

# Package ‘beachmat.hdf5’

May 15, 2024

**Version** 1.2.0

**Date** 2023-08-07

**Title** beachmat bindings for HDF5-backed matrices

**Description** Extends beachmat to support initialization of tatami matrices from HDF5-backed arrays. This allows C++ code in downstream packages to directly call the HDF5 C/C++ library to access array data, without the need for block processing via DelayedArray. Some utilities are also provided for direct creation of an in-memory tatami matrix from a HDF5 file.

**Encoding** UTF-8

**Imports** methods, beachmat, HDF5Array, DelayedArray, Rcpp

**Suggests** testthat, BiocStyle, knitr, rmarkdown, rhdf5, Matrix

**LinkingTo** Rcpp, beachmat, Rhdf5lib

**biocViews** DataRepresentation, DataImport, Infrastructure

**License** GPL-3

**NeedsCompilation** yes

**VignetteBuilder** knitr

**SystemRequirements** C++17, GNU make

**RoxygenNote** 7.2.3

**git\_url** <https://git.bioconductor.org/packages/beachmat.hdf5>

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** 4e2ef7a

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

**Repository** Bioconductor 3.19

**Date/Publication** 2024-05-15

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

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

## Contents

initializeCpp . . . . .	2
loadIntoMemory . . . . .	3
<b>Index</b>	<b>4</b>

---

initializeCpp	<i>Initialize HDF5-backed matrices.</i>
---------------	---

---

### Description

Initialize C++ representations of HDF5-backed matrices based on their **HDF5Array** representations.

### Usage

```
## S4 method for signature 'H5SparseMatrixSeed'
initializeCpp(x, ..., memorize = FALSE)
```

```
## S4 method for signature 'HDF5ArraySeed'
initializeCpp(x, ..., memorize = FALSE)
```

### Arguments

x	A <b>HDF5Array</b> seed object.
...	Further arguments, ignored.
memorize	Logical scalar specifying whether to load the matrix data in x into memory, if it has not already been loaded. See <a href="#">checkMemoryCache</a> for details.

### Value

An external pointer that can be used in any **tatami**-compatible function.

### Author(s)

Aaron Lun

### Examples

```
library(HDF5Array)
y <- matrix(runif(1000), ncol=20, nrow=50)
z <- as(y, "HDF5Array")
ptr <- initializeCpp(z)
```

---

loadIntoMemory	<i>Load a HDF5 matrix into memory</i>
----------------	---------------------------------------

---

### Description

Load a HDF5-backed matrix into memory as an external pointer to a **tatami**-compatible representation. This differs from the (default) behavior of `initializeCpp`, which only loads slices of the matrix on request.

### Usage

```
loadIntoMemory(x, force.integer = FALSE)
```

### Arguments

<code>x</code>	A <b>HDF5Array</b> -derived matrix or seed object.
<code>force.integer</code>	Whether to force floating-point values to be integers to reduce memory consumption.

### Value

An external pointer that can be used in **tatami**-based functions.

### Author(s)

Aaron Lun

### Examples

```
library(HDF5Array)
y <- matrix(runif(1000), ncol=20, nrow=50)
z <- as(y, "HDF5Array")
ptr <- loadIntoMemory(z)
```

# Index

`checkMemoryCache`, [2](#)

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

`initializeCpp,H5SparseMatrixSeed-method`  
(`initializeCpp`), [2](#)

`initializeCpp,HDF5ArraySeed-method`  
(`initializeCpp`), [2](#)

`loadIntoMemory`, [3](#)