

Package ‘alabaster.sce’

July 25, 2025

Title Load and Save SingleCellExperiment from File

Version 1.9.0

Date 2024-10-16

License MIT + file LICENSE

Description Save SingleCellExperiment into file artifacts, and load them back into memory.

This is a more portable alternative to serialization of such objects into RDS files.

Each artifact is associated with metadata for further interpretation;

downstream applications can enrich this metadata with context-specific properties.

Depends SingleCellExperiment, alabaster.base

Imports methods, alabaster.se, jsonlite

Suggests knitr, testthat, BiocStyle, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.2.3

biocViews DataImport, DataRepresentation

git_url <https://git.bioconductor.org/packages/alabaster.sce>

git_branch devel

git_last_commit 03a65cc

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-07-25

Author Aaron Lun [aut, cre]

Maintainer Aaron Lun <infinite.monkeys.with.keyboards@gmail.com>

Contents

readSingleCellExperiment	2
saveSingleCellExperiment	3

Index	4
--------------	----------

readSingleCellExperiment

Read a SingleCellExperiment from disk

Description

Read a [SingleCellExperiment](#) object from its on-disk representation. This is usually not directly called by users, but is instead called by dispatch in [readObject](#).

Usage

```
readSingleCellExperiment(path, metadata, ...)
```

Arguments

path	String containing a path to a directory, itself created using the saveObject method for SingleCellExperiment objects.
metadata	Named list of metadata for this object, see readObjectFile for details.
...	Further arguments passed to readRangedSummarizedExperiment and internal altReadObject calls.

Value

A [SingleCellExperiment](#) object.

Author(s)

Aaron Lun

See Also

["saveObject, SingleCellExperiment-method"](#), to save the SingleCellExperiment to disk.

Examples

```
# Mocking up an SCE:
mat <- matrix(rpois(10000, 10), ncol=10)
colnames(mat) <- letters[1:10]
rownames(mat) <- sprintf("GENE_%i", seq_len(nrow(mat)))

se <- SingleCellExperiment(list(counts=mat))
se$stuff <- LETTERS[1:10]
se$blah <- runif(10)
reducedDims(se) <- list(
  PCA=matrix(rnorm(ncol(se)*10), ncol=10),
  TSNE=matrix(rnorm(ncol(se)*2), ncol=2)
)
altExps(se) <- list(spikes=SummarizedExperiment(list(counts=mat[1:2,])))

# Staging it:
tmp <- tempfile()
saveObject(se, tmp)
readObject(tmp)
```

`saveSingleCellExperiment`*Save a SingleCellExperiment to disk*

Description

Save a [SingleCellExperiment](#) to its on-disk representation.

Usage

```
## S4 method for signature 'SingleCellExperiment'
saveObject(x, path, ...)
```

Arguments

<code>x</code>	A SingleCellExperiment object or one of its subclasses.
<code>path</code>	String containing the path to a directory in which to save <code>x</code> .
<code>...</code>	Further arguments to pass to the <code>RangedSummarizedExperiment</code> method.

Value

`x` is saved into `path` and `NULL` is invisibly returned.

Author(s)

Aaron Lun

See Also

[readSingleCellExperiment](#), to read the `SingleCellExperiment` back into the R session.

Examples

```
# Mocking up an SCE:
mat <- matrix(rpois(10000, 10), ncol=10)
colnames(mat) <- letters[1:10]
rownames(mat) <- sprintf("GENE_%i", seq_len(nrow(mat)))

se <- SingleCellExperiment(list(counts=mat))
se$stuff <- LETTERS[1:10]
se$blah <- runif(10)
reducedDims(se) <- list(
  PCA=matrix(rnorm(ncol(se)*10), ncol=10),
  TSNE=matrix(rnorm(ncol(se)*2), ncol=2)
)
altExps(se) <- list(spikes=SummarizedExperiment(list(counts=mat[1:2,])))

# Staging it:
tmp <- tempfile()
saveObject(se, tmp)
list.files(tmp, recursive=TRUE)
```

Index

altReadObject, [2](#)

loadSingleCellExperiment
 (readSingleCellExperiment), [2](#)

readObject, [2](#)

readObjectFile, [2](#)

readRangedSummarizedExperiment, [2](#)

readSingleCellExperiment, [2](#), [3](#)

saveObject, [2](#)

saveObject, SingleCellExperiment-method
 (saveSingleCellExperiment), [3](#)

saveSingleCellExperiment, [3](#)

SingleCellExperiment, [2](#), [3](#)

stageObject, SingleCellExperiment-method
 (saveSingleCellExperiment), [3](#)