

Package: cellxgene.census (via r-universe)

October 7, 2024

Title CZ CELLxGENE Discover Cell Census

Version 1.16.1

Description API to facilitate the use of the CZ CELLxGENE Discover Census. For more information about the API and the project visit <https://github.com/chanzuckerberg/cellxgene-census/>

License MIT + file LICENSE

URL <https://github.com/chanzuckerberg/cellxgene-census>

BugReports <https://github.com/chanzuckerberg/cellxgene-census/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Additional_repositories <https://tiledb-inc.r-universe.dev>

Imports aws.s3, dplyr, httr, jsonlite, methods, stats, tiledbsoma, tiledb

Suggests bit64, knitr, rmarkdown, SingleCellExperiment (>= 1.20.0), Seurat (>= 4.1.0), testthat (>= 3.0.0), withr

Config/testthat/edition 3

VignetteBuilder knitr

OS_type unix

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

RemoteUrl <https://github.com/chanzuckerberg/cellxgene-census>

RemoteRef v1.16.2

RemoteSha aec41967dbd74bd0e18b0b8a1350acde28fb5870

Contents

download_source_h5ad	2
get_census_mirror	3
get_census_mirror_directory	3

get_census_version_description	4
get_census_version_directory	4
get_presence_matrix	5
get_seurat	5
get_single_cell_experiment	7
get_source_h5ad_uri	9
new_SOMATfileDBContext_for_census	9
open_soma	10

Index	12
--------------	-----------

download_source_h5ad *Download source H5AD to local file name.*

Description

Download source H5AD to local file name.

Usage

```
download_source_h5ad(
  dataset_id,
  file,
  overwrite = FALSE,
  census_version = "stable",
  census = NULL
)
```

Arguments

dataset_id	The dataset_id of interest.
file	Local file name to store H5AD file.
overwrite	TRUE to allow overwriting an existing file.
census_version	The desired Census version.
census	An open Census handle for census_version. If not provided, then it will be opened and closed automatically; but it's more efficient to reuse a handle if calling download_source_h5ad() multiple times.

Examples

```
download_source_h5ad("0895c838-e550-48a3-a777-dbcd35d30272", "/tmp/data.h5ad", overwrite = TRUE)
```

`get_census_mirror` *Get locator information about a Census mirror*

Description

Get locator information about a Census mirror

Usage

```
get_census_mirror(mirror)
```

Arguments

`mirror` Name of the mirror.

Value

List with mirror information

Examples

```
get_census_mirror("AWS-S3-us-west-2")
```

`get_census_mirror_directory`
Get the directory of Census mirrors currently available

Description

Get the directory of Census mirrors currently available

Usage

```
get_census_mirror_directory()
```

Value

Nested list with information about available mirrors

Examples

```
get_census_mirror_directory()
```

get_census_version_description

Get release description for a Census version

Description

Get release description for a Census version

Usage

```
get_census_version_description(census_version)
```

Arguments

census_version The census version name.

Value

List with the release location and metadata

Examples

```
as.data.frame(get_census_version_description("stable"))
```

get_census_version_directory

Get the directory of Census releases currently available

Description

Get the directory of Census releases currently available

Usage

```
get_census_version_directory()
```

Value

Data frame of available cell census releases, including location and metadata.

Examples

```
get_census_version_directory()
```

get_presence_matrix *Read the feature dataset presence matrix.*

Description

Read the feature dataset presence matrix.

Usage

```
get_presence_matrix(census, organism, measurement_name = "RNA")
```

Arguments

census The census object from `cellxgene.census::open_soma()`.
organism The organism to query, usually one of `Homo sapiens` or `Mus musculus`
measurement_name The measurement object to query. Defaults to `RNA`.

Value

A `tiledbsoma::matrixZeroBasedView` object with dataset join id & feature join id dimensions, filled with 1s indicating presence. The sparse matrix is accessed with zero-based indexes since the join id's may be zero.

Examples

```
census <- open_soma()  
on.exit(census$close(), add = TRUE)  
print(get_presence_matrix(census, "Homo sapiens")$dim())
```

get_seurat *Export Census slices to Seurat*

Description

Convenience wrapper around `SOMAXperimentAxisQuery`, to build and execute a query, and return it as a Seurat object.

Usage

```

get_seurat(
  census,
  organism,
  measurement_name = "RNA",
  X_layers = c(counts = "raw", data = NULL),
  obs_value_filter = NULL,
  obs_coords = NULL,
  obs_column_names = NULL,
  obsm_layers = FALSE,
  var_value_filter = NULL,
  var_coords = NULL,
  var_column_names = NULL,
  var_index = "feature_id"
)

```

Arguments

census	The census object, usually returned by <code>cellxgene.census::open_soma()</code> .
organism	The organism to query, usually one of <code>Homo sapiens</code> or <code>Mus musculus</code> .
measurement_name	The measurement object to query. Defaults to <code>RNA</code> .
X_layers	A named character of X layers to add to the Seurat assay, where the names are the names of Seurat slots (counts or data) and the values are the names of layers within X.
obs_value_filter	A SOMA <code>value_filter</code> across columns in the obs dataframe, expressed as string.
obs_coords	A set of coordinates on the obs dataframe index, expressed in any type or format supported by <code>SOMADataFrame</code> 's <code>read()</code> method.
obs_column_names	Columns to fetch for the obs data frame.
obsm_layers	Names of arrays in <code>obsm</code> to add as the cell embeddings; pass <code>FALSE</code> to suppress loading in any dimensional reductions.
var_value_filter	Same as <code>obs_value_filter</code> but for <code>var</code> .
var_coords	Same as <code>obs_coords</code> but for <code>var</code> .
var_column_names	Columns to fetch for the var data frame.
var_index	Name of column in 'var' to add as feature names.

Value

A Seurat object containing the sensus slice.

Examples

```
## Not run:
census <- open_soma()
seurat_obj <- get_seurat(
  census,
  organism = "Homo sapiens",
  obs_value_filter = "cell_type == 'leptomeningeal cell'",
  var_value_filter = "feature_id %in% c('ENSG00000107317', 'ENSG00000106034')"
)

seurat_obj

census$close()

## End(Not run)
```

```
get_single_cell_experiment
```

Export Census slices to SingleCellExperiment

Description

Convenience wrapper around `SOMAXperimentAxisQuery`, to build and execute a query, and return it as a `SingleCellExperiment` object.

Usage

```
get_single_cell_experiment(
  census,
  organism,
  measurement_name = "RNA",
  X_layers = c(counts = "raw"),
  obs_value_filter = NULL,
  obs_coords = NULL,
  obs_column_names = NULL,
  obsm_layers = FALSE,
  var_value_filter = NULL,
  var_coords = NULL,
  var_column_names = NULL,
  var_index = "feature_id"
)
```

Arguments

<code>census</code>	The census object, usually returned by <code>cellxgene.census::open_soma()</code> .
<code>organism</code>	The organism to query, usually one of <code>Homo sapiens</code> or <code>Mus musculus</code>
<code>measurement_name</code>	The measurement object to query. Defaults to <code>RNA</code> .

<code>X_layers</code>	A character vector of X layers to add as assays in the main experiment; may optionally be named to set the name of the resulting assay (eg. <code>'X_layers = c(counts = "raw")'</code> will load in X layer “raw” as assay “counts”); by default, loads in all X layers
<code>obs_value_filter</code>	A SOMA <code>value_filter</code> across columns in the obs dataframe, expressed as string.
<code>obs_coords</code>	A set of coordinates on the obs dataframe index, expressed in any type or format supported by <code>SOMADataFrame</code> 's <code>read()</code> method.
<code>obs_column_names</code>	Columns to fetch for the obs data frame.
<code>obsm_layers</code>	Names of arrays in <code>obsm</code> to add as the cell embeddings; pass <code>FALSE</code> to suppress loading in any dimensional reductions.
<code>var_value_filter</code>	Same as <code>obs_value_filter</code> but for <code>var</code> .
<code>var_coords</code>	Same as <code>obs_coords</code> but for <code>var</code> .
<code>var_column_names</code>	Columns to fetch for the var data frame.
<code>var_index</code>	Name of column in <code>'var'</code> to add as feature names.

Value

A `SingleCellExperiment` object containing the sensus slice.

Examples

```
## Not run:
census <- open_soma()
sce_obj <- get_single_cell_experiment(
  census,
  organism = "Homo sapiens",
  obs_value_filter = "cell_type == 'leptomeningeal cell'",
  var_value_filter = "feature_id %in% c('ENSG00000107317', 'ENSG00000106034')"
)

sce_obj

census$close()

## End(Not run)
```

get_source_h5ad_uri *Locate source h5ad file for a dataset.*

Description

Locate source h5ad file for a dataset.

Usage

```
get_source_h5ad_uri(dataset_id, census_version = "stable", census = NULL)
```

Arguments

dataset_id The dataset_id of interest.

census_version The desired Census version.

census An open Census handle for census_version. If not provided, then it will be opened and closed automatically; but it's more efficient to reuse a handle if calling get_source_h5ad_uri() multiple times.

Value

A list with uri and optional s3_region.

Examples

```
get_source_h5ad_uri("0895c838-e550-48a3-a777-dbcd35d30272")
```

new_SOMATileDBContext_for_census
 Create SOMATileDBContext for Census

Description

Create a SOMATileDBContext suitable for using with open_soma(). Typically open_soma() creates a context automatically, but one can be created separately in order to set custom configuration options, or to share it between multiple open Census handles.

Usage

```
new_SOMATileDBContext_for_census(  
  census_version_description,  
  mirror = "default",  
  ...  
)
```

Arguments

census_version_description	The result of <code>get_census_version_description()</code> for the desired Census version.
mirror	The name of the intended census mirror (or <code>get_census_mirror_directory()[[name]]</code> to save the lookup), or <code>NULL</code> to configure for local file access.
...	Custom configuration options.

Value

SOMATileDBContext object for `open_soma()`.

Examples

```
census_desc <- get_census_version_description("stable")
ctx <- new_SOMATileDBContext_for_census(census_desc, "soma.init_buffer_bytes" = paste(4 * 1024**3))
census <- open_soma("stable", tiledbsoma_ctx = ctx)
census$close()
```

open_soma

Open the Census

Description

Open the Census

Usage

```
open_soma(
  census_version = "stable",
  uri = NULL,
  tiledbsoma_ctx = NULL,
  mirror = NULL
)
```

Arguments

census_version	The version of the Census, e.g., "stable".
uri	A URI containing the Census SOMA objects to open instead of a released version. (If supplied, takes precedence over <code>census_version</code> .)
tiledbsoma_ctx	A <code>tiledbsoma::SOMATileDBContext</code> built using <code>new_SOMATileDBContext_for_census()</code> . Optional (created automatically) if using <code>census_version</code> and the context does not need to be reused.
mirror	The Census mirror to access; one of names(<code>get_census_mirror_directory()</code>).

Value

Top-level `tiledbsoma::SOMACollection` object. After use, the census should be closed to release memory and other resources, usually with `on.exit(census$close(), add = TRUE)`. Closing the top-level census will also close all SOMA objects accessed through it.

Examples

```
census <- open_soma()  
as.data.frame(census$get("census_info")$get("summary")$read())$concat()  
census$close()
```

Index

`download_source_h5ad`, [2](#)

`get_census_mirror`, [3](#)

`get_census_mirror_directory`, [3](#)

`get_census_version_description`, [4](#)

`get_census_version_directory`, [4](#)

`get_presence_matrix`, [5](#)

`get_seurat`, [5](#)

`get_single_cell_experiment`, [7](#)

`get_source_h5ad_uri`, [9](#)

`new_SOMATileDBContext_for_census`, [9](#)

`open_soma`, [10](#)