

# Package ‘alabaster.spatial’

October 25, 2024

**Title** Save and Load Spatial 'Omics Data to/from File

**Description** Save SpatialExperiment objects and their images 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.

**Version** 1.5.3

**Date** 2024-10-16

**License** MIT + file LICENSE

**Depends** SpatialExperiment, alabaster.base

**Imports** methods, utils, grDevices, S4Vectors, alabaster.sce, rhdf5

**Suggests** testthat, knitr, rmarkdown, BiocStyle, DropletUtils, magick, png, digest

**VignetteBuilder** knitr

**RoxygenNote** 7.3.2

**biocViews** DataImport, DataRepresentation

**git\_url** <https://git.bioconductor.org/packages/alabaster.spatial>

**git\_branch** devel

**git\_last\_commit** 61461d9

**git\_last\_commit\_date** 2024-10-16

**Repository** Bioconductor 3.20

**Date/Publication** 2024-10-25

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

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

## Contents

loadSpatialImage . . . . .	2
readSpatialExperiment . . . . .	2
saveObject,SpatialExperiment-method . . . . .	3
stageSpatialImage . . . . .	4
<b>Index</b>	<b>6</b>

loadSpatialImage      *Load a spatial image*

---

### Description

Load an image as a [SpatialImage](#) or subclass thereof.

### Usage

```
loadSpatialImage(img.info, project)
```

### Arguments

`img.info`      Named list containing the metadata for this assay.  
`project`      Object specifying the project of interest.

### Value

A [SpatialImage](#) containing the image data (or a reference to it).

### Author(s)

Aaron Lun

### Examples

```
example(read10xVisium, echo=FALSE)
img <- imgData(spe)$data[[1]]

tmp <- tempfile()
dir.create(tmp)
meta <- stageObject(img, tmp, "whee")

out <- loadSpatialImage(meta, tmp)
```

---

readSpatialExperiment      *Read a SpatialExperiment from disk*

---

### Description

Read a [SpatialExperiment](#) object from its on-disk representation.

### Usage

```
readSpatialExperiment(path, metadata, ...)
```

**Arguments**

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

**Value**

A `SpatialExperiment` object.

**Author(s)**

Aaron Lun

**See Also**

"`saveObject,SpatialExperiment-method`", to save a `SpatialExperiment` to disk.

**Examples**

```
library(SpatialExperiment)
example(read10xVisium, echo=FALSE)

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

---

saveObject,SpatialExperiment-method  
*Save a spatial experiment*

---

**Description**

Save a `SpatialExperiment` object to its on-disk representation.

**Usage**

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

**Arguments**

x	A <code>SpatialExperiment</code> object.
path	String containing the path to a directory in which to save x.
...	Additional named arguments to pass to specific methods.

**Details**

Currently, only PNG and TIFF image formats are supported in the `imgData`. All other images will be re-saved as PNG.

**Value**

x is saved to path and NULL is invisibly returned.

**Author(s)**

Aaron Lun

**See Also**

[readSpatialExperiment](#), to read the SpatialExperiment back into the R session.

**Examples**

```
library(SpatialExperiment)
example(read10xVisium, echo=FALSE)

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

---

stageSpatialImage      *Stage images for upload*

---

**Description**

These methods are deprecated and are only documented here for back-compatibility purposes.

**Usage**

```
## S4 method for signature 'VirtualSpatialImage'
stageObject(x, dir, path, child = FALSE, ...)

## S4 method for signature 'StoredSpatialImage'
stageObject(x, dir, path, child = FALSE, ...)

## S4 method for signature 'RemoteSpatialImage'
stageObject(x, dir, path, child = FALSE, ...)
```

**Arguments**

x	A <a href="#">SpatialImage</a> object.
dir	String containing a path to a directory.
path	String containing a relative path inside a directory.
child	Logical scalar indicating whether x is a child of another object.
...	Further arguments, ignored.

## Details

Each of the different methods will take advantage of any existing files to avoid an actual save. For example, the [RemoteSpatialImage](#) method will download the file directly to path, while the [StoredSpatialImage](#) method will create a link or copy the file. The [SpatialImage](#) method will fall back to saving the raster directly as a PNG.

## Value

An image file is created at `file.path(dir, path)`, possibly after appending an appropriate file extension.

The return value should be a named list containing at least:

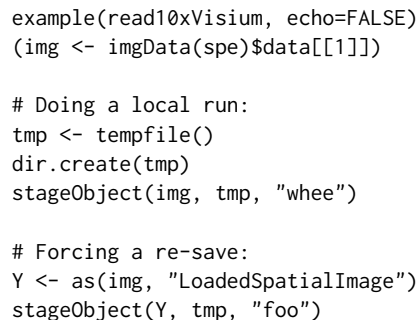
- `$schema`, a string specifying the schema to use to validate the metadata. This may have a `package` attribute to specify the package where the schema lives (in its `inst/schemas` directory).
- `path`, a string containing the path to the file containing the assay contents. This should start with the input path but can be followed by any necessary file extensions.
- `child`, whether this is a child resource of a larger object.

Other fields can be provided and will be included in the metadata, provided that they are recognized by the specified schema.

## Author(s)

Aaron Lun

## Examples

```
example(read10xVisium, echo=FALSE)
The examples show how to use the stageSpatialImage function. The first example shows a local run where a temporary directory is created and the image is staged. The second example shows how to force a re-save of the image as a LoadedSpatialImage object.

```
(img <- imgData(spe)$data[[1]])

# Doing a local run:
tmp <- tempfile()
dir.create(tmp)
stageObject(img, tmp, "whee")

# Forcing a re-save:
Y <- as(img, "LoadedSpatialImage")
stageObject(Y, tmp, "foo")
```


```

# Index

altReadObject, 3

imgData, 3

loadSpatialExperiment  
    (readSpatialExperiment), 2

loadSpatialImage, 2

readObjectFile, 3

readSingleCellExperiment, 3

readSpatialExperiment, 2, 4

RemoteSpatialImage, 5

saveObject, 3

saveObject, SpatialExperiment-method, 3

SpatialExperiment, 2, 3

SpatialImage, 2, 4, 5

stageObject, RemoteSpatialImage-method  
    (stageSpatialImage), 4

stageObject, SpatialExperiment-method  
    (saveObject, SpatialExperiment-method),  
    3

stageObject, StoredSpatialImage-method  
    (stageSpatialImage), 4

stageObject, VirtualSpatialImage-method  
    (stageSpatialImage), 4

stageSpatialImage, 4

StoredSpatialImage, 5