

# Package ‘alabaster.bumpy’

May 8, 2024

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

**Version** 1.4.0

**Date** 2024-01-01

**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, rhdf5, 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\_19

**git\_last\_commit** 31e3931

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-05-08

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

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

## Contents

|  |   |
|--|---|
| readBumpyDataFrameMatrix . . . . .               | 2 |
| saveObject,BumpyDataFrameMatrix-method . . . . . | 3 |
| useBumpyHDF5 . . . . .                           | 4 |

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

readBumpyDataFrameMatrix

*Read a BumpyDataFrameMatrix from disk*

---

## Description

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

## Usage

```
readBumpyDataFrameMatrix(path, metadata, ...)
```

## Arguments

|          |   |
|----------|---|
| path     | String containing a path to a directory, itself created using the <a href="#">saveObject</a> method for <a href="#">BumpyDataFrameMatrix</a> objects. |
| metadata | Named list of metadata for this object, see <a href="#">readObjectFile</a> for details.   |
| ...      | Further arguments passed to internal <a href="#">altReadObject</a> calls.   |

## 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))

# Saving it:
tmp <- tempfile()
saveObject(mat, tmp)

# Reading it:
readBumpyDataFrameMatrix(tmp)
```

---

`saveObject,BumpyDataFrameMatrix-method`*Save a BumpyDataFrameMatrix to disk*

---

## Description

Save a [BumpyDataFrameMatrix](#) to its on-disk representation.

## Usage

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

## Arguments

|                   |   |
|-------------------|---|
| <code>x</code>    | A <a href="#">BumpyDataFrameMatrix</a> object.                              |
| <code>path</code> | String containing the path to a directory in which to save <code>x</code> . |
| <code>...</code>  | Further arguments to pass to specific methods.                              |

## Value

`x` is saved into `path` and `NULL` is invisibly returned.

## 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))  
  
# Saving it:  
tmp <- tempfile()  
saveObject(mat, tmp)
```

---

`useBumpyHDF5`*Save BumpyMatrix data to HDF5*

---

**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

`altReadObject`, [2](#)

`BumpyDataFrameMatrix`, [2](#), [3](#)

`BumpyMatrix`, [4](#)

`loadBumpyDataFrameMatrix`  
(`readBumpyDataFrameMatrix`), [2](#)

`readBumpyDataFrameMatrix`, [2](#)

`readObjectFile`, [2](#)

`saveObject`, [2](#)

`saveObject`, `BumpyDataFrameMatrix`-method,  
[3](#)

`stageObject`, `BumpyDataFrameMatrix`-method  
(`saveObject`, `BumpyDataFrameMatrix`-method),  
[3](#)

`unlist`, [4](#)

`useBumpyHDF5`, [4](#)