Package ‘alabaster.se’

March 20, 2025

Title Load and Save SummarizedExperiments from File

Version 1.6.0

Date 2024-10-16

License MIT + file LICENSE

Description Save SummarizedExperiments 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 SummarizedExperiment, alabaster.base

Imports methods, alabaster.ranges, alabaster.matrix, BiocGenerics,
S4Vectors, IRanges, GenomicRanges, jsonlite

Suggests rmarkdown, knitr, testthat, BiocStyle

VignetteBuilder knitr

RoxygenNote 7.3.1

biocViews DataImport, DataRepresentation

git_url https://git.bioconductor.org/packages/alabaster.se

git_branch RELEASE_3_20

git_last_commit 48cf296

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2025-03-20

Author Aaron Lun [aut, cre]

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

Contents

`emptyRowRanges` *Is the* `rowRanges` *empty?*

Description

Check the `rowRanges` of a RangedSummarizedExperiment is empty, i.e., a GRangesList with no ranges.

Usage

```
emptyRowRanges(x)
```

Arguments

`x`	A RangedSummarizedExperiment object or the contents of its `rowRanges`.

Details

Metadata in `mcols` is ignored for the purpose of this discussion, as this can be moved to the `rowData(x)` of the base SummarizedExperiment class without loss. In other words, non-empty `mcols` will not be used to determine that the `rowRanges` is not empty. However, non-empty fields in the `metadata` or in the inner `mcols` of the GRanges will trigger a non-emptiness decision.

Value

A logical scalar indicating whether `x` has empty `rowRanges`.

Examples

```
emptyRowRanges(SummarizedExperiment())
emptyRowRanges(SummarizedExperiment(rowRanges=GRanges()))
emptyRowRanges(SummarizedExperiment(rowRanges=GRangesList()))
```

`readRangedSummarizedExperiment` *Read a RangedSummarizedExperiment from disk*

Description

Read a RangedSummarizedExperiment from its on-disk representation. This is usually not directly called by users, but is instead called by dispatch in `readObject`.

Usage

```
readRangedSummarizedExperiment(path, metadata, ...)
```

Arguments

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

Value

A RangedSummarizedExperiment object.

Author(s)

Aaron Lun

See Also

`"saveObject,RangedSummarizedExperiment-method"`, to save the RangedSummarizedExperiment to disk.

Examples

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

gr <- GRanges("chrA", IRanges(1:1000, width=10))
se <- SummarizedExperiment(list(counts=mat), rowRanges=gr)
se$stuff <- LETTERS[1:10]
rowData(se)$blah <- runif(1000)
metadata(se)$whee <- "YAY"

tmp <- tempfile()
saveObject(se, tmp)
readObject(tmp)
```

`readSummarizedExperiment`

Read a SummarizedExperiment from disk

Description

Read a SummarizedExperiment from its on-disk representation. This is usually not directly called by users, but is instead called by dispatch in `readObject`.

Usage

```
readSummarizedExperiment(path, metadata, ...)
```

Arguments

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

Value

A SummarizedExperiment object.

Author(s)

Aaron Lun

See Also

`"saveObject,SummarizedExperiment-method"`, to save the SummarizedExperiment to disk.

Examples

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

se <- SummarizedExperiment(list(counts=mat))
se$stuff <- LETTERS[1:10]
rowData(se)$blah <- runif(1000)
metadata(se)$whee <- "YAY"

tmp <- tempfile()
saveObject(se, tmp)
readObject(tmp)
```

`saveRangedSummarizedExperiment` *Save a RangedSummarizedExperiment to disk*

Description

Save a RangedSummarizedExperiment to its on-disk representation.

Usage

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

Arguments

`x`	A RangedSummarizedExperiment object or one of its subclasses.
`path`	String containing the path to a directory in which to save `x`.
`...`	Further arguments to pass to "`saveObject,SummarizedExperiment-method`" and internal `altSaveObject` calls.

Value

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

Author(s)

Aaron Lun

See Also

`readRangedSummarizedExperiment`, to read the RangedSummarizedExperiment back into the R session.

Examples

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

gr <- GRanges("chrA", IRanges(1:1000, width=10))
se <- SummarizedExperiment(list(counts=mat), rowRanges=gr)
se$stuff <- LETTERS[1:10]
rowData(se)$blah <- runif(1000)
metadata(se)$whee <- "YAY"

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

`saveSummarizedExperiment` *Save a SummarizedExperiment to disk*

Description

Save a SummarizedExperiment to its on-disk representation.

Usage

```
## S4 method for signature 'SummarizedExperiment'
saveObject(x, path, SummarizedExperiment.allow.dataframe.assay = FALSE, ...)
```

Arguments

`x`	A SummarizedExperiment object or one of its subclasses.
`path`	String containing the path to a directory in which to save `x`.
`SummarizedExperiment.allow.dataframe.assay`	Logical scalar indicating whether to allow data frames as assays of `x`.
`...`	Further arguments to pass to internal `altSaveObject` calls.

Details

By default, we consider the presence of data frames in the assays to be an error. Users should coerce these into an appropriate matrix type, e.g., a dense matrix or a sparse dgCMatrix. If a DataFrame as an assay is truly desired, users may set `options(alabaster.se.reject_data.frames=FALSE)` to skip the error. Note that this only works for DataFrame objects - data.frame objects will not be saved correctly.

Value

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

Author(s)

Aaron Lun

See Also

`readSummarizedExperiment`, to read the SummarizedExperiment back into the R session.

Examples

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

se <- SummarizedExperiment(list(counts=mat))
se$stuff <- LETTERS[1:10]
rowData(se)$blah <- runif(1000)
metadata(se)$whee <- "YAY"

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

Index