Package 'symDMatrix'

May 9, 2026

Version 2.1.1

License MIT + file LICENSE

Title Partitioned Symmetric Matrices

Description A matrix-like class to represent a symmetric matrix partitioned into file-backed blocks.

URL https://github.com/QuantGen/symDMatrix

BugReports https://github.com/QuantGen/symDMatrix/issues

Depends R (>= 3.0.2)

Imports methods, LinkedMatrix (>= 1.3.0), ff, bit

Suggests BGData, BEDMatrix, tinytest

NeedsCompilation no

Author Gustavo de los Campos [aut], Alexander Grueneberg [aut, cre]

Maintainer Alexander Grueneberg <cran@agrueneberg.info>

Repository CRAN

Date/Publication 2020-08-02 13:00:02 UTC

Contents

symDMatrix-package *A Package Providing Symmetric Matrices Partitioned into File-Backed Blocks*

Description

A Package Providing Symmetric Matrices Partitioned into File-Backed Blocks.

Example Dataset

The example dataset in the `extdata` folder is the G matrix of the dummy dataset that comes with the `BEDMatrix` package. It has been generated as follows:

```
library(BGData)
X <- BEDMatrix(system.file("extdata", "example.bed", package = "BEDMatrix"))
G <- getG_symDMatrix(X, blockSize = 17, folderOut = "inst/extdata")
```

To load the dataset:

```
load.symDMatrix(system.file("extdata", "G.RData", package = "symDMatrix"),
    readonly = TRUE)
```

To demonstrate the `as.symDMatrix` method for character vectors, `RData` files for each block have been generated:

```
for (i in 1:nBlocks(G)) {
    for (j in i:nBlocks(G)) {
        block <- G[[i]][[j]]
        save(block, file = paste0("inst/extdata/data_", i, "_", j, ".RData"))
    }
}
```

See Also

`symDMatrix-class` for the `symDMatrix` class. `BEDMatrix-package` for more information on the `BEDMatrix` package.

as.symDMatrix *Coerce an Object to a symDMatrix Object*

Description

Coerce an object to a symDMatrix object.

Usage

```
as.symDMatrix(x, ...)
```

Arguments

x	A numeric matrix.
...	Additional arguments.

Value

A symDMatrix object.

See Also

as.symDMatrix.matrix to coerce a matrix or as.symDMatrix.character to coerce a vector of path names to a symDMatrix object.

as.symDMatrix.character *Coerce a Character Vector to a symDMatrix Object*

Description

This function creates a symDMatrix object from a character vector of path names to RData files, each containing exactly one ff_matrix object that is used as a block, and is useful for distributed computing where each block is processed on a different node.

Usage

```
## S3 method for class 'character'
as.symDMatrix(x, ...)
```

Arguments

x	A character vector with path names to RData files.
...	Additional arguments (currently unused).

Details

The RData files must be ordered by block: G11, G12, G13, ..., G1q, G22, G23, ..., G2q, ..., Gqq. The matrix-like objects are initialized similarly to load.symDMatrix.

Value

A symDMatrix object.

See Also

list.files to create a character vector of file paths that match a certain pattern.

---

as.symDMatrix.matrix *Coerce a Matrix to a symDMatrix Object*

Description

This function creates a symDMatrix from a numeric matrix that is assumed to be symmetric.

Usage

```
## S3 method for class 'matrix'
as.symDMatrix(x, blockSize = 5000L, vmode = "double",
  folderOut = randomString(), ...)
```

Arguments

x	A symmetric numeric matrix.
blockSize	The number of rows and columns of each block. If NULL, a single block of the same dimensions as x will be created. Defaults to 5000.
vmode	The vmode used to store the data in the ff objects.
folderOut	A name for a folder where to store the data of the resulting symDMatrix object.
...	Additional arguments (currently unused).

Details

The input matrix is broken into blocks and each block is stored as an ff_matrix object. In addition, a metadata object called symDMatrix.RData is created to allow for easy reloading of the symDMatrix object.

Value

A symDMatrix object.

See Also

load.symDMatrix to reload the symDMatrix object.

blockIndex *Return the Block Structure of a symDMatrix Object*

Description

This function returns the block structure of a `symDMatrix` object and can be useful when implementing custom indexing techniques.

Usage

```
blockIndex(x)
```

Arguments

`x`	A `symDMatrix` object.

Value

A matrix with three columns: the block number, the start index and the end index.

blockSize *Return the Block Size of a symDMatrix Object*

Description

This function returns the block size of a `symDMatrix` object.

Usage

```
blockSize(x, last = FALSE)
```

Arguments

`x`	A `symDMatrix` object.
`last`	A boolean indicating whether to return the block size of the last (`TRUE`) column/row block or any of the other blocks (`FALSE`, default).

Details

The last block of a column/row may be smaller than the other blocks. Its size can be retrieved by setting `last` to `TRUE`.

Value

The block size of a `symDMatrix` object.

Examples

```
# Load example symDMatrix (G)
load.symDMatrix(system.file("extdata", "G.RData", package = "symDMatrix"), readonly = TRUE)

# Get the block size
blockSize(G)

# Get the block size of the trailing blocks
blockSize(G, last = TRUE)
```

`load.symDMatrix` *Load symDMatrix Objects from .RData Files*

Description

This function is similar to `load`, but it also initializes the `ff_matrix` blocks in the `symDMatrix` object.

Usage

```
load.symDMatrix(file, readonly = FALSE, envir = parent.frame())
```

Arguments

`file`	The name of an .RData file to be loaded.
`readonly`	Set to TRUE to forbid writing to existing files.
`envir`	The environment where to load the data.

`nBlocks` *Return the Number of Column/Row Blocks of a symDMatrix Object*

Description

This function returns the number of row blocks the original matrix has been partitioned into.

Usage

```
nBlocks(x)
```

Arguments

`x`	A `symDMatrix` object.

Value

The number of column/row blocks of a `symDMatrix` object.

Examples

```
# Load example symDMatrix (G)
load.symDMatrix(system.file("extdata", "G.RData", package = "symDMatrix"), readonly = TRUE)

# Get the number of row blocks the original matrix was partitioned into
nBlocks(G)
```

`symDMatrix` *Create a New symDMatrix Instance*

Description

This function constructs a new `symDMatrix` object.

Usage

```
symDMatrix(...)
```

Arguments

`...`	`ColumnLinkedMatrix` objects containing blocks that inherit from `ff_matrix`.

Details

Several structural checks are performed on the passed blocks: there must be at least one block, the blocks must be of type `ColumnLinkedMatrix`, and the number of blocks must be consistent across the `ColumnLinkedMatrix` objects. Each block must inherit from `ff_matrix` and have the same number of rows or columns as blocks in the same row or column, respectively. Non-final blocks have to be square, unless if there is only a single block, in which case that block also has to be square.

Value

A `symDMatrix` object.

See Also

`as.symDMatrix` to create a `symDMatrix` object from other objects.

Examples

```
# Generate a symmetric matrix
X <- cov(matrix(data = rnorm(25), nrow = 5, ncol = 5))

# Break this matrix into blocks X11, X12, X22
# X21 can be stored as a virtual transpose of X12
X11 <- ff::as.ff(X[1:3, 1:3])
X12 <- ff::as.ff(X[1:3, 4:5])
X22 <- ff::as.ff(X[4:5, 4:5])
```

```
X21 <- ff::vt(X12)

# Create a symDMatrix from blocks
S <- symDMatrix(
    LinkedMatrix::ColumnLinkedMatrix(X11, X12),
    LinkedMatrix::ColumnLinkedMatrix(X21, X22)
)
nBlocks(S)
blockSize(S)
blockSize(S, last = TRUE)
```

`symDMatrix-class` *A Matrix-Like Class to Represent a Symmetric Matrix Partitioned into File-Backed Blocks*

Description

A `symDMatrix` is a symmetric matrix partitioned into file-backed blocks. This approach allows for very large symmetric matrices, commonly found for example when computing genetic relationship matrices on large cohorts. A `symDMatrix` object behaves similarly to a regular `matrix` by implementing key methods such as `[`, `dim`, and `dimnames`.

Details

The `symDMatrix` class is a `RowLinkedMatrix` that nests multiple `ColumnLinkedMatrix` objects containing blocks of type `ff_matrix`. Because the matrix is symmetric, only the diagonal and upper-triangular blocks need to be stored, but for more efficient queries, the lower-triangular blocks are virtual transposes of their diagonal counterparts.

See Also

`symDMatrix` to create a `symDMatrix` object from scratch, or preferably, `as.symDMatrix` to create a `symDMatrix` object from other objects.

Examples

```
# Get the path to the example symmetric matrix
path <- system.file("extdata", "G.RData", package = "symDMatrix")

# Load the example symDMatrix object (G)
load.symDMatrix(path, readonly = TRUE)

# Get the dimensions
dim(G)

# Get the row names
rownames(G)

# Get the column names
```

```
colnames(G)

# Extract the diagonal
diag(G)

# Extract rows and columns
G[1, ]
G[1:3, ]
G["per0_per0", ]
G[c("per0_per0", "per1_per1", "per2_per2"), ]
```

Index