

# Package ‘alabaster.bumpy’

September 29, 2023

**Title** Save and Load BumpyMatrices to/from file

**Version** 1.0.0

**Date** 2023-02-28

**Description** Save BumpyMatrix objects 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.

**License** MIT + file LICENSE

**Depends** BumpyMatrix, alabaster.base

**Imports** methods, Matrix, BiocGenerics, S4Vectors, IRanges

**Suggests** BiocStyle, rmarkdown, knitr, testthat, jsonlite

**VignetteBuilder** knitr

**RoxygenNote** 7.2.3

**biocViews** DataImport, DataRepresentation

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

**git\_branch** RELEASE\_3\_17

**git\_last\_commit** da9aa0c

**git\_last\_commit\_date** 2023-04-25

**Date/Publication** 2023-09-29

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

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

## R topics documented:

loadBumpyDataFrameMatrix . . . . .	2
stageObject,BumpyDataFrameMatrix-method . . . . .	3
useBumpyHDF5 . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

`loadBumpyDataFrameMatrix`*Load a BumpyDataFrameMatrix object*

---

### Description

Load a [BumpyDataFrameMatrix](#) object based on its metadata.

### Usage

```
loadBumpyDataFrameMatrix(assay.info, project)
```

### Arguments

<code>assay.info</code>	Named list of metadata for the <a href="#">BumpyDataFrameMatrix</a> , generated by the corresponding <a href="#">stageObject</a> method.
<code>project</code>	Any argument accepted by the acquisition functions, see <a href="#">?acquireFile</a> . By default, this should be a string containing the path to a staging directory.

### Value

A [BumpyDataFrameMatrix](#) object.

### Author(s)

Aaron Lun

### Examples

```
# Mocking up a BumpyMatrix.
library(BumpyMatrix)
library(S4Vectors)
df <- DataFrame(x=runif(100), y=runif(100))
f <- factor(sample(letters[1:20], nrow(df), replace=TRUE), letters[1:20])
out <- S4Vectors::split(df, f)
mat <- BumpyMatrix(out, c(5, 4))

# Staging it:
tmp <- tempfile()
dir.create(tmp)
meta <- stageObject(mat, tmp, "BUMPY")

# Loading it:
loadBumpyDataFrameMatrix(meta, tmp)
```

---

stageObject,BumpyDataFrameMatrix-method  
*Stage BumpyDataFrameMatrix objects*

---

### Description

Staging method for [BumpyDataFrameMatrix](#) objects. See [?stageObject](#) for more details.

### Usage

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

### Arguments

x	A <a href="#">BumpyDataFrameMatrix</a> object.
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 <a href="#">Details</a> .
child	Logical scalar indicating whether x is a child of a larger object.

### Value

A named list of metadata, to be used by [.writeMetadata](#).

### Author(s)

Aaron Lun

### See Also

[useBumpyHDF5](#), to control whether the concatenated data frame is saved as a CSV or HDF5 file.

---

useBumpyHDF5	<i>Save BumpyMatrix data to HDF5</i>
--------------	--------------------------------------

---

### Description

Use HDF5 for the underlying data frame, i.e., obtained after [unlisting](#) the [BumpyMatrix](#). This is less intuitive than a CSV but preserves the precision of floating-point numbers.

### Usage

```
useBumpyHDF5(use)
```

**Arguments**

`use` Logical scalar indicating whether to save in HDF5.

**Value**

If `use` is missing, a logical scalar is returned indicating whether data should be saved in HDF5.

If `use` is provided, it is used to set the corresponding flag globally. The previous value of the flag is returned invisibly.

**Examples**

```
useBumpyHDF5()  
  
old <- useBumpyHDF5(FALSE)  
useBumpyHDF5()  
  
# Setting it back.  
useBumpyHDF5(old)
```

# Index

`.writeMetadata`, 3  
`acquireFile`, 2  
`BumpyDataFrameMatrix`, 2, 3  
`BumpyMatrix`, 3  
`loadBumpyDataFrameMatrix`, 2  
`stageObject`, 2, 3  
`stageObject`, `BumpyDataFrameMatrix`-method,  
3  
`unlist`, 3  
`useBumpyHDF5`, 3, 3