Package: annmatrix (via r-universe)

September 9, 2024

Title Annotated Matrix: Matrices with Persistent Row and Column Annotations

Version 0.1.2

Maintainer Karolis Koncevičius <`karolis.koncevicius@gmail.com`>

Description Implements persistent row and column annotations for R matrices. The annotations associated with rows and columns are preserved after subsetting, transposition, and various other matrix-specific operations. Intended use case is for storing and manipulating genomic datasets which typically consist of a matrix of measurements (like gene expression values) as well as annotations about rows (i.e. genomic locations) and annotations about columns (i.e. meta-data about collected samples). But 'annmatrix' objects are also expected to be useful in various other contexts.

Depends R (>= 4.3.0)

Imports utils, methods, stats

License GPL-2

Encoding UTF-8

LazyData true

URL `https://github.com/karoliskoncevicius/annmatrix`

BugReports `https://github.com/karoliskoncevicius/annmatrix/issues`

RoxygenNote 7.2.3

Repository https://karoliskoncevicius.r-universe.dev

RemoteUrl https://github.com/karoliskoncevicius/annmatrix

RemoteRef HEAD

RemoteSha d41cc0773a90659e5607380db17cede3ccecb91f

Contents

`annmatrix` *annmatrix Objects and Basic Functionality*

Description

Annotated matrix is a regular matrix with additional functionality for attaching persistent information about row and column entries. Annotations associated with rows and columns are preserved after subsetting, transposition, and various other matrix-specific operations.

Intended 'annmatrix' use case is for storing and manipulating genomic datasets that typically consist of a matrix of measurements (like gene expression values) as well as annotations about rows (i.e. genomic locations) and annotations about columns (i.e. meta-data about collected samples). But `annmatrix` objects are also expected be useful in various other contexts.

Usage

```
annmatrix(x, rann, cann)

rowanns(x, names)

rowanns(x, names) <- value

colanns(x, names)

colanns(x, names) <- value

`@.annmatrix`(object, name)

## S3 replacement method for class 'annmatrix'
@(object, name) <- value

## S3 method for class 'annmatrix'
```

```
x$name

## S3 replacement method for class 'annmatrix'
x$name <- value
```

Arguments

`x, object`	an R object.
`rann`	annotation `data.frame` for rows of the `annmatrix` object.
`cann`	annotation `data.frame` for columns of the `annmatrix` object.
`names`	a character vector of existing row/column annotation names.
`value`	a value that will be assigned to row/column annotation field.
`name`	a name of an existing row/column annotation.

Details

`annmatrix()` constructs an annmatrix. The function expects `x` to be a `matrix` and `rowanns` and `colanns` to be of class `data.frame`. If the passed objects are of a different classes they will be converted via the use of `as.matrix` and `as.data.frame`.

`rowanns` and `colanns` returns the selected field from column and row annotations respectively. When the selected field is not specified the whole annotation `data.frame` is returned.

`@` and `$` are convenience shortcuts for selecting annotations. `X@value` selects an existing column from row annotations while `X$value` selects a column from column annotations. An empty selection of `X@''` and `X$''` will return the whole annotation data.frame for rows and columns respectively.

`rowanns<-` and `colanns<-` functions can be used to replace the column and row annotations respectively. When the selected field is not specified the whole annotation `data.frame` is replaced.

`@<-` and `$<-` are convenience shortcuts for the above (see Examples). A replacement of an empty value - `X@'' <- df` and `X$'' <- df` will replace the whole annotation data.frame.

Value

`annmatrix` returns an R object of class ’annmatrix’.

Author(s)

Karolis Koncevičius

See Also

`as.annmatrix`

Examples

```
# construct annmatrix object
x <- matrix(rnorm(20*10), 20, 10)

coldata <- data.frame(group  = rep(c("case", "control"), each = 5),
                      gender = sample(c("M", "F"), 10, replace = TRUE))

rowdata <- data.frame(chr = sample(c("chr1", "chr2"), 20, replace = TRUE),
                      pos = runif(20, 0, 1000000))

X <- annmatrix(x, rowdata, coldata)

is.matrix(x)
is.matrix(X)

is.annmatrix(x)
is.annmatrix(X)

# manipulating annotations without using shortcuts
rowanns(X)
colanns(X)

rowanns(X, "chr")
rowanns(X, "gene") <- letters[1:20]
rowanns(X, c("chr", "gene"))
rowanns(X, "gene") <- NULL
rowanns(X)

colanns(X, "group")
colanns(X, "age") <- 1:10*10
colanns(X, "age")
colanns(X, "age") <- NULL
colanns(X, "age")

# more convenient
X@''
X@chr
X@gene <- letters[1:20]
X@gene
X@gene <- NULL
X@gene

X$''
X$group
X$age <- 1:10*10
X$age
X$age <- NULL
X$age
X$'' <- data.frame(id = 1:10, name = LETTERS[1:10])
X$name

# annotations are preserved after subsetting
```

```
Y <- X[X@chr == "chr1", X$name %in% c("A", "B", "C")]
Y@chr
Y$''

Y[, 1]
Y[, 1, drop = FALSE]
```

`autocomplete` *Auto Complete Functions for annmatrix Class*

Description

Function used to select autocomplete options for dollar ‘$‘ and at ‘@‘ operators.

Usage

```
## S3 method for class 'annmatrix'
.DollarNames(x, pattern = "")

## S3 method for class 'annmatrix'
.AtNames(x, pattern = "")
```

Arguments

`x`	annmatrix object.
`pattern`	a regular expression used to select possible auto-completion names.

Value

A set of possible auto-completion names for row (`@`) or column (`$`) annotation fields.

Author(s)

Karolis Koncevičius

`bind` *Bind several annmatrix Objects Together*

Description

Implementation of `rbind` and `cbind` methods for annmatrix objects.

Usage

```
## S3 method for class 'annmatrix'
rbind(...)

## S3 method for class 'annmatrix'
cbind(...)
```

Arguments

`...`	(generalized) vector or matrix objects

Details

All the inputs are bound together in exact same way as if all the annmatrix objects were regular matrices. Then, the row and column annotations of the supplied annmatrix objects are merged together. Missing annotations are filled in using 'NA' values.

For demonstration purposes only `rbind` will be described with `cbind` behaving accordingly.

1) Matrix. The obtained matrix will be exactly the same as calling `rbind` with all annmatrix objects replaced by regular matrices.

2) Column annotations. When `rbind` is called the matrices are all assumed to have the same set of columns. Hence, the column annotations are assumed to be shared between all provided annmatrix objects. Thus, in order to retain all possible column annotations, they are merged together. This way any column annotation field present in at least one of the provided annmatrix objects will be present in the final result. In case of conflicts, when the same annotation field is present in multiple annmatrix objects but contains different values, the first occuring instance is selected and an appropriate warning is displayed. Non-annmatrix objects are assumed to share the column annotations present in supplied annmatrix objects.

3) Row annotations. When `rbind` is called the matrices are assumed to have a separate unique set of rows. Hence no conflicts between annotation values are possible for row annotations. In order to retain all possible row annotations, row annotations are merged together. Thus, the resulting row annotation data frame will have as many fields as there were unique row annotation fields among all the provided annmatrix objects. Unlike with column annotations, if a particular annmatrix only had a subset of the final collection of annotation fields, then the missing fields are added and the annotation is filled with NA values. All the rows associated with non-annmatrix objects will have missing (NA) values for all the annotation fields.

Value

a single annmatrix object resulting from binding all the inputs together

Author(s)

Karolis Koncevičius

Examples

```
# construct annmatrix object
x <- matrix(rnorm(20*10), 20, 10)
```

```
coldata <- data.frame(group  = rep(c("case", "control"), each = 5),
                      gender = sample(c("M", "F"), 10, replace = TRUE))

rowdata <- data.frame(chr = sample(c("chr1", "chr2"), 20, replace = TRUE),
                      pos = runif(20, 0, 1000000))

X <- annmatrix(x, rowdata, coldata)

X1 <- X[1:10,]
X2 <- X[11:20,]
all.equal(X, rbind(X1, X2))

X1 <- X[,1:5]
X2 <- X[,6:10]
all.equal(X, cbind(X1, X2))

X11 <- X[1:10, 1:5]
X21 <- X[11:20, 1:5]
X12 <- X[1:10, 6:10]
X22 <- X[11:20, 6:10]
all.equal(X, cbind(rbind(X11, X21), rbind(X12, X22)))
```

`convert` *Convert annmatrix Objects to and from Other Types*

Description

Methods for turning R objects to class annmatrix and vice versa.

Usage

```
as.annmatrix(x)

## Default S3 method:
as.annmatrix(x)

## S3 method for class 'matrix'
as.annmatrix(x)

## S3 method for class 'annmatrix'
as.matrix(x, ...)

is.annmatrix(x)
```

Arguments

`x`	an R object.
`...`	additional arguments to be passed to or from methods.

Details

`as.annmatrix` will attempt to convert an object to annmatrix.

`as.matrix` will turn an `annmatrix` object into a regular matrix.

`is.annmatrix` checks if the object is an instance of `annmatrix`.

Value

`is.annmatrix` returns TRUE if object is of class ’annmatrix’ and FALSE otherwise. `as.annmatrix` methods return an object of class ’annmatrix’. `as.matrix` returns a regular matrix.

Author(s)

Karolis Koncevičius

Examples

```
# construct annmatrix object
x <- matrix(rnorm(20*10), 20, 10)
X <- as.annmatrix(x)

X$group  <- rep(c("case", "control"), each = 5)
X$gender <- sample(c("M", "F"), 10, replace = TRUE)
X@chr    <- sample(c("chr1", "chr2"), 20, replace = TRUE)
X@pos    <- runif(20, 0, 1000000)

is.matrix(x)
is.matrix(X)

is.annmatrix(x)
is.annmatrix(X)

as.matrix(X)
```

`groupgenerics` *Group Generic Functions for annmatrix Class*

Description

The functions listed here work under the hood and are almost never called by the user.

Usage

```
## S3 method for class 'annmatrix'
Ops(e1, e2)

## S3 method for class 'annmatrix'
chooseOpsMethod(x, y, mx, my, cl, reverse)
```

Arguments

`e1, e2`	annmatrix objects.
`x, y`	The objects being dispatched on by the group generic.
`mx, my`	The methods found for objects 'x' and 'y'.
`cl`	The call to the group generic.
`reverse`	A logical value indicating whether 'x' and 'y' are reversed from the way they were supplied to the generic.

Value

An object of class 'annmatrix'.

Author(s)

Karolis Koncevičius

`matrixgenerics` *Matrix Generic Functions for annmatrix Class*

Description

Matrix cross-product operator implemented for annmatrix class

Usage

```
## S3 method for class 'annmatrix'
x %*% y
```

Arguments

`x, y`	numeric or complex matrices or vectors.

Details

The resulting matrix will be the same as a product between two regular matrices. If present annmatrix row annotations will be carried over from the first matrix `x` while the annotations for rows will be carried over from the second matrix `y`.

Value

an object of class 'annmatrix'.

Author(s)

Karolis Koncevičius

Examples

```
# construct annmatrix object
x <- matrix(rnorm(20*10), 20, 10)

coldata <- data.frame(group  = rep(c("case", "control"), each = 5),
                      gender = sample(c("M", "F"), 10, replace = TRUE))

rowdata <- data.frame(chr = sample(c("chr1", "chr2"), 20, replace = TRUE),
                      pos = runif(20, 0, 1000000))

X <- annmatrix(x, rowdata, coldata)

res <- 1:20 %*% X
res$group

res <- X %*% 1:10
res@chr

res <- t(X) %*% X
res@group
res$group
```

`pca` *Principal Component Analysis for annmatrix Objects*

Description

Performs principal component analysis on annmatrix objects and preserves row and column annotations in scores and loadings.

Usage

```
## S3 method for class 'annmatrix'
prcomp(
  x,
  retx = TRUE,
  center = TRUE,
  scale. = FALSE,
  tol = NULL,
  rank. = NULL,
  ...
)
```

Arguments

`x`	annmatrix object.

`retx`	logical indicator whether the rotated variables should be returned (defaults to TRUE).
`center`	logical value indicating whether variables should be centered (defaults to TRUE).
`scale.`	logical value indicating whether variables should be scaled (defaults to FALSE).
`tol`	tolerance value indicating magnitude below which components will be omitted.
`rank.`	number specifying the maximal rank (max number of principal components to be used).
`...`	arguments passed to or from other methods.

Details

The resulting loadings ('rotation') and scores ('x') are turned into annmatrix objects with additional row and column annotations. Row annotations are copied from the original input matrix while column annotations contain computed information about each principal component. The added information contains: 1) principal component number, 2) standard deviation, 3) variance, 4) fraction of variance explained.

Value

prcom object with rotation and x matrices turned into annmatrix

Author(s)

Karolis Koncevičius

Examples

```
# construct annmatrix object
x <- matrix(rnorm(20*10), 20, 10)

coldata <- data.frame(group  = rep(c("case", "control"), each = 5),
                      gender = sample(c("M", "F"), 10, replace = TRUE))

rowdata <- data.frame(chr = sample(c("chr1", "chr2"), 20, replace = TRUE),
                      pos = runif(20, 0, 1000000))

X <- annmatrix(x, rowdata, coldata)

pca <- prcomp(t(X))
pca$rotation
pca$rotation$''

scores <- t(pca$rotation) %*% X
scores@var_explained
scores$gender
```

`print` *Print annmatrix Object*

Description

Functions that print an annmatrix object

Usage

```
## S3 method for class 'annmatrix'
print(x, nrow = 5, ncol = 5, digits = getOption("digits"), ...)
```

Arguments

`x`	a matrix.
`nrow`	number of rows to display (default is 5).
`ncol`	number of columns to display (default is 5).
`digits`	number of digits to display (default set to getOptions("digits")).
`...`	further arguments passed to or from methods.

Details

annmatrix objects are printed in a shortened form (5 rows and 5 columns by default). In addition the function displays information about available meta-data for rows and columns.

Value

invisibly returns annmatrix object.

Author(s)

Karolis Koncevičius

Examples

```
# construct annmatrix object
x <- matrix(rnorm(20*10), 20, 10)

coldata <- data.frame(group  = rep(c("case", "control"), each = 5),
                      gender = sample(c("M", "F"), 10, replace = TRUE))

rowdata <- data.frame(chr = sample(c("chr1", "chr2"), 20, replace = TRUE),
                      pos = runif(20, 0, 1000000))

X <- annmatrix(x, rowdata, coldata)

print(X)
print(X, 10, 5)
```

```
print(X, 2, 2)

# also works with a list-based matrix
x <- matrix(list(mtcars, iris3, USAccDeaths, rivers), ncol=2)
print(x)
X <- annmatrix(x)
print(X)
```

`scale` *Scaling and Centering of annmatrix Objects*

Description

Centers and scales the columns of an annmatrix object.

Usage

```
## S3 method for class 'annmatrix'
scale(x, center = TRUE, scale = TRUE)
```

Arguments

`x`	annmatrix object.
`center`	either a logical value or a numeric vector of length equal to the number of columns of 'x' (default is TRUE).
`scale`	either a logical value or a numeric vector of length equal to the number of columns of 'x' (default is TRUE).

Details

Behaves exactly as `scale` on a regular matrix with the preservation of 'annmatrix' class being the only difference.

Value

The centered and/or scaled annmatrix object with additional attributes "scaled:center" and "scaled:scale" holding the numbers used for centering and scaling of each column.

Author(s)

Karolis Koncevičius

See Also

`scale.default`

Examples

```
# construct annmatrix object
x <- matrix(rnorm(20*10), 20, 10)

coldata <- data.frame(group  = rep(c("case", "control"), each = 5),
                      gender = sample(c("M", "F"), 10, replace = TRUE))

rowdata <- data.frame(chr = sample(c("chr1", "chr2"), 20, replace = TRUE),
                      pos = runif(20, 0, 1000000))

X <- annmatrix(x, rowdata, coldata)

scale(X)
scale(X, center = colMeans(X))
```

`stack` *Stack an annmatrix object*

Description

Turns annmatrix into a data frame by transforming the matrix along with row and column annotations into separate data frame columns.

Usage

```
## S3 method for class 'annmatrix'
stack(x, ...)
```

Arguments

`x`	annmatrix object.
`...`	further arguments passed to or from methods.

Value

transposed annmatrix object

Author(s)

Karolis Koncevičius

Examples

```
# construct annmatrix object
x <- matrix(rnorm(20*10), 20, 10)

coldata <- data.frame(group  = rep(c("case", "control"), each = 5),
                      gender = sample(c("M", "F"), 10, replace = TRUE))

rowdata <- data.frame(chr = sample(c("chr1", "chr2"), 20, replace = TRUE),
                      pos = runif(20, 0, 1000000))

X <- annmatrix(x, rowdata, coldata)

# stack all information into a long-format data.frame
Y <- stack(X)
```

`subset` *Subset annmatrix Objects*

Description

Methods for selecting a set of rows or columns from annmatrix while keeping the associated annotations intact.

Usage

```
## S3 method for class 'annmatrix'
x[i, j, ..., drop = TRUE]
```

Arguments

`x`	an R object.
`i`	subset for rows.
`j`	subset for columns.
`...`	further arguments passed to or from methods.
`drop`	if TRUE (default) subsetting a single row or column will returned a vector.

Details

`X[i,j]` returns a selected subset of annmatrix object. Row and column annotations are preserved and subsetted where needed. In the special case when only one column or row is selected in order to be consistent with the `matrix` behavior the dimensions of matrix are dropped and a vector is returned. Just like in the case of matrices the additional argument `drop=FALSE` can be provided in order to return a proper matrix instead.

Value

A selected subset of the provided 'annmatrix' object.

Author(s)

Karolis Koncevičius

See Also

`as.annmatrix`

Examples

```
# construct annmatrix object
x <- matrix(rnorm(20*10), 20, 10)

coldata <- data.frame(group  = rep(c("case", "control"), each = 5),
                      gender = sample(c("M", "F"), 10, replace = TRUE))

rowdata <- data.frame(chr = sample(c("chr1", "chr2"), 20, replace = TRUE),
                      pos = runif(20, 0, 1000000))

X <- annmatrix(x, rowdata, coldata)

# annotations are preserved after subsetting
Y <- X[X@chr == "chr1", X$group == "case"]
Y@chr
Y$''

Y[, 1]
Y[, 1, drop = FALSE]
```

`transpose` *Transposing annmatrix Objects*

Description

Transpose annmatrix along with the associated row and column annotations.

Usage

```
## S3 method for class 'annmatrix'
t(x)
```

Arguments

`x`	annmatrix object.

Value

transposed annmatrix object with appropriately adjusted row and column annotations.

Author(s)

Karolis Koncevičius

Examples

```
# construct annmatrix object
x <- matrix(rnorm(20*10), 20, 10)

coldata <- data.frame(group  = rep(c("case", "control"), each = 5),
                      gender = sample(c("M", "F"), 10, replace = TRUE))

rowdata <- data.frame(chr = sample(c("chr1", "chr2"), 20, replace = TRUE),
                      pos = runif(20, 0, 1000000))

X <- annmatrix(x, rowdata, coldata)

# transposes the main matrix along with row and column annotations
Xt <- t(X)

print(X)
print(Xt)

X@chr
Xt$chr
```

Index