Package ‘scFeatures’

May 13, 2024

Title scFeatures: Multi-view representations of single-cell and spatial data for disease outcome prediction

Version 1.4.0

Description scFeatures constructs multi-view representations of single-cell and spatial data. scFeatures is a tool that generates multi-view representations of single-cell and spatial data through the construction of a total of 17 feature types. These features can then be used for a variety of analyses using other software in Bioconductor.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

biocViews CellBasedAssays, SingleCell, Spatial, Software, Transcriptomics

Depends R (>= 4.2.0)


Suggests knitr, S4Vectors, survival, survminer, BiocStyle, ClassifyR, org.Hs.eg.db, clusterProfiler

VignetteBuilder knitr

URL https://sydneybiox.github.io/scFeatures/

https://github.com/SydneyBioX/scFeatures/

BugReports https://github.com/SydneyBioX/scFeatures/issues

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

git_branch RELEASE_3_19

git_last_commit b3d1905
**example_scrnaseq**

**Example of scRNA-seq data**

**Description**

This is a subsampled version of the melanoma patients dataset as used in our manuscript. The original dataset is available at: [https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE120575](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE120575).
get_num_cell_per_spot

Usage

data("example_scrnaseq")

Format

exmaple_scrnaseq:
A Seurat object with 3523 genes and 550 cells. Some of the key metadata columns are:
  celltype  cell type of the cell
  sample   patient ID of the cell
  condition whether the patient is a responder or non-responder

Source


Description

This function takes a list object containing spatial transcriptomics matrix as input and estimates the relative number of cells per spot in the data. The number of cells is estimated as the library size scaled to the range from 1 to 100. This value stored in the number_cells attribute.

Usage

get_num_cell_per_spot(alldata)

Arguments

  alldata A list object containing spatial transcriptomics

Value

  a vector with the relative number of cells in each spot.

Examples

  utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq@assays$RNA@data
data <- list(data = data)
number_of_cells <- get_num_cell_per_spot(data)
remove_mito_ribo Remove mitochondrial and ribosomal genes, and other highly correlated genes

Description

This function removes mitochondria and ribosomal genes and genes highly correlated with these genes, as mitochondria and ribosomal genes are typically not interesting to look at.

Usage

remove_mito_ribo(alldata)

Arguments

alldata A list object containing expression data

Value

The list object with the mitochondrial and ribosomal genes and other highly correlated genes removed

Examples

utils::data("example_scRNAseq", package = "scFeatures")
data <- example_scRNAseq
data <- list(data = data@assays$RNA$data)
data <- remove_mito_ribo(data)

run_association_study_report

Create an association study report in HTML format

Description

This function takes the feature matrix generated by scFeatures as input and creates an HTML report containing the results of the association study. The report is saved to the specified output folder.

Usage

run_association_study_report(scfeatures_result, output_folder)
Arguments

- **scfeatures_result**: a named list storing the scFeatures feature output. Note that the names of the list should be one or multiple of the following: proportion_raw, proportion_logit, proportion_ratio, gene_mean_celltype, gene_prop_celltype, gene_cor_celltype, pathway_gsva, pathway_mean, pathway_prop, CCI, gene_mean_aggregated, gene_cor_aggregated, and gene_prop_aggregated.

- **output_folder**: the path to the folder where the HTML report will be saved.

Value

- an HTML file, saved to the directory defined in the `output_folder` argument.

Examples

```r
## Not run:
output_folder <- tempdir()
utils::data("scfeatures_result", package = "scFeatures")
run_association_study_report(scfeatures_result, output_folder )

## End(Not run)
```

---

**run_CCI**

*Generate cell cell communication score*

**Description**

This function calculates the ligand receptor interaction score using SingleCellSignalR. The output features are in the form of celltype a \rightarrow celltype b \rightarrow ligand 1 \rightarrow receptor 2, which indicates the interaction between ligand 1 in celltype a and receptor 2 in celltype b. It supports scRNA-seq.

**Usage**

```r
run_CCI(data, type = "scrna", ncores = 1)
```

**Arguments**

- **data**: A list object containing data matrix and celltype and sample vector.
- **type**: input data type, either scrna, spatial_p, or spatial_t
- **ncores**: number of cores

**Value**

- a matrix of samples x features. The features are in the form of ligand 1 receptor 2 celltype a, ligand 1 receptor 2 celltype b ... etc, with the numbers representing cell-cell interaction probability.
run_celltype_interaction

Generate cell type interaction

Description
This function calculates the pairwise distance between cell types for a sample by using the coordinates and cell types of the cells. We find the nearest neighbours of each cell and the cell types of these neighbours. These are considered as spatial interaction pairs. The cell type composition of the spatial interaction pairs are used as features. The function supports spatial proteomics and spatial transcriptomics.

Usage
run_celltype_interaction(data, type = "spatial_p", ncores = 1)

Arguments
- data: A list object containing data matrix and celltype and sample vector.
- type: The type of dataset, either "scRNA", "spatial_t", or "spatial_p".
- ncores: Number of cores for parallel processing.

Value
A dataframe of samples x features. The features are in the form of protein 1 vs protein 2, protein 1 vs protein 3 ... etc, with the numbers representing the proportion of each interaction pairs in a given sample.

Examples
```R
utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[1:1000, 1:100]
celltype <- data$celltype
sample <- data$sample
data <- as.matrix(data@assays$RNA@data)
alldata <- scFeatures:::formatData(data = data, celltype = celltype, sample = sample)
feature_CCI <- run_CCI(alldata, type = "scRNA", ncores = 1)
```
spatialCoords <- list(x, y)
alldata <- scFeatures:::formatData(data = data, sample = sample, celltype = celltype, 
spatialCoords = spatialCoords)

feature_celltype_interaction <- run_celltype_interaction( 
  alldata, type = "spatial_p", ncores = 1
)

---

run_gene_cor

**Generate overall aggregated gene correlation**

**Description**

This function computes the correlation of gene expression across samples. The user can specify the genes of interest, or let the function use the top variable genes by default. The function supports scRNA-seq, spatial proteomics, and spatial transcriptomics.

**Usage**

```r
run_gene_cor(
  data, 
  type = "scrna", 
  genes = NULL, 
  num_top_gene = NULL, 
  ncores = 1
)
```

**Arguments**

- `data`: A list object containing data matrix and celltype and sample vector.
- `type`: The type of dataset, either "scrna", "spatial_t", or "spatial_p".
- `genes`: Default to NULL, in which case the top variable genes will be used. If provided by user, need to be in the format of a list containing the genes of interest, eg, `genes <- c("GZMA", "GZMK", "CCR7", "RPL38")`
- `num_top_gene`: Number of top variable genes to use when genes is not provided. Defaults to 5.
- `ncores`: Number of cores for parallel processing.

**Value**

A dataframe of samples x features. The features are in the form of gene 1, gene 2 ... etc, with the numbers representing the proportion that the gene is expressed across all cells.
**Examples**

```r
utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[1:100, 1:200]
celltype <- data$celltype
sample <- data$sample
data <- data@assays$RNA@data

alldata <- scFeatures:::formatData(data = data, celltype = celltype, sample = sample)
feature_gene_cor <- run_gene_cor(
    alldata, type = "scrna", num_top_gene = 5, ncores = 1)
```

---

**run_gene_cor_celltype  Generate cell type specific gene expression correlation**

**Description**

This function computes the correlation of expression of a set of genes for each cell type in the input data. The input data can be of three types: 'scrna', 'spatial_p' or 'spatial_t'. If the genes parameter is not provided by the user, the top variable genes will be selected based on the num_top_gene parameter (defaults to 100).

**Usage**

```r
run_gene_cor_celltype( data, type = "scrna", genes = NULL, num_top_gene = NULL, ncores = 1 )
```

**Arguments**

- **data**
  A list object containing data matrix and celltype and sample vector.

- **type**
  The type of dataset, either "scrna", "spatial_t", or "spatial_p".

- **genes**
  Optional dataframe with 2 columns: 'marker' and 'celltype'. The 'marker' column should contain the genes of interest (e.g. 'S100A11', 'CCL4'), and the 'celltype' column should contain the celltype that the gene expression is to be computed from (e.g. 'CD8', 'B cells'). If not provided, the top variable genes will be used based on the num_top_gene parameter.

- **num_top_gene**
  Number of top genes to use when genes is not provided. Defaults to 5.

- **ncores**
  Number of cores for parallel processing.
run_gene_mean

Value

A dataframe of samples x features. The features are in the form of gene 1 vs gene 2 cell type a, gene 1 vs gene 3 cell type b ... etc, with the numbers representing the correlation of the two given genes in the given cell type.

Examples

```r
utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[1:50, 1:20]
celltype <- data$celltype
sample <- data$sample
data <- data@assays$RNA$data

alldata <- scFeatures:::formatData(data = data, celltype = celltype, sample = sample )

feature_gene_cor_celltype <- run_gene_cor_celltype(  
alldata,  
type = "srna", num_top_gene = 5, ncores = 1  )
```

Description

This function computes the mean expression of genes across samples. The user can specify the genes of interest, or let the function use the top variable genes by default. The function supports scRNA-seq, spatial proteomics, and spatial transcriptomics.

Usage

```r
run_gene_mean(  
data,  
type = "srna",  
genesis = NULL,  
um_top_gene = NULL,  
ncores = 1  )
```

Arguments

data A list object containing data matrix and celltype and sample vector.
type The type of dataset, either "srna", "spatial_t", or "spatial_p".
genesis Default to NULL, in which case the top variable genes will be used. If provided by user, need to be in the format of a list containing the genes of interest, eg, genesis <- c("GZMA", "GZMK", "CCR7", "RPL38")
run_gene_mean_celltype

num_top_gene  Number of top variable genes to use when genes is not provided. Defaults to 1500.

t瑙es  Number of cores for parallel processing.

Value

A dataframe of samples x features. The features are in the form of gene 1, gene 2 ... etc, with the numbers representing averaged gene expression across all cells.

Examples

```r
utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq
celltype <- data$celltype
sample <- data$sample
data <- data@assays$RNA@data

alldata <- scfeatures:::formatData(data = data, celltype = celltype, sample = sample)
feature_gene_mean <- run_gene_mean(
alldata,
type = "scrna", num_top_gene = 150, ncores = 1
)
```

---

**run_gene_mean_celltype**

Generate cell type specific gene mean expression

---

**Description**

This function computes the mean expression of a set of genes for each cell type in the input data. The input data can be of three types: 'scrna', 'spatial_p' or 'spatial_t'. If the genes parameter is not provided by the user, the top variable genes will be selected based on the num_top_gene parameter (defaults to 100).

**Usage**

```r
run_gene_mean_celltype(
data,
type = "scrna",
genes = NULL,
um_top_gene = NULL,
ncores = 1
)
```
run_gene_prop

Arguments

- **data**: A list object containing data matrix and celltype and sample vector.
- **type**: The type of dataset, either "scRNA", "spatial_t", or "spatial_p".
- **genes**: Optional dataframe with 2 columns: 'marker' and 'celltype'. The 'marker' column should contain the genes of interest (e.g. 'S100A11', 'CCL4'), and the 'celltype' column should contain the celltype that the gene expression is to be computed from (e.g. 'CD8', 'B cells'). If not provided, the top variable genes will be used based on the num_top_gene parameter.
- **num_top_gene**: Number of top genes to use when genes is not provided. Defaults to 100.
- **ncores**: Number of cores for parallel processing.

Value

A dataframe of samples x features. The features are in the form of gene 1 celltype a, gene 2 celltype b ... etc, with the number representing average gene expression of the given gene across the cells of the the given celltype.

Examples

```r
utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[1:200, 1:200]
celltype <- data$celltype
sample <- data$sample
data <- data@assays$RNA@data

alldata <- scFeatures:::formatData(data = data, celltype = celltype, sample = sample )

feature_gene_mean_celltype <- run_gene_mean_celltype(
  alldata,
  type = "scRNA", num_top_gene = 100, ncores = 1
)
```

---

**run_gene_prop**  
*Generate overall aggregated gene proportion expression*

Description

This function computes the proportion of gene expression across samples. The user can specify the genes of interest, or let the function use the top variable genes by default. The function supports scRNA-seq, spatial proteomics, and spatial transcriptomics.
run_gene_prop_celltype

Usage

run_gene_prop(
    data,
    type = "scrna",
    genes = NULL,
    num_top_gene = NULL,
    ncores = 1
)

Arguments

data A list object containing data matrix and celltype and sample vector.

type The type of dataset, either "scrna", "spatial_t", or "spatial_p".

genes Default to NULL, in which case the top variable genes will be used. If provided by user, need to be in the format of a list containing the genes of interest, eg, genes <- c(GZMA", "GZMK", "CCR7", "RPL38")

num_top_gene Number of top variable genes to use when genes is not provided. Defaults to 1500.

ncores Number of cores for parallel processing.

Value

A dataframe of samples x features The features are in the form of gene 1 vs gene 2, gene 1 vs gene 3 ... etc, with the numbers representing correlation of gene expressions.

Examples

utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[1:50, 1:20]
celltype <- data$celltype
sample <- data$sample
data <- data@assays$RNA@data

alldata <- scFeatures:::formatData(data = data, celltype = celltype, sample = sample)
feature_gene_prop <- run_gene_prop(alldata, type = "scrna", num_top_gene = 10, ncores = 1)

run_gene_prop_celltype

Generate cell type specific gene proportion expression

Description

This function computes the proportion of expression of a set of genes for each cell type in the input data. The input data can be of three types: 'scrna', 'spatial_p' or 'spatial_t'. If the genes parameter is not provided by the user, the top variable genes will be selected based on the num_top_gene parameter (defaults to 100).
run_gene_prop_celltype

Usage

```r
run_gene_prop_celltype(
  data,
  type = "scrna",
  genes = NULL,
  num_top_gene = NULL,
  ncores = 1
)
```

Arguments

data A list object containing data matrix and celltype and sample vector.
type The type of dataset, either "scrna", "spatial_t", or "spatial_p".
genesis Optional dataframe with 2 columns: 'marker' and 'celltype'. The 'marker' column should contain the genes of interest (e.g. 'S100A11', 'CCL4'), and the 'celltype' column should contain the celltype that the gene expression is to be computed from (e.g. 'CD8', 'B cells'). If not provided, the top variable genes will be used based on the num_top_gene parameter.
num_top_gene Number of top genes to use when genes is not provided. Defaults to 100.
ncores Number of cores for parallel processing.

Value

A dataframe of samples x features. The features are in the form of gene 1 celltype a, gene 2 celltype b ... etc, with the number representing proportion of gene expression of the given gene across the cells of the the given celltype.

Examples

```r
utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[,1:20]
celltype <- data$celltype
sample <- data$sample
data <- data@assays$RNA@data

alldata <- scFeatures:::formatData(data = data, celltype = celltype, sample = sample)

feature_gene_prop_celltype <- run_gene_prop_celltype(
  alldata,
  type = "scrna", num_top_gene = 100, ncores = 1
)
```
run_L_function

**Description**

This function calculates L-statistics to measure spatial autocorrelation. L value greater than zero indicates spatial attraction of the pair of proteins whereas L value less than zero indicates spatial repulsion. The function supports spatial proteomics and spatial transcriptomics.

**Usage**

```r
run_L_function(data, type = "spatial_p", ncores = 1)
```

**Arguments**

- `data` A list object containing data matrix and celltype and sample vector.
- `type` The type of dataset, either "scrna", "spatial_t", or "spatial_p".
- `ncores` Number of cores for parallel processing.

**Value**

A dataframe of samples x features. The features are in the form of protein 1 vs protein 2, protein 1 vs protein 3 ... etc, with the numbers representing the L values.

**Examples**

```r
utils::data("example_scrnaseq", package = "scFeatures")
celltype <- example_scrnaseq$celltype
data <- example_scrnaseq$assays$RNA$data
sample <- sample(c("patient1", "patient2", "patient3"), ncol(data), replace = TRUE)
x <- sample(1:100, ncol(data), replace = TRUE)
y <- sample(1:100, ncol(data), replace = TRUE)
spatialCoords <- list(x, y)
alldata <- scFeatures:::formatData(data = data, sample = sample, celltype = celltype,
spatialCoords = spatialCoords)

feature_L_function <- run_L_function(alldata, type = "spatial_p", ncores = 1)
```
**Description**

This function calculates Moran’s I to measure spatial autocorrelation, which an indication of how strongly the feature (i.e., genes/proteins) expression values in a sample cluster or disperse. A value closer to 1 indicates clustering of similar values and a value closer to -1 indicates clustering of dissimilar values. A value of 0 indicates no particular clustering structure, i.e., the values are spatially distributed randomly. The function supports spatial proteomics and spatial transcriptomics.

**Usage**

```r
run_Morans_I(data, type = "spatial_p", ncores = 1)
```

**Arguments**

- `data`: A list object containing `data` matrix and `celltype` and `sample` vector.
- `type`: The type of dataset, either "scrna", "spatial_t", or "spatial_p".
- `ncores`: Number of cores for parallel processing.

**Value**

A dataframe of samples x features. The features are in the form of protein 1, protein 2 ... etc, with the numbers representing Moran’s value.

**Examples**

```r
utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[1:50, 1:20]
celltype <- data$celltype
data <- data@assays$RNA$data
sample <- sample( c("patient1", "patient2", "patient3"), ncol(data) , replace= TRUE )
x <- sample(1:100, ncol(data) , replace = TRUE)
y <- sample(1:100, ncol(data) , replace = TRUE)
spatialCoords <- list( x , y)
alldata <- scFeatures:::formatData(data = data, sample = sample, celltype = celltype, spatialCoords = spatialCoords )

feature_Morans_I <- run_Morans_I(alldata, type = "spatial_p", ncores = 1)
```
run_nn_correlation

Generate nearest neighbour correlation

Description

This function calculates the nearest neighbour correlation for each feature (e.g., proteins) in each sample. This is calculated by taking the correlation between each cell and its nearest neighbours cell for a particular feature. This function supports spatial proteomics, and spatial transcriptomics.

Usage

run_nn_correlation(data, type = "spatial_p", num_top_gene = NULL, ncores = 1)

Arguments

data A list object containing data matrix and celltype and sample vector.
type The type of dataset, either "scRNA", "spatial_t", or "spatial_p".
num_top_gene Number of top variable genes to use when genes is not provided. Defaults to 1500.
ncores Number of cores for parallel processing.

Value

a dataframe of samples x features The features are in the form of protein 1, protein 2 ... etc, with the numbers representing Pearson’s correlation.

Examples

utils::data("example_scRNaseq", package = "scFeatures")
data <- example_scRNaseq[1:50, 1:20]
celltype <- data$celltype
data <- data@assays$RNA@data
sample <- sample( c("patient1", "patient2", "patient3"), ncol(data), replace = TRUE )
x <- sample(1:100, ncol(data), replace = TRUE)
y <- sample(1:100, ncol(data), replace = TRUE)
spatialCoords <- list(x, y)
alldata <- scFeatures:::formatData(data = data, sample = sample, celltype = celltype, spatialCoords = spatialCoords)
feature_nn_correlation <- run_nn_correlation(
  alldata, type = "spatial_p", ncores = 1
)
run_pathway_gsva

Generate pathway score using gene set enrichment analysis

Description

This function calculates pathway scores for a given input dataset and gene set using gene set enrichment analysis (GSVA). It supports scRNA-seq, spatial proteomics and spatial transcriptomics. It currently supports two pathway analysis methods: ssgsea and aucell. By default, it uses the 50 hallmark gene sets from msigdb. Alternatively, users can provide their own gene sets of interest in a list format.

Usage

run_pathway_gsva(
  data,
  method = "ssgsea",
  geneset = NULL,
  species = "Homo sapiens",
  type = "scrna",
  subsample = TRUE,
  ncores = 1
)

Arguments

data              A list object containing data matrix and celltype and sample vector.
method            Type of pathway analysis method, currently support ssgsea and aucell
geneset           By default (when the geneset argument is not specified), we use the 50 hallmark gene set from msigdb. The users can also provide their geneset of interest in a list format, with each list entry containing a vector of the names of genes in a gene set. eg, geneset <- list("pathway_a" = c("CAPN1", ...), "pathway_b" = c("PEX6"))
species           Whether the species is "Homo sapiens" or "Mus musculus". Default is "Homo sapiens".
type              The type of dataset, either "scrna", "spatial_t", or "spatial_p".
subsample         Whether to subsample, either TRUE or FALSE. For larger datasets (eg, over 30,000 cells), the subsample function can be used to increase speed.
ncores            Number of cores for parallel processing.

Value

A dataframe of samples x features. The features are in the form of pathway 1 celltype a, pathway 2 celltype b ... etc, with the number representing the gene set enrichment score of a given pathway in cells from a given celltype.
Examples

```r
utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[, 1:20]
celltype <- data$celltype
sample <- data$sample
data <- data@assays$RNA@data

alldata <- scFeatures:::formatData(data = data, celltype = celltype, sample = sample)

feature_pathway_gsva <- run_pathway_gsva(
alldata,
genesiset = NULL, species = "Homo sapiens",
type = "scrna", subsample = FALSE, ncores = 1)
```

run_pathway_mean  Generate pathway score using expression level

Description

This function calculates pathway scores for a given dataset and gene set using gene expression levels. It supports scRNA-seq, spatial transcriptomics and spatial proteomics and spatial transcriptomics). By default, it uses the 50 hallmark gene sets from msigdb. Alternatively, users can provide their own geneset of interest in a list format.

Usage

```r
run_pathway_mean(
data,
genesiset = NULL,
species = "Homo sapiens",
type = "scrna",
ncores = 1)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>A list object containing data matrix and celltype and sample vector.</td>
</tr>
<tr>
<td>genesiset</td>
<td>By default (when the geneset argument is not specified), we use the 50 hallmark gene set from msigdb. The users can also provide their geneset of interest in a list format, with each list entry containing a vector of the names of genes in a gene set. eg, geneset &lt;- list(&quot;pathway_a&quot; = c(&quot;CANS1&quot;, ...), &quot;pathway_b&quot; = c(&quot;PEX6&quot;))</td>
</tr>
<tr>
<td>species</td>
<td>Whether the species is &quot;Homo sapiens&quot; or &quot;Mus musculus&quot;. Default is &quot;Homo sapiens&quot;.</td>
</tr>
<tr>
<td>type</td>
<td>The type of dataset, either &quot;scrna&quot;, &quot;spatial_t&quot;, or &quot;spatial_p&quot;.</td>
</tr>
<tr>
<td>ncores</td>
<td>Number of cores for parallel processing.</td>
</tr>
</tbody>
</table>
Value

A dataframe of samples x features. The features are in the form of pathway 1 celltype a, pathway 2 celltype b... etc, with the number representing the averaged expression of a given pathway in cells from a given celltype.

Examples

```r
utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[1:500, 1:200]
celltype <- data$celltype
sample <- data$sample
data <- data@assays$RNA$data

alldata <- scFeatures:::formatData(data = data, celltype = celltype, sample = sample)
feature_pathway_mean <- run_pathway_mean(
alldata,
genesiset = NULL, species = "Homo sapiens",
type = "scrna", ncores = 1)
```

Description

This function calculates pathway scores for a given input dataset and gene set using the proportion of gene expression levels. It supports scRNA-seq, spatial transcriptomics and spatial proteomics. By default, it uses the 50 hallmark gene sets from msigdb. Alternatively, users can provide their own gene sets of interest in a list format.

Usage

```r
run_pathway_prop(
data,
genesiset = NULL,
species = "Homo sapiens",
type = "scrna",
ncores = 1)
```

Arguments

data A list object containing data matrix and celltype and sample vector.
geneset: By default (when the geneset argument is not specified), we use the 50 hallmark gene set from msigdb. The users can also provide their geneset of interest in a list format, with each list entry containing a vector of the names of genes in a gene set. E.g., geneset <- list("pathway_a" = c("CANS1", ...), "pathway_b" = c("PEX6"))

species: Whether the species is "Homo sapiens" or "Mus musculus". Default is "Homo sapiens".

type: The type of dataset, either "scrna", "spatial_t", or "spatial_p".

ncores: Number of cores for parallel processing.

Value

da dataframe of samples x features. The features are in the form of pathway 1 celltype a, pathway 2 celltype b... etc, with the number representing the proportion of expression of a given pathway in cells from a given celltype.

Examples

```r
utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[1:100, 1:100]
celltype <- data$celltype
sample <- data$sample
data <- data@assays$RNA$data

alldata <- scFeatures:::formatData(data = data, celltype = celltype, sample = sample)

feature_pathway_prop <- run_pathway_prop(
alldata,
genesiset = NULL, species = "Homo sapiens",
type = "scrna", ncores = 1
)
```

---

**Description**

This function calculates the proportions of cells belonging to each cell type, and applies a logit transformation to the proportions. The input data must contain sample and celltype metadata column. The function supports scRNA-seq and spatial proteomics. The function returns a dataframe with samples as rows and cell types as columns.

**Usage**

```r
run_proportion_logit(data, type = "scrna", ncores = 1)
```
run_proportion_ratio

Arguments

data A list object containing data matrix and celltype and sample vector.
type The type of dataset, either "scrna", "spatial_t", or "spatial_p".
ncores Number of cores for parallel processing.

Value

A dataframe of samples x features. The features are in the form of celltype a, celltype b, with the number representing proportions.

Examples

utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[1:50, 1:20]
celltype <- data$celltype
sample <- data$sample
data <- data@assays$RNA@data

alldata <- scFeatures:::formatData(data = data, celltype = celltype, sample = sample)

feature_proportion_logit <- run_proportion_logit(
alldata,
  type = "scrna", ncores = 1
)

run_proportion_ratio Generate cell type proportion ratio

Description

This function calculates pairwise cell type proportion ratio for each sample and applies a logit transformation to the proportions. The input data must contain sample and celltype metadata column. The function supports scRNA-seq and spatial proteomics. The function returns a dataframe with samples as rows and cell types as columns.

Usage

run_proportion_ratio(data, type = "scrna", ncores = 1)

Arguments

data A list object containing data matrix and celltype and sample vector.
type The type of dataset, either "scrna", "spatial_t", or "spatial_p".
ncores Number of cores for parallel processing.
run_proportion_raw

Value

A dataframe of samples x features. The features are in the form of celltype a vs celltype b, celltype a vs celltype c, with the number representing the ratio between the two cell types.

Examples

```r
utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[1:50, 1:20]
celltype <- data$celltype
sample <- data$sample
data <- data@assays$RNA@data

alldata <- scFeatures:::formatData(data = data, celltype = celltype, sample = sample)

feature_proportion_ratio <- run_proportion_ratio(
  alldata,
  type = "scrna", ncores = 1
)
```

Description

This function calculates the proportions of cells belonging to each cell type. The input data must contain sample and celltype metadata column. The function supports scRNA-seq and spatial proteomics. The function returns a dataframe with samples as rows and cell types as columns.

Usage

```r
run_proportion_raw(data, type = "scrna", ncores = 1)
```

Arguments

- `data`: A list object containing data matrix and celltype and sample vector.
- `type`: The type of dataset, either "scrna", "spatial_t", or "spatial_p".
- `ncores`: Number of cores for parallel processing.

Value

A dataframe of samples x features. The features are in the form of celltype a, celltype b, with the number representing proportions.
Examples

```r
utils::data("example_scrnaseq", package = "scFeatures")
data <- example_scrnaseq[1:50, 1:20]

celltype <- data$celltype
sample <- data$sample
data <- data@assays$RNA@data

alldata <- scFeatures:::formatData(data = data, celltype = celltype, sample = sample)

feature_proportion_raw <- run_proportion_raw(
  alldata,
  type = "scrna", ncores = 1
)
```

scFeatures  
Wrapper function to run all feature types in scFeatures

Description

The scFeatures function generates a variety of features from a Seurat object containing single cell RNA-sequencing data. By default, all feature types will be generated and returned in a single list containing multiple data frames.

Usage

```r
scFeatures(
  data = NULL,
  sample = NULL,
  celltype = NULL,
  spatialCoords = NULL,
  spotProbability = NULL,
  feature_types = NULL,
  type = "scrna",
  ncores = 1,
  species = "Homo sapiens",
  celltype_genes = NULL,
  aggregated_genes = NULL,
  geneset = NULL
)
```

Arguments

data  input data, a matrix of genes by cells
sample  a vector of sample information
celltype  a vector of cell type information
spatialCoords   a list of two vectors containing the x and y coordinates of each cell
spotProbability a matrix of spot probability, each row represents a celltype and each column represents a spot
feature_types   vector containing the name of the feature types to generate, options are "proportion_raw", "proportion_logit", "proportion_ratio", "gene_mean_celltype", "gene_prop_celltype", "gene_cor_celltype", "pathway_gsva", "pathway_mean", "pathway_prop", "CCI", "gene_mean_aggregated", "gene_prop_aggregated", "gene_cor_aggregated", "L_stats", "celltype_interaction", "morans_I", "nn_correlation". If no value is provided, all the above feature types will be generated.
type            input data type, either "scrna" (stands for single-cell RNA-sequencing data), "spatial_p" (stands for spatial proteomics data), or "spatial_t" (stands for single cell spatial data)
ncores          number of cores, default to 1
species         either "Homo sapiens" or "Mus musculus". Defaults to "Homo sapiens" if no value provided

celltype_genes  the genes of interest for celltype specific gene expression feature category, If no value is provided, the top variable genes will be used
aggregated_genes the genes of interest for overall aggregated gene expression feature category, If no value is provided, the top variable genes will be used
geneset         the geneset of interest for celltype specific pathway feature category, If no value is provided, the 50 hallmark pathways will be used

Value

a list of dataframes containing the generated feature matrix in the form of sample x features

Examples

def scfeatures_result <- scFeatures(data, celltype = celltype, sample = sample, type = "scrna", feature_types = "proportion_raw")

scfeatures_result  Example of scFeatures() output

Description

This is an example output of the scFeatures() function for example_scdnaseq.
Usage

\texttt{data("scfeatures_result")}

Format

\texttt{scfeatures_result:}

A list with two dataframes. In each dataframe the columns are each patient and the rows are the feature values. The first dataframe contains the feature type "proportion_raw". The second dataframe contains the feature type "proportion_logit".
Index

* datasets
  example_scrnaseq, 2
  scfeatures_result, 24
example_scrnaseq, 2
get_num_cell_per_spot, 3
remove_mito_ribo, 4
run_association_study_report, 4
run_CCI, 5
run_celltype_interaction, 6
run_gene_cor, 7
run_gene_cor_celltype, 8
run_gene_mean, 9
run_gene_mean_celltype, 10
run_gene_prop, 11
run_gene_prop_celltype, 12
run_L_function, 14
run_Morans_I, 15
run_nn_correlation, 16
run_pathway_gsva, 17
run_pathway_mean, 18
run_pathway_prop, 19
run_proportion_logit, 20
run_proportion_ratio, 21
run_proportion_raw, 22
scFeatures, 23
scfeatures_result, 24