

Package ‘scFeatures’

June 5, 2023

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

Version 1.0.0

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)

Imports DelayedArray, DelayedMatrixStats, EnsDb.Hsapiens.v79, EnsDb.Mmusculus.v79, GSVA, Seurat, ape, glue, dplyr, ensemblDb, gtools, msigdb, proxyC, reshape2, spatstat.explore, spatstat.geom, tidyr, AUCell, BiocParallel, SpatialExperiment, SummarizedExperiment, rmarkdown, methods, stats, DT, cli, SingleCellSignalR, MatrixGenerics

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

VignetteBuilder knitr

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

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

git_branch RELEASE_3_17

git_last_commit b8b24b5

git_last_commit_date 2023-04-25

Date/Publication 2023-06-04

Author Yue Cao [aut, cre],
Yingxin Lin [aut],
Ellis Patrick [aut],
Pengyi Yang [aut],
Jean Yee Hwa Yang [aut]

Maintainer Yue Cao <yue.cao@sydney.edu.au>

R topics documented:

example_scrnaseq	2
get_num_cell_per_spot	3
makeSeurat	4
process_data	5
remove_mito_ribo	6
run_association_study_report	6
run_CCI	7
run_celltype_interaction	8
run_gene_cor	9
run_gene_cor_celltype	10
run_gene_mean	11
run_gene_mean_celltype	12
run_gene_prop	13
run_gene_prop_celltype	14
run_L_function	15
run_Morans_I	16
run_nn_correlation	17
run_pathway_gsva	18
run_pathway_mean	19
run_pathway_prop	20
run_proportion_logit	21
run_proportion_ratio	22
run_proportion_raw	23
scFeatures	23
scfeatures_result	25
Index	26

example_scrnaseq	<i>Example of scRNA-seq data</i>
------------------	----------------------------------

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>.

Usage

```
data("example_scrnaseq")
```

Format

example_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

<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE120575>

get_num_cell_per_spot	<i>Estimate a relative number of cells per spot for spatial transcriptomics data</i>
-----------------------	--

Description

This function takes a spatial transcriptomics data 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(data)
```

Arguments

data spatial transcriptomics data in Seurat object.

Value

the object with the relative number of cells/spot stored in the number_cells attribute.

Examples

```
data("example_scrnaseq" , package = "scFeatures")
data <- example_scrnaseq
data$celltype <- NULL

data <- get_num_cell_per_spot(data)
```

makeSeurat

*Format data into Seurat object structured for scFeatures functions***Description**

This function is used to convert a SingleCellExperiment, SpatialExperiment or a Seurat object into Seurat object containing all required fields and structured for scFeatures functions.

Usage

```
makeSeurat(
  data,
  sample = NULL,
  celltype = NULL,
  assay = NULL,
  spatialCoords = NULL,
  spotProbability = NULL
)
```

Arguments

data	input data, either a SingleCellExperiment or SpatialExperiment object. The object needs to contain a column named "sample" and a column named "celltype". Alternatively, user can provide the name of the column containing sample and celltype into the sample and celltype argument. When passing as SingleCellExperiment or SpatialExperiment, by default we use the assay stored in "log-count". Alternatively, user can specify the assay to use in the assay argument. If users want to construct features from the spatial category, by default we need columns called "x_cord" and "y_cord". Alternatively, please specify the relevant column in the spatialCoords argument. For spot-based spatial transcriptomics, we also requires a matrix containing cell type prediction probability of each spot, in the format of celltype x spot
sample	a vector providing sample identifier for each cell. If not provided, we assume the data contain a metadata column "sample" for running scFeatures.
celltype	a vector providing celltype identifier. If not provided, we assume the data contain a metadata column "celltype" for running scFeatures.
assay	the assay identifier if using a SingleCellExperiment or SpatialExperiment object.
spatialCoords	the spatialCoords identifiers provided in a list of two vectors, if users want to construct features from the spatial category. If not provided, we assume the data contain the metadata columns "x_cord" and "y_cord" for constructing spatial features.
spotProbability	a matrix in the format of celltype x spot, where each entry is the prediction probability of that cell type for each spot. This is needed by spatial transcriptomics data.

Value

A Seurat dataset containing required metadata for running scFeatures.

Examples

```
data("example_scrnaseq" , package = "scFeatures")
data <- example_scrnaseq
coordinate <- list(x = rep(1, ncol(data)), y = rep(1, ncol(data)))
data <- makeSeurat(data, spatialCoords = coordinate)
```

process_data

data pre-processing

Description

This function takes a Seurat object as input and does data cleaning and pre-processing. For example, it replaces the "+" and "-" signs in the celltype column with "plus" and "minus", respectively. It also removes patients that have less than 10 cells across all cell types. If the normalise argument is set to TRUE, the function will normalize the data using the Seurat::NormalizeData function.

Usage

```
process_data(data, normalise = TRUE)
```

Arguments

data	input data, a Seurat object.
normalise	a logical value indicating whether to normalize the data or not. Default value is TRUE.

Value

a Seurat object

Examples

```
data("example_scrnaseq" , package = "scFeatures")
data <- example_scrnaseq
data <- process_data(data, normalise = FALSE)
```

remove_mito_ribo	<i>Remove mitochondrial and ribosomal genes, and other highly correlated genes</i>
------------------	--

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(data)
```

Arguments

data	A Seurat object containing expression data
------	--

Value

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

Examples

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

data <- remove_mito_ribo(data)
```

run_association_study_report	<i>Create an association study report in HTML format</i>
------------------------------	--

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

```
output_folder <- tempdir()
data("scfeatures_result" , package = "scFeatures")
run_association_study_report(scfeatures_result, output_folder )
```

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 -> celltype b – ligand 1 -> receptor 2 , which indicates the interaction between ligand 1 in celltype a and receptor 2 in celltype b.

It supports scRNA-seq.

Usage

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

Arguments

`data` input data, a Seurat object containing celltype and sample label

`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.

Examples

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

feature_CCI <- run_CCI(data, type = "scrna" , ncores = 1 )
```

```
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 Seurat object containing celltype and sample label
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 give sample.

Examples

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

x <- sample(1:100, ncol(data) , replace = TRUE)
y <- sample(1:100, ncol(data) , replace = TRUE)
data <- makeSeurat(data, spatialCoords = list(x,y))
data$sample <- sample( c("patient1", "patient2", "patient3"), ncol(data) , replace= TRUE )

feature_celltype_interaction <- run_celltype_interaction(
  data,
  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

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

Arguments

data	A Seurat object containing celltype and sample label
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, <code>genes <- c("GZMA", "GZMK", "CCR7", "RPL38")</code>
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

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

feature_gene_cor <- run_gene_cor(
  data, 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

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

Arguments

data	A Seurat object containing celltype and sample label
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.

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

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

feature_gene_cor_celltype <- run_gene_cor_celltype(
  data,
  type = "scrna", num_top_gene = 5, ncores = 1
)
```

run_gene_mean	<i>Generate overall aggregated mean expression</i>
---------------	--

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

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

Arguments

data	A Seurat object containing celltype and sample label
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, <code>genes <- c("GZMA", "GZMK", "CCR7", "RPL38")</code>
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, gene 2 ... etc, with the numbers representing averaged gene expression across all cells.

Examples

```
data("example_scrnaseq" , package = "scFeatures")
data <- example_scrnaseq

feature_gene_mean <- run_gene_mean(
  data,
  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

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

Arguments

data	A Seurat object containing celltype and sample label
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

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

feature_gene_mean_celltype <- run_gene_mean_celltype(
  data,
```

```

    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.

Usage

```

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

```

Arguments

data	A Seurat object containing cell type and sample label
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, <code>genes <- c("GZMA", "GZMK", "CCR7", "RPL38")</code>
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

```

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

feature_gene_prop <- run_gene_prop(data, type = "scrna", num_top_gene = 1500, 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).

Usage

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

Arguments

data	A Seurat object containing celltype and sample label
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 proportion of gene expression of the given gene across the cells of the the given celltype.

Examples

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

feature_gene_prop_celltype <- run_gene_prop_celltype(
  data,
```

```

    type = "scrna", num_top_gene = 100, ncores = 1
  )

```

run_L_function

Generate L stats

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

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

Arguments

data	A Seurat object containing celltype and sample label
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

```

data("example_scrnaseq" , package = "scFeatures")
data <- example_scrnaseq

x <- sample(1:100, ncol(data) , replace = TRUE)
y <- sample(1:100, ncol(data) , replace = TRUE)
data <- makeSeurat(data, spatialCoords = list(x,y))
data$sample <- sample( c("patient1", "patient2", "patient3"), ncol(data) , replace= TRUE )

feature_L_function <- run_L_function(data, type = "spatial_p", ncores = 1)

```

run_Morans_I

*Generate Moran's I***Description**

This function calculates Moran's I to measure spatial autocorrelation, which is an indication of how strongly the feature (ie, 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, ie, the values are spatially distributed randomly. The function supports spatial proteomics and spatial transcriptomics.

Usage

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

Arguments

data	A Seurat object containing celltype and sample label
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

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

x <- sample(1:100, ncol(data) , replace = TRUE)
y <- sample(1:100, ncol(data) , replace = TRUE)
data <- makeSeurat(data, spatialCoords = list(x,y))
data$sample <- sample( c("patient1", "patient2", "patient3"), ncol(data) , replace= TRUE )

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

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

run_nn_correlation	<i>Generate nearest neighbour correlation</i>
--------------------	---

Description

This function calculates the nearest neighbour correlation for each feature (eg, 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 Seurat object containing celltype and sample label
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

```
data("example_scrnaseq" , package = "scFeatures")
data <- example_scrnaseq[1:50, 1:20]
x <- sample(1:100, ncol(data) , replace = TRUE)
y <- sample(1:100, ncol(data) , replace = TRUE)
data <- makeSeurat(data, spatialCoords = list(x,y))
data$sample <- sample( c("patient1", "patient2", "patient3"), ncol(data) , replace= TRUE )

feature_nn_correlation <- run_nn_correlation(
  data,
  type = "spatial_p", ncores = 1
)
```

run_pathway_gsva	<i>Generate pathway score using gene set enrichment analysis</i>
------------------	--

Description

This function calculates pathway scores for a given input dataset and gene set using gene set enrichment analysis (GSEA). 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 Seurat object containing celltype and sample label
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

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

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

run_pathway_mean	<i>Generate pathway score using expression level</i>
------------------	--

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 gene sets of interest in a list format.

Usage

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

Arguments

data	A Seurat object containing celltype and sample label
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("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

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

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

feature_pathway_mean <- run_pathway_mean(
  data,
  geneset = NULL, species = "Homo sapiens",
  type = "scrna", ncores = 1
)
```

run_pathway_prop	<i>Generate pathway score using proportion of expression</i>
------------------	--

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 and spatial transcriptomics). 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_prop(
  data,
  geneset = NULL,
  species = "Homo sapiens",
  type = "scrna",
  ncores = 1
)
```

Arguments

data	A Seurat object containing celltype and sample label
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("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

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 proportion of expression of a given pathway in cells from a given celltype.

Examples

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

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

run_proportion_logit *Generate cell type proportions, with logit transformation*

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

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

Arguments

data	A Seurat object containing celltype and sample label
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

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

feature_proportion_logit <- run_proportion_logit(
  data,
  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 Seurat object containing celltype and sample label
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 vs celltype b, celltype a vs celltype c, with the number representing the ratio between the two cell types.

Examples

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

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

run_proportion_raw	<i>Generate cell type proportion raw</i>
--------------------	--

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

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

Arguments

data	A Seurat object containing celltype and sample label
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

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

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

scFeatures	<i>Wrapper function to run all feature types in scFeatures</i>
------------	--

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

```

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

```

Arguments

<code>data</code>	input data, a Seurat object containing "sample" and "celltype" column. "x_cord" and "y_cord" is also required for constructing the features in the spatial metrics category.
<code>feature_types</code>	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.
<code>type</code>	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)
<code>ncores</code>	number of cores , default to 1
<code>species</code>	either "Homo sapiens" or "Mus musculus". Defaults to "Homo sapiens" if no value provided
<code>celltype_genes</code>	the genes of interest for celltype specific gene expression feature category If no value is provided, the top variable genes will be used
<code>aggregated_genes</code>	the genes of interest for overall aggregated gene expression feature category If no value is provided, the top variable genes will be used
<code>geneset</code>	the geneset of interest for celltype specific pathway feature category If no value is provided, the 50 hallmark pathways will be used
<code>sample</code>	the sample identifier if using a SingleCellExperiment
<code>celltype</code>	the celltype identifier if using a SingleCellExperiment
<code>assay</code>	the assay identifier if using a SingleCellExperiment
<code>spatialCoords</code>	the spatialCoords identifiers if using a SingleCellExperiment

Value

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

Examples

```
data("example_scrnaseq" , package = "scFeatures")
data <- example_scrnaseq
scfeatures_result <- scFeatures(data, type = "scrna", feature_types = "proportion_raw")
```

scfeatures_result	<i>Example of scFeatures() output</i>
-------------------	---------------------------------------

Description

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

Usage

```
data("scfeatures_result")
```

Format

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, [25](#)

example_scrnaseq, [2](#)

get_num_cell_per_spot, [3](#)

makeSeurat, [4](#)

process_data, [5](#)

remove_mito_ribo, [6](#)

run_association_study_report, [6](#)

run_CCI, [7](#)

run_celltype_interaction, [8](#)

run_gene_cor, [9](#)

run_gene_cor_celltype, [10](#)

run_gene_mean, [11](#)

run_gene_mean_celltype, [12](#)

run_gene_prop, [13](#)

run_gene_prop_celltype, [14](#)

run_L_function, [15](#)

run_Morans_I, [16](#)

run_nn_correlation, [17](#)

run_pathway_gsva, [18](#)

run_pathway_mean, [19](#)

run_pathway_prop, [20](#)

run_proportion_logit, [21](#)

run_proportion_ratio, [22](#)

run_proportion_raw, [23](#)

scFeatures, [23](#)

scfeatures_result, [25](#)