

Package ‘alabaster.mae’

September 20, 2023

Title Load and Save MultiAssayExperiments

Version 1.0.0

Date 2023-02-20

License MIT + file LICENSE

Description Save MultiAssayExperiments 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 MultiAssayExperiment, alabaster.base

Imports methods, alabaster.se, S4Vectors

Suggests testthat, knitr, SummarizedExperiment, BiocParallel,
BiocStyle, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.2.3

biocViews DataImport, DataRepresentation

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

git_branch RELEASE_3_17

git_last_commit acbca50

git_last_commit_date 2023-04-25

Date/Publication 2023-09-20

Author Aaron Lun [aut, cre]

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

R topics documented:

loadMultiAssayExperiment	2
stageObject,MultiAssayExperiment-method	3
Index	5

`loadMultiAssayExperiment`*Load a MultiAssayExperiment*

Description

Load a dataset as a [MultiAssayExperiment](#), based on the metadata generated by the corresponding [stageObject](#) method.

Usage

```
loadMultiAssayExperiment(  
  ds.info,  
  project,  
  experiments = NULL,  
  BPPARAM = NULL,  
  include.nested = TRUE  
)
```

Arguments

<code>ds.info</code>	Named list containing the metadata for this object.
<code>project</code>	Any argument accepted by the acquisition functions, see ?acquireFile . By default, this should be a string containing the path to a staging directory.
<code>experiments</code>	Character or integer vector specifying the subset of experiments to load. If NULL, all experiments are loaded.
<code>BPPARAM</code>	A BiocParallelParam object indicating how loading should be parallelized across multiple experiments. If NULL, loading is done serially.
<code>include.nested</code>	Logical scalar indicating whether to include nested DataFrames in the <code>colData</code> of the output.

Value

A [MultiAssayExperiment](#) object.

Author(s)

Aaron Lun

Examples

```
library(SummarizedExperiment)  
  
# Mocking up an MAE  
mat <- matrix(rnorm(1000), ncol=10)  
colnames(mat) <- letters[1:10]  
rownames(mat) <- sprintf("GENE_%i", seq_len(nrow(mat)))
```

```
se <- SummarizedExperiment(list(counts=mat))

library(MultiAssayExperiment)
mae <- MultiAssayExperiment(list(gene=se))

# Staging it:
tmp <- tempfile()
dir.create(tmp)
info <- stageObject(mae, tmp, "dataset")

# Loading it back in:
loadMultiAssayExperiment(info, tmp)
```

stageObject,MultiAssayExperiment-method
Stage a dataset

Description

Save the metadata and annotations of a [MultiAssayExperiment](#) in a staging directory.

Usage

```
## S4 method for signature 'MultiAssayExperiment'
stageObject(
  x,
  dir,
  path,
  child = FALSE,
  sm.name = "sample_mapping",
  sd.name = "sample_data",
  meta.name = "dataset.json"
)
```

Arguments

x	A MultiAssayExperiment object or one of its subclasses.
dir	String containing the path to the staging directory.
path	String containing a prefix of the relative path inside dir where x is to be saved. The actual path used to save x may include additional components, see Details .
child	Logical scalar indicating whether x is a child of a larger object.
sm.name	String containing the prefix of the sample mapping file.
sd.name	String containing the prefix of the sample data file.
meta.name	String containing the name of the metadata file.

Details

`meta.name` is only needed to set up the output path, for consistency with the `stageObject` contract. Callers should make sure to write the metadata to the same document by using `.writeMetadata` with `meta.only=TRUE`.

Value

A named list containing the metadata for this dataset. The contents of `x` are saved into a path subdirectory inside `dir`.

Author(s)

Aaron Lun

Examples

```
# Mocking up an MAE
mat <- matrix(rnorm(1000), ncol=10)
colnames(mat) <- letters[1:10]
rownames(mat) <- sprintf("GENE_%i", seq_len(nrow(mat)))

library(SummarizedExperiment)
se <- SummarizedExperiment(list(counts=mat))

library(MultiAssayExperiment)
mae <- MultiAssayExperiment(list(gene=se))

# Staging it:
tmp <- tempfile()
dir.create(tmp)
stageObject(mae, tmp, "dataset")
```

Index

`.writeMetadata`, [4](#)
`acquireFile`, [2](#)
`loadMultiAssayExperiment`, [2](#)
`MultiAssayExperiment`, [2](#), [3](#)
`stageObject`, [2](#), [4](#)
`stageObject`, `MultiAssayExperiment`-method,
[3](#)
`stageObject`, `SampleMapFrame`-method
(`stageObject`, `MultiAssayExperiment`-method),
[3](#)