

# Package ‘alabaster.se’

December 23, 2024

**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** 2024-12-23

**Author** Aaron Lun [aut, cre]

**Maintainer** Aaron Lun <[infinite.monkeys.with.keyboards@gmail.com](mailto:infinite.monkeys.with.keyboards@gmail.com)>

## Contents

emptyRowRanges . . . . .	2
readRangedSummarizedExperiment . . . . .	2
readSummarizedExperiment . . . . .	3
saveRangedSummarizedExperiment . . . . .	4
saveSummarizedExperiment . . . . .	5
<b>Index</b>	<b>7</b>

---

emptyRowRanges	<i>Is the rowRanges empty?</i>
----------------	--------------------------------

---

### 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	<i>Read a RangedSummarizedExperiment from disk</i>
--------------------------------	--

---

### 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 <code>saveObject</code> method for <code>RangedSummarizedExperiment</code> objects.
metadata	Named list of metadata for this object, see <code>readObjectFile</code> for details.
...	Further arguments passed to <code>readSummarizedExperiment</code> and internal <code>altReadObject</code> 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 <code>saveObject</code> method for <code>SummarizedExperiment</code> objects.
metadata	Named list of metadata for this object, see <code>readObjectFile</code> for details.
...	Further arguments passed to internal <code>altReadObject</code> 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 <a href="#">RangedSummarizedExperiment</a> object or one of its subclasses.
path	String containing the path to a directory in which to save x.
...	Further arguments to pass to " <a href="#">saveObject, SummarizedExperiment-method</a> " and internal <a href="#">altSaveObject</a> 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 <a href="#">SummarizedExperiment</a> 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 <a href="#">altSaveObject</a> 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

`altReadObject`, [3](#), [4](#)  
`altSaveObject`, [5](#), [6](#)

`DataFrame`, [6](#)

`emptyRowRanges`, [2](#)

`GRanges`, [2](#)  
`GRangesList`, [2](#)

`loadSummarizedExperiment`  
    (`readSummarizedExperiment`), [3](#)

`mcols`, [2](#)  
`metadata`, [2](#)

`options`, [6](#)

`RangedSummarizedExperiment`, [2–5](#)  
`readObject`, [2](#), [3](#)  
`readObjectFile`, [3](#), [4](#)  
`readRangedSummarizedExperiment`, [2](#), [5](#)  
`readSummarizedExperiment`, [3](#), [3](#), [6](#)  
`rowData`, [2](#)  
`rowRanges`, [2](#)

`saveObject`, [3](#), [4](#)  
`saveObject`, `RangedSummarizedExperiment`-method  
    (`saveRangedSummarizedExperiment`),  
    [4](#)  
`saveObject`, `SummarizedExperiment`-method  
    (`saveSummarizedExperiment`), [5](#)  
`saveRangedSummarizedExperiment`, [4](#)  
`saveSummarizedExperiment`, [5](#)  
`stageObject`, `RangedSummarizedExperiment`-method  
    (`saveRangedSummarizedExperiment`),  
    [4](#)  
`stageObject`, `SummarizedExperiment`-method  
    (`saveSummarizedExperiment`), [5](#)  
`SummarizedExperiment`, [2–6](#)