Package 'TimeSeriesExperiment'

October 12, 2021

Type Package

Title Analysis for short time-series data

Version 1.11.1

Date 2021-08-31

Description TimeSeriesExperiment is a visualization and analysis toolbox for short time course data. The package includes dimensionality reduction, clustering, two-sample differential expression testing and gene ranking techniques. Additionally, it also provides methods for retrieving enriched pathways.

License MIT + file LICENSE

Depends R (>= 4.1), S4Vectors (>= 0.19.23), SummarizedExperiment (>= 1.11.6)

VignetteBuilder knitr

Imports dynamicTreeCut, dplyr, edgeR, DESeq2, ggplot2 (>= 3.0.0), graphics, Hmisc, limma, methods, magrittr, proxy, stats, tibble, tidyr, vegan, viridis, utils

Suggests Biobase, BiocFileCache (>= 1.5.8), circlize, ComplexHeatmap, GO.db, grDevices, grid, knitr, org.Mm.eg.db, org.Hs.eg.db, MASS, RColorBrewer, rmarkdown, UpSetR,

Collate TimeSeriesExperiment-class.R TimeSeriesExperiment-methods.R utils.R pca.R clustering.R differential_expression.R pathway_over_representation.R plots.R data.R

Encoding UTF-8

RoxygenNote 7.1.1

biocViews TimeCourse, Sequencing, RNASeq, Microbiome, GeneExpression, ImmunoOncology, Transcription, Normalization, DifferentialExpression, PrincipalComponent, Clustering, Visualization, Pathways

BugReports https://github.com/nlhuong/TimeSeriesExperiment/issues

URL https://github.com/nlhuong/TimeSeriesExperiment

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

git_branch master

git_last_commit fcb85c7

git_last_commit_date 2021-08-31

Date/Publication 2021-10-12

Author Lan Huong Nguyen [cre, aut] (<https://orcid.org/0000-0003-3397-0380>)

Maintainer Lan Huong Nguyen <nlhuong90@gmail.com>

R topics documented:

`.addLagsToTimeSeries` *Add differences to the time-course data.*

Description

This function add lag difference features to the time-course data.

Usage

```
.addLagsToTimeSeries(timeseries, lambda)
```

Arguments

`timeseries`	a data matrix or data.frame where each column corresponds to consecutive time point.
`lambda`	Weights for each lag difference, for time-course data. Length of `lambda` specifies number of lags to include. By default lag of order one and two are included with coefficients 0.5 and 0.25 respectively.

Value

a data matrix with added difference lags.

`.imputeData` *Imputing missing samples*

Description

Estimates assay values for missing timepoints

Usage

```
.imputeData(raw.data, sample.data)
```

Arguments

`raw.data`	Raw input data with columns corresponding to samples (observations) and rows to features
`sample.data`	Optional. A `data.frame` object were rows are samples (observations) and columns are sample attributes (e.g. group/condition, replicate, timepoint)

Details

Sometimes datasets include a missing sample. Here we impute the expression values by taking the mean expression of two samples from the same group and replicate (same individual) at two times surrounding the time of the missing sample

Value

list of modified raw.data and sample.data

`addLags` *Add differences to the time-course data.*

Description

This function concatenates lags to time-course data stored in elements of the `timeSeries` slot.

Usage

```
addLags(object, lambda = c(0.5, 0.25))
```

Arguments

`object`	A `TimeSeriesExperiment` object
`lambda`	Weights for each lag difference, for time-course data. Length of `lambda` specifies number of lags to include. By default lag of order one and two are included with coefficients 0.5 and 0.25 respectively.

Value

Returns `TimeSeriesExperiment` object with lags added to elements in `timeSeries` slot.

Examples

```
data("endoderm_small")
endoderm_small <- collapseReplicates(endoderm_small)
endoderm_small <- makeTimeSeries(endoderm_small)
endoderm_small <- addLags(endoderm_small)
head(timeSeries(endoderm_small, "ts"))
head(timeSeries(endoderm_small, "ts_collapsed"))
```

assayCollapsed *Collapsed data.*

Description

Assay and colData collapsed over replicates. The values can be computed with set with collapseReplicates function.

Usage

```
assayCollapsed(object, ...)

## S4 method for signature 'TimeSeriesExperiment'
assayCollapsed(object)

assayCollapsed(object, ...) <- value

## S4 replacement method for signature 'TimeSeriesExperiment'
assayCollapsed(object) <- value

colDataCollapsed(object, ...)

## S4 method for signature 'TimeSeriesExperiment'
colDataCollapsed(object)

colDataCollapsed(object, ...) <- value

## S4 replacement method for signature 'TimeSeriesExperiment'
colDataCollapsed(object) <- value
```

Arguments

object	a TimeSeriesExperiment object.
...	argiments to other functions.
value	a numerical matrix

Value

a DataFrame

Examples

```
data("endoderm_small")
endoderm_small <- collapseReplicates(endoderm_small)
head(assayCollapsed(endoderm_small))

data("endoderm_small")
```

```
endoderm_small <- collapseReplicates(endoderm_small)
head(colDataCollapsed(endoderm_small))
```

`assignClusterDynamic` *Cluster assignment using dynamic branch cutting.*

Description

This function computes the cluster assignment for hierarchical clustering results using dynamic branch cutting.

Usage

```
assignClusterDynamic(hclst, max_height = 0.9, ...)
```

Arguments

`hclst`	an object of class `hclust` representing the clustering of features of X.
`max_height`	a fraction of the total tree height used to compute `maxTreeHeight` argument for `cutreeDynamic` function from `dynamicTreeCut` package.
`...`	other parameters for `cutreeDynamic`.

Value

a data.frame containing cluster assignments.

`assignClusterStatic` *Cluster assignment using static branch cutting.*

Description

This function computes the cluster assignment for hierarchical clustering results using static branch cutting.

Usage

```
assignClusterStatic(hclst, h = NULL, k = NULL)
```

Arguments

`hclst`	an object of class `hclust` representing the clustering of features of X.
`h`	a the fraction of the max tree height at which to cut and assign labels.
`k`	an integer scalar or vector with the desired number of groups

Value

a data.frame containing cluster assignments.

clusterAssignment *Cluster analysis results*

Description

Getter methods for clusterAssignment slot of a TimeSeriesExperiment object. The slot is a list with with elements named: 'settings', 'hclust', 'cluster_map', 'clust_centroids' storing results from running clusterTimeSeries function.

Usage

```
clusterAssignment(object, ...)

## S4 method for signature 'TimeSeriesExperiment'
clusterAssignment(object, name = NULL)

clusterMap(object, ...)

## S4 method for signature 'TimeSeriesExperiment'
clusterMap(object)
```

Arguments

object	a TimeSeriesExperiment object.
...	argiments to other functions.
name	one of elements of 'clusterAssignment' slot: 'settings', 'hclust', 'cluster_map', 'clust_centroids'. If NULL, all elements are returned.

Value

a data.frame

Examples

```
data("endoderm_small")
endoderm_small <- clusterTimeSeries(endoderm_small)
clusterAssignment(endoderm_small, name = 'settings')
head(clusterAssignment(endoderm_small, name = 'final_cluster_map'))
head(clusterAssignment(endoderm_small, name = 'clust_centroids'))
```

clusterData *Cluster time series data.*

Description

Perform cluster assignment using hierarchical clustering and static or dynamic branch cutting. If 'subset' is specified the clsutering is performed on the subset of the data, and the rest of the rows are assigned based on the distances to the centroids of computed clusters.

Usage

```
clusterData(
  X,
  dist = "euclidean",
  dynamic = FALSE,
  hclust_params = list(),
  static_cut_params = list(h = 0.5),
  dynamic_cut_params = list(max_height = 0.9)
)
```

Arguments

X	a data matrix or data.frame where rows are time series.
dist	the distance metric for the dissimilarity used for clustering.
dynamic	whether dynamic branch cutting should be done for cluster assignment.
hclust_params	parameters for hclust function.
static_cut_params	parameters for assignClusterStatic.
dynamic_cut_params	parameters for assignClusterDynamic.

Value

a list with the hclust object, as well as clust_map and clust_centroids data.frames.

Examples

```
data("endoderm_small")
endoderm_small <- normalizeData(endoderm_small)
X <- SummarizedExperiment::assays(endoderm_small)$norm
clust_res <- clusterData(X)
head(clust_res$clust_centroids)
head(clust_res$clust_map)
```

clusterTimeSeries *Cluster time series features.*

Description

Find the cluster assignment for timecourse features. Clustering computed for top "n.top.feat" features most variable over time in each of the selected "groups" using time-series expression (collpased over replicates). The cluster assignment of the remaining genes is based on the distance to the closest cluster centroid previously obtained. Hierarchical clustering is performed and both static and dynamic branch cutting algorithm are available for assigning cluster membership.

Usage

```
clusterTimeSeries(
  object,
  n.top.feat = 1000,
  groups.selected = "all",
  lambda = c(0.5, 0.25),
  clust.params = list()
)
```

Arguments

object	A TimeSeriesExperiment object
n.top.feat	A number of top most variable time-course features to use for clustering.
groups.selected	One or multiple groups from object@group to take into account when aggregating time-course features.
lambda	Weights for each lag difference, for time-course data. Length of lambda specifies number of lags to include. By default lag of order one and two are included with coefficients 0.5 and 0.25 respectively.
clust.params	A list contating arguments for hierarchical clustering. For details see clusterData.

Value

a TimeSeriesExperiment object with cluster assignment stored in cluster.map slot.

Examples

```
data("endoderm_small")
endoderm_small <- clusterTimeSeries(endoderm_small)
head(clusterMap(endoderm_small))
```

colData<-,TimeSeriesExperiment,ANY-method
Column data for TimeSeriesExperiment

Description

colData() slots holds information on individual samples including corresponding sample name in column 'sample' as well as time, group and replicate.

Usage

```
## S4 replacement method for signature 'TimeSeriesExperiment,ANY'
colData(x, ...) <- value
```

Arguments

x	a TimeSeriesExperiment object
...	argiments to other functions.
value	a DataFrame with new sample information

Details

The setter also updates the information in timepoint, replicate and group slots and resets the time-series analysis results to NULL.

Value

a S4Vectors::DataFrame

Examples

```
data("endoderm_small")
head(colData(endoderm_small))
newdf <- colData(endoderm_small)
newdf$random <- sample(ncol(endoderm_small), ncol(endoderm_small))
colData(endoderm_small) <- newdf
head(colData(endoderm_small))
```

collapseReplicates *Collapse data over replicates.*

Description

This function aggregates the data over replicates, i.e. returns collapse data for each group and at each time point.

Usage

```
collapseReplicates(object, FUN = mean)
```

Arguments

object	A TimeSeriesExperiment object
FUN	the aggreagate function. Default is mean

Value

Returns TimeSeriesExperiment object after collapsing over replicates. Collapsed data is stored sample.data.collapsed and data.collapsed slots.

Examples

```
data("endoderm_small")
endoderm_small <- collapseReplicates(endoderm_small)
assayCollapsed(endoderm_small)[1:10, 1:6]
```

colnames<-,TimeSeriesExperiment-method *Row and column name getters and setters for* TimeSeriesExperiment *object.*

Description

colnames(), rownames() can be used to reset the dimensions, column and row names respectively. The data will be updated across all slots of TimeSeriesExperiment.

Usage

```
## S4 replacement method for signature 'TimeSeriesExperiment'
colnames(x) <- value

## S4 replacement method for signature 'TimeSeriesExperiment'
rownames(x) <- value
```

Arguments

x	a TimeSeriesExperiment object.
value	a character vector or a list of two character vectors with new dimension names.

Details

Setting colnames() automatically updates information in dimensionReduction slot.

Value

a character vector

Examples

```
data("endoderm_small")
head(colnames(endoderm_small))
colnames(endoderm_small) <- paste0("Smp", 1:ncol(endoderm_small))
head(colnames(endoderm_small))
```

---

dataToTimeSeries *Data to time-series*

Description

Function that splits data to time series for each replicate included.

Usage

```
dataToTimeSeries(X, timepoint, group = NULL, replicate = NULL)
```

Arguments

X	a data matrix or data.frame where columns correspond to samples and rows to features.
timepoint	a vector of length equal to ncol(X) indicating the time variable corresponding to the sample (data column).
group	a vector of length equal to ncol(X) indicating the group membership corresponding to the sample (data column).
replicate	a vector of length equal to ncol(X) indicating the replicate variable corresponding to the sample (data column).

Value

a data.frame with ordered time series for each replicate.

Examples

```
X <- matrix(rnorm(1000), ncol = 50)
group <- rep(c("A", "B"), each = 25)
replicate <- rep(paste0("rep", 1:5), each = 5)
time <- rep(1:5, 10)
tc <- dataToTimeSeries(X, time, replicate, group)
head(tc)
```

---

differentialExpression

Differential Expression for TimeSeriesExperiment

Description

Getter method for differentialExpression slot of a TimeSeriesExperiment. The slot is a list with differential expression results possibly containing elements named 'timepoint_de', and 'trajectory_de' computed with timepointDE and trajectoryDE functions.

Usage

```
differentialExpression(object, ...)

## S4 method for signature 'TimeSeriesExperiment'
differentialExpression(object, name = NULL)
```

Arguments

object	a TimeSeriesExperiment object.
...	argiments to other functions.
name	one of elements of 'differentialExpression' slot: 'timepoint_de', 'trajectory_de'. If NULL, all elements are returned.

Value

a data.frame or a list of data.frames

Examples

```
data("endoderm_small")
endoderm_small <- trajectoryDE(endoderm_small)
head(differentialExpression(endoderm_small, "trajectory_de"))
```

dimensionReduction *Dimension reduction results*

Description

Getter methods for dimensionReduction slot of a TimeSeriesExperiment object. The slot is a list of data.frames: 'pca_sample', 'pca_feature' and 'pca_eigs' storing results from a PCA projection.

Usage

```
dimensionReduction(object, ...)

## S4 method for signature 'TimeSeriesExperiment'
dimensionReduction(object, name = NULL)
```

Arguments

object	a TimeSeriesExperiment object.
...	argiments to other functions.
name	one of elements of 'dimensionReduction' slot: 'pca_sample', 'pca_feature' and 'pca_eigs' for returning the entire list. If NULL, all elements are returned.

Value

a data.frame or a list of data.frames

Examples

```
data("endoderm_small")
endoderm_small <- runPCA(endoderm_small)
head(dimensionReduction(endoderm_small, "pca_sample")[, 1:3])
```

endoderm_small *Endoderm differentiation*

Description

A 'TimeSeriesExperiment' object contating gene expression data, feature and sample data from a study by Blake et al. "A Comparative Study Of Endoderm Differentiation In Humans And Chimpanzees". Here we include only a small fraction of the data including top 250 most variable genes, to serve as a toy exapmle for TimeSeriesExperiment.

Usage

```
data(endoderm_small)
```

Format

A TimeSeriesExperiment object

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE98411

References

Blake et al. (2017) (bioRxiv)

---

filterFeatures *Filter features*

Description

Filters TimeSeriesExperiment object to keep only chosen features. All relevant slots are updates.

Usage

```
filterFeatures(object, features)
```

Arguments

object	TimeSeriesExperiment object
features	features (genes) to keep

Details

The slots for collapsed data and time series formatted data are filtered accordingly, but dimensionReduction, clusterAssignment and differentialExpression are reset to NULL as different set of features would output in different results.

Value

"TimeSeriesExperiment" object

Examples

```
data("endoderm_small")
features <- 1:100
endoderm_small <- filterFeatures(endoderm_small, features)
endoderm_small
```

groups *Group information*

Description

Group information

Usage

```
groups(object, ...)

## S4 method for signature 'TimeSeriesExperiment'
groups(object)

groups(object, ...) <- value

## S4 replacement method for signature 'TimeSeriesExperiment'
groups(object) <- value
```

Arguments

object	a TimeSeriesExperiment object.
...	argiments to other functions.
value	a character vector with new group membership

Value

a character vector

Examples

```
data("endoderm_small")
head(groups(endoderm_small))
groups(endoderm_small) <- sample(c("A", "B"),
    ncol(endoderm_small), replace = TRUE)
head(groups(endoderm_small))
```

makeTimeSeries *Convert data to time-series*

Description

This function converts the wide data matrix to time-course long `data.frame` format where each row gives data values over time (at each time point) for each feature, group, and replicate.

Usage

```
makeTimeSeries(
  object,
  feature.trans.method = "var_stab",
  var.stabilize.method = "asinh"
)
```

Arguments

`object`	A `TimeSeriesExperiment` object
`feature.trans.method`	Method for feature normalization. Default "none". Currently supports only "none" (no transformation), "scale_feat_sum" (scaling by feature sum), or "var_stab" (variance stabilization). Default is "var_stab".
`var.stabilize.method`	Method for variance stabilization (VST). Currently, supports "none" (no VST), "log1p" (log plus one), "asinh" (inverse hyperbolic sine) or "deseq" (varianceStabilizingTransformat function from `DESeq2` package). Default is "log1p".

Value

Returns `TimeSeriesExperiment` object after conversion to time-course format. Converted data is stored in `timecourse.data` slot.

Examples

```
data("endoderm_small")
endoderm_small <- makeTimeSeries(endoderm_small)
names(timeSeries(endoderm_small))
head(timeSeries(endoderm_small)[[1]])
```

`makeTimeSeriesExperimentFromExpressionSet` *TimeSeriesExperiment constructor from ExpressionSet*

Description

`TimeSeriesExperiment` constructor initializes the `TimeSeriesExperiment` object from `ExpressionSet` and populates the time, replicate, and group slots.

Usage

```
makeTimeSeriesExperimentFromExpressionSet(
  eset,
  timepoint = NULL,
  group = NULL,
  replicate = NULL
)
```

Arguments

`eset`	ExpressionSet object
`timepoint`	a vector indicating timepoint at which each sample was collected or a character string equal to one of the column names of a supplied `ExpressionSet`.
`group`	a vector indicating a group membership for each sample or a character string equal to one of the column names of a supplied `ExpressionSet` If not specified, the group is set to 'G1' for each sample.
`replicate`	a vector indicating a replicate id of each sample or a character string equal to one of the column names of a supplied `ExpressionSet` If not specified, the replicate is set to 'R1' for each sample.

Details

`TimeSeriesExperiment` is an extension of `SummarizedExperiment` class.

Value

Returns an initialized TimeSeriesExperiment object.

Examples

```
raw <- matrix(runif(3000), ncol = 30)
pheno.data <- data.frame(
   time = rep(rep(1:5, each = 3), 2),
   replicate = rep(1:3, 10),
   group = rep(1:2, each = 15))
feature.data <- data.frame(
   feature = paste0("F", 1:100)
```

```
)
## Not run:
    library(Biobase)
    test_eset <- ExpressionSet(
         raw, phenoData = AnnotatedDataFrame(pheno.data),
         featureData = AnnotatedDataFrame(feature.data))
    test_TimeSeriesExperiment <- makeTimeSeriesExperimentFromExpressionSet(
       test_eset, timepoint = "time", group = "group",
       replicate = "replicate")
    test_TimeSeriesExperiment

## End(Not run)
```

`makeTimeSeriesExperimentFromSummarizedExperiment` *TimeSeriesExperiment constructor from SummarizedExperiment*

Description

`TimeSeriesExperiment` constructor initializes the object from SummarizedExperiment::SummarizedExperiment and populates the time, replicate, and group slots.

Usage

```
makeTimeSeriesExperimentFromSummarizedExperiment(
  se,
  timepoint = NULL,
  group = NULL,
  replicate = NULL
)
```

Arguments

`se`	SummarizedExperiment::SummarizedExperiment object
`timepoint`	a vector indicating timepoint at which each sample was collected or a character string equal to one of the column names of a supplied `SummarizedExperiment`.
`group`	a vector indicating a group membership for each sample or a character string equal to one of the column names of a supplied `SummarizedExperiment` If not specified, the group is set to 'G1' for each sample.
`replicate`	a vector indicating a replicate id of each sample or a character string equal to one of the column names of a supplied `SummarizedExperiment` If not specified, the replicate is set to 'R1' for each sample.

Details

`TimeSeriesExperiment` is an extension of `SummarizedExperiment` class.

Value

Returns an initialized `TimeSeriesExperiment` object.

Examples

```
raw <- matrix(runif(3000), ncol = 30)
pheno.data <- data.frame(
   time = rep(rep(1:5, each = 3), 2),
   replicate = rep(1:3, 10),
   group = rep(1:2, each = 15))
feature.data <- data.frame(
   feature = paste0("F", 1:100)
)
test_sumexp <- SummarizedExperiment::SummarizedExperiment(
   assays = list(counts = raw),
   rowData = feature.data, colData = pheno.data)
test_TimeSeriesExperiment <-
makeTimeSeriesExperimentFromSummarizedExperiment(
   test_sumexp, timepoint = "time", group = "group",
   replicate = "replicate")
test_TimeSeriesExperiment
```

---

meltMatrix *Melt data matrix.*

Description

Convert data matrix to a long format

Usage

```
meltMatrix(X)
```

Arguments

`X`	a data matrix.

Value

a data matrix in a long format.

Examples

```
Z <- matrix(rnorm(100), 20)
Z.m <- meltMatrix(Z)
head(Z.m)
```

normalizeData *Normalize Data*

Description

Normalize data the assay data.

Usage

```
normalizeData(
  object,
  sample.norm.method = "scale_common_factor",
  column.scale.factor = 1e+06
)
```

Arguments

object	A TimeSeriesExperiment object or a data matrix/data frame.
sample.norm.method	Method for sample normalization. Currently supports only scaling to a common factor, "scale_common_factor" which with column.scale.factor = 1e+06 is equivalent to CPM normalization.
column.scale.factor	Sets the scale factor for sample-level normalization

Value

Returns TimeSeriesExperiment object after normalization. Normalized data is stored data slot.

Examples

```
data("endoderm_small")
names(assays(endoderm_small))
endoderm_small <- normalizeData(endoderm_small)
assays(endoderm_small)$norm[1:10, 1:6]
```

pathwayEnrichment *Pathway enrichment testing.*

Description

This is a wrapper around goana function from limma package for testing over-representation of gene ontology (GO) terms or KEGG pathways in sets of genes.

Usage

```
pathwayEnrichment(
  object,
  features,
  species,
  feature_column = "feature",
  universe = NULL,
  clustered = TRUE,
  kegg = FALSE,
  ontology = c("BP", "CC", "MF"),
  fltr_DE = 0.1,
  fltr_N = 500,
  fltr_P.DE = 0.05,
  ...
)
```

Arguments

object	A TimeSeriesExperiment object.
features	A vector of ENTREZID for enrichment testing.
species	A character string specifying the species. See goana for details.
feature_column	the feature column in 'feature.data' slot holding ENTREZIDs, by deafult the 'feature' column.
universe	A vector of genes in the universe. By default all the genes in the 'raw.data' slot.
clustered	Whether features should be grouped based on cluster assignment (stored in object@cluster.features$cluster_map) and tested as separate sets.
kegg	Whether KEGG pathways should be used instead of gene ontology (GO)
ontology	character vector of ontologies to be included in output. Elements should be one or more of "BP", "CC" or "MF".
fltr_DE	A scalar fraction of the number of genes in tested set to use as a threshold for filtering genes based on "DE" column (the number of genes in the DE set). Default is 0.1, i.e. at least 0.1 genes in the set must be present in the pathway.
fltr_N	A number of genes used as a threshold to filter out all pathway terms of size greater than the threshold. Default is 500.
fltr_P.DE	A p-value threshold to filter out terms in the enrichment results. Default is 0.05.
...	other parameters for goana or limma::kegga().

Value

a data.frame or list of data.frames with enrichment results.

Examples

```
data("endoderm_small")
selected_genes <- c('114299', '2825', '3855', '221400', '7941',
```

```
                    '6164', '1292', '6161', '6144', '23521')
enrich_res <- pathwayEnrichment(
  object = endoderm_small, clustered = FALSE,
  features = selected_genes,
  species = "Hs", ontology = "BP", fltr_DE = 0,
  fltr_N = Inf, fltr_P.DE = 0.05)
head(enrich_res)
```

`plotEnrichment` *Plot enrichment results.*

Description

Plotting top most enriched terms found with DE methods and tested for overrepresentation in GO/KEGG db using goana/kegga from limma package.

Usage

```
plotEnrichment(enrich, n_max = 15)
```

Arguments

`enrich`	a data matrix or data.frame with enrichment result - outputs from `pathwayEnrichment` function or `goana`, `limma::kegga()`. Must contain columns Term, DE, and P.DE.
`n_max`	max number of terms to show

Value

a ggplot object

Examples

```
data("endoderm_small")
selected_genes <- c('114299', '2825', '3855', '221400', '7941',
                    '6164', '1292', '6161', '6144', '23521')
enrich_res <- pathwayEnrichment(
  object = endoderm_small, clustered = FALSE,
  features = selected_genes,
  species = "Hs", ontology = "BP", fltr_DE = 0,
  fltr_N = Inf, fltr_P.DE = 0.05)
plotEnrichment(enrich = enrich_res, n_max = 15)
```

plotHeatmap *Plot a heatmap*

Description

Generates a heatmap of expression data.

Usage

```
plotHeatmap(
  object,
  num.feat = 200,
  scale = TRUE,
  feat_desc = "feature",
  sample_desc = "sample"
)
```

Arguments

object	A TimeSeriesExperiment object
num.feat	Number of top most features to use.
scale	Whether to scale the data (by features) before plotting.
feat_desc	One of the column names from feature_data(object) to describe the features.
sample_desc	ne of the column names from sample_data(object) to describe the samples.

Value

Returns a ggplot2 objet.

Examples

```
data("endoderm_small")
endoderm_small <- normalizeData(endoderm_small)
## Not run:
    plotHeatmap(endoderm_small)

## End(Not run)
```

plotSamplePCA *Plot a standard PCA*

Description

Generates a standard PCA plot of observations in the dataset.

Usage

```
plotSamplePCA(object, axis = c(1, 2), col.var = NULL, ...)
```

Arguments

object	A TimeSeriesExperiment object
axis	An integer vector indicating principal components to use for plotting, by default 1:2.
col.var	A character string indicating a column from sample_data(object) which should be used for coloring the points. By default NULL.
...	other parameters ggplot.

Value

Returns a ggplot2 objet.

Examples

```
data("endoderm_small")
endoderm_small <- runPCA(endoderm_small)
plotSamplePCA(endoderm_small, col.var = "group")
```

plotTimeSeries *Plot selected time series.*

Description

Plotting expression over time for selected genes curve and colors correspond to distinct groups.

Usage

```
plotTimeSeries(
  object,
  features = rownames(object),
  trans = FALSE,
  smooth = TRUE,
  ncol = 5,
  scales = "free"
)
```

Arguments

object	A TimeSeriesExperiment object.
features	A vector of names of selected features to plot.
trans	A boolean indicating whether (TRUE) transformed, variance stabilized, assay values should be printed or (FALSE) just normalized by sample depth.
smooth	If TRUE a smoothed line is plotted for each gene and each group, else a piecewise linear average (over replicates) curve is plotted.
ncol	An integer indicating the number of columns for facetting. Default is 5.
scales	character scalar indecating facet scales, by default "free".

Value

list of ggplot objects

Examples

```
data("endoderm_small")
feat_to_plot <- rownames(endoderm_small)[1:10]
plotTimeSeries(endoderm_small, features = feat_to_plot, smooth = FALSE)
```

---

plotTimeSeriesClusters

Plot (time) series over clusters.

Description

Plots timecourse (aggregated over replicates) feature data faceted by computed clusters and experimental group.

Usage

```
plotTimeSeriesClusters(
  object,
  features = NULL,
  transparency = 0.5,
  ncol = 4,
  scales = "free"
)
```

Arguments

object	A TimeSeriesExperiment object.
features	A vector of names of selected features to plot.
transparency	transparency of trajectory lines.
ncol	number of columns in the factet plot.
scales	character scalar indecating facet scales, by default "free".

Value

ggplot object

Examples

```
data("endoderm_small")
endoderm_small <- clusterTimeSeries(endoderm_small)
plotTimeSeriesClusters(endoderm_small)
```

`plotTimeSeriesPCA` *Overlay (time) series over PCA grid*

Description

PCA plot for data features, with time-series levels overlayed on top.

Usage

```
plotTimeSeriesPCA(
  object,
  axis = c(1, 2),
  m = 20,
  n = 20,
  group.highlight = NULL,
  linecol = NULL,
  ...
)
```

Arguments

`object`	A `TimeSeriesExperiment` object.
`axis`	An integer vector indicating principal components to use for plotting, by default 1:2.
`m`	a number of tiles in a grid in the horizontal direction.
`n`	a number of tiles in a grid in the vertical direction.
`group.highlight`	An optional character string indicating the group subset for which the time-course trends should be plotted. By default all time-course trends are plotted for all groups.
`linecol`	a vector indicating the color of the gene profile trend line, different for each group.
`...`	other parameters for the line plots.

Value

None

Examples

```
data("endoderm_small")
endoderm_small <- runPCA(endoderm_small)
plotTimeSeriesPCA(endoderm_small)
```

---

replicates *Replicate information*

Description

Replicate information

Usage

```
replicates(object, ...)

## S4 method for signature 'TimeSeriesExperiment'
replicates(object)

replicates(object, ...) <- value

## S4 replacement method for signature 'TimeSeriesExperiment'
replicates(object) <- value
```

Arguments

object	a TimeSeriesExperiment object.
...	argiments to other functions.
value	a character vector with new replicate ids.

Value

a character vector

Examples

```
data("endoderm_small")
head(replicates(endoderm_small))
replicates(endoderm_small) <- sample(c("R1", "R2", "R3"),
    ncol(endoderm_small), replace = TRUE)
head(replicates(endoderm_small))
```

rowData<-,TimeSeriesExperiment-method *Row data for* TimeSeriesExperiment

Description

rowData() holds information on individual features including corresponding feature name in column 'feature'.

Usage

```
## S4 replacement method for signature 'TimeSeriesExperiment'
rowData(x, ...) <- value
```

Arguments

x	a TimeSeriesExperiment object.
...	argiments to other functions.
value	a DataFrame with new feature data

Value

a S4Vectors::DataFrame

Examples

```
data("endoderm_small")
head(rowData(endoderm_small))
rowData(endoderm_small) <- data.frame(
    feature = rowData(endoderm_small)$feature,
    random = sample(nrow(endoderm_small), nrow(endoderm_small)))
head(rowData(endoderm_small))
```

runPCA *Run PCA scores.*

Description

Compute the projection of either samples or time series features onto a PCA space. In case, of PCA for features, the PCA can be computed for individual sample group as indicated or for samples from all groups. In either, case the data is first collapsed over replicates, so that each gene is represented as a vector of a single time course.

Usage

```
runPCA(
  object,
  collapse.replicates = FALSE,
  groups.selected = NULL,
  var.stabilize.method = "log1p"
)
```

Arguments

object	A TimeSeriesExperiment object
collapse.replicates	Whether PCA should be computed on the data with replicates aggregated.
groups.selected	An optional character string indicating a particular group of samples PCA should be applied to. By default set to NULL and all groups are included.
var.stabilize.method	Method for variance stabilization (VST). Currently, supports "none" (no VST), "log1p" (log plus one), "asinh" (inverse hyperbolic sine) or "deseq" (varianceStabilizingTransformat function from DESeq2 package). Default is "log1p".

Value

Returns TimeSeriesExperiment object with PCA results in dim.red slot, a lists containing matrices of coordinates 'pca_sample', and 'pca_features', as well as a vector 'pca_eigs'.

Examples

```
data("endoderm_small")
endoderm_small <- runPCA(endoderm_small)
head(dimensionReduction(endoderm_small, "pca_sample")[, 1:5])
head(dimensionReduction(endoderm_small, "pca_eigs"))
```

---

show,TimeSeriesExperiment-method

show method for TimeSeriesExperiment

Description

show method for TimeSeriesExperiment

Usage

```
## S4 method for signature 'TimeSeriesExperiment'
show(object)
```

Arguments

object	A TimeSeriesExperiment object

Value

nothing, just prints to console

---

timepointDE *Differential timepoint expression testing.*

Description

This is a wrapper around lmFit and voom functions from limma package for testing differential expression at specified timepoints.

Usage

```
timepointDE(object, timepoints = "all", min_gene_sum = 1, alpha = 0.05)
```

Arguments

object	A TimeSeriesExperiment object.
timepoints	Vector of timepoints to test at.
min_gene_sum	A scalar for filtering sparse genes before DE testing. Default is 1.
alpha	A scalar for level of significance. Default is 0.05.

Value

a TimeSeriesExperiment object with timepoint differential expression testing results stored in 'timepoint_de' element in diff.expr slot.

Examples

```
data("endoderm_small")
endoderm_small <- timepointDE(endoderm_small, timepoint = 1.0)
head(differentialExpression(endoderm_small, "timepoint_de")$`1`)
```

timepoints *Timepoint information*

Description

Timepoint information

Usage

```
timepoints(object, ...)

## S4 method for signature 'TimeSeriesExperiment'
timepoints(object)

timepoints(object, ...) <- value

## S4 replacement method for signature 'TimeSeriesExperiment'
timepoints(object) <- value
```

Arguments

object	a TimeSeriesExperiment object.
...	argiments to other functions.
value	a numeric vector with new time information.

Value

a numeric vector

Examples

```
data("endoderm_small")
head(timepoints(endoderm_small))
timepoints(endoderm_small) <- sample(1:ncol(endoderm_small))
head(timepoints(endoderm_small))
```

timeSeries *Time series formatted data.*

Description

Getter and setter methods for timeSeries slot of a TimeSeriesExperiment object.

Usage

```
timeSeries(object, ...)

## S4 method for signature 'TimeSeriesExperiment'
timeSeries(object, name = NULL)

timeSeries(object, ...) <- value

## S4 replacement method for signature 'TimeSeriesExperiment'
timeSeries(object) <- value
```

Arguments

object	a TimeSeriesExperiment object.
...	argiments to other functions.
name	a character string, one of 'ts', 'ts_with_lags', 'ts_collapsed' and 'ts_collapsed_with_lags'. If NULL, all elements are returned.
value	replacement list

Details

timeSeries slot is a list with 'ts' and (optionally) 'ts_collapsed' storing data formatted as time-series/time-courses.

Value

a data.frame

Examples

```
data("endoderm_small")
endoderm_small <- makeTimeSeries(endoderm_small)
head(timeSeries(endoderm_small))
head(timeSeries(endoderm_small, name = 'ts'))
```

TimeSeriesExperiment-class

Time Series Experiment Class

Description

TimeSeriesExperiment is an extension of SummarizedExperiment class with the following new slots (in addition to SummarizedExperiment slots).

Constructor for 'TimeSeriesExperiment' object which stores same data as SummarizedExperiment but also slots for time, group, replicate and other time-series formated data useful for applyting data-analysis.

Usage

```
TimeSeriesExperiment(
  ...,
  timepoint = numeric(0),
  group = character(0),
  replicate = character(0)
)
```

Arguments

`...`	For SummarizedExperiment:: SummarizedExperiment, S4 methods `list` and `matrix`, arguments identical to those of the `SimpleList` method.
`timepoint`	a vector indicating timepoint at which each sample was collected or a character string equal to one of the column names of a supplied colData.
`group`	a vector indicating a group membership for each sample or a character string equal to one of the column names of a supplied colData. If not specified, the group is set to 'G1' for each sample.
`replicate`	a vector indicating a replicate id of each sample or a character string equal to one of the column names of a supplied colData. If not specified, the replicate is set to 'R1' for each sample.

Details

The `TimeSeriesExperiment` class is an object (the main data container) used for in the time series/course experiment analysis. It stores all relevant information associated with the dataset, including the raw data, group, replicate and time associated with each sample (column of the data). The object includes also slots for results from some class-specific methods.

`TimeSeriesExperiment` constructor initializes the `TimeSeriesExperiment` TimeSeriesExperiment object and populates the time, replicate, and group slots.

Value

Returns an initialized TimeSeriesExperiment object.

Slots

`timepoint` A vector indicating the time-point of each sample collection.

`group` A vector indicating the group membership of each sample.

`replicate` A vector indicating the replicate id of each sample.

`assayCollapsed` A matrix with assay data aggregated over replicates.

`colDataCollapsed` A DataFrame where rows correspond to samples aggregated over replicates and columns indicate group membership and time-point.

`timeSeries` A list of time-course formatted data. Each element of the list is a DataFrame with the first three columns indicating feature, group, replicate, and the remaining ones storing the assay data at consecutive time points. `TimeSeriesExperiment` methods will typically generate elements of the list named: 'ts', 'ts_collapsed'.

`dimensionReduction` A list of results from applying dimmensionality reduction methods; elements named by technique used.

`clusterAssignment` A list of results from clustering of the time-series features (rows) containing elements: 'settings', 'hclust', 'clust_map' and 'clust_centroids'.

`differentialExpression` A list of results from differential expression analysis. Either from point-wise ('timepoint_de') or trajectory ('trajectory_de') differential expression analysis.

Examples

```
raw <- matrix(runif(3000), ncol = 30)
timepoint <- rep(rep(1:5, each = 3), 2)
replicate <- rep(1:3, 10)
group <- rep(1:2, each = 15)
test_TimeSeriesExperiment <- TimeSeriesExperiment(
    assays = list(raw),
    timepoint = timepoint,
    replicate = replicate,
    group = group)
test_TimeSeriesExperiment
```

---

trajectoryDE *Differential trajectory testing.*

Description

Performs differential trajectory testing for timecourse data using `adonis` method.

Usage

```
trajectoryDE(
  object,
  dist_method = "euclidean",
  p_adj_method = "BH",
  lambda = c(0.5, 0.25),
  verbose = TRUE,
  ...
)
```

Arguments

`object`	A `TimeSeriesExperiment` object.
`dist_method`	the name of any method used in vegdist to calculate pairwise distances, "euclidean" by defaults.
`p_adj_method`	a correction method. See details in `p.adjust`. Default is "BH".

lambda	Weights for each lag difference, for time-course data. Length of lambda specifies number of lags to include. Default is c(0.5, 0.25) for lag 1 and 2. Used only if 'timecourse.data' slot not initialized.
verbose	whether code comments should be printed. Default is TRUE.
...	other options to adonis function from vegan.

Value

a data.frame with adonis results for all features.

Examples

```
data("endoderm_small")
endoderm_small <- makeTimeSeries(endoderm_small)
## Not run:
   endoderm_small <- trajectoryDE(endoderm_small)
   head(differentialExpression(endoderm_small, "trajectory_de"))

## End(Not run)
```

---

varianceStabilization *Variance stabilization.*

Description

This function performs variance stabilization on assay data.

Usage

```
varianceStabilization(X, method = "asinh")
```

Arguments

X	an assay data matrix or data.frame where columns correspond
method	Method for variance stabilization (VST). Currently, supports "none" (no VST), "log1p" (log plus one), "asinh" (inverse hyperbolic sine) or "deseq" (varianceStabilizingTransformat function from DESeq2 package). Default is "log1p".

Value

Returns a varianced stabilized data matrix.

Examples

```
X <- sapply(exp(rlnorm(10)), function(m) rnbinom(20, size = 1, mu = m))
head(X)
Y <- varianceStabilization(X, method = "asinh")
head(Y)
```

Index