Package 'tradeSeq'

April 15, 2020

Type Package

Title trajectory-based differential expression analysis for sequencing data

Date 2019-03-15

Version 1.0.1

Description tradeSeq provides a flexible method for finding genes that are differentially expressed along one or multiple trajectories, using a variety of tests suited to answer questions of interest, e.g. the discovery of genes that whose expression is associated with pseudotime, or who are differentially expressed (in a specific region) along the trajectory. It fits a generalized additive model (GAM) for each gene, and performs inference on the parameters of the GAM.

License MIT + file LICENSE

Encoding UTF-8

LazyData false

URL https://statomics.github.io/tradeSeq/index.html

Depends R (>= 3.6)

Collate 'AllGenerics.R' 'utils.R' 'associationTest.R' 'clusterExpressionPatterns.R' 'data.R' 'diffEndTest.R' 'earlyDETest.R' 'evaluateK.R' 'fitGAM.R' 'getSmootherPvalues.R' 'getSmootherTestStats.R' 'patternTest.R' 'plotGeneCount.R' 'plotSmoothers.R' 'startVsEndTest.R'

RoxygenNote 6.1.1

Imports mgcv, edgeR, SingleCellExperiment, SummarizedExperiment, slingshot, magrittr, RColorBrewer, clusterExperiment, BiocParallel, pbapply, ggplot2, princurve, methods, S4Vectors

Suggests knitr, rmarkdown, cowplot, dplyr, tidyr, testthat

VignetteBuilder knitr

biocViews Clustering, Regression, TimeCourse, DifferentialExpression, GeneExpression, RNASeq, Sequencing, Software, SingleCell, Transcriptomics, MultipleComparison, Visualization

BugReports https://github.com/statOmics/tradeSeq/issues

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

git_branch RELEASE_3_10

git_last_commit eb046aa

git_last_commit_date 2020-04-03

Date/Publication 2020-04-14

Author Koen Van den Berge [aut],
Hector Roux de Bezieux [aut, cre]
(<https://orcid.org/0000-0002-1489-8339>),
Kelly Street [ctb],
Lieven Clement [ctb],
Sandrine Dudoit [ctb]

Maintainer Hector Roux de Bezieux <hector.rouxdebezieux@berkeley.edu>

R topics documented:

`associationTest` *Perform statistical test to check whether gene expression is constant across pseudotime within a lineage*

Description

Assess whether gene expression is associated with pseudotime.

Usage

```
associationTest(models, ...)

## S4 method for signature 'SingleCellExperiment'
associationTest(models, global = TRUE,
  lineages = FALSE)

## S4 method for signature 'list'
associationTest(models, global = TRUE,
  lineages = FALSE)
```

Arguments

models	the list of GAMs, typically the output from fitGAM.
...	parameters including:
global	If TRUE, test for all lineages simultaneously.
lineages	If TRUE, test for all lineages independently.

Value

A matrix with the wald statistic, the number of df and the p-value associated with each gene for all the tests performed. If the testing procedure was unsuccessful, the procedure will return NA test statistics and p-values.

Examples

```
data(gamList, package = "tradeSeq")
associationTest(gamList, global = TRUE, lineages = TRUE)
```

---

celltype *A vector defining cell types, used in the package vignette.*

Description

This object contains a vector that define the cell type for each cell in the data described in Paul et al. (2015).

Usage

```
celltype
```

Format

An object of class character of length 2660.

Details

#' @references Franziska Paul, Yaara Arkin, Amir Giladi, DiegoAdhemar Jaitin, Ephraim Kenigsberg, Hadas KerenShaul, Deborah Winter, David Lara-Astiaso, Meital Gury, Assaf Weiner, Eyal David, Nadav Cohen, FeliciaKathrineBratt Lauridsen, Simon Haas, Andreas Schlitzer, Alexander Mildner, Florent Ginhoux, Steen Jung, Andreas Trumpp, BoTorben Porse, Amos Tanay, and Ido Amit. Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors. Cell, 163(7):1663–1677, 12 2015. ISSN 0092- 8674. doi: 10.1016/J.CELL.2015.11.013. URL https://www.sciencedirect.com
ii/S0092867415014932?via

clusterExpressionPatterns *Cluster gene expression patterns.*

Description

Cluster genes in clusters with similar expression patterns along the trajectory.

Usage

```
## S4 method for signature 'SingleCellExperiment'
clusterExpressionPatterns(models, nPoints,
  genes, reduceMethod = "PCA", nReducedDims = 10, minSizes = 6,
  ncores = 1, random.seed = 176201, verbose = TRUE, ...)

## S4 method for signature 'list'
clusterExpressionPatterns(models, nPoints, genes,
  reduceMethod = "PCA", nReducedDims = 10, minSizes = 6,
  ncores = 1, random.seed = 176201, verbose = TRUE, ...)
```

Arguments

models	The list of GAMs, typically the output from fitGAM.
nPoints	The number of points to use for clustering the expression patterns.
genes	A numerical or character vector specifying the genes from models that should be clustered.
reduceMethod	Method used before running the clustering methods. Passed to RSEC
nReducedDims	Number of dimensions kept after reduceMethod. Passed to RSEC
minSizes	Number of dimensions kept after reduceMethod. Passed to RSEC
ncores	Number of cores to use. Passed to RSEC
random.seed	Passed to RSEC
verbose	Passed to RSEC
...	Additional arguments to be passed to RSEC.

Details

This method adopts the RSEC function from the clusterExperiment package to perform consensus clustering.

Value

A list containing the scaled fitted values yhatScaled(for plotting) and a ClusterExperiment object.

Examples

```
data(gamList, package = "tradeSeq")
clusterExpressionPatterns(gamList, 200, seq_len(11))
```

countMatrix *A count matrix, used in the package vignette.*

Description

This object contains the gene expression counts from the data described in Paul et al. (2015).

Usage

```
countMatrix
```

Format

An object of class `dgCMatrix` with 240 rows and 2660 columns.

Details

#' @references Franziska Paul, Yaara Arkin, Amir Giladi, DiegoAdhemar Jaitin, Ephraim Kenigsberg, Hadas KerenShaul, Deborah Winter, David Lara-Astiaso, Meital Gury, Assaf Weiner, Eyal David, Nadav Cohen, FeliciaKathrineBratt Lauridsen, Simon Haas, Andreas Schlitzer, Alexander Mildner, Florent Ginhoux, Steen Jung, Andreas Trumpp, BoTorben Porse, Amos Tanay, and Ido Amit. Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors. Cell, 163(7):1663–1677, 12 2015. ISSN 0092- 8674. doi: 10.1016/J.CELL.2015.11.013. URL https://www.sciencedirect.com
ii/S0092867415014932?via

crv *A SlingshotDataset object, used in the package vignette.*

Description

This dataset contains the Slingshot trajectory from the data described in Paul et al. (2015).

Usage

```
crv
```

Format

An object of class `SlingshotDataSet` of length 1.

References

Franziska Paul, Yaara Arkin, Amir Giladi, DiegoAdhemar Jaitin, Ephraim Kenigsberg, Hadas KerenShaul, Deborah Winter, David Lara-Astiaso, Meital Gury, Assaf Weiner, Eyal David, Nadav Cohen, FeliciaKathrineBratt Lauridsen, Simon Haas, Andreas Schlitzer, Alexander Mildner, Florent Ginhoux, Steen Jung, Andreas Trumpp, BoTorben Porse, Amos Tanay, and Ido Amit. Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors. Cell, 163(7):1663–1677, 12 2015. ISSN 0092- 8674. doi: 10.1016/J.CELL.2015.11.013. URL https://www.sciencedirect.com/science/article/
ii/S0092867415014932?via

`diffEndTest` *Perform statistical test to check for DE between final stages of every lineage.*

Description

Assess differential expression between the end points of lineages of a trajectory.

Usage

```
diffEndTest(models, ...)

## S4 method for signature 'SingleCellExperiment'
diffEndTest(models, global = TRUE,
  pairwise = FALSE, l2fc = 0)

## S4 method for signature 'list'
diffEndTest(models, global = TRUE, pairwise = FALSE,
  l2fc = 0)
```

Arguments

`models`	Typically the output from `fitGAM`, either a list of fitted GAM models, or an object of `SingleCellExperiment` class.
`...`	parameters including:
`global`	If TRUE, test for all pairwise comparisons simultaneously.
`pairwise`	If TRUE, test for all pairwise comparisons independently.
`l2fc`	Numeric: log2 fold change threshold to test against. Note, that this only applies to the pairwise comparisons, the global test will be unaffected.

Details

The `l2fc` argument allows to test against a particular fold change threshold. For example, if the interest lies in discovering genes that are differentially expressed with an absolute log2 fold change cut off above 1, i.e. a fold change of at least 2, then one can test for this by setting `l2fc=1` as argument to the function.

Value

A matrix with the wald statistic, the number of df and the p-value associated with each gene for all the tests performed. If the testing procedure was unsuccessful, the procedure will return NA test statistics and p-values.

Examples

```
data(gamList, package = "tradeSeq")
diffEndTest(gamList, global = TRUE, pairwise = TRUE)
```

`earlyDETest` *Perform test of early differences between lineages*

Description

Perform test of early differences between lineages

Perform test of early differences between lineages

Usage

```
earlyDETest(models, ...)

## S4 method for signature 'SingleCellExperiment'
earlyDETest(models, global = TRUE,
  pairwise = FALSE, knots = NULL, nPoints = 100)

## S4 method for signature 'list'
earlyDETest(models, global = TRUE, pairwise = FALSE,
  knots = NULL, nPoints = 100)
```

Arguments

`models`	the list of GAMs, typically the output from `fitGAM`.
`...`	parameters including:
`global`	If TRUE, test for all pairwise comparisons simultaneously.
`pairwise`	If TRUE, test for all pairwise comparisons independently.
`knots`	A vector of length 2 specifying the knots before and after the branching of interest.
`nPoints`	the number of points to be compared between lineages.

Details

To help the user in choosing which knots to use when defining the branching, the `plotGeneCount` function has a models optional parameter that can be used to visualize where the knots are.

Value

A matrix with the wald statistic, the number of df and the p-value associated with each gene for all the tests performed.

Examples

```
data(gamList, package = "tradeSeq")
earlyDETest(gamList, knots = c(1, 2), global = TRUE, pairwise = TRUE)
```

evaluateK *Evaluate the optimal number of knots required for fitGAM.*

Description

Evaluate the optimal number of knots required for fitGAM.

Evaluate an appropriate number of knots.

Usage

```
evaluateK(counts, ...)

## S4 method for signature 'matrix'
evaluateK(counts, k = 3:10, nGenes = 500,
  sds = NULL, pseudotime = NULL, cellWeights = NULL, U = NULL,
  weights = NULL, offset = NULL, aicDiff = 2, verbose = TRUE,
  control = mgcv::gam.control(), sce = FALSE, family = "nb", ...)
```

Arguments

counts	the count matrix.
...	parameters including:
k	The range of knots to evaluate. '3:10' by default.
nGenes	The number of genes to use in the evaluation. Genes will be randomly selected. 500 by default.
sds	Slingshot object containing the lineages.
pseudotime	a matrix of pseudotime values, each row represents a cell and each column represents a lineage.
cellWeights	a matrix of cell weights defining the probability that a cell belongs to a particular lineage. Each row represents a cell and each column represents a lineage.
U	the design matrix of fixed effects. The design matrix should not contain an intercept to ensure identifiability.
weights	Optional: a matrix of weights with identical dimensions as the counts matrix. Usually a matrix of zero-inflation weights.
offset	Optional: the offset, on log-scale. If NULL, TMM is used to account for differences in sequencing depth, see fitGAM.
aicDiff	Used for selecting genes with significantly varying AIC values over the range of evaluated knots to make the barplot output. Default is set to 2, meaning that only genes whose AIC range is larger than 2 will be used to check for the optimal number of knots through the barplot visualization that is part of the output of this function.
verbose	logical, should progress be verbose?
control	Control object for GAM fitting, see mgcv::gam.control().
sce	Logical, automatically set by the function and should not be altered by the user.
family	The distribution assumed, currently only "nb" (negative binomial) is supported.

Value

A plot of average AIC value over the range of selected knots, and a matrix of AIC values for the selected genes (rows) and the range of knots (columns).

Examples

```
## This is an artifical example, please check the vignette for a realistic one.
set.seed(8)
data(sds, package="tradeSeq")
loadings <- matrix(runif(2000*2,-2,2), nrow=2, ncol=2000)
counts <- round(abs(t(slingshot::reducedDim(sds) %*% loadings)))+100
aicK <- evaluateK(counts = counts, sds=sds,
                  nGenes=100, k=3:5, verbose=FALSE)
```

`fitGAM` *fitGAM*

Description

This fits the NB-GAM model as described in Van den Berge et al.[2019]

Usage

```
fitGAM(counts, ...)

## S4 method for signature 'matrix'
fitGAM(counts, sds = NULL, pseudotime = NULL,
  cellWeights = NULL, U = NULL, weights = NULL, offset = NULL,
  nknots = 6, verbose = TRUE, parallel = FALSE,
  BPPARAM = BiocParallel::bpparam(), control = mgcv::gam.control(),
  sce = FALSE, family = "nb")
```

Arguments

`counts`	the count matrix.
`...`	parameters including:
`sds`	an object of class `SlingshotDataSet`, typically obtained after running Slingshot. If this is provided, `pseudotime` and `cellWeights` arguments are derived from this object.
`pseudotime`	a matrix of pseudotime values, each row represents a cell and each column represents a lineage.
`cellWeights`	a matrix of cell weights defining the probability that a cell belongs to a particular lineage. Each row represents a cell and each column represents a lineage.
`U`	the design matrix of fixed effects. The design matrix should not contain an intercept to ensure identifiability.
`weights`	a matrix of weights with identical dimensions as the `counts` matrix. Usually a matrix of zero-inflation weights.
`offset`	the offset, on log-scale. If NULL, TMM is used to account for differences in sequencing depth., see `edgeR::calcNormFactors`. Alternatively, this may also be a matrix of the same dimensions as the expression matrix.

nknots	Number of knots used to fit the GAM. Defaults to 6.
verbose	Logical, should progress be printed?
parallel	Logical, defaults to FALSE. Set to TRUE if you want to parallellize the fitting.
BPPARAM	object of class bpparamClass that specifies the back-end to be used for computations. See bpparam in BiocParallel package for details.
control	Variables to control fitting of the GAM, see gam.control.
sce	Should output be of SingleCellExperiment class? This argument should not be changed by users.
family	The assumed distribution for the response, set to "nb" by default.

Value

A list of length the number of genes (number of rows of counts). Each element of the list is either a gamObject if the fiting procedure converged, or an error message.

Examples

```
set.seed(8)
data(crv, package="tradeSeq")
data(countMatrix, package="tradeSeq")
gamList <- fitGAM(counts = as.matrix(countMatrix),
                  sds = crv,
                  nknots = 5)
```

---

gamList *A list of GAM models, used to demonstrate the various tests.*

Description

A list of 11 gamObject obtained by fitting 10 genes on 15 cells randomly assigned to lineages with random pseudotimes.

Usage

```
gamList
```

Format

Can be re-obtained by runing the code in the example section of fitGAM.

getSmootherPvalues *Get smoother p-value as returned by* mgcv.

Description

Return smoother p-values from the mgcv package.

Usage

```
getSmootherPvalues(models)
```

Arguments

models	the GAM models, typically the output from fitGAM. Note that this function only works when models is a list.

Value

a matrix with the p-value associated with each lineage's smoother. The matrix has one row per gene where the fitting procedure converged.

Examples

```
data(gamList, package = "tradeSeq")
getSmootherPvalues(gamList)
```

getSmootherTestStats *Get smoother Chi-squared test statistics.*

Description

Return test statistics from the mgcv package.

Usage

```
getSmootherTestStats(models)
```

Arguments

models	the GAM models, typically the output from fitGAM. Note that this function only works when models is a list.

Value

a matrix with the wald statistics associated with each lineage's smoother. The matrix has one row per gene where the fitting procedure converged.

Examples

```
data(gamList, package = "tradeSeq")
getSmootherPvalues(gamList)
```

patternTest *Assess differential expression pattern between lineages.*

Description

Assess differences in expression patterns between lineages.

Usage

```
patternTest(models, ...)

## S4 method for signature 'list'
patternTest(models, global = TRUE, pairwise = FALSE,
  nPoints = 100)

## S4 method for signature 'SingleCellExperiment'
patternTest(models, global = TRUE,
  pairwise = FALSE, nPoints = 100)
```

Arguments

models	the list of GAMs, typically the output from fitGAM.
...	parameters including:
global	If TRUE, test for all pairwise comparisons simultaneously.
pairwise	If TRUE, test for all pairwise comparisons independently.
nPoints	the number of points to be compared between lineages.

Value

A matrix with the wald statistic, the number of df and the p-value associated with each gene for all the tests performed. If the testing procedure was unsuccessful, the procedure will return NA test statistics and p-values.

Examples

```
data(gamList, package = "tradeSeq")
patternTest(gamList, global = TRUE, pairwise = TRUE)
```

plotGeneCount *Plot the gene in reduced dimension space*

Description

Plot the gene in reduced dimension space

Usage

```
plotGeneCount(curve, counts = NULL, gene = NULL, clusters = NULL,
  models = NULL, title = NULL)
```

Arguments

curve	The output from a lineage computation
counts	the count matrix.
gene	The name of gene for which you want to plot the count or the row number of that gene in the count matrix. Alternatively, one can specify the cluster arguments
clusters	The assignation of each cell to a cluster. Used to color the plot. Either clusters or gene and counts must be supplied.
models	the list of GAMs, typically the output from fitGAM. Used to display the knots.
title	Title for the plot.

Details

If both gene and clusters arguments are supplied, the plot will be colored according to gene count level.

Value

A ggplot object

Examples

```
set.seed(97)
library(slingshot)
data(crv, package="tradeSeq")
data(countMatrix, package="tradeSeq")
rd <- slingshot::reducedDim(crv)
cl <- kmeans(rd, centers = 7)$cluster
lin <- slingshot::getLineages(rd, clusterLabels = cl, start.clus = 4)
crv <- slingshot::getCurves(lin)
counts <- as.matrix(countMatrix)
gamList <- fitGAM(counts = counts,
 pseudotime = slingPseudotime(crv, na = FALSE),
 cellWeights = slingCurveWeights(crv))
plotGeneCount(crv, counts, gene = "Mpo")
```

---

plotSmoothers *Plot the log-transformed counts and the fitted values for a particular gene along all lineages*

Description

Plot the smoothers estimated by tradeSeq.

Usage

```
plotSmoothers(models, ...)

## S4 method for signature 'gam'
plotSmoothers(models, nPoints = 100, lwd = 2,
  size = 2/3, xlab = "Pseudotime", ylab = "Log(expression + 1)",
  border = TRUE, alpha = 1)
```

```
## S4 method for signature 'SingleCellExperiment'
plotSmoothers(models, counts, gene,
  nPoints = 100, lwd = 2, size = 2/3, xlab = "Pseudotime",
  ylab = "Log(expression + 1)", border = TRUE, alpha = 1)
```

Arguments

models	Either the SingleCellExperiment object obtained after running fitGAM, or the specific GAM model for the corresponding gene, if working with the list output of tradeSeq.
...	parameters including:
nPoints	The number of points used to extraplolate the fit
lwd	Line width of the smoother. Passed to geom_line
size	Character expansion of the data points. Passed to geom_point
xlab	x-axis label. Passed to labs
ylab	y-axis label. Passed to labs
border	logical: should a white border be drawn around the mean smoother.
alpha	Numeric between 0 and 1, determines the transparancy of data points, see scale_color_viridis_d.
counts	The matrix of gene expression counts.
gene	Gene name of gene to plot.

Value

A ggplot object

Examples

```
data(gamList, package = "tradeSeq")
plotSmoothers(gamList[[4]])
```

---

sds *A SlingshotDataset object, used in the package unit tests.*

Description

This dataset contains the toy example from the Slingshot R package vignette.

Usage

```
sds
```

Format

An object of class SlingshotDataSet of length 1.

Source

https://bioconductor.org/packages/release/bioc/html/slingshot.html

`startVsEndTest` *Perform statistical test to check for DE between starting point and the end stages of every lineage*

Description

Assess differential expression between the start and end points of a lineage.

Usage

```
startVsEndTest(models, ...)

## S4 method for signature 'SingleCellExperiment'
startVsEndTest(models, global = TRUE,
  lineages = FALSE, pseudotimeValues = NULL)

## S4 method for signature 'list'
startVsEndTest(models, global = TRUE,
  lineages = FALSE, pseudotimeValues = NULL)
```

Arguments

`models`	the list of GAMs, typically the output from `fitGAM`.
`...`	parameters including:
`global`	If TRUE, test for all lineages simultaneously.
`lineages`	If TRUE, test for all lineages independently.
`pseudotimeValues`	a vector of length 2, specifying two pseudotime values to be compared against each other, for every lineage of the trajectory. @details Note that this test assumes that all lineages start at a pseudotime value of zero, which is the starting point against which the end point is compared.

Value

A matrix with the wald statistic, the number of df and the p-value associated with each gene for all the tests performed. If the testing procedure was unsuccessful, the procedure will return NA test statistics and p-values. If both `global` and `lineages` are TRUE, then a matrix of p-values is returned.

Examples

```
data(gamList, package = "tradeSeq")
startVsEndTest(gamList, global = TRUE, lineages = TRUE)
```

Index