

# Package ‘bigalgebra’

February 19, 2015

**Version** 0.8.4

**Date** 2014-04-15

**Title** BLAS routines for native R matrices and big.matrix objects.

**Author** Michael J. Kane, Bryan Lewis, and John W. Emerson

**Maintainer** Michael J. Kane <bigmemoryauthors@gmail.com>

**Imports** methods

**Depends** bigmemory (>= 4.0.0)

**LinkingTo** bigmemory, BH

**Description** This package provides arithmetic functions for R matrix and big.matrix objects.

**License** LGPL-3 | Apache License 2.0

**Copyright** (C) 2014 Michael J. Kane, Bryan Lewis, and John W. Emerson

**URL** <http://www.bigmemory.org>

**LazyLoad** yes

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2014-04-16 09:21:50

## R topics documented:

bigalgebra-package . . . . .	2
balgebra-methods . . . . .	3
daxpy . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

bigalgebra-package      *Arithmetic routines for native R matrices and big.matrix objects.*

---

## Description

This package provides arithmetic functions for native R matrices and `big.matrix` objects.

## Details

This package provides arithmetic functions for native R matrices and `big.matrix` objects.

The package defines a number of global options that begin with `bigalgebra`. They include:

Option	Default value
<code>bigalgebra.temp_pattern</code>	"matrix_"
<code>bigalgebra.tempdir</code>	<code>tempdir</code>
<code>bigalgebra.mixed_arithmetic_returns_R_matrix</code>	TRUE
<code>bigalgebra.DEBUG</code>	FALSE

The `bigalgebra.tempdir` option must be a function that returns a temporary directory path used to big matrix results of BLAS and LAPACK operations. The default value is simply the default R `tempdir` function.

The `bigalgebra.temp_pattern` is a name prefix for file names of generated big matrix objects output as a result of BLAS and LAPACK operations.

The `bigalgebra.mixed_arithmetic_returns_R_matrix` option determines whether arithmetic operations involving an R matrix or vector and a `big.matrix` matrix or vector return a big matrix (when the option is FALSE), or return a normal R matrix (TRUE).

The package is built, by default, with R's native BLAS libraries, which use 32-bit signed integer indexing. The default build is limited to vectors of at most  $2^{31} - 1$  entries and matrices with at most  $2^{31} - 1$  rows and  $2^{31} - 1$  columns (note that standard R matrices are limited to  $2^{31} - 1$  total entries).

The package includes a reference BLAS implementation that supports 64-bit integer indexing, relaxing the limitation on vector lengths and matrix row and column limits. Installation of this package with the 64-bit reference BLAS implementation may be performed from the command-line install:

```
REFBLAS=1 R CMD INSTALL bigalgebra
```

where "bigalgebra" is the source package (for example, `bigalgebra_0.8.4.tar.gz`).

The package may also be build with user-supplied external BLAS and LAPACK libraries, in either 32- or 64-bit varieties. This is an advanced topic that requires additional Makevars modification, and may include adjustment of the low-level calling syntax depending on the library used.

Feel free to contact us for help installing and running the package.

**Author(s)**

Michael J. Kane, Bryan Lewis, John W. Emerson

Maintainers: Mike and Jay <bigmemoryauthors@gmail.com>

**References**

<http://www.netlib.org/blas/>

<http://www.netlib.org/lapack/>

**See Also**

[bigmemory](#), [big.matrix](#)

**Examples**

```
# Testing the development of the user-friendly operators:
# if you have any problems, please email us! - Jay & Mike 4/29/2010

library("bigmemory")
A <- big.matrix(5,4,init=1)
B <- big.matrix(4,4,init=2)

C <- A
D <- A[]

print(C - D)      # Compare the results (subtraction of an R matrix from a
                  # big.matrix)

# The next example illustrates mixing R and big.matrix objects. It returns by
# default (see # options("bigalgebra.mixed_arithmetic_returns_R_matrix"))
D <- matrix(rnorm(16),4)
E <- A
```

---

balgebra-methods

*Class "big.matrix" arithmetic methods*

---

**Description**

Arithmetic operations for big.matrices

**Methods**

**%\*%** signature{x="big.matrix", y="big.matrix"}: ...

**%\*%** signature{x="matrix", y="big.matrix"}: ...

**%\*%** signature{x="big.matrix", y="matrix"}: ...

**Arith** signature{x="big.matrix", y="big.matrix"}: ...

**Arith** signature{x="big.matrix", y="matrix"}: ...

**Arith** signature{x="matrix", y="big.matrix"}: ...

**Arith** signature{x="big.matrix", y="numeric"}: ...

**Arith** signature{x="numeric", y="big.matrix"}: ...

## Notes

Miscellaneous arithmetic methods for matrices and big.matrices. See also options("bigalgebra.mixed\_arithmetic\_retu

## Author(s)

B. W. Lewis <blewis@illposed.net>

---

daxpy

*BLAS daxpy functionality*

---

## Description

This function implements the function  $Y := A * X + Y$  where  $X$  and  $Y$  may be either native double-precision valued  $\mathbb{R}$  matrices or numeric vectors, or double-precision valued `big.matrix` objects, and  $A$  is a scalar.

## Usage

daxpy(A=1, X, Y)

## Arguments

A	Optional numeric scalar value to scale the matrix $X$ by, with a default value of 1.
X	Required to be either a native $\mathbb{R}$ <code>matrix</code> or numeric vector, or a <code>big.matrix</code> object
Y	Optional native $\mathbb{R}$ <code>matrix</code> or numeric vector, or a <code>big.matrix</code> object

## Details

At least one of either  $X$  or  $Y$  must be a `big.matrix`. All values must be of type double (the only type presently supported by the `bigalgebra` package).

This function is rarely necessary to use directly since the `bigalgebra` package defines standard arithmetic operations and scalar multiplication. It is more efficient to use `daxpy` directly when both scaling and matrix addition are required, in which case both operations are performed in one step.

**Value**

The output value depends on the classes of input values X and Y and on the value of the global option `bigalgebra.mixed_arithmetic_returns_R_matrix`.

If X and Y are both big matrices, or Y is missing, `options("bigalgebra.mixed_arithmetic_returns_R_matrix")` is FALSE, then a `big.matrix` is returned. The returned `big.matrix` is backed by a temporary file mapping that will be deleted when the returned result is garbage collected by R (see the examples).

Otherwise, a standard R matrix is returned. The dimensional shape of the output is taken from X. If input X is dimensionless (that is, lacks a dimension attribute), then the output is a column vector.

**Author(s)**

Michael J. Kane

**References**

<http://www.netlib.org/blas/daxpy.f>

**See Also**

[bigmemory](#)

**Examples**

```
require(bigmemory)
A = matrix(1, nrow=3, ncol=2)
B = big.matrix(nrow=3, ncol=2, type='double', init=1)
C = B + B # C is a new big matrix
D = A + B # D defaults to a regular R matrix, to change this, set the option:
          # options(bigalgebra.mixed_arithmetic_returns_R_matrix=FALSE)
E = daxpy(A=1.0, X=B, Y=B) # Same kind of result as C
print(C[])
print(D)
print(E[])

# The C and E big.matrix file backings will be deleted when garbage collected:
# (We enable debugging to see this explicitly)
options(bigalgebra.DEBUG=TRUE)
rm(C,E)
gc()
```

# Index

## \*Topic **package**

- bigalgebra-package, [2](#)
- %%,big.matrix, big.matrix-method  
(balgebra-methods), [3](#)
- %%,big.matrix, matrix-method  
(balgebra-methods), [3](#)
- %%,matrix, big.matrix-method  
(balgebra-methods), [3](#)
  
- Arith, big.matrix, big.matrix-method  
(balgebra-methods), [3](#)
- Arith, big.matrix, matrix-method  
(balgebra-methods), [3](#)
- Arith, big.matrix, numeric-method  
(balgebra-methods), [3](#)
- Arith, matrix, big.matrix-method  
(balgebra-methods), [3](#)
- Arith, numeric, big.matrix-method  
(balgebra-methods), [3](#)
  
- balgebra-methods, [3](#)
- big.matrix, [2-4](#)
- bigalgebra (bigalgebra-package), [2](#)
- bigalgebra-package, [2](#)
- bigmemory, [3, 5](#)
  
- daxpy, [4](#)
  
- matrix, [4](#)