-course data
data(TCsimData)
pData(TCsimData)

## define function parameters
df <- 3
reference <- "T1"
toPlot <- rownames(TCsimData)[1:10]

splinePlot(eSetObject = TCsimData, df, reference, toPlot)
```

`TCsimData` *Simulated time-course gene expression data set*

Description

Simulated data set of gene expression temporal profiles of 2000 genes. Data contain expression values in 8 time points after applying 2 differrent types of perturbation/treatment (`"T1"` and `"T2"`).

Usage

```
data(TCsimData)
```

Format

An object of class `ExpressionSet` with 2000 observations per time point and perturbation/treatment. Phenotypic data of the object (`pData(TCsimData)`) contain 4 columns with the following variables:

`SampleName` names of the samples

`Time` time points when samples were collected

`Treatment` types of treatment

`Replicate` names of replicates

Details

Simulation experiments were conducted using GeneNetWeaver (GNW), an open-source simulator for the DREAM challenges. The *in silico* expression data were simulated based on the network structure of a 2000-gene sub-network from the Reactome functional interaction network. The sub-network was converted to a dynamical network model without autoregulatory interactions (self-loops). The data were simulated based on the *"ODEs"* model. Two types of time-series experiments were chosen: *"Time Series as in DREAM4"* and *"Multifactorial"*. Gene expression data were simulated for 48 time points after perturbations. For more details see GNW User Manual.

The names of *"Time Series as in DREAM4"* and *"Multifactorial"* simulation experiments were changed to `"T1"` and `"T2"`, respectively. From the generated data sets, eight time points are provided (1, 4, 8, 16, 24, 32, 40 and 48). The numbers correspond to the same time units after perturbation (e.g. minutes, hours, days, ect.). Replicates for both time-course experiments were generated by the addition of the normally distributed random errors with a standard deviation of 0.05 to the expression values for each time point. Subsequently, the entire dataset was normalized between 0 and 1.

Value

An object of class `ExpressionSet`.

References

Marbach, D., Schaffter, T., Mattiussi, C., and Floreano, D. (2009). Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology* 2(16), 229-239.

Schaffter, T., Marbach, D., and Roulet G. (2010). GNW User Manual. http://gnw.sourceforge.net

Reactome project. Reactome Functional Interaction Network. Retrieved September 25, 2015 from http://www.reactome.org/

Examples

```
data(TCsimData)
pData(TCsimData)
head(exprs(TCsimData),3)
```

Index

∗ **datasets**
    TCsimData, 10
∗ **degree distribution**
    networkProperties, 2
∗ **differential expression**
    splineDiffExprs, 5
    splineNetRecon, 7
∗ **gene set enrichment analysis**
    pathEnrich, 3
∗ **pathway enrichment analysis**
    pathEnrich, 3
∗ **scale-free properties**
    networkProperties, 2
∗ **spline**
    splineDiffExprs, 5
    splinePlot, 8
∗ **time-course data**
    splineDiffExprs, 5
    splineNetRecon, 7
    splinePlot, 8

FIs, *3*

limma, *6*, *9*

networkProperties, 2

pathEnrich, 3

splineDiffExprs, 5
splineNetRecon, 7
splinePlot, 8

TCsimData, 10