# Package 'matter'

October 18, 2017

Title A framework for rapid prototyping with binary data on disk

Type Package

Version 1.2.0

Author Kylie A. Bemis <k. bemis@northeastern.edu="">  Maintainer Kylie A. Bemis <k. bemis@northeastern.edu="">  Description Memory-efficient reading, writing, and manipulation of structured binary data on disk as vectors, matrices, and arrays.  License Artistic-2.0  Depends methods, stats, biglm  Imports BiocGenerics, irlba, utils  Suggests BiocStyle, testthat  Collate matterGenerics.R utils.R drle.R atoms.R matter.R matter_vec.R matter_mat.R stats.R apply.R bigglm.R prcomp.R  biocViews Software, Infrastructure  URL https://github.com/kuwisdelu/matter  NeedsCompilation yes  R topics documented:  apply</k.></k.>	Date 2016-10-11
Description Memory-efficient reading, writing, and manipulation of structured binary data on disk as vectors, matrices, and arrays.  License Artistic-2.0  Depends methods, stats, bigIm  Imports BiocGenerics, irlba, utils  Suggests BiocStyle, testthat  Collate matterGenerics.R utils.R drle.R atoms.R matter.R matter_vec.R matter_mat.R stats.R apply.R biggIm.R prcomp.R  biocViews Software, Infrastructure  URL https://github.com/kuwisdelu/matter  NeedsCompilation yes  R topics documented:  apply 2 biggIm 3 delayed-ops 4 drle-class 5 matter_class 5 matter_ex-data 8 matter_mat-class 8 matter_mat-class 8 matter_mat-class 9 matter_wec-class 9 matter_vec-class 9 matter_vec-class 9 matter_vec-class 9 more more more matter_vec-class 9 more more matter_vec-class 9 more more more more more more more more	Author Kylie A. Bemis <k.bemis@northeastern.edu></k.bemis@northeastern.edu>
structured binary data on disk as vectors, matrices, and arrays.  License Artistic-2.0  Depends methods, stats, bigIm  Imports BiocGenerics, irlba, utils  Suggests BiocStyle, testthat  Collate matterGenerics.R utils.R drle.R atoms.R matter.R matter_vec.R matter_mat.R stats.R apply.R biggIm.R prcomp.R  biocViews Software, Infrastructure  URL https://github.com/kuwisdelu/matter  NeedsCompilation yes  R topics documented:  apply 2 biggIm 3 delayed-ops 4 drle-class 5 matter-class 5 matter_ex-data 8 matter_mat-class 8 matter_mat-class 8 matter_wec-class 9 matter_vec-class 9 matter_	Maintainer Kylie A. Bemis <k.bemis@northeastern.edu></k.bemis@northeastern.edu>
Depends methods, stats, bigIm Imports BiocGenerics, irlba, utils Suggests BiocStyle, testthat Collate matterGenerics.R utils.R drle.R atoms.R matter.R matter_vec.R matter_mat.R stats.R apply.R biggIm.R prcomp.R biocViews Software, Infrastructure URL https://github.com/kuwisdelu/matter NeedsCompilation yes  R topics documented:  apply	
Imports BiocGenerics, irlba, utils  Suggests BiocStyle, testthat  Collate matterGenerics.R utils.R drle.R atoms.R matter.R matter_vec.R matter_mat.R stats.R apply.R bigglm.R prcomp.R  biocViews Software, Infrastructure  URL https://github.com/kuwisdelu/matter  NeedsCompilation yes  R topics documented:  apply 2 bigglm 3 delayed-ops 4 drle-class 5 matter-class 6 matter_ex-data 8 matter_ex-data 8 matter_mat-class 8 matter_wec-class 10 prcomp 12 scale 13 summary-stats 14	License Artistic-2.0
Suggests BiocStyle, testthat  Collate matterGenerics.R utils.R drle.R atoms.R matter.R matter_vec.R matter_mat.R stats.R apply.R bigglm.R prcomp.R  biocViews Software, Infrastructure  URL https://github.com/kuwisdelu/matter  NeedsCompilation yes  R topics documented:  apply	<b>Depends</b> methods, stats, biglm
Collate matterGenerics.R utils.R drle.R atoms.R matter_R matter_vec.R matter_mat.R stats.R apply.R bigglm.R prcomp.R biocViews Software, Infrastructure  URL https://github.com/kuwisdelu/matter NeedsCompilation yes  R topics documented:  apply	Imports BiocGenerics, irlba, utils
matter_mat.R stats.R apply.R bigglm.R prcomp.R biocViews Software, Infrastructure  URL https://github.com/kuwisdelu/matter NeedsCompilation yes  R topics documented:  apply 2 bigglm 3 delayed-ops 4 drle-class 5 matter-class 6 matter_ex-data 8 matter_mat-class 8 matter_wec-class 10 prcomp 12 scale 13 summary-stats 14	Suggests BiocStyle, testthat
URL https://github.com/kuwisdelu/matter NeedsCompilation yes  R topics documented:  apply	
NeedsCompilation yes         R topics documented:         apply       2         bigglm       3         delayed-ops       4         drle-class       5         matter-class       6         matter_ex-data       8         matter_mat-class       8         matter_vec-class       10         prcomp       12         scale       13         summary-stats       14	biocViews Software, Infrastructure
bigglm       3         delayed-ops       4         drle-class       5         matter-class       6         matter_ex-data       8         matter_mat-class       8         matter_vec-class       10         prcomp       12         scale       13         summary-stats       14	NeedsCompilation yes
Index 16	bigglm       3         delayed-ops       4         drle-class       5         matter-class       6         matter_ex-data       8         matter_mat-class       8         matter_vec-class       10         prcomp       12         scale       13
	Index 16

2 apply

apply

Apply Functions Over "matter" Matrices

# Description

An implementation of apply for matter\_mat matrices.

# Usage

```
## S4 method for signature 'matter_mat'
apply(X, MARGIN, FUN, ...)
```

# Arguments

X A matter\_mat object.

MARGIN Must be 1 or 2 for matter\_mat matrices, where '1' indicates rows and '2' indi-

cates columns. The dimension names can also be used if X has dimnames set.

FUN The function to be applied.

... Additional arguments to be passed to FUN.

#### **Details**

Because FUN must be executed by the interpreter in the appropriate R environment, the full row or column will be loaded into memory. The chunksize of X is ignored. For summary statistics, functions like colMeans, matter\_mat-method and rowMeans, matter\_mat-method offer greater control over memory pressure.

## Value

See apply for details.

### Author(s)

Kylie A. Bemis

#### See Also

apply

```
x \leftarrow matter(1:100, nrow=10, ncol=10)
apply(x, 2, summary)
```

bigglm 3

	bigglm	Using "biglm" with "matter"	
--	--------	-----------------------------	--

#### **Description**

This method allows matter\_mat matrices to be used with the bigglm function from the "biglm" package.

#### Usage

```
## S4 method for signature 'formula,matter_mat'
bigglm(formula, data, ..., chunksize = NULL, fc = NULL)
```

#### **Arguments**

formula A model formula.

data A matter matrix with column names.

chunksize An integer giving the maximum number of rows to process at a time. If left

NULL, this will be calculated by dividing the chunksize of data by the number

of variables in the formula.

fc Either column indices or names of variables which are factors.

... Additional options passed to bigglm.

### Value

An object of class bigglm.

# Author(s)

Kylie A. Bemis

#### See Also

bigglm

```
set.seed(1)

x <- matter_mat(rnorm(1000), nrow=100, ncol=10)

colnames(x) <- c(paste0("x", 1:9), "y")

fm <- paste0("y ~ ", paste0(paste0("x", 1:9), collapse=" + "))
fm <- as.formula(fm)

fit <- bigglm(fm, data=x, chunksize=50)
coef(fit)</pre>
```

4 delayed-ops

delayed-ops

Delayed Operations on "matter" Objects

#### **Description**

Some arithmetic operations are available as delayed operations on matter objects. With these operations, no data is changed on disk, and the operation is only executed when elements of the object are actually accessed.

#### **Details**

Currently the following operations are supported:

```
'Arith': '+', '-', '*', '/', '^'
'Math': 'exp', 'log', 'log2', 'log10'
```

Delayed operations are applied at the C++ layer immediately after the elements are read from disk. This means that operations that are implemented in C and/or C++ for efficiency (such as summary statistics) will also reflect the execution of the delayed operations.

#### Value

A new matter object with the registered delayed operation. Data on disk is not modified; only object metadata is changed.

#### Author(s)

Kylie A. Bemis

## See Also

Arith, Math

```
x <- matter(1:100, length=100)
y <- x + 1
x[1:10]
y[1:10]
mean(x)
mean(y)</pre>
```

drle-class 5

drle-class

Delta Run Length Encoding

#### **Description**

The drle class stores delta-run-length-encoded vectors. These differ from other run-length-encoded vectors provided by other packages in that they allow for runs of values that each differ by a common difference (delta).

#### Usage

```
## Instance creation
drle(x, cr_threshold = 0)
is.drle(x)
## Additional methods documented below
```

## **Arguments**

x An integer or nu

An integer or numeric vector to convert to delta run length encoding for drle();

an object to test if it is of class drle for is.drle().

cr\_threshold

The compression ratio threshold to use when converting a vector to delta run length encoding. The default (0) always converts the object to drle. Values of cr\_threshold < 1 correspond to compressing even when the output will be larger than the input (by a certain ratio). For values > 1, compression will only take place when the output is (approximately) at least cr\_threshold times smaller.

#### Value

An object of class drle.

#### Slots

```
values: The values that begin each run.lengths: The length of each run.deltas: The difference between the values of each run.
```

#### **Creating Objects**

drle instances can be created through drle().

#### Methods

Standard generic methods:

```
x[i]: Get or set elements of the uncompressed vector.
```

length(x): Get the length of the uncompressed vector.

c(x, ...): Combine vectors.

6 matter-class

#### Author(s)

Kylie A. Bemis

#### See Also

```
[base]{rle}
```

#### **Examples**

```
## Create a drle vector
x <- c(1,1,1,1,1,6,7,8,9,10,21,32,33,34,15)
y <- drle(x)

# Check that their elements are equal
x == y[]</pre>
```

matter-class

Vectors, Matrices, and Arrays Stored on Disk

## **Description**

The matter class and its subclasses are designed for easy on-demand read/write access to binary on-disk data structures, and working with them as vectors, matrices, and arrays.

#### Usage

```
## Instance creation
matter(...)
## Additional methods documented below
```

## **Arguments**

... Arguments passed to subclasses.

## Value

An object of class matter.

# Slots

data: This slot stores the information about locations of the data on disk and within the files.

datamode: The storage mode of the accessed data when read into R. This should a 'character' vector of length one with value 'integer' or 'numeric'.

paths: A 'character' vector of the paths to the files where the data are stored.

filemode: The read/write mode of the files where the data are stored. This should be 'rb' for read-only access, or 'rb+' for read/write access.

chunksize: The maximum number of elements which should be loaded into memory at once. Used by methods implementing summary statistics and linear algebra. Ignored when explicitly subsetting the dataset.

matter-class 7

length: The length of the data.

dim: Either 'NULL' for vectors, or an integer vector of length one of more giving the maximal indices in each dimension for matrices and arrays.

names: The names of the data elements for vectors.

dimnames: Either 'NULL' or the names for the dimensions. If not 'NULL', then this should be a list of character vectors of the length given by 'dim' for each dimension. This is always 'NULL' for vectors.

ops: Delayed operations to be applied on atoms.

# **Creating Objects**

matter is a virtual class and cannot be instantiated directly, but instances of its subclasses can be created through matter().

#### Methods

```
Class-specific methods:
```

```
atomdata(x): Access the 'data' slot.
adata(x): An alias for atomdata(x).
datamode(x), datamode(x) <- value: Get or set 'datamode'.
paths(x), paths(x) <- value: Get or set 'paths'.
filemode(x), filemode(x) <- value: Get or set 'filemode'.
chunksize(x), chunksize(x) <- value: Get or set 'filemode'.
Standard generic methods:
length(x), length(x) <- value: Get or set 'length'.
dim(x), dim(x) <- value: Get or set 'dim'.
names(x), names(x) <- value: Get or set 'names'.
dimnames(x), dimnames(x) <- value: Get or set 'dimnames'.</pre>
```

# Author(s)

Kylie A. Bemis

## See Also

```
matter_vec, matter_mat
```

```
## Create a matter_vec vector
x <- matter(1:100, length=100)
x[]

## Create a matter_mat matrix
x <- matter(1:100, nrow=10, ncol=10)
x[]</pre>
```

8 matter\_mat-class

matter_ex-data Exam	ples for "matter" package
---------------------	---------------------------

#### **Description**

Example data for the "matter" package for use in vignettes.

#### Usage

```
data(matter_ex)
```

#### Value

None. Loads objects required to build vignettes.

matter\_mat-class

Matrices Stored on Disk

# Description

The matter\_mat class implements on-disk matrices.

# Usage

# Arguments

data	An optional data vector which will be initially written to the data on disk if provided.
datamode	A 'character' vector giving the storage mode of the data on disk. Allowable values are 'short', 'int', 'long', 'float', and 'double'.
paths	A 'character' vector of the paths to the files where the data are stored. If 'NULL', then a temporary file is created using tempfile.
filemode	The read/write mode of the files where the data are stored. This should be 'rb' for read-only access, or 'rb+' for read/write access.
offset	A vector giving the offsets in number of bytes from the beginning of each file in 'paths', specifying the start of the data to be accessed for each file.
extent	A vector giving the length of the data for each file in 'paths', specifying the number of elements of size 'datamode' to be accessed from each file.
nrow	An optional number giving the total number of rows.

matter\_mat-class 9

ncol An optional number giving the total number of columns.

rowMaj Whether the data should be stored in row-major order (as opposed to column-

major order) on disk. Defaults to 'FALSE', for efficient access to columns. Set

to 'TRUE' for more efficient access to rows instead.

dimnames The names of the matrix dimensions.

... Additional arguments to be passed to constructor.

#### Value

An object of class matter\_mat.

#### **Slots**

data: This slot stores the information about locations of the data on disk and within the files.

datamode: The storage mode of the accessed data when read into R. This should a 'character' vector of length one with value 'integer' or 'numeric'.

paths: A 'character' vector of the paths to the files where the data are stored.

filemode: The read/write mode of the files where the data are stored. This should be 'rb' for read-only access, or 'rb+' for read/write access.

chunksize: The maximum number of elements which should be loaded into memory at once. Used by methods implementing summary statistics and linear algebra. Ignored when explicitly subsetting the dataset.

length: The length of the data.

dim: Either 'NULL' for vectors, or an integer vector of length one of more giving the maximal indices in each dimension for matrices and arrays.

names: The names of the data elements for vectors.

dimnames: Either 'NULL' or the names for the dimensions. If not 'NULL', then this should be a list of character vectors of the length given by 'dim' for each dimension. This is always 'NULL' for vectors.

#### Extends

matter

## **Creating Objects**

matter\_mat instances can be created through matter\_mat() or matter().

#### Methods

Standard generic methods:

```
x[i,j], x[i,j] <- value: Get or set the elements of the matrix.
```

x %\*% y: Matrix multiplication. At least one matrix must be an in-memory R matrix (or vector).

crossprod(x, y): Alias for t(x) %\*% y.

tcrossprod(x, y): Alias for x %\*% t(y).

cbind(x, ...), rbind(x, ...): Combine matrices by row or column.

t(x): Transpose a matrix. This is a quick operation which only changes metadata and does not touch the on-disk data.

10 matter\_vec-class

#### Author(s)

Kylie A. Bemis

## See Also

matter

# **Examples**

```
x <- matter_mat(1:100, nrow=10, ncol=10)
x[]</pre>
```

matter\_vec-class

Vectors Stored on Disk

# Description

The matter\_vec class implements on-disk vectors.

# Usage

# **Arguments**

data	An optional data vector which will be initially written to the data on disk if provided.
datamode	A 'character' vector giving the storage mode of the data on disk. Allowable values are 'short', 'int', 'long', 'float', and 'double'.
paths	A 'character' vector of the paths to the files where the data are stored. If 'NULL', then a temporary file is created using tempfile.
filemode	The read/write mode of the files where the data are stored. This should be 'rb' for read-only access, or 'rb+' for read/write access.
offset	A vector giving the offsets in number of bytes from the beginning of each file in 'paths', specifying the start of the data to be accessed for each file.
extent	A vector giving the length of the data for each file in 'paths', specifying the number of elements of size 'datamode' to be accessed from each file.
length	An optional number giving the total length of the data across all files, equal to the sum of 'extent'. This is ignored and calculated automatically if 'extent' is specified.
names	The names of the data elements.
	Additional arguments to be passed to constructor.

matter\_vec-class 11

#### Value

An object of class matter\_vec.

#### **Slots**

data: This slot stores the information about locations of the data on disk and within the files.

datamode: The storage mode of the *accessed* data when read into R. This is a 'character' vector of length one with value 'integer' or 'numeric'.

paths: A 'character' vector of the paths to the files where the data are stored.

filemode: The read/write mode of the files where the data are stored. This should be 'rb' for read-only access, or 'rb+' for read/write access.

chunksize: The maximum number of elements which should be loaded into memory at once. Used by methods implementing summary statistics and linear algebra. Ignored when explicitly subsetting the dataset.

length: The length of the data.

dim: Either 'NULL' for vectors, or an integer vector of length one of more giving the maximal indices in each dimension for matrices and arrays.

names: The names of the data elements for vectors.

dimnames: Either 'NULL' or the names for the dimensions. If not 'NULL', then this should be a list of character vectors of the length given by 'dim' for each dimension. This is always 'NULL' for vectors.

#### **Extends**

matter

# **Creating Objects**

matter\_vec instances can be created through matter\_vec() or matter().

#### Methods

Standard generic methods:

```
x[i], x[i] <- value: Get or set the elements of the vector.
```

c(x, ...): Combine vectors.

t(x): Transpose a vector (to a row matrix). This is a quick operation which only changes metadata and does not touch the on-disk data.

#### Author(s)

Kylie A. Bemis

#### See Also

matter

```
x <- matter_vec(1:100, length=100)
x[]</pre>
```

12 prcomp

prcomp

Principal Components Analysis for "matter" Matrices

#### **Description**

This method allows computation of a truncated principal components analysis of a matter\_mat matrix using the implicitly restarted Lanczos method irlba.

# Usage

```
## S4 method for signature 'matter_mat'
prcomp(x, n = 3, retx = TRUE, center = TRUE, scale. = FALSE, ...)
```

# **Arguments**

x	A matter matrix.
n	The number of principal componenets to return, must be less than $min(dim(x))$ .
retx	A logical value indicating whether the rotated variables should be returned.
center	A logical value indicating whether the variables should be shifted to be zero-centered, or a centering vector of length equal to the number of columns of x. The centering is performed implicitly and does not change the data-on-disk in x.
scale.	A logical value indicating whether the variables should be scaled to have unit variance, or a scaling vector of length equal to the number of columns of x. The scaling is performed implicitly and does not change the data-on-disk in x.
	Additional options passed to irlba.

#### Value

An object of class 'prcomp'. See ?prcomp for details.

# Note

The 'tol' truncation argument found in the default prcomp method is not supported. In place of the truncation tolerance in the original function, the argument n explicitly gives the number of principal components to return. A warning is generated if the argument 'tol' is used.

# Author(s)

```
Kylie A. Bemis
```

#### See Also

```
bigglm
```

```
set.seed(1)

x <- matter_mat(rnorm(1000), nrow=100, ncol=10)
prcomp(x)</pre>
```

scale 13

	_
SCA	ء (

Scaling and Centering of "matter" Matrices

# Description

An implementation of scale for matter\_mat matrices.

# Usage

```
## S4 method for signature 'matter_mat'
scale(x, center = TRUE, scale = TRUE)
```

# **Arguments**

X	A matter_mat object.
center	Either a logical value or a numeric vector of length equal to the number of columns of $\dot{x}$ .
scale	Either a logical value or a numeric vector of length equal to the number of

columns of 'x'.

#### **Details**

See scale for details.

#### Value

A matter\_mat object with the appropriate 'scaled:center' and 'scaled:scale' attributes set. No data on disk is changed, but the scaling will be applied any time the data is read. This includes but is not limited to loading data elements via subsetting, summary statistics methods, and matrix multiplication.

# Author(s)

Kylie A. Bemis

## See Also

scale

```
x <- matter(1:100, nrow=10, ncol=10)
scale(x)</pre>
```

14 summary-stats

summary-stats

Summary Statistics for "matter" Objects

#### **Description**

These functions efficiently calculate summary statistics for matter objects. For matrices, they operate efficiently on both rows and columns.

#### Usage

```
## S4 method for signature 'matter'
mean(x, na.rm)
## S4 method for signature 'matter'
sum(x, na.rm)
## S4 method for signature 'matter'
sd(x, na.rm)
## S4 method for signature 'matter'
var(x, na.rm)
## S4 method for signature 'matter_mat'
colMeans(x, na.rm)
## S4 method for signature 'matter_mat'
colSums(x, na.rm)
## S4 method for signature 'matter_mat'
colSds(x, na.rm)
## S4 method for signature 'matter_mat'
colVars(x, na.rm)
## S4 method for signature 'matter_mat'
rowMeans(x, na.rm)
## S4 method for signature 'matter_mat'
rowSums(x, na.rm)
## S4 method for signature 'matter_mat'
rowSds(x, na.rm)
## S4 method for signature 'matter_mat'
rowVars(x, na.rm)
```

## **Arguments**

```
x A matter object.
na.rm If TRUE, remove NA values before summarizing.
```

#### **Details**

These summary statistics methods operate on chunks of data (equal to the chunksize of x) which are loaded into memory and then freed before reading the next chunk.

For row and column summaries on matrices, the iteration scheme is dependent on the layout of the data. Column-major matrices will always be iterated over by column, and row-major matrices will always be iterated over by row. Row statistics on column-major matrices and column statistics on row-major matrices are calculated iteratively.

The efficiency of these methods is entirely dependent on the chunksize of x. Larger chunks will yield faster calculations, but greater memory usage. The row and column summary methods may be more or less efficient than the equivalent call to apply, depending on the chunk size.

summary-stats 15

Varsiance and standard deviation are calculated using a running sum of squares formula which can be calculated iteratively and is accurate for large floating-point datasets (see reference).

#### Value

For mean, sum, sd, and var, a single number. For the column summaries, a vector of length equal to the number of columns of the matrix. For the row summaries, a vector of length equal to the number of rows of the matrix.

## Author(s)

Kylie A. Bemis

#### References

B. P. Welford, "Note on a Method for Calculating Corrected Sums of Squares and Products," Technometrics, vol. 4, no. 3, pp. 1-3, Aug. 1962.

#### See Also

```
colSums, colMeans, rowSums, rowMeans
```

```
x <- matter(1:100, nrow=10, ncol=10)
sum(x)
mean(x)
var(x)
sd(x)

colSums(x)
colMeans(x)
colVars(x)
colSds(x)

rowSums(x)
rowMeans(x)
rowWars(x)
rowVars(x)
rowSds(x)</pre>
```

# Index

*Topic <b>IO</b>	(delayed-ops), 4
matter-class, 6	+,matter_matr,numeric-method
matter_mat-class, 8	(delayed-ops), 4
matter_vec-class, 10	+,matter_vec,numeric-method
*Topic arith	(delayed-ops), 4
delayed-ops, 4	+,numeric,matter_matc-method
*Topic <b>array</b>	(delayed-ops), 4
matter-class, 6	+,numeric,matter_matr-method
matter_mat-class, 8	(delayed-ops), 4
matter_vec-class, 10	+,numeric,matter_vec-method
*Topic <b>classes</b>	(delayed-ops), 4
drle-class, 5	-,matter_matc,numeric-method
matter-class, 6	(delayed-ops), 4
matter_mat-class, 8	-,matter_matr,numeric-method
matter_vec-class, 10	(delayed-ops), 4
*Topic datasets	-,matter_vec,numeric-method
matter_ex-data, 8	(delayed-ops), 4
*Topic <b>methods</b>	-,numeric,matter_matc-method
apply, 2	(delayed-ops), 4
delayed-ops, 4	-,numeric,matter_matr-method
scale, 13	(delayed-ops), 4
summary-stats, 14	-,numeric,matter_vec-method
*Topic <b>models</b>	(delayed-ops), 4
bigglm, 3	/,matter_matc,numeric-method
*Topic <b>multivariate</b>	(delayed-ops), 4
prcomp, 12	/,matter_matr,numeric-method
*Topic <b>regression</b>	(delayed-ops), 4
bigglm, 3	/,matter_vec,numeric-method
*Topic <b>univar</b>	(delayed-ops), 4
summary-stats, 14	/,numeric,matter_matc-method
*,matter_matc,numeric-method	(delayed-ops), 4
(delayed-ops), 4	/,numeric,matter_matr-method
*,matter_matr,numeric-method	(delayed-ops), 4
(delayed-ops), 4	/,numeric,matter_vec-method
*,matter_vec,numeric-method	(delayed-ops), 4
(delayed-ops), 4	[,atoms,ANY,ANY,ANY-method
*,numeric,matter_matc-method	(matter-class), 6
(delayed-ops), 4	[,drle,ANY,missing,missing-method
*,numeric,matter_matr-method	(drle-class), 5
(delayed-ops), 4	[,drle,missing,missing,missing-method
*,numeric,matter_vec-method	(drle-class), 5
(delayed-ops), 4	[,matter_mat,ANY,ANY,logical-method
+,matter_matc,numeric-method	(matter_mat-class), 8

INDEX 17

[,matter_mat,ANY,ANY,missing-method	(delayed-ops), 4
<pre>(matter_mat-class), 8</pre>	^,matter_matr,numeric-method
[,matter_mat,ANY,missing,logical-method	(delayed-ops), 4
(matter_mat-class), 8	^,matter_vec,numeric-method
[,matter_mat,ANY,missing,missing-method	(delayed-ops), 4
(matter_mat-class), 8	^,numeric,matter_matc-method
[,matter_mat,missing,ANY,logical-method	(delayed-ops), 4
(matter_mat-class), 8	^,numeric,matter_matr-method
[,matter_mat,missing,ANY,missing-method	(delayed-ops), 4
<pre>(matter_mat-class), 8</pre>	^,numeric,matter_vec-method
[,matter_mat,missing,missing,missing-method	(delayed-ops), 4
<pre>(matter_mat-class), 8</pre>	
[,matter_mat-method(matter_mat-class),	adata (matter-class), 6
8	adata, matter-method (matter-class), 6
[,matter_vec,ANY,missing,ANY-method	apply, 2, 2, 14
(matter_vec-class), 10	apply, matter_mat-method (apply), 2
[,matter_vec,missing,missing,ANY-method	Arith, 4
(matter_vec-class), 10	atomdata (matter-class), 6
[,matter_vec-method (matter_vec-class),	atomdata, matter-method (matter-class), (
10	atomuata, matter -metriou (matter -class), (
<pre>[&lt;-,matter_mat,ANY,ANY,ANY-method</pre>	bigglm, 3, 3, 12
(matter_mat-class), 8	bigglm, formula, matter_mat-method
[<-, matter_mat, ANY, missing, ANY-method	(bigglm), 3
(matter_mat-class), 8	bigglm.out (matter_ex-data), 8
[<-, matter_mat, missing, ANY, ANY-method	SISSIM. Out (matter lex data), o
(matter_mat-class), 8	c,atoms-method(matter-class),6
[<-,matter_mat,missing,missing,ANY-method	c,drle-method (drle-class), 5
(matter_mat-class), 8	c, matter-method (matter-class), 6
[<-,matter_mat-method	
(matter_mat-class), 8	<pre>c,matter_vec-method (matter_vec-class), 10</pre>
[<-,matter_vec,ANY,missing,ANY-method	
(matter_vec-class), 10	cbind, matter-method (matter_mat-class),
[<-,matter_vec,missing,missing,ANY-method	obunicai za (mattam alaga) 6
(matter_vec-class), 10	chunksize (matter-class), 6
	chunksize, matter-method (matter-class),
[<-,matter_vec-method	0
(matter_vec-class), 10	chunksize<- (matter-class), 6
[[,atoms-method (matter-class), 6	chunksize<-,matter-method
%*%, matrix, matter_mat-method	(matter-class), 6
(matter_mat-class), 8	class:drle (drle-class), 5
%*%, matter, matter-method	<pre>class:matter (matter-class), 6</pre>
$(matter\_mat-class), 8$	<pre>class:matter_mat (matter_mat-class), 8</pre>
%*%,matter_mat,matrix-method	<pre>class:matter_vec (matter_vec-class), 10</pre>
<pre>(matter_mat-class), 8</pre>	colMeans, 15
%*%,matter_matc,numeric-method	colMeans,matter_mat-method
<pre>(matter_mat-class), 8</pre>	(summary-stats), 14
%*%, matter_matr, numeric-method	colSds (summary-stats), 14
<pre>(matter_mat-class), 8</pre>	colSds,matter_mat-method
%*%, numeric, matter_matc-method	(summary-stats), 14
(matter_mat-class), 8	colSums, 15
<pre>%*%,numeric,matter_matr-method</pre>	colSums,matter_mat-method
(matter_mat-class), 8	(summary-stats), 14
^,matter_matc,numeric-method	colVars (summary-stats), 14

18 INDEX

colVars,matter_mat-method	<pre>log2,matter_mat-method(delayed-ops), 4</pre>
(summary-stats), 14	<pre>log2, matter_vec-method (delayed-ops), 4</pre>
crossprod, ANY, matter-method	
(matter_mat-class), 8	Math, 4
crossprod, matter, ANY-method	matter, 3, 4, 6, 9–12, 14
(matter_mat-class), 8	matter (matter-class), 6
(	matter-class, 6
data:matter_ex (matter_ex-data), 8	matter_ex (matter_ex-data), 8
datamode (matter-class), 6	matter_ex-data, 8
datamode, atoms-method (matter-class), 6	
	matter_mat, 2, 3, 7, 9, 12, 13
datamode, matter-method (matter-class), 6	<pre>matter_mat (matter_mat-class), 8</pre>
datamode<- (matter-class), 6	matter_mat-class, 8
datamode<-,atoms-method(matter-class),	<pre>matter_matc (matter_mat-class), 8</pre>
6	matter_matc-class (matter_mat-class), 8
datamode<-,matter-method	matter_matr (matter_mat-class), 8
(matter-class), 6	matter_matr-class(matter_mat-class), 8
delayed-ops, 4	matter_vec, 7, <i>11</i>
dim, matter-method (matter-class), 6	matter_vec (matter_vec-class), 10
dim<-, matter-method (matter-class), 6	matter_vec-class, 10
dimnames, matter-method (matter-class), 6	mean (summary-stats), 14
dimnames<-, matter, ANY-method	mean, matter-method (summary-stats), 14
(matter-class), 6	mean.matter(summary-stats), 14
drle,5	, , , , , , , , , , , , , , , , , , ,
drle (drle-class), 5	names, matter-method (matter-class), 6
drle-class, 5	names<-,matter-method (matter-class), 6
ui ie Ciass, 5	Traines , marrer metrioa (marrer etass), o
<pre>exp,matter_mat-method(delayed-ops), 4</pre>	paths (matter-class), 6
exp, matter_vec-method (delayed-ops), 4	paths, matter-method (matter-class), 6
exp, matter_vec-method (delayed-ops), 4	paths<- (matter-class), 6
filomodo (mattar-alaca) 6	
filemode (matter-class), 6	paths<-, matter-method (matter-class), 6
filemode, matter-method (matter-class), 6	prcomp, 12, 12
filemode<- (matter-class), 6	prcomp, matter_mat-method (prcomp), 12
filemode<-,matter-method	$prcomp.out(matter\_ex-data), 8$
(matter-class), 6	
	rbind, matter-method (matter_mat-class),
irlba, <i>12</i>	8
is.drle(drle-class), 5	rowMeans, 15
	rowMeans,matter_mat-method
length, atoms-method (matter-class), 6	(summary-stats), 14
<pre>length,drle-method(drle-class),5</pre>	rowSds (summary-stats), 14
length, matter-method (matter-class), 6	rowSds,matter_mat-method
<pre>length&lt;-,matter-method(matter-class), 6</pre>	(summary-stats), 14
log, matter_matc, numeric-method	rowSums, 15
(delayed-ops), 4	rowSums, matter_mat-method
log, matter_matc-method (delayed-ops), 4	(summary-stats), 14
log, matter_matr, numeric-method	rowVars (summary-stats), 14
(delayed-ops), 4	
	rowVars, matter_mat-method
log, matter_matr-method (delayed-ops), 4	(summary-stats), 14
log, matter_vec, numeric-method	1- 12 12
(delayed-ops), 4	scale, 13, 13
<pre>log,matter_vec-method(delayed-ops), 4</pre>	scale, matter_mat-method (scale), 13
<pre>log10,matter_mat-method(delayed-ops), 4</pre>	scale.matter(scale), 13
<pre>log10, matter_vec-method (delayed-ops), 4</pre>	sd (summary-stats), 14

INDEX 19

```
sd, matter-method (summary-stats), 14
sum (summary-stats), 14
\verb"sum,matter-method" (\verb"summary-stats"), 14
summary-stats, 14
t,matter_matc-method
          (matter_mat-class), 8
t,matter_matr-method
          (matter_mat-class), 8
t,matter_vec-method(matter_vec-class),
          10
t.matter(matter_mat-class), 8
tcrossprod, ANY, matter-method
          (matter_mat-class), 8
tcrossprod, matter, ANY-method
          (matter_mat-class), 8
\texttt{tempfile}, \textcolor{red}{8}, \textcolor{red}{10}
var (summary-stats), 14
\textit{var}, \textit{matter-method} \, (\textit{summary-stats}), \, 14
```