Package ‘dmSC’

May 31, 2024

Title DeMayo Lab Single-Cell Utilities

Version 2023.11.16.1-0

Description A repository for useful functions that the DeMayo lab uses to process single-cell RNAseq data. These functions are subject to change as procedures change.

License `use_mit_license()`, `use_gpl3_license()` or friends to pick a license

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports Seurat,
dplyr,
Matrix,
magrittr,
biomaRt,
data.table,
ggplot2

Suggests Seurat (>= 4.4.0),
dplyr (>= 1.1.3),
Matrix (>= 1.6.1.1),
magrittr (>= 2.0.3),
biomaRt (>= 2.56.1),
data.table (>= 1.14.8),
R (>= 3.5.0),
ggplot2,
sf,
knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

adobo_impute 2
append_datapath 3
apply_ensembl_seurat 3
apply_mapper 4
check_if_data_exists 5

create_data_folder . . . 5
create_default_mapper_table . . . 5
create_split_heatmap . . . 6
eliminate_mapper_all_duplicates . . . 7
eliminate_mapper_key_duplicates . . . 7
eliminate_mapper_value_duplicates . . . 8
get_datapath . . . 8
impute_st . . . 9
install_data . . . 9
ipa_dotplot . . . 10
list_data . . . 11
preprocessing_function . . . 11
set_datapath . . . 12
seurat_to_sc_ref . . . 12
volcano_plot . . . 13

Index **15**

adobo_impute — *Call cell types in a Seurat object using adobo, a Python package that calls cell types using a marker gene list.*

Description

Call cell types in a Seurat object using adobo, a Python package that calls cell types using a marker gene list.

Usage

```
adobo_impute(
  st_obj,
  gene_symbols,
  cell_types,
  cluster_slot = NULL,
  out_cluster_slot = "adobo_clusters",
  symbol_switch = NULL,
  new_idents = FALSE
)
```

Arguments

`st_obj` Seurat object to process.

`gene_symbols` A vector of gene symbols corresponding to the marker genes for each cluster.

`cell_types` A vector of cell types corresponding to the marker genes for each cluster.

`cluster_slot` The slot name in the Seurat dataset corresponding to cell identities or cluster names. The default behavior is Idents(st_obj).

`symbol_switch` Whether to switch ENSEMBL to MGI symbols for "mouse" or "human". Default is to not switch (None). Input gene format must be gene symbols in that case.

`new_idents` Whether to set the adobo output results as new cell idents. Default is FALSE.

Value

The modified Seurat object with adobo clusters.

`append_datapath` *Add search locations for dmSC datasets.*

Description

Add search locations for dmSC datasets.

Usage

```
append_datapath(...)
```

`apply_ensembl_seurat` *Apply ENSEMBL naming to Seurat objects*

Description

Given a Seurat object with a specific name and assay, this function renames the genes in the assay (assuming the gene names are MGI symbols) by replacing them with their ENSEMBL IDs (ENSMUSG...).

Usage

```
apply_ensembl_seurat(st, mart = NULL, mapper_table = NULL)
```

Arguments

`st`	The Seurat object. Make sure that the default assay is the one that is to be processed.
`mart`	If mapper_table is NULL, the mart to be used for fetching ENSEMBL IDs and MGI symbols. Default is NULL. If mapper_table is not NULL, this parameter will be ignored.
`mapper_table`	The mapping table to use. Default is NULL; if a table is not provided, then one will be created using the `create_default_mapper_table` function.

Value

A Seurat object with renamed symbols in a new assay with 'ENSEMBL' prepended to the old assay name. This new assay will be the default.

apply_mapper *Map using a mapper_table*

Description

Given a list of gene symbols or ENSEMBL IDs and a mapping table describing the appropriate relationships between them, converts the entries of to_map into their corresponding values.

Usage

```
apply_mapper(to_map, to_ensembl = FALSE, mart = NULL, mapper_table = NULL)
```

Arguments

to_map	A vector containing values to map from.
to_ensembl	If mapper_table is NULL, whether ENSEMBL IDs should be the keys (FALSE) or values (TRUE) of the created mapping table. Default is FALSE. If mapper_table is not NULL, this parameter will be ignored.
mart	If mapper_table is NULL, the mart to be used for fetching ENSEMBL IDs and MGI symbols. Default is NULL. If mapper_table is not NULL, this parameter will be ignored. Uses package biomaRt.
mapper_table	The mapping table to use. Default is NULL; if a table is not provided, then one will be created using the create_default_mapper_table function.

Details

Note that to_map must contain entries that fall within the keys of the mapping table. These keys will then be turned into their corresponding values within the mapping table. There may be duplicate keys or values in the mapping table, but a mapping table containing a one-to-one mapping between keys and values is recommended for compatibility with other preprocessing functions.

Note that this can be used outside the specific context of ENSEMBL IDs and MGI symbols.

UPDATE: The GetBM function may already carry out these functions. The functionality of the method may be replaced in the future.

Value

A vector with mapped values (or 'NA' in the corresponding slot if the element in to_map is not in the mapping table).

`check_if_data_exists` *Check if dmSC datasets are already installed. Not exported.*

Description

Check if dmSC datasets are already installed. Not exported.

Usage

```
check_if_data_exists(data_nm)
```

Value

Whether a data name exists in the available search paths.

`create_data_folder` *Create the data folder for holding dmSC datasets.*

Description

Create the data folder for holding dmSC datasets.

Usage

```
create_data_folder(verbose = FALSE)
```

`create_default_mapper_table` *Default mapping table creation*

Description

Creates a default mapping table mapping between MGI symbols and ENSEMBL IDs. Keys are in the first column, values are in the second. One-to-many and one-to-one relations are tossed out to avoid problems with preprocessing_function.

Usage

```
create_default_mapper_table(mart = NULL, to_ensembl = FALSE)
```

Arguments

`mart`	The mart to be used for fetching ENSEMBL IDs and MGI symbols. Default is NULL. @param to_ensembl Whether ENSEMBL IDs should be the keys (FALSE) or values (TRUE). Default is FALSE. @returns A list containing the following: * A one-to-one mapping table based on the description above ('mapper_table'), * A data.table containing the duplicated keys that were removed ('duplicate_keys'), * A data.table containing the duplicated values that were removed ('duplicate_values').

Details

Currently creates a one-to-one mapping table based on Mus Musculus genes and ENSEMBL version 102.

Note that we could allow one-to-one and many-to-one mappings, but NOT one-to- many. However, the assumption toward the end of preprocessing_function would be broken and we would need to write code that pools values with the same ENSEMBL ID.

`create_split_heatmap` *Generate a grouped heatmap for a specific list of genes and corresponding expression data. Just a wrapper around pheatmap::pheatmap.*

Description

Generate a grouped heatmap for a specific list of genes and corresponding expression data. Just a wrapper around pheatmap::pheatmap.

Usage

```
create_split_heatmap(
  assay,
  assay_data_columns = NULL,
  type_column = "TYPE",
  symbol_column = "SYMBOL",
  gap_length = 3,
  cluster_rows = FALSE,
  cluster_cols = FALSE,
  border_color = NA,
  ...
)
```

Arguments

`assay` The assay to include. Should be a 2D table with a column of gene symbols and at least one column of corresponding gene expression values for that symbol. Can also include a 'type column', or a column including the categories under which each gene falls (if you're trying to split a heatmap based on specific categories of genes). Ideally a data.table or a data.frame.

`assay_data_columns` A vector of names specifying the data columns within assay. The default (NULL) just takes all columns besides the symbol and type columns.

`type_column` The name of the column of assay specifying the category that each gene falls into. If NULL, then no types are assumed and the heatmap won't be split. Default is "TYPE".

`symbol_column` The column of assay specifying gene symbols. Default is "SYMBOL".

`gap_length` The space between splits on the split heatmap. Default is 3.

`cluster_rows` Whether to cluster heatmap rows within splits. Default is FALSE.

`cluster_cols` Whether to cluster columns across all heatmap entries, disregarding splits. Default is FALSE.

`border_color`	The color of border to use on heatmaps. Default is NA, or no color.
`...`	Other arguments to be passed into pheatmap::pheatmap.

Value

Heatmap object as output by pheatmap::pheatmap.

`eliminate_mapper_all_duplicates`

Eliminate all duplicates in mapping table

Description

Eliminates duplicate values in all columns (both keys and values) of a mapper_table, returning both the de-duplicated mapper_table and data.frames consisting of the removed duplicate keys/values (and the number of occurrences for each duplicate key/value).

Usage

```
eliminate_mapper_all_duplicates(mapper_table)
```

Arguments

`mapper_table`	The mapper_table whose values are to be de-duplicated.

Value

A list containing the following: * A deduplicated mapping table based on the description above ('mapper_table'), * A data.table containing the duplicated keys that were removed ('duplicate_keys'). * A data.table containing the duplicated values that were removed ('duplicate_values').

`eliminate_mapper_key_duplicates`

Eliminate duplicates in mapping table keys

Description

Eliminates duplicate values in the 'keys' (first column) of a mapper_table, returning both the de-duplicated mapper_table and a data.frame consisting of the removed duplicate keys (and the number of occurrences for each duplicate key).

Usage

```
eliminate_mapper_key_duplicates(mapper_table)
```

Arguments

`mapper_table`	The mapper_table whose keys are to be de-duplicated.

Value

A list containing the following: * A deduplicated mapping table based on the description above ('mapper_table'), * A data.table containing the duplicated keys that were removed ('duplicate_keys').

eliminate_mapper_value_duplicates

Eliminate duplicates in mapping table values

Description

Eliminates duplicate values in the 'values' (second column) of a mapper_table, returning both the de-duplicated mapper_table and a data.frame consisting of the removed duplicate values (and the number of occurrences for each duplicate value).

Usage

```
eliminate_mapper_value_duplicates(mapper_table)
```

Arguments

`mapper_table`	The mapper_table whose values are to be de-duplicated.

Details

At the end of the day, analogous to `eliminate_mapper_key_duplicates(rev(mapper_table))`.

Value

A list containing the following: * A deduplicated mapping table based on the description above ('mapper_table'), * A data.table containing the duplicated values that were removed ('duplicate_values').

get_datapath

View search locations for dmSC datasets.

Description

View search locations for dmSC datasets.

Usage

```
get_datapath()
```

Value

A vector describing the current search path.

impute_st *Impute cell types for Seurat object tgt_st with Seurat object ref_st.*

Description

Impute cell types for Seurat object tgt_st with Seurat object ref_st.

Usage

```
impute_st(
  tgt_st,
  ref_st,
  out_nm = "singler_clusters",
  tgt_clusters_nm = NULL,
  ref_clusters_nm = NULL
)
```

Arguments

`tgt_st`	Target Seurat object to impute cell types against.
`ref_st`	Reference Seurat object against which to impute cell types.
`out_nm`	Column to create imputed clusters under in tgt_st.
`tgt_clusters_nm`	Column in tgt_st containing clusters. If NULL, Seurat::Idents(tgt_st) is used.
`ref_clusters_nm`	Column in ref_st containing clusters. If NULL, Seurat::Idents(ref_st) is used.

Value

Target Seurat object with extra column containing imputed clusters.

install_data *Install dmSC datasets from local locations or online locations.*

Description

Install dmSC datasets from local locations or online locations.

Usage

```
install_data(nm, ..., reinstall = FALSE, verbose = FALSE)
```

Arguments

`nm`	Name of the dataset to install. Do not include extension.
`...`	Names of additional datasets to install. Optional.
`reinstall`	Whether to re-copy the dataset to your installed data files if the requested dataset is installed. Default FALSE.
`verbose`	Whether to log messages. Default FALSE.

ipa_dotplot — *A simple function for creating dot plots for gene ontology (GO) or Ingenuity Pathway Analysis (IPA) plots. Code taken from enrichplot: https://github.com/YuLab-SMU/enrichplot/blob/devel/R/dotplot.R#L200-L207*

Description

A simple function for creating dot plots for gene ontology (GO) or Ingenuity Pathway Analysis (IPA) plots. Code taken from enrichplot: https://github.com/YuLab-SMU/enrichplot/blob/devel/R/dotplot.R#L200-L207

Usage

```
ipa_dotplot(
  tbl,
  z_score_colname,
  ratio_colname,
  neg_log_p_colname,
  x_axis_label_colname = NULL,
  y_axis_label_colname = NULL
)
```

Arguments

`tbl` A data.frame or similar containing gene symbols, z-scores, gene set ratios, and -log10(p) values. Both of these should be output or obtainable from GO enrichment analysis or IPA results.

`z_score_colname` Name of the column containing results for z-scores.

`ratio_colname` Name of the column containing gene set ratios.

`neg_log_p_colname` Name of the column containing -log10(p) values.

`x_axis_label_colname` Name of the column containing gene symbols for plotting along the x-axis. This will create a horizontal plot. Note that either this or y_axis_label_colname should be set but not both.

`y_axis_label_colname` Name of the column containing gene symbols for potting along the y-axis. This will create a vertical plot. Note that either this or y_axis_label_colname should be set but not both.

Value

A ggplot2-compatible dotplot summarizing the input data.

`list_data` *List all installable dmSC datasets.*

Description

List all installable dmSC datasets.

Usage

```
list_data()
```

Value

A vector of all installable file names.

`preprocessing_function` *Standard preprocessing function for single-cell data*

Description

DEPRECATED: Recommended to just do preprocessing yourself so you have more granular control over parameters. See the source code of this function for an idea/starting point if still interested.

Usage

```
preprocessing_function(st, mart = NULL, mapper_table = NULL)
```

Arguments

`st` The Seurat single-cell file to be preprocessed. Read above for how to ensure this file is structured.

`mart` If mapper_table is NULL, the mart to be used for fetching ENSEMBL IDs and MGI symbols. Default is NULL. If mapper_table is not NULL, this parameter will be ignored.

`mapper_table` The mapping table to use. Default is NULL; if a table is not provided, then one will be created using the `create_default_mapper_table` function.

Details

TO DO:

- Make mitochondrial gene identification case-insensitive.
- Manual assay specification in addition to just using the default assay.
- Preserve variable features in the ENSEMBL assay since we just change the gene names.
- It seems that this function does not work when multiple assays are present. Fix that.

Preprocess a Seurat single-cell dataset using the following techniques:

- Remove genes that are expressed in fewer than 5% of cells.
- Remove genes that are not expressed or have no variance in expression across all cells.
- Remove cells with UMI counts, feature counts, mitochondrial gene percent, and ribosomal gene percent outside 3 mean absolute deviations from the median.
- Remove cells with no ENSEMBL ID counterpart.

Note that in the Seurat dataset, mitochondrial genes must begin with 'mt-' and ribosomal genes must begin with 'Rp'. The genes also cannot be ENSEMBL IDs to begin with. The gene name requirements are case-sensitive. Some mitochondrial genes may begin with CAPITAL 'MT-' based on the source; this will cause problems. Also ensure that the assay of interest in the Seurat object is marked as the default assay, since this will be the assay that will be used.

Value

The preprocessed Seurat single-cell file. A new assay will be created with the key 'ENSEMBL' appended onto the end of the default assay of the input Seurat file.

`set_datapath` *Set the search locations for dmSC datasets.*

Description

Set the search locations for dmSC datasets.

Usage

```
set_datapath(...)
```

Arguments

`...`	An array of paths to search for dmSC datasets.

`seurat_to_sc_ref` *Obtain a single-cell reference matrix compatible with CIBERSORTx from a Seurat object.*

Description

Note that this function is basically just calling GetAssayData and then reassigning the column names of the resultant table.

Usage

```
seurat_to_sc_ref(st_obj, cluster_colname = NULL, ...)
```

Arguments

`st_obj` The Seurat object to analyze.

`cluster_colname` The name of the column in st_obj[[]] that contains cluster information. If NULL, information obtained from Seurat::Idents(st_obj). Default is NULL.

`assay` The name of the assay to pull from. If NULL, gets the default assay from the Seurat object. Default is NULL.

Value

A data.frame representing the single-cell reference matrix. Careful, since these can get pretty large in size!

`volcano_plot` *Plot a nice-looking volcano plot with minimal effort. Thanks, Elvis.*

Description

Plot a nice-looking volcano plot with minimal effort. Thanks, Elvis.

Usage

```
volcano_plot(
  df,
  logfc_column,
  logfc_cutoff,
  padj_column,
  padj_cutoff = 0.05,
  point_shape = 21,
  hline_linetype = "dashed",
  vlines_linetype = "dashed"
)
```

Arguments

`df` The data frame to input. This would ideally be the output of a program like DESeq2, containing logarithmic fold changes and adjusted p-values.

`logfc_column` The column name containing the logarithmic fold-changes.

`logfc_cutoff` The logarithmic fold-change cutoff that determines "high" or "low" values.

`padj_column` The column name containing adjusted p-values.

`padj_cutoff` The default p-value cutoff. Default is 0.05.

`point_shape` The shape of the points to plot. Default is 21, corrsponding to circles. See ggplot2 for more information.

`hline_linetype` The type of boundary to plot between significant and nonsignificant points. Default is a dashed line.

`vlines_linetype` The type of boundary to plot between high/low points and points with lower logarithmic fold changes. Default is a dashed line.

Value

A ggplot2-compatible volcano plot.

Index

adobo_impute, 2
append_datapath, 3
apply_ensembl_seurat, 3
apply_mapper, 4

check_if_data_exists, 5
create_data_folder, 5
create_default_mapper_table, 5
create_split_heatmap, 6

eliminate_mapper_all_duplicates, 7
eliminate_mapper_key_duplicates, 7
eliminate_mapper_value_duplicates, 8

get_datapath, 8

impute_st, 9
install_data, 9
ipa_dotplot, 10

list_data, 11

preprocessing_function, 11

set_datapath, 12
seurat_to_sc_ref, 12

volcano_plot, 13