# Package ‘BufferedMatrix’

May 15, 2024

**Version** 1.69.0

**Title** A matrix data storage object held in temporary files

**Author** Ben Bolstad <bmb@bmbolstad.com>

**Maintainer** Ben Bolstad <bmb@bmbolstad.com>

**Depends** R (>= 2.6.0), methods

**Description** A tabular style data object where most data is stored outside main memory. A buffer is used to speed up access to data.

**License** LGPL (>= 2)

**URL** https://github.com/bmbolstad/BufferedMatrix

**Collate** allGenerics.R BufferedMatrix.R as.BufferedMatrix.R createBufferedMatrix.R

**LazyLoad** yes

**biocViews** Infrastructure

**git_url** https://git.bioconductor.org/packages/BufferedMatrix

**git_branch** devel

**git_last_commit** d422a05

**git_last_commit_date** 2024-04-30

**Repository** Bioconductor 3.20

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

## Contents

## as.BufferedMatrix *Check or Coerce object to BufferedMatrix*

### Description

'as.BufferedMatrix' will coerce the supplied object into a BufferedMatrix. 'is.BufferedMatrix' checks whether the supplied argument is a BufferedMatrix.

### Usage

```
as.BufferedMatrix(x, bufferrows=1, buffercols=1,directory=getwd())
is.BufferedMatrix(x)
```

### Arguments

| | |
|---|---|
| x | an R object |
| bufferrows | number of rows to be buffered if the row buffer is activated |
| buffercols | number of columns to be buffered |
| directory | path to directory where temporary files should be stored |

### Details

These functions are useful for converting between R `matrix` objects and `BufferedMatrix` objects.

### Author(s)

B. M. Bolstad <bmb@bmbolstad.com>

## BufferedMatrix-class *Class BufferedMatrix*

### Description

This is a class representation of a buffered matrix (of numeric data). In this case data is primarily stored outide main memory in temporary files.

### Objects from the Class

Objects can be created using the function `createBufferedMatrix`

### Slots

`rawBufferedMatrix`**:** a pointer to an external structure used to access and store the matrix data.

`rownames`**:** rownames for the matrix.

`colnames`**:** colnames for the matrix.

## Methods

**ncol** `signature(object = "BufferedMatrix")`: Returns the number of columns in the matrix

**nrow** `signature(object = "BufferedMatrix")`: Returns the number of rows in the matrix

**dim** `signature(object = "BufferedMatrix")`: Returns the dimensions of the matrix

**buffer.dim** `signature(object = "BufferedMatrix")`: Returns the number of columns and the number of rows to be stored in the buffer

**set.buffer.dim** `signature(object = "BufferedMatrix")`: Set the buffer size or resize it

**[** `signature(object = "BufferedMatrix")`: matrix accessor

**[<-** `signature(object = "BufferedMatrix")`: matrix replacer

**show** `signature(object = "BufferedMatrix")`: prints basic information about the BufferedMatrix out to screen

**is.RowMode** `signature(object = "BufferedMatrix")`: returns `TRUE` if the row buffer is active and `FALSE` otherwise.

**is.ColMode** `signature(object = "BufferedMatrix")`: returns `TRUE` if the row buffer is inactive and `FALSE` otherwise.

**RowMode** `signature(object = "BufferedMatrix")`: Activate the row buffer.

**ColMode** `signature(object = "BufferedMatrix")`: Deactivate the row buffer

**duplicate** `signature(object = "BufferedMatrix")`: Make a copy of the BufferedMatrix

**prefix** `signature(object = "BufferedMatrix")`: return the initial part of the string used for temporary files

**directory** `signature(object = "BufferedMatrix")`: return the location where temporary files are stored

**filenames** `signature(object = "BufferedMatrix")`: return the fully pathed filenames for each column in the matrix

**ewApply** `signature(object = "BufferedMatrix")`: apply a function elementwise

**exp** `signature(object = "BufferedMatrix")`: Compute the exponential elementwise of the matrix

**sqrt** `signature(object = "BufferedMatrix")`: Compute the square-root elementwise of the matrix

**pow** `signature(object = "BufferedMatrix")`: Compute $x^power$ elementwise of the matrix

**log** `signature(object = "BufferedMatrix")`: Compute logarithm elementwise of the matrix

**colMax** `signature(object = "BufferedMatrix")`: Returns a vector containing maximums by column

**rowMax** `signature(object = "BufferedMatrix")`: Returns a vector containing maximums by row

**colMeans** `signature(object = "BufferedMatrix")`: Returns a vector containing means by column

**rowMeans** `signature(object = "BufferedMatrix")`: Returns a vector containing means by row

**colMin** `signature(object = "BufferedMatrix")`: Returns a vector containing minimums by column

**rowMin** `signature(object = "BufferedMatrix")`: Returns a vector containing minimums by row

**colVars** `signature(object = "BufferedMatrix")`: Returns a vector containing sample variances by column

**rowVars** `signature(object = "BufferedMatrix")`: Returns a vector containing sample variances by row

**colSd** `signature(object = "BufferedMatrix")`: Returns a vector containing sample standard deviations by column

**rowSd** `signature(object = "BufferedMatrix")`: Returns a vector containing sample standard deviations by row

**colSums** `signature(object = "BufferedMatrix")`: Returns a vector containing sum by column

**rowSums** `signature(object = "BufferedMatrix")`: Returns a vector containing sum by row

**colMedians** `signature(object = "BufferedMatrix")`: Returns a vector containing medians by column

**rowMedians** `signature(object = "BufferedMatrix")`: Returns a vector containing medians by row. Best only used when the matrix is in RowMode (otherwise it is extremely slow)

**Max** `signature(object = "BufferedMatrix")`: Returns the maximum of all elements in the matrix

**Min** `signature(object = "BufferedMatrix")`: Returns the minimum of all elements in the matrix

**Var** `signature(object = "BufferedMatrix")`: Returns the sample variance of all elements in the matrix

**Sd** `signature(object = "BufferedMatrix")`: Returns the sample standard deviations of all elements in the matrix

**Sum** `signature(object = "BufferedMatrix")`: Returns the sum of all elements in the matrix

**mean** `signature(object = "BufferedMatrix")`: Returns the mean of all elements in the matrix

**colApply** `signature(object = "BufferedMatrix")`: apply a function columnwise. Returns either a vector or BufferedMatrix.

**rowApply** `signature(object = "BufferedMatrix")`: apply a function row-wise. Returns either a vector or BufferedMatrix.

**as.matrix** `signature(object = "BufferedMatrix")`: coerce BufferedMatrix into a regular R `matrix`

**subBufferedMatrix** `signature(object = "BufferedMatrix")`: gets data from BufferedMatrix and returns it in another BufferedMatrix

**rownames** `signature(object = "BufferedMatrix")` : access the row names

**colnames** `signature(object = "BufferedMatrix")` : access the column names

**rownames<-** `signature(object = "BufferedMatrix")` : replace the row names

**colnames<-** `signature(object = "BufferedMatrix")` : replace the column names

**dimnames** `signature(object = "BufferedMatrix")` : Access the row and column names

**dimnames** `signature(object = "BufferedMatrix")` : Replace the row and column names

**ReadOnlyMode** `signature(object = "BufferedMatrix")` : Toggles the Read Only mode on and off

**is.ReadOnlyMode** `signature(object = "BufferedMatrix")` : Finds out if it is in Read Only Mode

**memory.usage** `signature(object = "BufferedMatrix")` : Give amount of RAM currently in use by BufferedMatrix object

**disk.usage** `signature(object = "BufferedMatrix")` : Give amount of disk space currently in use by BufferedMatrix object

`as(matrix,BufferedMatrix)`**:** Coerce `matrix` to `BufferedMatrix`.

`as(BufferedMatrix,matrix)`**:** Coerce the `Buffered` to `matrix`.

`AddColumn`**:** Add an additional column to the matrix. Will be all empty (set to 0)

`MoveStorageDirectory`**:** Move the temporary files used to store the matrix from one location to another

## Author(s)

B. M. Bolstad <bmb@bmbolstad.com>

---

`createBufferedMatrix` *createBufferedMatrix*

---

## Description

Creates a Buffered Matrix object

## Usage

```
createBufferedMatrix(rows, cols=0, bufferrows=1, buffercols=1,prefix="BM",directory=getwd())
```

## Arguments

| | |
|---|---|
| `rows` | Number of rows in the matrix |
| `cols` | Initial number of coulmns in the matrix |
| `bufferrows` | number of rows to be buffered if the row buffer is activated |
| `buffercols` | number of columns to be buffered |
| `prefix` | String to be used as start of name for any temporary files |
| `directory` | path to directory where temporary files should be stored |

## Author(s)

B. M. Bolstad <bmb@bmbolstad.com>

# Index