Package 'tradeSeq'

April 16, 2025

Type Package

Title trajectory-based differential expression analysis for sequencing data

Date 2019-03-17

Version 1.22.0

Description tradeSeq provides a flexible method for fitting regression models that can be used to find genes that are differentially expressed along one or multiple lineages in a trajectory. Based on the fitted models, it uses a variety of tests suited to answer different questions of interest, e.g. the discovery of genes for which expression is associated with pseudotime, or which are differentially expressed (in a specific region) along the trajectory. It fits a negative binomial 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' 'conditionTest.R' 'data.R' 'diffEndTest.R' 'earlyDETest.R' 'evaluateK.R' 'fitGAM.R' 'getSmootherPvalues.R' 'getSmootherTestStats.R' 'nknots.R' 'patternTest.R' 'plotGeneCount.R' 'plotSmoothers.R' 'predictCells.R' 'predictSmooth.R' 'startVsEndTest.R'

RoxygenNote 7.2.1

Imports mgcv, edgeR, SingleCellExperiment, SummarizedExperiment, slingshot, magrittr, RColorBrewer, BiocParallel, Biobase, pbapply, igraph, ggplot2, princurve, methods, S4Vectors, tibble, Matrix, TrajectoryUtils, viridis, matrixStats, MASS

Suggests knitr, rmarkdown, testthat, covr, clusterExperiment, DelayedMatrixStats

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_21

git_last_commit e8f3144

git_last_commit_date 2025-04-15

Repository Bioconductor 3.21

Date/Publication 2025-04-15

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

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

Contents

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

Description

This test assesses whether average gene expression is associated with pseudotime.

Usage

```
associationTest(models, ...)

## S4 method for signature 'SingleCellExperiment'
associationTest(
  models,
  global = TRUE,
  lineages = FALSE,
  l2fc = 0,
  nPoints = 2 * tradeSeq::nknots(models),
  contrastType = "start",
  inverse = ifelse(l2fc == 0, "Chol", "eigen")
)

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

Arguments

`models`	The fitted GAMs, typically the output from `fitGAM`.
`...`	parameters including:
`global`	If TRUE, test for all lineages simultaneously.
`lineages`	If TRUE, test for all lineages independently.
`l2fc`	The log2 fold change threshold to test against. Note, that this will affect both the global test and the pairwise comparisons.
`nPoints`	The number of points used per lineage to set up the contrast. Defaults to 2 times the number of knots. Note that not all points may end up being actually used in the inference; only linearly independent contrasts will be used.
`contrastType`	The contrast used to impose the log2 fold-change threshold. Defaults to `"start"`. Three options are possible: - If `"start"`, the starting point of each lineage is used to compare against all other points, and the fold-change threshold is applied on these contrasts. - If `"end"`, the procedure is similar to `"start"`, except that the reference point will now be the endpoint of each lineage rather than the starting point. - If `"consecutive"`, then consecutive points along each lineage will be used as contrasts. This is the original way the `associationTest` was implemented and is kept for backwards compatibility. If a fold change threshold has been set, we recommend users to use either the `"start"` or `"end"` options.

`inverse` Method to use to invert variance-covariance matrix of contrasts. Usually you do not want to change this. Options are - `"Chol"` for Cholesky decomposition using `chol2inv`. This is the default when `l2fc=0`. - `"eigen"` for eigendecomposition using `eigen`. - `"QR"` for QR decomposition using `qr.solve`. - `"generalized"` for Moore-Pennrose generalized inverse using `MASS::ginv`.

Details

If a log2 fold-change threshold has not been set, we use the QR decompositon through `qr.solve` to invert the variance-covariance matrix of the contrasts. If instead a log2 fold chalnge-threshold has been set, we invert that matrix using the Cholesky decomposition through `chol2inv`.

Value

A matrix with the wald statistic, the degrees of freedom and the (unadjusted) p-value associated with each gene for all the tests performed. If the testing procedure was unsuccessful for a particular gene, `NA` values will be returned for that gene.

Examples

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

`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

```
data(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/scie
ii/S0092867415014932?via

`clusterExpressionPatterns`

Cluster gene expression patterns.

Description

Cluster genes in clusters that have similar expression patterns along all lineages in the trajectory. By default, this function uses the `clusterExperiment` package to do the clustering. If another clustering method is of interest, one can extract fitted values to use for clustering, see details in the vignette.

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 fitted 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	Dimensionality reduction method used before running the clustering methods. Passed to RSEC. Defaults to PCA.
nReducedDims	Number of dimensions kept after reduceMethod. Passed to RSEC.
minSizes	Minimum size of clusters. 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, containing the clustering results.

Examples

```
## Not run:
data(gamList, package = "tradeSeq")
clusterExpressionPatterns(models = gamList, nPoints = 200, genes = seq_len(11),
                          verbose = FALSE)

## End(Not run)
```

`conditionTest` *Assess differential expression patterns between conditions within a lineage.*

Description

Assess differential expression patterns between conditions within a lineage.

Assess differential expression patterns between conditions within a lineage.

Usage

```
conditionTest(models, ...)

## S4 method for signature 'SingleCellExperiment'
conditionTest(
  models,
  global = TRUE,
  pairwise = FALSE,
  lineages = FALSE,
  knots = NULL,
  l2fc = 0,
  eigenThresh = 0.01
)
```

Arguments

`models`	The fitted GAMs, typically the output from `fitGAM`. For `conditionTest`, these are required to be a `singleCellExperiment` object.
`...`	parameters including:
`global`	If TRUE, test for all pairwise comparisons simultaneously, i.e. test for DE between all conditions in all lineages.
`pairwise`	If TRUE, return output for all comparisons between pairs of conditions. Both `global` and `pairwise` can be TRUE.
`lineages`	If TRUE, return output for all comparisons within each lineage. Both `global` and `lineages` can be TRUE. If both `lineages` and `pairwise` are TRUE, the function returns output for all pairs of conditions within each lineage.
`knots`	Default to NULL. Otherwise, a vector of length 2 specifying the smallest and largest knots that are contrasted between conditions.
`l2fc`	The log2 fold change threshold to test against. Note, that this will affect both the global test and the pairwise comparisons.
`eigenThresh`	Eigenvalue threshold for inverting the variance-covariance matrix of the coefficients to use for calculating the Wald test statistics. Lower values are more lenient to adding more information but also decrease computational stability. This argument should in general not be changed by the user but is provided for back-compatability. Set to `1e-8` to reproduce results of older version of `tradeSeq`.

Value

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

Examples

```
## artificial example
data(crv, package = "tradeSeq")
data("countMatrix", package = "tradeSeq")
conditions <- factor(sample(1:2, size = ncol(countMatrix), replace = TRUE))
sce <- fitGAM(as.matrix(countMatrix), sds = crv, conditions = conditions)
res <- conditionTest(sce)
```

---

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

```
data(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/scie
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

```
data(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 average expression at 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	The fitted 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.
l2fc	The log2 fold change threshold to test against. Note, that this will affect both the global test and the pairwise comparisons.

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. Also, for each possible pairwise comparision, the observed log fold changes. If the testing procedure was unsuccessful, the procedure will return NA test statistics, fold changes 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 differential expression patterns between lineages in a user-defined region based on the knots of the smoothers.

Usage

```
earlyDETest(models, ...)

## S4 method for signature 'SingleCellExperiment'
earlyDETest(
  models,
  global = TRUE,
  pairwise = FALSE,
  knots = NULL,
  nPoints = 2 * nknots(models),
```

```
  l2fc = 0,
  eigenThresh = 0.01
)

## S4 method for signature 'list'
earlyDETest(
  models,
  global = TRUE,
  pairwise = FALSE,
  knots = NULL,
  nPoints = 2 * nknots(models),
  l2fc = 0,
  eigenThresh = 0.01
)
```

Arguments

models	The fitted 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 at the start and end of the region of interest.
nPoints	The number of points to be compared between lineages. Defaults to twice the number of knots
l2fc	The log2 fold change threshold to test against. Note, that this will affect both the global test and the pairwise comparisons.
eigenThresh	Eigenvalue threshold for inverting the variance-covariance matrix of the coefficients to use for calculating the Wald test statistics. Lower values are more lenient to adding more information but also decrease computational stability. This argument should in general not be changed by the user but is provided for back-compatability. Set to 1e-8 to reproduce results of older version of 'tradeSeq'.

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. Also, for each possible pairwise comparision, the observed log fold changes. If the testing procedure was unsuccessful, the procedure will return NA test statistics, fold changes and p-values.

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,
  conditions = NULL,
  plot = TRUE,
  weights = NULL,
  offset = NULL,
  aicDiff = 2,
  verbose = TRUE,
  parallel = FALSE,
  BPPARAM = BiocParallel::bpparam(),
  control = mgcv::gam.control(),
  family = "nb",
  gcv = FALSE,
  ...
)

## S4 method for signature 'dgCMatrix'
evaluateK(
  counts,
  k = 3:10,
  nGenes = 500,
  sds = NULL,
  pseudotime = NULL,
```

```
  cellWeights = NULL,
  plot = TRUE,
  U = NULL,
  weights = NULL,
  offset = NULL,
  aicDiff = 2,
  verbose = TRUE,
  conditions = NULL,
  control = mgcv::gam.control(),
  parallel = FALSE,
  BPPARAM = BiocParallel::bpparam(),
  family = "nb",
  gcv = FALSE,
  ...
)

## S4 method for signature 'SingleCellExperiment'
evaluateK(
  counts,
  k = 3:10,
  nGenes = 500,
  sds = NULL,
  pseudotime = NULL,
  cellWeights = NULL,
  plot = TRUE,
  U = NULL,
  weights = NULL,
  offset = NULL,
  aicDiff = 2,
  verbose = TRUE,
  conditions = NULL,
  parallel = FALSE,
  BPPARAM = BiocParallel::bpparam(),
  control = mgcv::gam.control(),
  family = "nb",
  gcv = FALSE,
  ...
)
```

Arguments

counts	The count matrix, genes in rows and cells in columns.
...	parameters including:
k	The range of knots to evaluate. '3:10' by default. See details.
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.
conditions	This argument is in beta phase and should be used carefully. If each lineage consists of multiple conditions, this argument can be used to specify the conditions. tradeSeq will then fit a condition-specific smoother for every lineage.
plot	Whether to display diagnostic plots. Default to TRUE.
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?
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	Control object for GAM fitting, see mgcv::gam.control().
family	The distribution assumed, currently only "nb" (negative binomial) is supported.
gcv	(In development). Logical, should a GCV score also be returned?

Details

The number of parameter to evaluate (and therefore the runtime) scales in k * the number of lineages. Moreover, we have found that, in practice, values of k above 12-15 rarely lead to improved result, not matter the complexity of the trajectory being considered. As such, we recommand that user proceed with care when setting k to value higher than 15.

Value

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

Examples

```
## This is an artificial example, please check the vignette for a realistic one.
set.seed(8)
library(slingshot)
data(sds, package="tradeSeq")
```

```
loadings <- matrix(runif(2000*2, -2, 2), nrow = 2, ncol = 2000)
counts <- round(abs(t(slingshot:::slingReducedDim(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]. There are two ways to provide the required input in `fitGAM`. See Details and the vignette.

Usage

```
fitGAM(counts, ...)

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

## S4 method for signature 'dgCMatrix'
fitGAM(
  counts,
  sds = NULL,
  pseudotime = NULL,
  cellWeights = NULL,
  conditions = NULL,
  U = NULL,
  genes = seq_len(nrow(counts)),
```

```
  weights = NULL,
  offset = NULL,
  nknots = 6,
  verbose = TRUE,
  parallel = FALSE,
  BPPARAM = BiocParallel::bpparam(),
  control = mgcv::gam.control(),
  sce = TRUE,
  family = "nb",
  gcv = FALSE
)

## S4 method for signature 'SingleCellExperiment'
fitGAM(
  counts,
  U = NULL,
  genes = seq_len(nrow(counts)),
  conditions = NULL,
  weights = NULL,
  offset = NULL,
  nknots = 6,
  verbose = TRUE,
  parallel = FALSE,
  BPPARAM = BiocParallel::bpparam(),
  control = mgcv::gam.control(),
  sce = TRUE,
  family = "nb",
  gcv = FALSE
)
```

Arguments

counts	The count matrix of expression values, with genes in rows and cells in columns. Can be a matrix or a sparse matrix.
...	parameters including:
sds	an object of class SlingshotDataSet or PseudotimeOrdering, 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. If only a single lineage, provide a matrix with one column containing all values of 1.
conditions	This argument is in beta phase and should be used carefully. If each lineage consists of multiple conditions, this argument can be used to specify the conditions. tradeSeq will then fit a condition-specific smoother for every lineage.

U	The design matrix of fixed effects. The design matrix should not contain an intercept to ensure identifiability.
genes	The genes on which to run fitGAM. Default to all the genes. If only a subset of the genes is indicated, normalization will be done using all the genes but the smoothers will be computed only for the subset.
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 vector with length equal to the number of cells.
nknots	Number of knots used to fit the GAM. Defaults to 6. It is recommended to use the 'evaluateK' function to guide in selecting an appropriate number of knots.
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	Logical: should output be of SingleCellExperiment class? This is recommended to be TRUE. If sds argument is specified, it will always be set to TRUE
family	The assumed distribution for the response. Is set to "nb" by default.
gcv	(In development). Logical, should a GCV score also be returned?

Details

fitGAM supports four different ways to input the required objects:

- "Count matrix, matrix of pseudotime and matrix of cellWeights." Input count matrix using counts argument and pseudotimes and cellWeights as a matrix, with number of rows equal to number of cells, and number of columns equal to number of lineages.
- "Count matrix and Slingshot input."Input count matrix using counts argument and Slingshot object using sds argument.
- "SingleCellExperiment Object after running slingshot on the object." Input SingleCellExperiment Object using counts argument.
- "CellDataSet object after running the orderCells function." Input CellDataSet Object using counts argument.

Value

If sce=FALSE, returns a list of length equal to 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. If sce=TRUE, returns a singleCellExperiment object with the tradeSeq results stored in the rowData, colData and metadata.

Examples

```
set.seed(8)
data(crv, package="tradeSeq")
data(countMatrix, package="tradeSeq")
sceGAM <- 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

```
data(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)
```

nknots *knots*

Description

Get the number of knots used for the fit

Usage

```
nknots(models, ...)

## S4 method for signature 'SingleCellExperiment'
nknots(models)

## S4 method for signature 'list'
nknots(models)
```

Arguments

models	The fitted GAMs, typically the output from fitGAM.
...	parameters including:

Value

A numeric, the number of nknots

Examples

```
data(gamList, package = "tradeSeq")
nknots(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 = 2 * nknots(models),
  l2fc = 0,
  eigenThresh = 0.01
)

## S4 method for signature 'SingleCellExperiment'
patternTest(
  models,
  global = TRUE,
  pairwise = FALSE,
  nPoints = 2 * nknots(models),
  l2fc = 0,
  eigenThresh = 0.01
)
```

Arguments

models	The fitted GAMs, typically the output from fitGAM.
...	parameters including:
global	If TRUE, test for all pairwise comparisons simultaneously. If models contains conditions (i.e. fitGAM was run with the conditions argument), then we compare the within-lineage average across conditions, between lineages.

pairwise	If TRUE, test for all pairwise comparisons, between lineages.
nPoints	The number of points to be compared between lineages. Defaults to twice the number of knots
l2fc	The log2 fold change threshold to test against. Note, that this will affect both the global test and the pairwise comparisons.
eigenThresh	Eigenvalue threshold for deciding on the rank of the variance-covariance matrix of the contrasts defined by 'patternTest', and to use for calculating the Wald test statistics. Lower values are more lenient to adding more information but also decrease computational stability. This argument should in general not be changed by the user but is provided for back-compatability. Set to `1e-8` to reproduce results of older version of 'tradeSeq'.

Value

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

Examples

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

---

plotGeneCount *Plot gene expression in reduced dimension.*

Description

Plot the gene in reduced dimensional space.

Usage

```
plotGeneCount(curve, ...)

## S4 method for signature 'SlingshotDataSet'
plotGeneCount(
  curve,
  counts = NULL,
  gene = NULL,
  clusters = NULL,
  models = NULL,
  title = NULL
)

## S4 method for signature 'PseudotimeOrdering'
plotGeneCount(
```

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

## S4 method for signature 'SingleCellExperiment'
plotGeneCount(
  curve,
  counts = NULL,
  gene = NULL,
  clusters = NULL,
  models = NULL,
  title = NULL
)
```

Arguments

curve	One of the following: • A PseudotimeOrdering or SlingshotDataSet object. The output from trajectory inference using Slingshot. • A SingleCellExperiment object. The output from trajectory inference using Slingshot. • A CellDataset object.
...	parameters including:
counts	The count matrix, genes in rows and cells in columns. Only needed if the input is of the type PseudotimeOrdering or SlingshotDataSet and the gene argument is not NULL.
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 clusters argument.
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 fitted GAMs, typically the output from fitGAM. Used to display the knots. Does not work with a CellDataset object as input.
title	Title for the plot.

Details

If both gene and clusters arguments are supplied, the plot will be colored according to gene count level. If none are provided, the function will fail.

Value

A ggplot object

Examples

```
set.seed(97)
library(slingshot)
data(crv, package="tradeSeq")
data(countMatrix, package="tradeSeq")
rd <- slingReducedDim(crv)
cl <- kmeans(rd, centers = 7)$cluster
lin <- getLineages(rd, clusterLabels = cl, start.clus = 4)
crv <- 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,
  sample = 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,
  sample = 1,
  pointCol = NULL,
  curvesCols = NULL,
  plotLineages = TRUE,
  lineagesToPlot = NULL
)
```

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 extrapolate the fit. Defaults to 100.
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 transparency of data points, see scale_color_viridis_d.
sample	Numeric between 0 and 1, use to subsample the cells when there are too many so that it can plot faster.
counts	The matrix of gene expression counts.
gene	Gene name or row in count matrix of gene to plot.
pointCol	Plotting colors for each cell. Can be either character vector of length 1, denoting a variable in the colData(models) to color cells by, or a vector of length equal to the number of cells.
curvesCols	Plotting colors for each curve Should be a list of colors of the exact same length as the number of curves, i.e. the number of lineages (if there is no conditions) or the number of lineages by the number of conditions. In the second case, the colors are grouped by condition (lineage 1 - condition 1, lineage 1 - condition 2,...).
plotLineages	Logical, should the mean smoothers for each lineage be plotted?
lineagesToPlot	A vector of integers referring to the lineages to be plotted.

Value

A ggplot object

Examples

```
set.seed(8)
data(crv, package="tradeSeq")
data(countMatrix, package="tradeSeq")
counts <- as.matrix(countMatrix)
sce <- fitGAM(counts = counts,
                  sds = crv,
                  nknots = 5)
plotSmoothers(sce, counts, rownames(counts)[1])
# Show only first lineage curve
curvesCols <- c("#440154FF", "transparent")
plotSmoothers(sce, counts, rownames(counts)[1], curvesCols = curvesCols,
  border = FALSE)
# Show only first curve and cells assigned to first lineage
plotSmoothers(sce, counts, rownames(counts)[1], curvesCols = curvesCols,
  border = FALSE) +
  ggplot2::scale_color_manual(values = curvesCols)
```

`plot_evalutateK_results` *Evaluate an appropriate number of knots.*

Description

Evaluate an appropriate number of knots.

Usage

```
plot_evalutateK_results(aicMat, k = NULL, aicDiff = 2)
```

Arguments

`aicMat`	The output from `evaluateK`
`k`	The range of knots to evaluate. '3:10' by default. Extracted from the column names by default
`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.

Examples

```
## This is an artificial example, please check the vignette for a realistic one.
set.seed(8)
data(sds, package="tradeSeq")
library(slingshot)
loadings <- matrix(runif(2000*2, -2, 2), nrow = 2, ncol = 2000)
```

```
counts <- round(abs(t(slingshot:::slingReducedDim(sds) %*% loadings))) + 100
aicK <- evaluateK(counts = counts, sds = sds, nGenes = 100,
                  k = 3:5, verbose = FALSE, plot = FALSE)
plot_evalutateK_results(aicK, k = 3:5)
```

---

predictCells *predictCells*

Description

Get fitted values for each cell.

Usage

```
predictCells(models, ...)

## S4 method for signature 'SingleCellExperiment'
predictCells(models, gene)

## S4 method for signature 'list'
predictCells(models, gene)
```

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:
gene	Gene name of gene for which to extract fitted values.

Details

Using the gene expression model of tradeSeq available at https://www.nature.com/articles/s41467-020-14766-3#Sec2. the output of predictCells returns the η_{gi} value for each cell.

Value

A vector of fitted values.

Examples

```
data(gamList, package = "tradeSeq")
predictCells(models = gamList, gene = 1)
```

predictSmooth *predictSmooth*

Description

Get smoothers estimated by tradeSeq along a grid. This function does not return fitted values but rather the predicted mean smoother, for a user-defined grid of points.

Usage

```
predictSmooth(models, ...)

## S4 method for signature 'SingleCellExperiment'
predictSmooth(models, gene, nPoints = 100, tidy = TRUE)

## S4 method for signature 'list'
predictSmooth(models, gene, nPoints = 100)
```

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:
gene	Either a vector of gene names or an integer vector, corresponding to the row(s) of the gene(s).
nPoints	The number of points used to create the grid along the smoother for each lineage. Defaults to 100.
tidy	Logical: return tidy output. If TRUE, returns a data.frame specifying lineage, gene, pseudotime and value of estimated smoother. If FALSE, returns matrix of predicted smoother values, where each row is a gene and each column is a point on the uniform grid along the lineage. For example, if the trajectory consists of 2 lineages and nPoints=100, then the returned matrix will have 2*100 columns, where the first 100 correspond to the first lineage and columns 101-200 to the second lineage.

Details

Using the gene expression model of tradeSeq available at https://www.nature.com/articles/s41467-020-14766-3#Sec2. the output of predictSmooth returns the η_{gi} value for equally space values of pseudotimes, and a constant value for U_i and N_i (arbitraly, we select the values of $i = 1$).

Value

A matrix with estimated averages.

A vector of fitted values.

Examples

```
data(gamList, package = "tradeSeq")
predictSmooth(models = gamList, gene = 1)
```

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

Description

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

Usage

```
data(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

This function assesses differential expression between the average expression of the start and end points of a lineage.

Usage

```
startVsEndTest(models, ...)

## S4 method for signature 'SingleCellExperiment'
startVsEndTest(
  models,
  global = TRUE,
  lineages = FALSE,
  pseudotimeValues = NULL,
  l2fc = 0
)
```

```
## S4 method for signature 'list'
startVsEndTest(
  models,
  global = TRUE,
  lineages = FALSE,
  pseudotimeValues = NULL,
  l2fc = 0
)
```

Arguments

models	The fitted 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.
l2fc	The log2 fold change threshold to test against. Note, that this will affect both the global test and the pairwise comparisons.

Value

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

Examples

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

Index