

Package ‘alabaster.spatial’

June 4, 2026

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 1.13.0

Date 2026-02-01

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 7.3.3

biocViews DataImport, DataRepresentation

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

git_branch devel

git_last_commit 3c905d3

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-06-04

Author Aaron Lun [aut, cre]

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

Contents

loadSpatialImage	2
readSpatialExperiment	2
saveObject,SpatialExperiment-method	3
stageSpatialImage	4
Index	6

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, SpatialExperiment.reuse.images = "link", ...)
```

Arguments

x	A <code>SpatialExperiment</code> object.
path	String containing the path to a directory in which to save x.
<code>SpatialExperiment.reuse.images</code>	Whether to re-use existing files on disk (e.g., by copying or linking to them in path) when an image in <code>imgData(x)</code> contains a reference to its original file location (e.g., via <code>imgSource</code>). This can be any of the values described for <code>action=</code> in <code>cloneFile</code> .
...	Further arguments to pass to internal <code>saveObject</code> calls.

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

<code>x</code>	A SpatialImage object.
<code>dir</code>	String containing a path to a directory.
<code>path</code>	String containing a relative path inside a directory.
<code>child</code>	Logical scalar indicating whether <code>x</code> is a child of another object.
<code>...</code>	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:

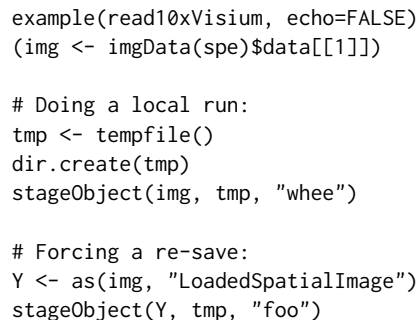
- `$schema`, a string specifying the schema to use to validate the metadata. This may have a `package` attribute to specify the package where the schema lives (in its `inst/schemas` directory).
- `path`, a string containing the path to the file containing the assay contents. This should start with the 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 code block contains two examples of using the stageSpatialImage function. The first example shows a local run where a temporary directory is created and the function is called with 'whee' as the path. The second example shows forcing a re-save by converting the image to a 'LoadedSpatialImage' object and then calling stageObject with a new path 'foo'.  
(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

cloneFile, 3

imgData, 3, 4

imgSource, 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