Package: scooter (via r-universe)

October 24, 2024

Title Streamlined scRNA-Seq Analysis Pipeline

Version 0.0.0.9004

Description Streamlined scRNA-Seq analysis pipeline.

License MIT + file LICENCE

Encoding UTF-8

LazyData true

Depends R (>= 4.1)

Imports cowplot, data.table, dplyr, ggplot2, ggsci, glue, irlba, magrittr, Matrix, matrixStats, methods, purrr, RColorBrewer, readr, rlang (>= 1.0.0), Rtsne, sctransform, Seurat (>= 4.0.2), SeuratObject, stringr, tibble, tidyr, uwot

Suggests roxygen2, testthat

RoxygenNote 7.2.1

Repository https://igordot.r-universe.dev

RemoteUrl https://github.com/igordot/scooter

RemoteRef HEAD

RemoteSha 2a639459d3848e111717624497797441cfbf1747

Contents

`add_seurat_assay` *Add assay to Seurat object.*

Description

Add assay to Seurat object.

Usage

```
add_seurat_assay(seurat_obj, assay, counts_matrix, log_file = NULL)
```

Arguments

`seurat_obj`	Seurat object.
`assay`	Seurat assay to add the matrix to.
`counts_matrix`	Raw counts matrix.
`log_file`	Filename for the log file.

Value

Seurat object of cells found in both the existing object and new data.

as_data_frame_seurat *Function to extract data from Seurat object.*

Description

Function to extract data from Seurat object.

Usage

```
as_data_frame_seurat(
  seurat_obj,
  assay = NULL,
  slot = NULL,
  features = NULL,
  reduction = NULL,
  metadata = TRUE
)
```

Arguments

seurat_obj	A Seurat object.
assay	Assay such as RNA.
slot	Slot such as counts. Default is scale.data.
features	Features from assay.
reduction	Character vector of reduction types.
metadata	Boolean. To grab metadata or not

Value

A metadata file merged on cell identifiers.

calculate_clusters *Run dimensionality reduction, pca, tse, and umap*

Description

Run dimensionality reduction, pca, tse, and umap

Usage

```
calculate_clusters(
  pcs,
  num_dim,
  log_file,
  num_neighbors = 30,
  res = NULL,
  algorithm = 3
)
```

Arguments

pcs	Data.
num_dim	Number of PCs to use for tsne and umap.
log_file	log file.
num_neighbors	Number of neighbors to use for umap.
res	Resolution
algorithm	See Seurat::FindClusters()

Value

.

---

calculate_mito_pct *Calculate mitochondrial percentage from Seurat object.*

Description

Calculate mitochondrial percentage from Seurat object.

Usage

```
calculate_mito_pct(seurat_obj)
```

Arguments

seurat_obj	A Seurat object.

Value

Seurat object.

calculate_variance *Get variable genes and scale data.*

Description

Get variable genes and scale data.

Usage

```
calculate_variance(
  seurat_obj,
  assay = "RNA",
  nfeatures = 2000,
  log_file = NULL
)
```

Arguments

seurat_obj	Seurat object.
assay	Assay.
nfeatures	Number of variable features to output.
log_file	A log file.

Value

A named list of the top features, and the scaled data.

calc_clust_averages *Get cluster averages.*

Description

Get cluster averages.

Usage

```
calc_clust_averages(metadata, data, group)
```

Arguments

metadata	Metadata.
data	Gene expression data.
group	Column in metadata.

Value

.

`check_identity_column` *Check identity of the Seurat object.*

Description

Check identity of the Seurat object.

Usage

```
check_identity_column(seurat_obj, identity_column)
```

Arguments

`seurat_obj`	A Seurat object.
`identity_column`	The name of the identity column to pull from object metadata.

Value

The name of the identity column, potentially corrected if resolution.

`create_color_vect` *Function to create a color vector.*

Description

Function to create a color vector.

Usage

```
create_color_vect(seurat_obj, group = "orig.ident")
```

Arguments

`seurat_obj`	A Seurat object.
`group`	Assay such as RNA.

Value

A vector of colors.

`create_seurat_obj` *Create a new Seurat object from a matrix.*

Description

Create a new Seurat object from a matrix.

Usage

```
create_seurat_obj(
  counts_matrix,
  assay = "RNA",
  min_cells = 1,
  min_genes = 1,
  log_file = NULL,
  project = "proj"
)
```

Arguments

`counts_matrix`	A matrix of raw counts.
`assay`	Seurat assay to add the data to.
`min_cells`	Include genes/features detected in at least this many cells.
`min_genes`	Include cells where at least this many genes/features are detected.
`log_file`	Filename for the logfile.
`project`	Project name for Seurat object.

Value

Seurat object.

`differential_expression_global` *Calculate differential expression for one group versus all*

Description

Calculate differential expression for one group versus all

Usage

```
differential_expression_global(
  data,
  metadata,
  metadata_column,
  log_fc_threshold = 0.5,
  min.pct = 0.1,
  test.use = "wilcox",
  out_path = ".",
  write = FALSE,
  log_file = NULL
)
```

Arguments

data	Gene expression data.
metadata	Metadata.
metadata_column	Column in metadata.
log_fc_threshold	Log fc threshold.
min.pct	Minimum percentage of cells a gene must be expressed in to be tested.
test.use	Test to use.
out_path	output path.
write	Boolean to write or not.
log_file	log file.

Value

.

---

differential_expression_paired

Calculate differential expression between specific groups

Description

Calculate differential expression between specific groups

Usage

```
differential_expression_paired(
  data,
  metadata,
  metadata_column,
  list_groups = NULL,
  log_fc_threshold = 0.5,
  min.pct = 0.1,
  test.use = "wilcox",
  out_path = ".",
  write = FALSE,
  log_file = NULL
)
```

Arguments

data	Gene expression data.
metadata	Metadata.
metadata_column	Column in metadata.
list_groups	dataframe of groups to compare in the metadata column.
log_fc_threshold	Log fc threshold.
min.pct	Minimum percentage of cells a gene must be expressed in to be tested.
test.use	Test to use.
out_path	output path.
write	Boolean to write or not.
log_file	log file.

Value

.

---

differential_expression_per_cluster

Calculate differentially expressed genes within each subpopulation/cluster

---

Description

Calculate differentially expressed genes within each subpopulation/cluster

Usage

```
differential_expression_per_cluster(
  seurat_obj,
  cluster_column,
  group_column,
  test = "wilcox",
  out_path = ".",
  write = TRUE,
  log_file = NULL
)
```

Arguments

seurat_obj	Gene expression data.
cluster_column	Metadata column specifying the groups to split by.
group_column	Metadata column specifying the groups for differential expressin within each split.
test	Statistical method to use.
out_path	Output path.
write	Boolean to save results to disk.
log_file	log file.

Value

.

---

filter_data *Filter cells based on the number of genes and mitochondrial reads.*

Description

Filter out cells based on minimum and maximum number of genes and maximum percentage mitochondrial reads. If cutoffs are not provided, the min_genes will be the 0.02 quantile, and the max genes will be 0.98 quantile and the mitochondrial percentage will be 10

Usage

```
filter_data(
  data,
  log_file = NULL,
  min_genes = NULL,
  max_genes = NULL,
  max_mt = 10
)
```

Arguments

data	A tibble with metadata.
log_file	Log file.
min_genes	Minimum number of genes per cell.
max_genes	Maximim number of genes per cell.
max_mt	Maximum percentage of mitochondrial reads per cell.

Value

Filtered data

---

geneset_score *Get geneset scores.*

Description

Get geneset scores.

Usage

```
geneset_score(module_tbl, counts_raw, min_cpm = 0, limit_pct = 1)
```

Arguments

module_tbl	geneset table.
counts_raw	Raw counts
min_cpm	.
limit_pct	.

Value

.

get_color_scheme *Determine the color scheme.*

Description

Determine the color scheme. Can be specified for samples or clusters to avoid confusion.

Usage

```
get_color_scheme(type = "clusters")
```

Arguments

type	Type of scheme ("samples" or "clusters").

Value

A vector of colors.

get_dr_point_size *Determine the point size for reduced dimensions scatter plots (smaller for larger datasets).*

Description

Determine the point size for reduced dimensions scatter plots (smaller for larger datasets).

Usage

```
get_dr_point_size(num_cells)
```

Arguments

num_cells	Number of cells (points on the plot) or a Seurat object.

Value

Numeric point size.

get_test_counts_matrix *Get an example counts matrix.*

Description

Get a small matrix of raw counts from the PBMC dataset.

Usage

```
get_test_counts_matrix()
```

Value

A matrix of raw counts.

Examples

```
pbmc_mat <- get_test_counts_matrix()
```

import_mtx *Read in 10x Genomics Cell Ranger Matrix Market format data.*

Description

Read in 10x Genomics Cell Ranger Matrix Market format data.

Usage

```
import_mtx(data_path, gene_column = 2, log_file = NULL)
```

Arguments

data_path	Path to directory that holds the files output from 10x.
gene_column	The column with the gene names.
log_file	Filename for the log file.

Value

Named list of matrices. One matrix for each data type as specified in the third column of the features.tsv file. As of Oct 3rd 2019, the two options are ‘Gene Expression‘ and ‘Antibody Capture‘

`load_sample_counts_matrix` *Read in Gene Expression and Antibody Capture data from a 10x Genomics Cell Ranger sparse matrix or from a text file.*

Description

Read in Gene Expression and Antibody Capture data from a 10x Genomics Cell Ranger sparse matrix or from a text file.

Usage

```
load_sample_counts_matrix(sample_name, path, log_file = NULL)
```

Arguments

`sample_name`	A character that will be used as a prefix for all cell names.
`path`	Path to directory containing 10x matrix, or path to a text file.
`log_file`	Filename for the log file.

Value

Named list of matrices. One matrix for each data type. Currently the only two data types are 'Gene Expression' and 'Antibody Capture'

`log_normalize_data` *Log normalize data.*

Description

Log normalize data.

Usage

```
log_normalize_data(data, log_file = NULL)
```

Arguments

`data`	A seurat object.
`log_file`	log file.

Value

normalized data

`merge_metadata` *Function to merge two metadata tables together.*

Description

Function to merge two metadata tables together.

Usage

```
merge_metadata(metadata1, metadata2, log_file = NULL)
```

Arguments

`metadata1`	A Seurat object or a tibble containing metadata with either a column called "cell" with cell IDs or rownames with cell IDs.
`metadata2`	A tibble containing metadata with either a column called "cell" with cell IDs or rownames with cell IDs.
`log_file`	A log filename.

Value

A metadata file merged on cell identifiers.

`normalize_data` *Normalize data*

Description

Normalize data

Usage

```
normalize_data(
  data,
  method,
  nfeatures = 2000,
  metadata = NULL,
  assay = NULL,
  log_file = NULL
)
```

Arguments

data	Input data.
method	Normalization method ("log" or "sct").
nfeatures	.
metadata	.
assay	.
log_file	Log file.

Value

Normalized data

---

plot_distribution *Plot the distribution of specified features/variables.*

Description

Plot the distribution of specified features/variables.

Usage

```
plot_distribution(data, features, grouping, color_scheme = NULL)
```

Arguments

data	Seurat object or metadata.
features	Vector of features to plot (such as "nGene", "nUMI", "percent.mito").
grouping	X.
color_scheme	(optional) Vector of colors.

Value

A vector of colors.

run_dr *Run dimensionality reduction, pca, tse, and umap*

Description

Run dimensionality reduction, pca, tse, and umap

Usage

```
run_dr(
  data,
  dr_method,
  prefix,
  assay = NULL,
  var_features = FALSE,
  features = NULL,
  graph = NULL,
  num_dim_use = NULL,
  reduction = NULL,
  num_neighbors = NULL,
  num_pcs = NULL,
  ...
)
```

Arguments

data	Data to use for dimensionality reduction.
dr_method	Dimensionality reduction method.
prefix	.
assay	.
var_features	.
features	.
graph	.
num_dim_use	.
reduction	.
num_neighbors	.
num_pcs	.

Value

list of dimensionality reduced/

See Also

run_pca(), run_tsne(), run_umap()

run_pca *Run PCA*

Description

Run PCA

Usage

```
run_pca(data, num_pcs, prefix = "PC_")
```

Arguments

data	A tibble with metadata.
num_pcs	Maximim number of genes per cell.
prefix	suffix.

Value

named list of feature loadings, cell embeddings, sdev, output from pca

run_tsne *Run TSNE*

Description

Run TSNE

Usage

```
run_tsne(data, seed.use = 1, dim.embed = 2, prefix = "tSNE_")
```

Arguments

data	Data to run tsne on.
seed.use	seed to use.
dim.embed	Number of tsne embeddings to return.
prefix	suffix.

Value

tsne.

run_umap *Run UMAP*

Description

Run UMAP

Usage

```
run_umap(data, num_neighbors, min_dist = 0.3, prefix = "UMAP_")
```

Arguments

data	Data to run UMAP on.
num_neighbors	Number of neighbors.
min_dist	Distance metric.
prefix	Prefix.

Value

umap

save_seurat_counts_matrix *Function to write Seurat counts matrix to csv.*

Description

Function to write Seurat counts matrix to csv.

Usage

```
save_seurat_counts_matrix(
  seurat_obj,
  proj_name = "",
  label = "",
  out_dir = ".",
  assay = "RNA",
  slot = "data",
  log_file = NULL
)
```

Arguments

seurat_obj	A Seurat object.
proj_name	Name of the project that will be the prefix of the file name.
label	An optional label for the file.
out_dir	Directory in which to save csv.
assay	The assay within the Seurat object to retrieve data from.
slot	The slot within the Seurat object to retrieve data from.
log_file	A log filename.

Value

A csv file in the out_dir.

---

sctransform_data *SCT normalize data.*

Description

SCT normalize data.

Usage

```
sctransform_data(counts, metadata, nfeatures, log_file = NULL)
```

Arguments

counts	Raw counts.
metadata	Metadata for each cell.
nfeatures	Number of variable features to output.
log_file	A log file.

Value

A named list of the vst output, the final scaled data, and the top variable genes.

set_identity *Set identity of the Seurat object.*

Description

Wrapper for SeuratObject::Idents() with extra safety checks.

Usage

```
set_identity(seurat_obj, identity_column)
```

Arguments

seurat_obj	A Seurat object.
identity_column	The name of the identity column to pull from object metadata.

Value

A Seurat object with an updated identity set.

write_message *Small function to write to message and to log file.*

Description

Small function to write to message and to log file.

Usage

```
write_message(message_str, log_file = NULL)
```

Arguments

message_str	A string to write as a message.
log_file	A log filename.

Value

A message and writes the message to the specified log file.

Examples

```
write_message(message_str = "Finished Step 1", log_file = "log.file.txt")
```

Index