Package 'FateID'

July 21, 2025

Title Quantification of Fate Bias in Multipotent Progenitors

Version 0.2.2

Date 2022-06-14

Author Dominic Grün <dominic.gruen@gmail.com>

Maintainer Dominic Grün <dominic.gruen@gmail.com>

Description
Application of 'FateID' allows computation and visualization of cell fate bias for multi-lineage single cell transcriptome data. Herman, J.S., Sagar, Grün D. (2018) <DOI:10.1038/nmeth.4662>.

Depends R (>= 3.5.0)

Imports graphics, grDevices, locfit, matrixStats, pheatmap, princurve, randomForest, RColorBrewer, Rtsne, som, stats, umap, utils

Suggests DESeq2, knitr, rmarkdown

VignetteBuilder knitr

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-06-14 11:20:02 UTC

Contents

`compdr` *Computation of dimensional reduction representations*

Description

This function computes dimensional reduction representations to a specified number of dimensions using a number of different algorithms: t-SNE, cmd, diffusion maps, umap

Usage

```
compdr(
  x,
  z = NULL,
  m = c("tsne", "cmd", "umap"),
  k = 2,
  tsne.perplexity = 30,
  umap.pars = NULL,
  seed = 12345
)
```

Arguments

`x`	expression data frame with genes as rows and cells as columns. Gene IDs should be given as row names and cell IDs should be given as column names. This can be a reduced expression table only including the features (genes) to be used in the analysis.
`z`	Matrix containing cell-to-cell distances to be used in the fate bias computation. Default is `NULL`. In this case, a correlation-based distance is computed from `x` by `1 - cor(x)`
`m`	a vector of dimensional reduction representations to be computed. The following representations can be computed: `cmd` (classical multidimensional scaling), `dm` (diffusion map), `tsne` (t-SNE map), `umap` (umap). The default value of m is `c("cmd","tsne","umap")`. Any subset of methods can be selected.

k	vector of integers representing the dimensions for which the dimensional reduction representations will be computed. Default value is 2.
tsne.perplexity	positive number. Perplexity used in the t-SNE computation. Default value is 30.
umap.pars	umap parameters. See **umap** package, umap.defaults. Default is NULL and umap.defaults are used. umap.pars$input is automatically set to "dist", since the umap is computed for the distance object.
seed	integer seed for initialization. If equal to NULL then each run will yield slightly different results due to the randomness of the random forest algorithm. Default is NULL

Value

A two-dimensional list with the dimensional reduction representation stored as data frames as components. Component names for the first dimension are given by one of the following algorithms:

cmd	classical multidimensional scaling computed by the cmdscale function of the **stats** package.
tsne	t-SNE map computed by the Rtsne function of the **Rtsne** package.
umap	umap computed by the umap function of the **umap** package.

Component names of the second dimension are a concatenation of a capital D and an integer number of the dimension. There is one component for each dimension in k.

Examples

```
x <- intestine$x
dr <- compdr(x,z=NULL,m="cmd",k=2,tsne.perplexity=30)
plot(dr[["cmd"]][["D2"]],pch=20,col="grey")
```

---

diffexpnb *Function for differential expression analysis*

Description

This function performs differential expression analysis between two sets of single cell transcriptomes. The inference is based on a noise model or relies on the DESeq2 approach.

Usage

```
diffexpnb(
  x,
  A,
  B,
  DESeq = FALSE,
  method = "pooled",
```

```
  norm = FALSE,
  vfit = NULL,
  locreg = FALSE,
  ...
)
```

Arguments

x	expression data frame with genes as rows and cells as columns. Gene IDs should be given as row names and cell IDs should be given as column names. This can be a reduced expression table only including the features (genes) to be used in the analysis. This input has to be provided if g (see below) is given and corresponds to a valid gene ID, i. e. one of the rownames of x. The default value is NULL. In this case, cluster identities are highlighted in the plot.
A	vector of cell IDs corresponding column names of x. Differential expression in set A versus set B will be evaluated.
B	vector of cell IDs corresponding column names of x. Differential expression in set A versus set B will be evaluated.
DESeq	logical value. If TRUE, then **DESeq2** is used for the inference of differentially expressed genes. In this case, it is recommended to provide non-normalized input data x. Default value is FALSE
method	either "per-condition" or "pooled". If DESeq is not used, this parameter determines, if the noise model is fitted for each set separately ("per-condition") or for the pooled set comprising all cells in A and B. Default value is "pooled".
norm	logical value. If TRUE then the total transcript count in each cell is normalized to the minimum number of transcripts across all cells in set A and B. Default value is FALSE.
vfit	function describing the background noise model. Inference of differentially expressed genes can be performed with a user-specified noise model describing the expression variance as a function of the mean expression. Default value is NULL.
locreg	logical value. If FALSE then regression of a second order polynomial is perfomed to determine the relation of variance and mean. If TRUE a local regression is performed instead. Default value is FALSE.
...	additional arguments to be passed to the low level function DESeqDataSetFromMatrix.

Value

If DESeq equals TRUE, the function returns the output of **DESeq2**. In this case list of the following two components is returned:

cds	object returned by the **DESeq2** function DESeqDataSetFromMatrix.
res	data frame containing the results of the **DESeq2** analysis.

Otherwise, a list of three components is returned:

vf1	a data frame of three columns, indicating the mean m, the variance v and the fitted variance vm for set A.

vf2 a data frame of three columns, indicating the mean m, the variance v and the fitted variance vm for set B.

res a data frame with the results of the differential gene expression analysis with the structure of the DESeq output, displaying mean expression of the two sets, fold change and log2 fold change between the two sets, the p-value for differential expression (pval) and the Benjamini-Hochberg corrected false discovery rate (padj).

Examples

```
x <- intestine$x
y <- intestine$y
v <- intestine$v

tar <- c(6,9,13)
fb <- fateBias(x,y,tar,z=NULL,minnr=5,minnrh=10,nbfactor=5,use.dist=FALSE,seed=NULL,nbtree=NULL)

thr <- .3

A <- rownames(fb$probs)[fb$probs[,"t6"]  > .3]
B <- rownames(fb$probs)[fb$probs[,"t13"] > .3]
de <- diffexpnb(v,A=A,B=B)
```

---

fateBias *Computation of fate bias*

Description

This function computes fate biases for single cells based on expression data from a single cell sequencing experiment. It requires a clustering partition and a target cluster representing a commited state for each trajectory.

Usage

```
fateBias(
  x,
  y,
  tar,
  z = NULL,
  minnr = NULL,
  minnrh = NULL,
  adapt = TRUE,
  confidence = 0.75,
  nbfactor = 5,
  use.dist = FALSE,
  seed = NULL,
  nbtree = NULL,
```

```
  verbose = FALSE,
  ...
)
```

Arguments

x	expression data frame with genes as rows and cells as columns. Gene IDs should be given as row names and cell IDs should be given as column names. This can be a reduced expression table only including the features (genes) to be used in the analysis.
y	clustering partition. A vector with an integer cluster number for each cell. The order of the cells has to be the same as for the columns of x.
tar	vector of integers representing target cluster numbers. Each element of tar corresponds to a cluster of cells committed towards a particular mature state. One cluster per different cell lineage has to be given and is used as a starting point for learning the differentiation trajectory.
z	Matrix containing cell-to-cell distances to be used in the fate bias computation. Default is NULL. In this case, a correlation-based distance is computed from x by 1 - cor(x).
minnr	integer number of cells per target cluster to be selected for classification (test set) in each iteration. For each target cluster, the minnr cells with the highest similarity to a cell in the training set are selected for classification. If z is not NULL it is used as the similarity matrix for this step. Otherwise, 1-cor(x) is used. Default value is NULL and minnr is estimated as the minimum of and 20 and half the median of target cluster sizes.
minnrh	integer number of cells from the training set used for classification. From each training set, the minnrh cells with the highest similarity to the training set are selected. If z is not NULL it is used as the similarity matrix for this step. Default value is NULL and minnrh is estimated as the maximum of and 20 and half the median of target cluster sizes.
adapt	logical. If TRUE then the size of the test set for each target cluster is adapted based on the classification success in the previous iteration. For each target cluster, the number of successfully classified cells is determined, i.e. the number of cells with a minimum fraction of votes given by the confidence parameter for the target cluster, which gave rise to the inclusion of the cell in the test set (see minnr). Weights are then derived by dividing this number by the maximum across all clusters after adding a pseudocount of 1. The test set size minnr is rescaled for each cluster by the respective weight in the next iteration. Default is TRUE.
confidence	real number between 0 and 1. See adapt parameter. Default is 0.75.
nbfactor	positive integer number. Determines the number of trees grown for each random forest. The number of trees is given by the number of columns of th training set multiplied by nbfactor. Default value is 5.
use.dist	logical value. If TRUE then the distance matrix is used as feature matrix (i. e. z if not equal to NULL and 1-cor(x) otherwise). If FALSE, gene expression values in x are used. Default is FALSE.

seed	integer seed for initialization. If equal to NULL then each run will yield slightly different results due to the radomness of the random forest algorithm. Default is NULL
nbtree	integer value. If given, it specifies the number of trees for each random forest explicitely. Default is NULL.
verbose	logical. If TRUE, then print information to console.
...	additional arguments to be passed to the low level function randomForest.

Details

The bias is computed as the ratio of the number of random forest votes for a trajectory and the number of votes for the trajectory with the second largest number of votes. By this means only the trajectory with the largest number of votes will receive a bias >1. The siginifcance is computed based on counting statistics on the difference in the number of votes. A significant bias requires a p-value < 0.05. Cells are assigned to a trajectory if they exhibit a significant bias >1 for this trajectory.

Value

A list with the following three components:

probs	a data frame with the fraction of random forest votes for each cell. Columns represent the target clusters. Column names are given by a concatenation of t and target cluster number.
votes	a data frame with the number of random forest votes for each cell. Columns represent the target clusters. Column names are given by a concatenation of t and target cluster number.
tr	list of vectors. Each component contains the IDs of all cells on the trajectory to a given target cluster. Component names are given by a concatenation of t and target cluster number.
rfl	list of randomForest objects for each iteration of the classification.
trall	vector of cell ids ordered by the random forest iteration in which they have been classified into one of the target clusters.

Examples

```
x <- intestine$x
y <- intestine$y
tar <- c(6,9,13)
fb <- fateBias(x,y,tar,minnr=5,minnrh=20,adapt=TRUE,confidence=0.75,nbfactor=5)
head(fb$probs)
```

filterset *Function for filtering expression data*

Description

This function discards lowly expressed genes from the expression data frame.

Usage

```
filterset(x, n = NULL, minexpr = 2, minnumber = 1)
```

Arguments

x	expression data frame with genes as rows and cells as columns. Gene IDs should be given as row names and cell IDs should be given as column names.
n	ordered vector of cell IDs to be included. Cell IDs need to be column names of x. If not provided, then all cell IDs are included in arbitray order. Default value is NULL.
minexpr	positive real number. This is the minimum expression required for at least minnumber cells. All genes that do not fulfill this criterion are removed. The default value is 2.
minnumber	positive integer number. This is the minimum number of cells in which a gene needs to be expressed at least at a level of minexpr. All genes that do not fulfill this criterion are removed. The default value is 1.

Value

Reduced expression data frame with genes as rows and cells as columns in the same order as in n.

gene2gene *Comparative plot of the expression levels of two genes*

Description

This function produces a scatter plot of the expression levels of two genes. It allows plotting cells of selected clusters and permits highlighting of the fate bias.

Usage

```
gene2gene(
  x,
  y,
  g1,
  g2,
  clusters = NULL,
```

```
  fb = NULL,
  tn = NULL,
  col = NULL,
  tp = 1,
  plotnum = TRUE,
  seed = 12345
)
```

Arguments

x	expression data frame with genes as rows and cells as columns. Gene IDs should be given as row names and cell IDs should be given as column names. This can be a reduced expression table only including the features (genes) to be used in the analysis.
y	clustering partition. A vector with an integer cluster number for each cell. The order of the cells has to be the same as for the columns of x.
g1	gene id corresponding to a valid row names of x. Expression of gene g1 versus gene g2 will be plotted.
g2	gene id corresponding to a valid row names of x. Expression of gene g1 versus gene g2 will be plotted.
clusters	vector of valid cluster ids. Expression is displayed for cells in any of the clusters contained in clusters. If the argument is not given, cells of all clusters are displayed. Default value is NULL.
fb	fateBias object returned by the function fateBias. Default value is NULL. Only if both tn and fb are provided as input, the fate bias will be colour coded.
tn	name of a target cluster, i. e. concatenation of a t and the number of a target cluster. Has to correspond to a column name of fb$probs. The default value is NULL. Only if both tn and fb are provided as input, the fate bias will be colour coded.
col	optional vector of valid color names for all clusters in y ordered by increasing cluster number. Default value is NULL.
tp	Transparency of points in the plot. Default value is 1, i. e. non-transparent.
plotnum	logical value. If TRUE, then cluster numbers are displayed on top of the data points. Default value is TRUE.
seed	integer number. Random seed for determining colour scheme. Default is 12345.

Value

None

Examples

```
x <- intestine$x
y <- intestine$y
v <- intestine$v
```

```
tar <- c(6,9,13)
fb <- fateBias(x,y,tar,z=NULL,minnr=5,minnrh=10,nbfactor=5,use.dist=FALSE,seed=NULL,nbtree=NULL)
gene2gene(v,y,"Muc2__chr7","Apoa1__chr9")
gene2gene(v,y,"Muc2__chr7","Apoa1__chr9",fb=fb,tn="t6",plotnum=FALSE)
```

`getFeat` *Feature selection based on differentially expressed genes*

Description

This function performs a feature selection based on the inference of differentially expressed genes between each target cluster and all remaining cells.

Usage

```
getFeat(x, y, tar, fpv = 0.05, ...)
```

Arguments

`x`	expression data frame with genes as rows and cells as columns. Gene IDs should be given as row names and cell IDs should be given as column names. This can be a reduced expression table only including the features (genes) to be used in the analysis.
`y`	clustering partition. A vector with an integer cluster number for each cell. The order of the cells has to be the same as for the columns of `x`.
`tar`	vector of integers representing target cluster numbers. Each element of `tar` corresponds to a cluster of cells committed towards a particular mature state. One cluster per different cell lineage has to be given and is used as a starting point for learning the differentiation trajectory.
`fpv`	p-value cutoff for calling differentially expressed genes. This is a cutoff for the Benjamini-Hochberg corrected false discovery rate. Default value is 0.05.
`...`	additional arguments to be passed to the low level function `diffexpnb`.

Details

The function determines differentially expressed between the cells in each of the target clusters in comparison to the remaining cells by using `diffexpnb` function.

Value

A filtered expression table with features extracted based on differentially expressed genes.

Examples

```
x <- intestine$x
y <- intestine$y
tar <- c(6,9,13)
xf <- getFeat(x,y,tar,fpv=.05)
```

getPart *Inference of a cell type partition*

Description

This function performs an inference of a cell type partition based on the expression of marker genes.

Usage

```
getPart(x, FMarker, fthr = NULL, n = 25)
```

Arguments

x	expression data frame with genes as rows and cells as columns. Gene IDs should be given as row names and cell IDs should be given as column names. This can be a reduced expression table only including the features (genes) to be used in the analysis.
FMarker	list of vectors of gene IDs corresponding to valid rownames of x. The gene IDs within each component of FMarker are considered as marker genes of one of the cell types in the dataset. The aggregated expression of the genes for each component is compared to a threshold defined by the input argument fthr or n. All cells exceeding this threshold are assigned to a cluster representing cells with expression of the respective marker genes.
fthr	vector of real positive numbers. This vector has to have the same length as the list FMarker and contains a threshold for the aggregated expression of all genes in the corresponding component of FMarker. If NULL then a threshold is inferred from the n top-expressing cells for the genes in the respective component of FMarker.
n	positive integer number. For each component of FMarker the expression of all genes is aggregated in every cell and the n top-expressing cells are extracted. The average expression across these cell defines the expression threshold applied to infer the partitioning. Default value is 25.

Value

A list with the following three components:

part	A vector with a partitioning, i. e. cluster assignment for each cell.
tar	A vector with the numbers of target clusters. Cluster 1 comprises all cells with no enrichment of marker genes. The remaining clusters correspond to cell types up-regulating the markers in the list FMarker in the same order as in this list.
thr	A vector with expression threshold values applied for each component in the list FMarker in the same order as in this list.

Examples

```
x <- intestine$x
y <- intestine$y

FMarker <- list(c("Defa20__chr8","Defa24__chr8"), "Clca3__chr3", "Alpi__chr1")
xf <- getPart(x,FMarker,fthr=NULL,n=5)
```

---

getsom *Topological ordering of pseudo-temporal expression profiles*

Description

This function computes a topological ordering of pseudo-temporal expression profiles of all genes by using 1-dimensional self-organizing maps.

Usage

```
getsom(x, nb = 1000, alpha = 0.5)
```

Arguments

x	expression data frame with genes as rows and cells as columns. Gene IDs should be given as row names and cell IDs should be given as column names. The pseudo-temporal expression profile of each gene is defined by the order of cell IDs, i. e. columns, in x.
nb	positive integer number. Number of nodes of the self-organizing map. Default value is 1000.
alpha	positive real number. Pseudo-temporal expression profiles are derived by a local regression of expression values across the ordered cells using the function loess from the package **stats**. This is the parameter, which controls the degree of smoothing. Larger values return smoother profiles. Default value is 0.5.

Value

A list of the following three components:

som	a som object returned by the function som of the package **som**
x	pseudo-temporal expression profiles, i. e. the input expression data frame x after smoothing by running mean or local regression, respectivey, and normalization. The sum of smoothened gene expression values across all cells is normalized to 1.
zs	data frame of z-score transformed pseudo-temporal expression profiles.

Examples

```
x <- intestine$x
y <- intestine$y
v <- intestine$v

tar <- c(6,9,13)
fb <- fateBias(x,y,tar,z=NULL,minnr=5,minnrh=10,nbfactor=5,use.dist=FALSE,seed=NULL,nbtree=NULL)
dr <- compdr(x,z=NULL,m="cmd",k=2,tsne.perplexity=30)
pr <- prcurve(y,fb,dr,k=2,m="cmd",trthr=0.4,start=NULL)
n <- pr$trc[["t6"]]
fs  <- filterset(v,n,minexpr=2,minnumber=1)
s1d <- getsom(fs,nb=1000,alpha=.5)
```

`impGenes` *Extract genes with high importance values for random forest classification*

Description

This function extracts all genes with an importance value for classifying cells into a given target cluster exceeding a given threshold for at least one of the random forest iterationns.

Usage

```
impGenes(fb, tn, ithr = 0.02, zthr = 2)
```

Arguments

`fb`	fateBias object returned by the function `fateBias`. If `fb` is provided, then a principal curve is computed and shown in the plot. Default value is `NULL`. The curve is only displayed if g equal `NULL`.
`tn`	name of a target cluster, i. e. concatenation of a `t` and the number of a target cluster. Has to correspond to a column name of `fb$probs`.
`ithr`	positive real number. Threshold for the required importance measure (mean decrease in accuracy of classification upon removal, see **randomForest**) to include a gene into the output as important feature for classying cells in `tn`. Default value is 0.02.
`zthr`	positive real number. Threshold for the required z-score of the importance measure (importance divided by the standard deviation of importance) to include a gene into the output as important feature for classying cells in `tn`. Default value is 2.

Value

The function returns a list of two elements.

d	a data frame with mean importance values for all genes surviving the filtering by ithr and zthr. Columns correspond to random forest iterations, starting from the initial target cluster.
d	a data frame with the standard deviation of importance values for all genes surviving the filtering by ithr and zthr. Columns correspond to random forest iterations, starting from the initial target cluster.

The function produces a heatmap of d with hierarchical clustering of the rows using the function pheatmap from the **pheatmap** package.

Examples

```
x <- intestine$x
y <- intestine$y
tar <- c(6,9,13)
fb <- fateBias(x,y,tar,z=NULL,minnr=5,minnrh=10,nbfactor=5,use.dist=FALSE,seed=NULL,nbtree=NULL)
k <- impGenes(fb,"t6",ithr=.02,zthr=2)
```

---

intestine *Single-cell transcriptome data of intestinal epithelial cells*

Description

This dataset contains gene expression values, i. e. transcript counts, of 278 intestinal epithelial cells, and additional information on different cell types in this sample.

Usage

```
intestine
```

Format

A list of the following 5 components:

x data frame with genes as rows and cells as columns. This reduced data frame only contains expression of the most variable genes.

y vector containing a clustering partition of the 278 cells into different cell types

v data frame with genes as rows and cells as columns. This data frame contains expression of all genes.

fcol vector containing colour values for all clusters in y

Value

None

References

Grün et al. (2016) Cell Stem Cell 19(2): 266-77 (PubMed)

`plotdiffgenesnb` *Function for plotting differentially expressed genes*

Description

This is a plotting function for visualizing the output of the `diffexpnb` function as MA plot.

Usage

```
plotdiffgenesnb(
  x,
  pthr = 0.05,
  padj = TRUE,
  lthr = 0,
  mthr = -Inf,
  Aname = NULL,
  Bname = NULL,
  show_names = TRUE,
  ...
)
```

Arguments

`x`	output of the function `diffexpnb`.
`pthr`	real number between 0 and 1. This number represents the p-value cutoff applied for displaying differentially expressed genes. Default value is 0.05. The parameter `padj` (see below) determines if this cutoff is applied to the uncorrected p-value or to the Benjamini-Hochberg corrected false discovery rate.
`padj`	logical value. If `TRUE`, then genes with a Benjamini-Hochberg corrected false discovery rate lower than `pthr` are displayed. If `FALSE`, then genes with a p-value lower than `pthr` are displayed.
`lthr`	real number between 0 and Inf. Differentially expressed genes are displayed only for log2 fold-changes greater than `lthr`. Default value is 0.
`mthr`	real number between -Inf and Inf. Differentially expressed genes are displayed only for log2 mean expression greater than `mthr`. Default value is -Inf.
`Aname`	name of expression set `A`, which was used as input to `diffexpnb`. If provided, this name is used in the axis labels. Default value is `NULL`.
`Bname`	name of expression set `B`, which was used as input to `diffexpnb`. If provided, this name is used in the axis labels. Default value is `NULL`.
`show_names`	logical value. If `TRUE` then gene names displayed for differentially expressed genes. Default value is `FALSE`.
`...`	Additional arguments for function `plot`.

Value

None

Examples

```
x <- intestine$x
y <- intestine$y
v <- intestine$v

tar <- c(6,9,13)
fb <- fateBias(x,y,tar,z=NULL,minnr=5,minnrh=10,nbfactor=5,use.dist=FALSE,seed=NULL,nbtree=NULL)

thr <- .3

A <- rownames(fb$probs)[fb$probs[,"t6"]  > .3]
B <- rownames(fb$probs)[fb$probs[,"t13"] > .3]
de <- diffexpnb(v,A=A,B=B)
plotdiffgenesnb(de,pthr=.05)
```

`plotexpression` *Plotting of pseudo-temporal expression profiles*

Description

This function allows plotting pseudo-temporal expression profiles for single genes or groups of genes.

Usage

```
plotexpression(
  x,
  y,
  g,
  n,
  logsc = FALSE,
  col = NULL,
  name = NULL,
  cluster = FALSE,
  alpha = 0.5,
  types = NULL,
  cex = 3,
  ylim = NULL,
  map = TRUE,
  leg = TRUE,
  seed = 12345,
  ylab = NULL
)
```

Arguments

x	expression data frame with genes as rows and cells as columns. Gene IDs should be given as row names and cell IDs should be given as column names.
y	clustering partition. A vector with an integer cluster number for each cell. The order of the cells has to be the same as for the columns of x.
g	a gene ID corresponding to one of the rownames of x. It can also be a vector of gene IDs. In this case, the aggregated expression across all gene IDs is plotted.
n	ordered vector of cell IDs to be included. Cell IDs need to be column names of x.
logsc	logical value. If TRUE, then log2-transformed values are plotted. Default is FALSE and untransformed values are plotted.
col	optional vector of valid color names for all clusters in y ordered by increasing cluster number. Default value is NULL.
name	optional character string. This argument corresponds to a title for the plot. Default value is NULL. If not provided, and g is given, then name will equal g or g[1], respectively, if g is a vector of gene IDs.
cluster	logical value. If TRUE then the partitioning along the x-axis is indicated be vertical lines representing the boundaries of all positions with a given value in y. The average position across all cells in a cluster will be indicated on the x-axis.
alpha	positive real number. Pseudo-temporal expression profiles are derived by a local regression of expression values across the ordered cells using the function loess from the package **stats**. This is the parameter, which controls the degree of smoothing. Larger values return smoother profiles. Default value is 0.5.
types	optional vector with IDs for different subsets of cells in y, e. g. different batches. All cells with the same ID will be displayed by the same symbol and color. Default value is NULL
cex	size of data points. Default value is 3.
ylim	vector of two numerical values: lower and upper limit of values shown on the y-axis. Default value is NULL and the whole range is shown.
map	logical. If TRUE then data points are shown. Default value is TRUE.
leg	logical. If TRUE then axes and labels are shown. Default value is TRUE.
seed	integer number. Random seed for determining colour scheme. Default is 12345.
ylab	Optional label for the y-axis. Default is NULL and axis is labeled "norm. expression".

Value

None

Examples

```
x <- intestine$x
y <- intestine$y
```

```
v <- intestine$v
fcol <- intestine$col
tar <- c(6,9,13)
fb <- fateBias(x,y,tar,z=NULL,minnr=5,minnrh=10,nbfactor=5,use.dist=FALSE,seed=NULL,nbtree=NULL)
dr <- compdr(x,z=NULL,m="cmd",k=2,tsne.perplexity=30)
pr <- prcurve(y,fb,dr,k=2,m="cmd",trthr=0.4,start=NULL)
n <- pr$trc[["t6"]]
fs  <- filterset(v,n,minexpr=2,minnumber=1)
s1d <- getsom(fs,nb=1000,alpha=.5)
ps <- procsom(s1d,corthr=.85,minsom=3)
# plot average profile of all genes of node 1 in the self-organizing map
g <- names(ps$nodes)[ps$nodes == 1]
plotexpression(v,y,g,n,col=fcol,name="Node 1",cluster=FALSE,alpha=.5,types=NULL)
```

`plotexpressionProfile` *Plotting smoothed pseudo-temporal expression profiles for groups of genes*

Description

This function allows plotting loess-smoothed pseudo-temporal expression profiles for groups of genes. To display gene expression profiles on the same scale, row sums are normalized to one.

Usage

```
plotexpressionProfile(
  x,
  y,
  g,
  n,
  logsc = FALSE,
  col = NULL,
  name = NULL,
  cluster = FALSE,
  alpha = 0.5,
  lwd = 1,
  ylim = NULL,
  seed = 12345,
  ylab = NULL
)
```

Arguments

`x`	expression data frame with genes as rows and cells as columns. Gene IDs should be given as row names and cell IDs should be given as column names.

y	clustering partition. A vector with an integer cluster number for each cell. The order of the cells has to be the same as for the columns of x.
g	a gene ID corresponding to one of the rownames of x. It can also be vector of gene IDs. In this case, a separate profile is plotted for each gene in g.
n	ordered vector of cell IDs to be included. Cell IDs need to be column names of x.
logsc	logical value. If TRUE, then log2-transformed values are plotted. Default is FALSE and untransformed values are plotted.
col	optional vector of valid color names used for the profiles of all genes in g. Default value is NULL.
name	optional character string. This argument corresponds to a title for the plot. Default value is NULL. If not provided, and g is given, then name will equal g or g[1], respectively, if g is a vector of gene IDs.
cluster	logical value. If TRUE then the partitioning along the x-axis is indicated be vertical lines representing the boundaries of all positions with a given value in y. The average position across all cells in a cluster will be indicated on the x-axis.
alpha	positive real number. Pseudo-temporal expression profiles are derived by a local regression of expression values across the ordered cells using the function loess from the package **stats**. This is the parameter, which controls the degree of smoothing. Larger values return smoother profiles. Default value is 0.5.
lwd	line width of profiles. Default value is 1.
ylim	vector of two numerical values: lower and upper limit of values shown on the y-axis. Default value is NULL and the whole range is shown.
seed	integer number. Random seed for determining colour scheme. Default is 12345.
ylab	Optional label for the y-axis. Default is NULL and axis is labeled "norm. expression".

Value

None

Examples

```
x <- intestine$x
y <- intestine$y
v <- intestine$v
fcol <- intestine$col
tar <- c(6,9,13)
fb <- fateBias(x,y,tar,z=NULL,minnr=5,minnrh=10,nbfactor=5,use.dist=FALSE,seed=NULL,nbtree=NULL)
dr <- compdr(x,z=NULL,m="cmd",k=2,tsne.perplexity=30)
pr <- prcurve(y,fb,dr,k=2,m="cmd",trthr=0.4,start=NULL)
n <- pr$trc[["t6"]]
fs  <- filterset(v,n,minexpr=2,minnumber=1)
s1d <- getsom(fs,nb=1000,alpha=.5)
ps <- procsom(s1d,corthr=.85,minsom=3)
# plot average profile of all genes of node 1 in the self-organizing map
```

```
g <- sample(names(ps$nodes)[ps$nodes == 1],5)
plotexpressionProfile(v,y,g,n,col=fcol,name="Node 1",alpha=.2)
```

---

plotFateMap *Plot dimensional reduction representation of the expression data*

Description

This function plots a dimensional reduction representation using the output of the compdr function as input. It allows display of the input clusters as well as color coding of fate bias probabilities and gene expression.

Usage

```
plotFateMap(
  y,
  dr,
  x = NULL,
  g = NULL,
  n = NULL,
  prc = FALSE,
  logsc = FALSE,
  k = 2,
  m = "cmd",
  kr = NULL,
  col = NULL,
  fb = NULL,
  trthr = NULL,
  start = NULL,
  tp = 1,
  seed = 12345,
  ...
)
```

Arguments

y	clustering partition. A vector with an integer cluster number for each cell. The order of the cells has to be the same as for the columns of x.
dr	list of dimensional reduction representations returned by the function compdr.
x	expression data frame with genes as rows and cells as columns. Gene IDs should be given as row names and cell IDs should be given as column names. This can be a reduced expression table only including the features (genes) to be used in the analysis. This input has to be provided if g (see below) is given and corresponds to a valid gene ID, i. e. one of the rownames of x. The default value is NULL. In this case, cluster identities are highlighted in the plot.

g	either the name of one of the trajectories from fb or a gene ID corresponding to one of the rownames of x. In the latter case, the input argument x needs to be provided. A vector of gene IDs can also be provided. In this case, the aggregated expression across all gene IDs is plotted. If g equals E, then the entropy of fate bias is displayed. The default value is NULL. In this case, cluster identities are highlighted in the plot.
n	optional character string. This argument corresponds to a title for 2-dimensional plots. Default value is NULL. If not provided, and g is given, then n will equal g or g[1], respectively, if g is a vector of gene IDs.
prc	logical. If TRUE, then a principal curve is computed and returned. Default is FALSE.
logsc	logical. If TRUE, then gene expression of fate bias probabilities are plotted on a log2 scale. Default value is FALSE.
k	integer number for the dimension to be used. This dimension has to be present in dr. Only k=2 is allowed starting from version 0.1.9.
m	name of the dimensional reduction algorithms to be used for the principal curve computation. One of cmd, dm, tsne, umap. Default value is cmd. Has to be a component of dr, i.e. previously computed by compdr.
kr	integer vector. If k>3 then kr indicates the dimensions to be plotted (either two or three of all possible dimensions). Default value is NULL. In this case, kr is given by 1:min(k,3).
col	optional vector of valid color names for all clusters in y ordered by increasing cluster number. Default value is NULL.
fb	fateBias object returned by the function fateBias. If fb is provided, then a principal curve is computed and shown in the plot. Default value is NULL. The curve is only displayed if g equal NULL.
trthr	real value representing the threshold of the fraction of random forest votes required for the inclusion of a given cell for the computation of the principal curve. If NULL then only cells with a significant bias >1 are included for each trajectory. The bias is computed as the ratio of the number of votes for a trajectory and the number of votes for the trajectory with the second largest number of votes. By this means only the trajectory with the largest number of votes will receive a bias >1. The siginifcance is computed based on counting statistics on the difference in the number of votes. A significant bias requires a p-value < 0.05. Default value is NULL.
start	integer number representing a specified starting cluster number for all trajectories, i. e. a common progenitor cluster. The argument is optional. Default value is NULL.
tp	Transparency of points in the plot to allow better visibility of the principal curves. Default value is 1, i. e. non-transparent.
seed	integer number. Random seed for determining colour scheme. Default is 12345.
...	additional arguments to be passed to the low level function principal_curve.

Value

If fb is provided as input argument and prc equals TRUE then the output corresponds to the output of prcurve. Otherwise, only ouput is generated is g equals E. In this case a vector of fate bias entropies for all cells is given.

Examples

```
x <- intestine$x
y <- intestine$y
# v contains all genes (no feature selection like in x)
v <- intestine$v
fcol <- intestine$fcol
tar <- c(6,9,13)
fb <- fateBias(x,y,tar,z=NULL,minnr=5,minnrh=10,nbfactor=5,use.dist=FALSE,seed=NULL,nbtree=NULL)
dr <- compdr(x,z=NULL,m="cmd",k=2,tsne.perplexity=30)

# plot principal curves
pr <- plotFateMap(y,dr,k=2,prc=TRUE,m="cmd",col=fcol,fb=fb,trthr=0.25,start=NULL,tp=.5)

# plot expression of gene Apoa1__chr9
plotFateMap(y,dr,x=v,g="Apoa1__chr9",prc=FALSE,k=2,m="cmd",col=intestine$fcol)
```

---

plotheatmap *Heatmap of expression profiles*

Description

This function allows plotting of normalized or z-score transformed pseudo-temporal expression profiles and permits highlighting of partitioning along the x-axis and the y-axis

Usage

```
plotheatmap(
  x,
  xpart = NULL,
  xcol = NULL,
  xlab = TRUE,
  xgrid = FALSE,
  ypart = NULL,
  ycol = NULL,
  ylab = TRUE,
  ygrid = FALSE,
  cex = 1
)
```

Arguments

x	data frame with input data to show. Columns will be displayed on the x-axis and rows on the y-axis in the order given in x. For example, columns can correspond to cells in pseudo-temporal order and rows contain gene expression, i. e. rows can represent pseudo-temporal gene expression profiles.
xpart	optional vector with integer values indicating partitioning of the data points along the x-axis. For instance, xpart can be a cluster assignment of cell IDs. The order of the components has to be the same as for the columns in x. Default value is NULL.
xcol	optional vector with valid color names. The number of components has to be equal to the number of different values on xpart. If provided, these colors are used to highlight partitioning along the x-axis based on xpart. Default value is NULL.
xlab	logical value. If TRUE then the average position is indicated for each partition value along the x-axis. Default value is TRUE.
xgrid	logical value. If TRUE then the partitioning along the x-axis is indicated by vertical lines representing the boundaries of all positions with a given value in xpart.
ypart	optional vector with integer values indicating partitioning of the data points along the y-axis. For instance, ypart can be the assignment of gene IDs to nodes of a sel-organizing map. The order of the components has to be the same as for the rows in x. Default value is NULL.
ycol	optional vector with valid color names. The number of components has to be equal to the number of different values on ypart. If provided, these colors are used to highlight partitioning along the y-axis based on ypart. Default value is NULL.
ylab	logical value. If TRUE then the average position is indicated for each partition value along the y-axis. Default value is TRUE.
ygrid	logical value. If TRUE then the partitioning along the y-axis is indicated by horizontal lines representing the boundaries of all positions with a given value in ypart.
cex	positive real number. Size of axis labels. Default is 1.

Value

None

Examples

```
x <- intestine$x
y <- intestine$y
v <- intestine$v
fcol <- intestine$col

tar <- c(6,9,13)
fb <- fateBias(x,y,tar,z=NULL,minnr=5,minnrh=10,nbfactor=5,use.dist=FALSE,seed=NULL,nbtree=NULL)
```

```
dr <- compdr(x,z=NULL,m="cmd",k=2,tsne.perplexity=30)
pr <- prcurve(y,fb,dr,k=2,m="cmd",trthr=0.4,start=NULL)
n <- pr$trc[["t6"]]
fs  <- filterset(v,n,minexpr=2,minnumber=1)
s1d <- getsom(fs,nb=1000,alpha=.5)
ps <- procsom(s1d,corthr=.85,minsom=3)
plotheatmap(ps$all.e,xpart=y[n],xcol=fcol,ypart=ps$nodes,xgrid=FALSE,ygrid=TRUE,xlab=FALSE)
```

---

prcurve *Computation of a principal curve for a given dimensional reduction representation*

Description

This function computes a principal curve for a given dimensional reduction representation which is specified by component names of an object returned by `compdr` using the **princurve** package.

Usage

```
prcurve(y, fb, dr, k = 2, m = "cmd", trthr = NULL, start = NULL, ...)
```

Arguments

`y`	clustering partition. A vector with an integer cluster number for each cell. The order of the cells has to be the same as for the columns of x.
`fb`	fateBias object returned by the function `fateBias`.
`dr`	list of dimensional reduction representations returned by the function `compdr`.
`k`	integer number for the dimension to be used. This dimension has to be present in `dr`. Default value is 2.
`m`	name of the dimensional reduction algorithms to be used for the principal curve computation. One of `cmd`, `dm`, `tsne`, `umap`. Default value is `cmd`. Has to be a component of `dr`, i.e. previously computed by `compdr`.
`trthr`	real value representing the threshold of the fraction of random forest votes required for the inclusion of a given cell for the computation of the principal curve. If `NULL` then only cells with a significant bias >1 are included for each trajectory. The bias is computed as the ratio of the number of votes for a trajectory and the number of votes for the trajectory with the second largest number of votes. By this means only the trajectory with the largest number of votes will receive a bias >1. The siginifcance is computed based on counting statistics on the difference in the number of votes. A significant bias requires a p-value < 0.05. Default value is `NULL`.
`start`	integer number representing a specified starting cluster number for all trajectories, i. e. a common progenitor cluster. The argument is optional. Default value is `NULL`.
`...`	additional arguments to be passed to the low level function `principal_curve`.

Details

The function computes a principal curve for each differentiation trajectory by considering only cells that are assigned to the trajectory with a significant fate bias >1 or at least `trthr` of the random forest votes, respectively.

For simulateneous computation and plotting of the principal curve, see function `plotFateMap`.

Value

A list of the following two components:

`pr`	A list of principal curve objects produced by the `principal_curve` function from the **princurve** package. Each component corresponds to one differentiation trajectory giving rise to one of the target clusters from the `fb` object.
`trc`	A list of ordered cell IDs for each trajectory in `pr`.

Examples

```
x <- intestine$x
y <- intestine$y
tar <- c(6,9,13)
fb <- fateBias(x,y,tar,z=NULL,minnr=5,minnrh=10,nbfactor=5,use.dist=FALSE,seed=NULL,nbtree=NULL)
dr <- compdr(x,z=NULL,m="cmd",k=2,tsne.perplexity=30)
pr <- prcurve(y,fb,dr,k=2,m="cmd",trthr=0.25,start=NULL)
```

---

procsom *Processing of self-organizing maps for pseudo-temporal expression profiles*

Description

This function processes the self-organizing maps produced by the function `getsom`.

Usage

```
procsom(s1d, corthr = 0.85, minsom = 3)
```

Arguments

`s1d`	output of function `getsom`
`corthr`	correlation threshold, i. e. a real number between 0 and 1. The z-score of the average normalized pseudo-temporal expression profiles within each node of the self-organizing map is computed, and the correlation of these z-scores between neighbouring nodes is computed. If the correlation is greater than `corthr`, neighbouring nodes are merged. Default value is 0.85.
`minsom`	positive integer number. Nodes of the self-organizing map with less than `minsom` transcripts are discarded. Default value is 3.

Value

A list of the following seven components:

k	vector of Pearson's correlation coefficient between node i and node i+1 of the populated nodes of the self-organizing map.
nodes	vector with assignment of genes to nodes of the final self-organizing map (after merging). Components are node numbers and component names are gene IDs.
nodes.e	data frame with average normalized pseudo-temporal expression profile for each node, ordered by node number.
nodes.z	data frame with z-score transformed average normalized pseudo-temporal expression profile for each node, ordered by node number.
all.e	data frame with normalized pseudo-temporal expression profile for all genes in the self-organizing map, ordered by node number.
all.z	data frame with z-score transformed normalized pseudo-temporal expression profile for all genes in the self-organizing map, ordered by node number.
all.b	data frame with binarized pseudo-temporal expression profile for all genes in the self-organizing map, ordered by node number. Expression is 1 in cells with z-score > 1 and -1 in cells with z-score < -1, and 0 otherwise.

Examples

```
x <- intestine$x
y <- intestine$y
v <- intestine$v

tar <- c(6,9,13)
fb <- fateBias(x,y,tar,z=NULL,minnr=5,minnrh=10,nbfactor=5,use.dist=FALSE,seed=NULL,nbtree=NULL)
dr <- compdr(x,z=NULL,m="cmd",k=2,tsne.perplexity=30)
pr <- prcurve(y,fb,dr,k=2,m="cmd",trthr=0.4,start=NULL)
n <- pr$trc[["t6"]]
fs  <- filterset(v,n,minexpr=2,minnumber=1)
s1d <- getsom(fs,nb=1000,alpha=.5)
ps <- procsom(s1d,corthr=.85,minsom=3)
```

---

reclassify *Reclassification of cells*

Description

This function attempts to reassign additional cells in the dataset to one of the target clusters.

Usage

```
reclassify(
  x,
  y,
  tar,
  z = NULL,
  clthr = 0.75,
  nbfactor = 5,
  use.dist = FALSE,
  seed = NULL,
  nbtree = NULL,
  q = 0.9,
  ...
)
```

Arguments

x	expression data frame with genes as rows and cells as columns. Gene IDs should be given as row names and cell IDs should be given as column names. This can be a reduced expression table only including the features (genes) to be used in the analysis.
y	clustering partition. A vector with an integer cluster number for each cell. The order of the cells has to be the same as for the columns of x.
tar	vector of integers representing target cluster numbers. Each element of tar corresponds to a cluster of cells committed towards a particular mature state. One cluster per different cell lineage has to be given and is used as a starting point for learning the differentiation trajectory.
z	Matrix containing cell-to-cell distances to be used in the fate bias computation. Default is NULL. In this case, a correlation-based distance is computed from x by 1 - cor(x)
clthr	real number between zero and one. This is the threshold for the fraction of random forest votes required to assign a cell not contained within the target clusters to one of these clusters. The value of this parameter should be sufficiently high to only reclassify cells with a high-confidence assignment. Default value is 0.9.
nbfactor	positive integer number. Determines the number of trees grown for each random forest. The number of trees is given by the number of columns of th training set multiplied by nbfactor. Default value is 5.
use.dist	logical value. If TRUE then the distance matrix is used as feature matrix (i. e. z if not equal to NULL and 1-cor(x) otherwise). If FALSE, gene expression values in x are used. Default is FALSE.
seed	integer seed for initialization. If equal to NULL then each run will yield slightly different results due to the radomness of the random forest algorithm. Default is NULL
nbtree	integer value. If given, it specifies the number of trees for each random forest explicitely. Default is NULL.

q	real value between zero and one. This number specifies a threshold used for feature selection based on importance sampling. A reduced expression table is generated containing only features with an importance larger than the q-quantile for at least one of the classes (i. e. target clusters). Default value is 0.75.
...	additional arguments to be passed to the low level function randomForest.

Details

The function uses random forest based supervised learning to assign cells not contained in the target clusters to one of these clusters. All cells not within any of the target clusters which receive a fraction of votes larger than clthr for one of the target clusters will be reassigned to this cluster. Since this function is developed to reclassify cells only if they can be assigned with high confidence, a high value of clthr (e. g. > 0.75) should be applied.

Value

A list with the following three components:

part	A vector with the revised cluster assignment for each cell in the same order as in the input argument y.
rf	The random forest object generated for the reclassification, with enabled importance sampling (see **randomForest**).
xf	A filtered expression table with features extracted based on the important samples, only features with an importance larger than the q-quantile are for at least one of the classes are retained.

Examples

```
x <- intestine$x
y <- intestine$y
tar <- c(6,9,13)
rc <- reclassify(x,y,tar,z=NULL,nbfactor=5,use.dist=FALSE,seed=NULL,nbtree=NULL,q=.9)
```

Index