

Package: loupeR (via r-universe)

October 29, 2024

Title Converts Seurat objects to 10x Genomics Loupe files

Version 1.1.2

Description Converts Seurat objects to 10x Genomics Loupe files. This is a second line to make the package checker not complain.

License file LICENSE

Encoding UTF-8

Depends R (>= 4.0.0)

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0), Matrix, SeuratObject (>= 5.0.0)

Config/testthat/edition 3

Imports methods, Seurat, hdf5r

Repository <https://blaserlab.r-universe.dev>

RemoteUrl <https://github.com/10XGenomics/loupeR>

RemoteRef HEAD

RemoteSha a169417e7a0d59ae7ebba042a4f100de8efb8101

Contents

counts_matrix_from_assay	2
create_bugreport	2
create_bugreport_from_seurat	3
create_loupe	3
create_loupe_from_seurat	4
read_feature_ids_from_tsv	5
select_assay	6
select_clusters	6
select_projections	7
setup	7
validate_barcodes	8
validate_clusters	8
validate_count_mat	9
validate_projections	9

Index**10**

`counts_matrix_from_assay`*Extract the counts matrix from the Assay*

Description

Extract the counts matrix from the Assay

Usage

```
counts_matrix_from_assay(assay)
```

Arguments

`assay` A `SeuratObject::Assay` or `SeuratObject::Assay5`

Value

A sparse counts matrix

`create_bugreport`*Create a Bugreport from a count matrix, projections, and clusters*

Description

This bugreport can then be included when reaching out to 10xGenomics Support or when filing a Github ticket. This information should be included along with any other output when creating a Loupe file.

Usage

```
create_bugreport(  
  count_mat,  
  clusters,  
  projections,  
  assay_name = NULL,  
  seurat_obj_version = NULL,  
  skip_metadata = FALSE  
)
```

Arguments

count_mat	A sparse dgCMatix as is generated via Matrix::rsparsematrix. Rows are features, Columns are barcodes.
clusters	list of factors that hold information for each barcode
projections	list of matrices, all with dimensions (barcodeCount x 2)
assay_name	optional string that holds the Seurat Object assay name.
seurat_obj_version	optional string that holds the Seurat Object version. It is useful for debugging compatibility issues.
skip_metadata	optional logical which skips printing metadata

```
create_bugreport_from_seurat
```

Create a Bugreport from a Seurat Object

Description

This bugreport can then be included when reaching out to 10xGenomics Support or when filing a Github ticket. This information should be included along with any other output when creating a Loupe file.

Usage

```
create_bugreport_from_seurat(obj)
```

Arguments

obj	A Seurat Object
-----	-----------------

```
create_loupe
```

Create a Loupe file

Description

Create a Loupe file

Usage

```

create_loupe(
  count_mat,
  clusters = list(),
  projections = list(),
  output_dir = NULL,
  output_name = NULL,
  feature_ids = NULL,
  executable_path = NULL,
  force = FALSE,
  seurat_obj_version = NULL
)

```

Arguments

count_mat	A sparse dgCMatrix as is generated via Matrix::rsparsematrix. Rows are features, Columns are barcodes.
clusters	list of factors that hold information for each barcode
projections	list of matrices, all with dimensions (barcodeCount x 2)
output_dir	optional directory where the Loupe file will be written
output_name	optional name of the Loupe file with the extensions not included.
feature_ids	optional character vector that specifies the feature ids of the count matrix. Typically, these are the ensemble ids.
executable_path	optional path to the louter executable.
force	optional logical as to whether we should overwrite an already existing file
seurat_obj_version	optional string that holds the Seurat Object version. It is useful for debugging compatibility issues.

Value

TRUE on success, FALSE on error

create_loupe_from_seurat

Create a Loupe file from a Seurat Object

Description

create_loupe_from_seurat() passes the active counts matrix, reductions, and factors found in meta.data to create a Loupe file.

Usage

```
create_loupe_from_seurat(
  obj,
  output_dir = NULL,
  output_name = NULL,
  dedup_clusters = FALSE,
  feature_ids = NULL,
  executable_path = NULL,
  force = FALSE
)
```

Arguments

obj	A Seurat Object
output_dir	optional directory where the Loupe file will be written
output_name	optional name of the Loupe file with the extensions not included.
dedup_clusters	optional logical that will try to deduplicate all clusters that are numerically the same
feature_ids	optional character vector that specifies the feature ids of the count matrix. Typically, these are the ensemble ids.
executable_path	optional path to the louter executable.
force	optional logical as to whether we should overwrite an already existing file

Value

TRUE on success, FALSE on error

read_feature_ids_from_tsv

Read FeatureIds from 10x features.tsv.gz file

Description

Read FeatureIds from 10x features.tsv.gz file

Usage

```
read_feature_ids_from_tsv(tsv_path)
```

Arguments

tsv_path	character vector path to the features.tsv.gz file
----------	---

Value

A character vector of the feature ids

select_assay	<i>Select the "best" assay for conversion</i>
--------------	---

Description

Prioritizes the active assay, then RNA, and then the rest Usable assays must have a non empty count matrix

Usage

```
select_assay(obj)
```

Arguments

obj	A Seurat Object
-----	-----------------

Value

A list with the named Seurat Assay or NULL if not found

select_clusters	<i>Select clusters from the assay</i>
-----------------	---------------------------------------

Description

Select clusters from the assay

Usage

```
select_clusters(obj, dedup = FALSE)
```

Arguments

obj	A Seurat Object
dedup	logical to deduplicate clusters. Default TRUE.

Value

A list of factors

`select_projections` *Select projections from the assay*

Description

Select projections from the assay

Usage

```
select_projections(obj)
```

Arguments

`obj` A Seurat Object

Value

A list of matrices, all with dimensions (barcodeCount x 2)

`setup` *Setup eula and download executable*

Description

Setup eula and download executable

Usage

```
setup(executable_path = NULL)
```

Arguments

`executable_path`
optional string to a non default executable path

validate_barcodes	<i>Validate the format of the barcodes</i>
-------------------	--

Description

Validate the format of the barcodes

Usage

```
validate_barcodes(barcodes)
```

Arguments

barcodes a character vector

Value

A list with two elements:

- success: a logical value indicating success (TRUE) or failure (FALSE)
- msg: an optional error message (NULL if success is TRUE)

validate_clusters	<i>Validate the seurat clusters</i>
-------------------	-------------------------------------

Description

Validate the seurat clusters

Usage

```
validate_clusters(clusters, barcode_count)
```

Arguments

clusters list of factors that hold information for each barcode
barcode_count number of barcodes

Value

A list with two elements:

- success: a logical value indicating success (TRUE) or failure (FALSE)
- msg: an optional error message (NULL if success is TRUE)

validate_count_mat *Validate the seurat count matrix*

Description

Validate the seurat count matrix

Usage

```
validate_count_mat(count_mat, feature_ids = NULL)
```

Arguments

count_mat A sparse dgCMatrix as is generated via `Matrix::rsparsematrix`. Rows are features, Columns are barcodes.

feature_ids optional character vector that specifies the feature ids of the count matrix. Typically, these are the ensemble ids.

Value

A list with two elements:

- success: a logical value indicating success (TRUE) or failure (FALSE)
- msg: an optional error message (NULL if success is TRUE)

validate_projections *Validate the seurat projections*

Description

Validate the seurat projections

Usage

```
validate_projections(projections, barcode_count)
```

Arguments

projections list of matrices, all with dimensions (barcodeCount x 2)

barcode_count number of barcodes

Value

A list with two elements:

- success: a logical value indicating success (TRUE) or failure (FALSE)
- msg: an optional error message (NULL if success is TRUE)

Index

counts_matrix_from_assay, [2](#)
create_bugreport, [2](#)
create_bugreport_from_seurat, [3](#)
create_loupe, [3](#)
create_loupe_from_seurat, [4](#)

read_feature_ids_from_tsv, [5](#)

select_assay, [6](#)
select_clusters, [6](#)
select_projections, [7](#)
setup, [7](#)

validate_barcodes, [8](#)
validate_clusters, [8](#)
validate_count_mat, [9](#)
validate_projections, [9](#)