Package ‘singleCellHaystack’

Type Package

Title A Universal Differential Expression Prediction Tool for Single-Cell and Spatial Genomics Data

Version 1.0.0

Description One key exploratory analysis step in single-cell genomics data analysis is the prediction of features with different activity levels. For example, we want to predict differentially expressed genes (DEGs) in single-cell RNA-seq data, spatial DEGs in spatial transcriptomics data, or differentially accessible regions (DARs) in single-cell ATAC-seq data. 'singleCellHaystack' predicts differentially active features in single cell omics datasets without relying on the clustering of cells into arbitrary clusters. 'singleCellHaystack' uses Kullback-Leibler divergence to find features (e.g., genes, genomic regions, etc) that are active in subsets of cells that are non-randomly positioned inside an input space (such as 1D trajectories, 2D tissue sections, multi-dimensional embeddings, etc). For the theoretical background of 'singleCellHaystack' we refer to our original paper Vandenbon and Diez (Nature Communications, 2020) <doi:10.1038/s41467-020-17900-3> and our update Vandenbon and Diez (bioRxiv, 2022) <doi:10.1101/2022.11.13.516355>.

Imports methods, Matrix, splines, ggplot2, reshape2

Suggests knitr, rmarkdown, testthat, SummarizedExperiment, SingleCellExperiment, SeuratObject, cowplot, wrswoR, sparseMatrixStats, ComplexHeatmap, patchwork

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.2

VignetteBuilder knitr

NeedsCompilation no

Author Alexis Vandenbon [aut, cre] (<https://orcid.org/0000-0003-2180-5732>), Diego Diez [aut] (<https://orcid.org/0000-0002-2325-4893>)

Maintainer Alexis Vandenbon <alexis.vandenbon@gmail.com>

Repository CRAN

Date/Publication 2022-12-20 10:00:02 UTC
dat.expression

Single cell RNA-seq dataset.
**dat.tsne**

Single cell tSNE coordinates.

**default_bandwidth.nrd**

Default function given by function bandwidth.nrd in MASS. No changes were made to this function.

**Usage**

default_bandwidth.nrd(x)

**Arguments**

- `x` A numeric vector

**Value**

A suitable bandwidth.

**extract_row_dgRMatrix**

Returns a row of a sparse matrix of class dgRMatrix. Function made by Ben Bolker and Ott Toomet (see https://stackoverflow.com/questions/47997184/)

**Description**

Returns a row of a sparse matrix of class dgRMatrix. Function made by Ben Bolker and Ott Toomet (see https://stackoverflow.com/questions/47997184/)

**Usage**

extract_row_dgRMatrix(m, i = 1)

**Arguments**

- `m` A sparse matrix of class dgRMatrix
- `i` the index of the row to return
### extract_row_lgRMatrix

*Description*

Returns a row of a sparse matrix of class `lgRMatrix`. Function made by Ben Bolker and Ott Toomet (see https://stackoverflow.com/questions/47997184/)

*Usage*

```r
extract_row_lgRMatrix(m, i = 1)
```

*Arguments*

- `m`: a sparse matrix of class `lgRMatrix`
- `i`: the index of the row to return

*Value*

A row (logical vector) of the sparse matrix

---

### get_density

*Function to get the density of points with value TRUE in the (x,y) plot*

*Description*

Function to get the density of points with value TRUE in the (x,y) plot

*Usage*

```r
get_density(
  x,
  y,
  detection,
  rows.subset = 1:nrow(detection),
  high.resolution = FALSE
)
```
**get_dist_two_sets**

**Arguments**

- **x**
  - x-axis coordinates of cells in a 2D representation (e.g. resulting from PCA or t-SNE)

- **y**
  - y-axis coordinates of cells in a 2D representation

- **detection**
  - A logical matrix or dgRMatrix showing which genes (rows) are detected in which cells (columns)

- **rows.subset**
  - Indices of the rows of 'detection' for which to get the densities. Default: all.

- **high.resolution**
  - Logical: should high resolution be used? Default is FALSE.

**Value**

A 3-dimensional array (dim 1: genes/rows of expression, dim 2 and 3: x and y grid points) with density data

---

**get_dist_two_sets** *Calculate the pairwise Euclidean distances between the rows of 2 matrices.*

---

**Description**

Calculate the pairwise Euclidean distances between the rows of 2 matrices.

**Usage**

```
get_dist_two_sets(set1, set2)
```

**Arguments**

- **set1**
  - A numerical matrix.

- **set2**
  - A numerical matrix.

**Value**

A matrix of pairwise distances between the rows of 2 matrices.
get_D_KL

Calculates the Kullback-Leibler divergence between distributions.

Description

Calculates the Kullback-Leibler divergence between distributions.

Usage

get_D_KL(classes, parameters, reference.prob, pseudo)

Arguments

- classes: A logical vector. Values are T is the gene is expressed in a cell, F is not.
- parameters: Parameters of the analysis, as set by function 'get_parameters_haystack'
- reference.prob: A reference distribution to calculate the divergence against.
- pseudo: A pseudocount, used to avoid log(0) problems.

Value

A numerical value, the Kullback-Leibler divergence

g get_D_KL_continuous_highD

Calculates the Kullback-Leibler divergence between distributions for the high-dimensional continuous version of haystack.

Description

Calculates the Kullback-Leibler divergence between distributions for the high-dimensional continuous version of haystack.

Usage

get_D_KL_continuous_highD(
    weights,
    density.contributions,
    reference.prob,
    pseudo = 0
)
**get_D_KL_highD**

**Arguments**

- **weights**: A numerical vector with expression values of a gene.
- **density.contributions**: A matrix of density contributions of each cell (rows) to each center point (columns).
- **reference.prob**: A reference distribution to calculate the divergence against.
- **pseudo**: A pseudocount, used to avoid log(0) problems.

**Value**

A numerical value, the Kullback-Leibler divergence

---

**get_D_KL_highD**

*Calculates the Kullback-Leibler divergence between distributions for the high-dimensional version of haystack().*

---

**Description**

Calculates the Kullback-Leibler divergence between distributions for the high-dimensional version of haystack().

**Usage**

```
get_D_KL_highD(classes, density.contributions, reference.prob, pseudo = 0)
```

**Arguments**

- **classes**: A logical vector. Values are T if the gene is expressed in a cell, F is not.
- **density.contributions**: A matrix of density contributions of each cell (rows) to each center point (columns).
- **reference.prob**: A reference distribution to calculate the divergence against.
- **pseudo**: A pseudocount, used to avoid log(0) problems.

**Value**

A numerical value, the Kullback-Leibler divergence
get_euclidean_distance

*Calculate the Euclidean distance between x and y.*

**Description**

Calculate the Euclidean distance between x and y.

**Usage**

```
get_euclidean_distance(x, y)
```

**Arguments**

- **x**: A numerical vector.
- **y**: A numerical vector.

**Value**

A numerical value, the Euclidean distance.

---

get_grid_points

*A function to decide grid points in a higher-dimensional space*

**Description**

A function to decide grid points in a higher-dimensional space.

**Usage**

```
get_grid_points(input, method = "centroid", grid.points = 100)
```

**Arguments**

- **input**: A numerical matrix with higher-dimensional coordinates (columns) of points (rows).
- **method**: The method to decide grid points. Should be "centroid" (default) or "seeding".
- **grid.points**: The number of grid points to return. Default is 100.

**Value**

Coordinates of grid points in the higher-dimensional space.
get_log_p_D_KL

Estimates the significance of the observed Kullback-Leibler divergence by comparing to randomizations.

**Description**

Estimates the significance of the observed Kullback-Leibler divergence by comparing to randomizations.

**Usage**

```r
get_log_p_D_KL(T.counts, D_KL.observed, D_KL.randomized, output.dir = NULL)
```

**Arguments**

- `T.counts`: The number of cells in which a gene is detected.
- `D_KL.observed`: A vector of observed Kullback-Leibler divergences.
- `D_KL.randomized`: A matrix of Kullback-Leibler divergences of randomized datasets.
- `output.dir`: Optional parameter. Default is NULL. If not NULL, some files will be written to this directory.

**Value**

A vector of log10 p values, not corrected for multiple testing using the Bonferroni correction.

get_log_p_D_KL_continuous

Estimates the significance of the observed Kullback-Leibler divergence by comparing to randomizations for the continuous version of haystack.

**Description**

Estimates the significance of the observed Kullback-Leibler divergence by comparing to randomizations for the continuous version of haystack.

**Usage**

```r
get_log_p_D_KL_continuous(
  D_KL.observed,
  D_KL.randomized,
  all.coeffVar,
  train.coeffVar,
  output.dir = NULL,
  spline.method = "ns"
)
```
get_parameters_haystack

Function that decides most of the parameters that will be used during the "Haystack" analysis.

**Arguments**

- **D_KL.observed**: A vector of observed Kullback-Leibler divergences.
- **D_KL.randomized**: A matrix of Kullback-Leibler divergences of randomized datasets.
- **all.coeffVar**: Coefficients of variation of all genes. Used for fitting the Kullback-Leibler divergences.
- **train.coeffVar**: Coefficients of variation of genes that will be used for fitting the Kullback-Leibler divergences.
- **output.dir**: Optional parameter. Default is NULL. If not NULL, some files will be written to this directory.
- **spline.method**: Method to use for fitting splines "ns" (default): natural splines, "bs": B-splines.

**Value**

A vector of log10 p values, not corrected for multiple testing using the Bonferroni correction.

---

**Usage**

```r
get_parameters_haystack(x, y, high.resolution = FALSE)
```

**Arguments**

- **x**: x-axis coordinates of cells in a 2D representation (e.g. resulting from PCA or t-SNE)
- **y**: y-axis coordinates of cells in a 2D representation
- **high.resolution**: Logical: should high resolution be used? Default is FALSE.

**Value**

A list containing various parameters to use in the analysis.
get_reference  Get reference distribution

Description
Get reference distribution

Usage
get_reference(param, use.advanced.sampling = NULL)

Arguments
param Parameters of the analysis, as set by function 'get_parameters_haystack'
use.advanced.sampling    If NULL naive sampling is used. If a vector is given (of length = no. of cells) sampling is done according to the values in the vector.

Value
A list with two components, Q for the reference distribution and pseudo.

haystack The main Haystack function

Description
The main Haystack function

Usage
haystack(x, ...)

## S3 method for class 'matrix'
haystack(
  x,
  expression,
  weights.advanced.Q = NULL,
  dir.randomization = NULL,
  scale = TRUE,
  grid.points = 100,
  grid.method = "centroid",
  ...
)
## S3 method for class 'data.frame'
haystack(
  x,
  expression,
  weights.advanced.Q = NULL,
  dir.randomization = NULL,
  scale = TRUE,
  grid.points = 100,
  grid.method = "centroid",
  ...
)

## S3 method for class 'Seurat'
haystack(
  x,
  coord,
  assay = "RNA",
  slot = "data",
  dims = NULL,
  cutoff = 1,
  method = NULL,
  weights.advanced.Q = NULL,
  ...
)

## S3 method for class 'SingleCellExperiment'
haystack(
  x,
  assay = "counts",
  coord = "TSNE",
  dims = NULL,
  cutoff = 1,
  method = NULL,
  weights.advanced.Q = NULL,
  ...
)

### Arguments

- **x**
  - a matrix or other object from which coordinates of cells can be extracted.
- **...**
  - further parameters passed down to methods.
- **expression**
  - a matrix with expression data of genes (rows) in cells (columns)
- **weights.advanced.Q**
  - If NULL naive sampling is used. If a vector is given (of length = no. of cells) sampling is done according to the values in the vector.
- **dir.randomization**
  - If NULL, no output is made about the random sampling step. If not NULL, files related to the randomizations are printed to this directory.
scale Logical (default=TRUE) indicating whether input coordinates in x should be
scaled to mean 0 and standard deviation 1.
grid.points An integer specifying the number of centers (gridpoints) to be used for estimating
the density distributions of cells. Default is set to 100.
grid.method The method to decide grid points for estimating the density in the high-dimensional
space. Should be "centroid" (default) or "seeding".
coord name of coordinates slot for specific methods.
assay name of assay data for Seurat method.
slot name of slot for assay data for Seurat method.
dims dimensions from coord to use. By default, all.
cutoff cutoff for detection.
method choose between highD (default) and 2D haystack.

Value
An object of class "haystack"

haystack_2D The main Haystack function, for 2-dimensional spaces.

Description
The main Haystack function, for 2-dimensional spaces.

Usage
haystack_2D(
  x,
  y,
  detection,
  use.advanced.sampling = NULL,
  dir.randomization = NULL
)

Arguments
x x-axis coordinates of cells in a 2D representation (e.g. resulting from PCA or
t-SNE)
y y-axis coordinates of cells in a 2D representation
detection A logical matrix showing which genes (rows) are detected in which cells (columns)
use.advanced.sampling If NULL, naive sampling is used. If a vector is given (of length = no. of cells)
sampling is done according to the values in the vector.
dir.randomization If NULL, no output is made about the random sampling step. If not NULL, files
related to the randomizations are printed to this directory.
Value

An object of class "haystack"

Description

The main Haystack function, for higher-dimensional spaces and continuous expression levels.

Usage

haystack_continuous_highD(
  x,
  expression,
  grid.points = 100,
  weights.advanced.Q = NULL,
  dir.randomization = NULL,
  scale = TRUE,
  grid.method = "centroid",
  randomization.count = 100,
  n.genes.to.randomize = 100,
  selection.method.genes.to.randomize = "heavytails",
  grid.coord = NULL,
  spline.method = "ns"
)

Arguments

x Coordinates of cells in a 2D or higher-dimensional space. Rows represent cells, columns the dimensions of the space.

expression a matrix with expression data of genes (rows) in cells (columns)

grid.points An integer specifying the number of centers (grid points) to be used for estimating the density distributions of cells. Default is set to 100.

weights.advanced.Q (Default: NULL) Optional weights of cells for calculating a weighted distribution of expression.

dir.randomization If NULL, no output is made about the random sampling step. If not NULL, files related to the randomizations are printed to this directory.

scale Logical (default=TRUE) indicating whether input coordinates in x should be scaled to mean 0 and standard deviation 1.
The method to decide grid points for estimating the density in the high-dimensional space. Should be "centroid" (default) or "seeding".

Number of randomizations to use. Default: 100

Number of genes to use in randomizations. Default: 100

Method used to select genes for randomization.

matrix of grid coordinates.

Method to use for fitting splines "ns" (default): natural splines, "bs": B-splines.

An object of class "haystack", including the results of the analysis, and the coordinates of the grid points used to estimate densities.

# using the toy example of the singleCellHaystack package

# running haystack
res <- haystack(dat.tsne, dat.expression)
# list top 10 biased genes
show_result_haystack(res, n=10)
Arguments

x                      Coordinates of cells in a 2D or higher-dimensional space. Rows represent cells, columns the dimensions of the space.
detection              A logical matrix showing which genes (rows) are detected in which cells (columns)
grid.points            An integer specifying the number of centers (grid points) to be used for estimating the density distributions of cells. Default is set to 100.
use.advanced.sampling  If NULL naive sampling is used. If a vector is given (of length = no. of cells) sampling is done according to the values in the vector.
dir.randomization      If NULL, no output is made about the random sampling step. If not NULL, files related to the randomizations are printed to this directory.
scale                  Logical (default=TRUE) indicating whether input coordinates in x should be scaled to mean 0 and standard deviation 1.
grid.method            The method to decide grid points for estimating the density in the high-dimensional space. Should be "centroid" (default) or "seeding".

Value

An object of class "haystack", including the results of the analysis, and the coordinates of the grid points used to estimate densities.

Examples

# I need to add some examples.
# A toy example will be added too.

hclust_haystack

Function for hierarchical clustering of genes according to their expression distribution in 2D or multi-dimensional space

Description

Function for hierarchical clustering of genes according to their expression distribution in 2D or multi-dimensional space

Usage

hclust_haystack(
  x,
  expression,
  grid.coordinates,
  hclust.method = "ward.D",
  cor.method = "spearman",
  ...  
)
## S3 method for class 'matrix'

```r
hclust_haystack(
  x,
  expression,
  grid.coordinates,
  hclust.method = "ward.D",
  cor.method = "spearman",
  ...
)
```

## S3 method for class 'data.frame'

```r
hclust_haystack(
  x,
  expression,
  grid.coordinates,
  hclust.method = "ward.D",
  cor.method = "spearman",
  ...
)
```

### Arguments

- **x**: a matrix or other object from which coordinates of cells can be extracted.
- **expression**: expression matrix.
- **grid.coordinates**: coordinates of the grid points.
- **hclust.method**: method used with hclust.
- **cor.method**: method used with cor.
- **...**: further parameters passed down to methods.

---

**hclust_haystack_highD**  
*Function for hierarchical clustering of genes according to their distribution in a higher-dimensional space.*

---

### Usage

```r
hclust_haystack_highD(
  x,
  detection,
```
genes,  
method = "ward.D",  
grid.coordinates = NULL,  
scale = TRUE
)

Arguments

x Coordinates of cells in a 2D or higher-dimensional space. Rows represent cells, columns the dimensions of the space.
detection A logical matrix showing which genes (rows) are detected in which cells (columns)
genes A set of genes (of the 'detection' data) which will be clustered.
method The method to use for hierarchical clustering. See '?hclust' for more information. Default: "ward.D".
grid.coordinates Coordinates of grid points in the same space as 'x', to be used to estimate densities for clustering.
scale whether to scale data.

Value

An object of class hclust, describing a hierarchical clustering tree.

Examples

# to be added

---

hclust_haystack_raw Function for hierarchical clustering of genes according to their distribution on a 2D plot.

Description

Function for hierarchical clustering of genes according to their distribution on a 2D plot.

Usage

hclust_haystack_raw(x, y, detection, genes, method = "ward.D")

Arguments

x x-axis coordinates of cells in a 2D representation (e.g. resulting from PCA or t-SNE)
y y-axis coordinates of cells in a 2D representation
detection A logical matrix showing which genes (rows) are detected in which cells (columns)
genes A set of genes (of the 'detection' data) which will be clustered.
method The method to use for hierarchical clustering. See '?hclust' for more information. Default: "ward.D".
**Value**
An object of class hclust, describing a hierarchical clustering tree.

Based on the MASS kde2d() function, but heavily simplified; it’s just tcrossprod() now.

**Description**
Based on the MASS kde2d() function, but heavily simplified; it’s just tcrossprod() now.

**Usage**

```
kde2d_faster(dens.x, dens.y)
```

**Arguments**

- `dens.x`: Contribution of all cells to densities of the x-axis grid points.
- `dens.y`: Contribution of all cells to densities of the y-axis grid points.

**kmeans_haystack**
Function for k-means clustering of genes according to their expression distribution in 2D or multi-dimensional space

**Description**
Function for k-means clustering of genes according to their expression distribution in 2D or multi-dimensional space

**Usage**

```
kmeans_haystack(x, expression, grid.coordinates, k, ...)
## S3 method for class 'matrix'
kmeans_haystack(x, expression, grid.coordinates, k, ...)
## S3 method for class 'data.frame'
kmeans_haystack(x, expression, grid.coordinates, k, ...)
```

**Arguments**

- `x`: a matrix or other object from which coordinates of cells can be extracted.
- `expression`: expression matrix.
- `grid.coordinates`: coordinates of the grid points.
- `k`: number of clusters.
- `...`: further parameters passed down to methods.
kmeans_haystack_highD  
Function for k-means clustering of genes according to their distribution in a higher-dimensional space.

Description

Function for k-means clustering of genes according to their distribution in a higher-dimensional space.

Usage

kmeans_haystack_highD(
  x,  
detection,
  genes,
  grid.coordinates = NULL,
  k,
  scale = TRUE,
  ...
)

Arguments

x           Coordinates of cells in a 2D or higher-dimensional space. Rows represent cells, columns the dimensions of the space.
detection   A logical matrix showing which genes (rows) are detected in which cells (columns)
genes       A set of genes (of the 'detection' data) which will be clustered.
grid.coordinates
  Coordinates of grid points in the same space as 'x', to be used to estimate densities for clustering.
k           The number of clusters to return.
scale       whether to scale data.
...          Additional parameters which will be passed on to the kmeans function.

Value

An object of class kmeans, describing a clustering into 'k' clusters

Examples

# to be added
Function for k-means clustering of genes according to their distribution on a 2D plot.

Usage

kmeans_haystack_raw(x, y, detection, genes, k, ...)

Arguments

- **x**: x-axis coordinates of cells in a 2D representation (e.g. resulting from PCA or t-SNE)
- **y**: y-axis coordinates of cells in a 2D representation
- **detection**: A logical matrix showing which genes (rows) are detected in which cells (columns)
- **genes**: A set of genes (of the 'detection' data) which will be clustered.
- **k**: The number of clusters to return.
- **...**: Additional parameters which will be passed on to the kmeans function.

Value

An object of class kmeans, describing a clustering into 'k' clusters

Visualizing the detection/expression of a gene in a 2D plot

Usage

plot_gene_haystack(x, ...)

## S3 method for class 'matrix'
plot_gene_haystack(x, dim1 = 1, dim2 = 2, ...)

## S3 method for class 'data.frame'
plot_gene_haystack(x, dim1 = 1, dim2 = 2, ...)

## S3 method for class 'SingleCellExperiment'
plot_gene_haystack_raw

Visualizing the detection/expression of a gene in a 2D plot

Description

Visualizing the detection/expression of a gene in a 2D plot

Usage

plot_gene_haystack_raw(
  x,
  y,
  gene,
  expression,
Arguments

x x-axis coordinates of cells in a 2D representation (e.g. resulting from PCA or t-SNE)
y y-axis coordinates of cells in a 2D representation
gene name of a gene that is present in the input expression data, or a numerical index
eexpression a logical/numerical matrix showing detection/expression of genes (rows) in cells (columns)
detection an optional logical matrix showing detection of genes (rows) in cells (columns). If left as NULL, the density distribution of the gene is not plotted.
high.resolution logical (default: FALSE). If set to TRUE, the density plot will be of a higher resolution
point.size numerical value to set size of points in plot. Default is 1.
order.by.signal If TRUE, cells with higher signal will be put on the foreground in the plot. Default is FALSE.

Value

A plot

Description

Visualizing the detection/expression of a set of genes in a 2D plot

Usage

plot_gene_set_haystack(x, ...)

## S3 method for class 'matrix'
plot_gene_set_haystack(x, dim1 = 1, dim2 = 2, ...)

## S3 method for class 'data.frame'
plot_gene_set_haystack(x, dim1 = 1, dim2 = 2, ...)
## S3 method for class 'SingleCellExperiment'
plot_gene_set_haystack(
  x,
  dim1 = 1,
  dim2 = 2,
  assay = "counts",
  coord = "TSNE",
  ...
)

## S3 method for class 'Seurat'
plot_gene_set_haystack(
  x,
  dim1 = 1,
  dim2 = 2,
  assay = "RNA",
  slot = "data",
  coord = "tsne",
  ...
)

### Arguments

**x**

a matrix or other object from which coordinates of cells can be extracted.

**...**

further parameters passed to plot_gene_haystack_raw().

**dim1**

column index or name of matrix for x-axis coordinates.

**dim2**

column index or name of matrix for y-axis coordinates.

**assay**

name of assay data for Seurat method.

**coord**

name of coordinates slot for specific methods.

**slot**

name of slot for assay data for Seurat method.

---

**plot_gene_set_haystack_raw**

Visualizing the detection/expression of a set of genes in a 2D plot

---

**Description**

Visualizing the detection/expression of a set of genes in a 2D plot

**Usage**

plot_gene_set_haystack_raw(
  x,
  y,
Args

x x-axis coordinates of cells in a 2D representation (e.g. resulting from PCA or t-SNE)
y y-axis coordinates of cells in a 2D representation
genes Gene names that are present in the input expression data, or a numerical index. If NA, all genes will be used.
detection a logical matrix showing detection of genes (rows) in cells (columns)

high.resolution logical (default: TRUE). If set to FALSE, the density plot will be of a lower resolution
point.size numerical value to set size of points in plot. Default is 1.
order.by.signal If TRUE, cells with higher signal will be put on the foreground in the plot. Default is FALSE.

Value

A plot

Description

plot_rand_fit

Usage

plot_rand_fit(x, type = c("mean", "sd"))

## S3 method for class 'haystack'
plot_rand_fit(x, type = c("mean", "sd"))

Arguments

x haystack object.
type whether to plot mean or sd.
**read_haystack**

*Function to read haystack results from file.*

---

**Description**

Function to read haystack results from file.

**Usage**

```r
read_haystack(file)
```

**Arguments**

- `file` A file containing 'haystack' results to read

**Value**

An object of class "haystack"

---

**show_result_haystack**

*show_result_haystack*

---

**Description**

Shows the results of the 'haystack' analysis in various ways, sorted by significance. Priority of params is genes > p.value.threshold > n.

**Usage**

```r
show_result_haystack(
    res.haystack,
    n = NULL,
    p.value.threshold = NULL,
    gene = NULL
)
```

```r
## S3 method for class 'haystack'
show_result_haystack(
    res.haystack,
    n = NULL,
    p.value.threshold = NULL,
    gene = NULL
)
```
Arguments

res.haystack A 'haystack' result object.
n If defined, the top "n" significant genes will be returned. Default: NA, which shows all results.
p.value.threshold If defined, genes passing this p-value threshold will be returned.
gene If defined, the results of this (these) gene(s) will be returned.

Details

The output is a data.frame with the following columns: * D_KL the calculated KL divergence. * log.p.vals log10 p.values calculated from randomization. * log.p.adj log10 p.values adjusted by Bonferroni correction.

Value

A data.frame with 'haystack' results sorted by log.p.vals.

Examples

# using the toy example of the singleCellHaystack package

# running haystack
res <- haystack(dat.tsne, dat.expression)

# below are variations for showing the results in a table
# 1. list top 10 biased genes
show_result_haystack(res.haystack = res, n = 10)
# 2. list genes with p value below a certain threshold
show_result_haystack(res.haystack = res, p.value.threshold = 1e-10)
# 3. list a set of specified genes
set <- c("gene_497", "gene_386", "gene_275")
show_result_haystack(res.haystack = res, gene = set)

write_haystack

Function to write haystack result data to file.

Description

Function to write haystack result data to file.

Usage

write_haystack(res.haystack, file)

Arguments

res.haystack A 'haystack' result variable
file A file to write to
Index

* data
  dat.expression, 2
dat.tsne, 3

dat.expression, 2
dat.tsne, 3
default_bandwidth.nrd, 3

extract_row_dgRMatrix, 3
extract_row_lgRMatrix, 4

get_D_KL, 6
get_D_KL_continuous_highD, 6
get_D_KL_highD, 7
get_density, 4
get_dist_two_sets, 5
get_euclidean_distance, 8
get_grid_points, 8
get_log_p_D_KL, 9
get_log_p_D_KL_continuous, 9
get_parameters_haystack, 10
get_reference, 11

haystack, 11
haystack_2D, 13
haystack_continuous_highD, 14
haystack_highD, 15
hclust_haystack, 16
hclust_haystack_highD, 17
hclust_haystack_raw, 18

kde2d_faster, 19
kmeans_haystack, 19
kmeans_haystack_highD, 20
kmeans_haystack_raw, 21

plot_gene_haystack, 21
plot_gene_haystack_raw, 22
plot_gene_set_haystack, 23
plot_gene_set_haystack_raw, 24
plot_rand_fit, 25

read_haystack, 26
show_result_haystack, 26
write_haystack, 27