Package 'oompaBase'

May 9, 2026

Version 3.2.11

Date 2026-01-09

Title Class Unions, Matrix Operations, and Color Schemes for OOMPA

Depends R (>= 4.4)

Imports methods, graphics, grDevices, stats, cluster

Description Provides the class unions that must be
preloaded in order for the basic tools in the OOMPA (Object-Oriented
Microarray and Proteomics Analysis) project to be defined and loaded.
It also includes vectorized operations for row-by-row means,
variances, and t-tests. Finally, it provides new color schemes.
Details on the packages in the OOMPA project can be found at
<http://oompa.r-forge.r-project.org/>.

License Apache License (== 2.0)

LazyLoad yes

biocViews Infrastructure

URL http://oompa.r-forge.r-project.org/

NeedsCompilation no

Author Kevin R. Coombes [aut, cre]

Maintainer Kevin R. Coombes <krc@silicovore.com>

Repository CRAN

Date/Publication 2026-01-11 06:11:00 UTC

Contents

`ColorCodedPair-class` *Class "ColorCodedPair"*

Description

This class represents a vector of (x,y) pairs, each of which should be plotted in a specific color with a specific symbol.

Usage

```
ColorCodedPair(x, y, ccl)
## S4 method for signature 'ColorCodedPair,missing'
plot(x, y, ...)
```

Arguments

`x`	numeric vector (for `ColorCodedPair`) or an object of class `ColorCodedPair` (for `plot`)
`y`	numeric vector
`ccl`	list of `ColorCoding` objects
`...`	additional arguments are as in the underlying generic methods.

Details

It is often necessary with microarray data to produce multiple plots, where each point on the plot corresponds to a gene or a spot on the microarray. Across multiple plots, we often want to use symbols or colors to mark subsets of he genes with certain properties. The `ColorCodedPair` class works in tandem with the `ColorCoding` class to make it easier to maintain consistent plotting conventions across multiple graphs.

Value

The constructor returns a valid `ColorCodedPair` object.

The `plot` method invisibly returns the object being plotted.

Slots

`x` numeric vector

`y` numeric vector

`colorCodingList` list of `ColorCoding` objects

Methods

plot(object, ...) Plot the `ColorCodedPair` object, with appropriate colors and symbols (of course).

Author(s)

Kevin R. Coombes <krc@silicovore.com>, P. Roebuck <proebuck@mdanderson.org>

See Also

colorCode, ColorCoding

Examples

```
showClass("ColorCodedPair")

theta <- (0:360)*pi/180
x <- cos(theta)
y <- sin(theta)
xp <- x > 0
yp <- y > 0
colors <- list(ColorCoding(xp&yp, oompaColor$EXPECTED),
               ColorCoding(xp&!yp, oompaColor$OBSERVED),
               ColorCoding(!xp&yp, oompaColor$PERMTEST),
               ColorCoding(!xp&!yp, oompaColor$FITTED))
plot(ColorCodedPair(x, y, colors))

plot(ColorCodedPair(theta, x, colors))

plot(ColorCodedPair(theta, y, colors),
     xlab='angle in radians', ylab='sine', main='colored sine')
```

ColorCoding-class *Class "ColorCoding"*

Description

A class for associating plotting symbols and colors with a logical vector or with levels of a factor.

Usage

```
ColorCoding(v, color, mark = 1)
colorCode(fac, colorScheme = 1:length(levels(fac)), mark = 1)
```

Arguments

v	a logical vector
color	a string or integer representing a color
mark	an integer representing a plotting symbol, or list of plotting symbols that should be associated with levels of the factor fac
fac	A factor
colorScheme	A list of colors that should be associated with levels of the factor fac

Details

It is often necessary with microarray data to produce multiple plots, where each point on the plot corresponds to a gene or a spot on the microarray. Across multiple plots, we often want to use symbols or colors to mark subsets of he genes with certain properties. The `ColorCoding` class works in tandem with the `ColorCodedPair` class to make it easier to maintain consistent plotting conventions across multiple graphs.

Value

The constructor returns a valid object of the `ColorCoding` class.

The `colorCode` function provides a simple interface to associate colors and symbols with the levels of a factor. It returns a list of valid `ColorCoding` objects that can be passed directly to the constructor of a `ColorCodedPair` object.

Objects from the Class

Although objects can be created using `new`, the preferred method is to use the constructor function, `ColorCoding`. To create a set of colors and symbols associated with different levels of a factor, use the `colorCode` function.

Slots

`v` a logical vector

`color` a string or integer representing a color

`mark` an integer representing a plotting symbol

Methods

There are no specialized methods for objects of this class; all activities can be performed by directly accessing the documented slots.

Author(s)

Kevin R. Coombes <krc@silicovore.com>, P. Roebuck <proebuck@mdanderson.org>

See Also

`ColorCodedPair`

Examples

```
showClass("ColorCoding")

theta <- (0:360)*pi/180
x <- cos(theta)
y <- sin(theta)
xp <- x > 0
yp <- y > 0
colors <- list(ColorCoding(xp&yp, oompaColor$BORING),
```

```
                ColorCoding(xp&!yp, oompaColor$TOP.TEN),
                ColorCoding(!xp&yp, oompaColor$BOTTOM.TEN),
                ColorCoding(!xp&!yp, oompaColor$CONFIDENCE.CURVE))
plot(ColorCodedPair(x, y, colors))

plot(ColorCodedPair(theta, x, colors))

plot(ColorCodedPair(theta, y, colors),
     xlab='angle in radians', ylab='sine', main='colored sine')

fac <- factor(rep(c('left', 'right'), c(180, 181)))
colors <- colorCode(fac, c('blue', 'red'))
plot(ColorCodedPair(x, y, colors))

## cleanup
rm(x, y, xp, yp, theta, colors, fac)

colorList <- c(oompaColor$BORING, oompaColor$SIGNIFICANT,
                oompaColor$EXPECTED, oompaColor$OBSERVED,
                oompaColor$PERMTEST, oompaColor$FITTED,
                oompaColor$CENTRAL.LINE, oompaColor$CONFIDENCE.CURVE,
                oompaColor$BAD.REPLICATE, oompaColor$WORST.REPLICATE,
                oompaColor$FOLD.DIFFERENCE, oompaColor$BAD.REPLICATE.RATIO,
                oompaColor$TOP.TEN, oompaColor$BOTTOM.TEN,
                oompaColor$TOP.TEN.SOLO, oompaColor$BOTTOM.TEN.SOLO
                )
plot(c(1,4), c(1,4), type='n')
for (i in 1:4) {
  for (j in 1:4) {
    points(i,j, col=colorList[i + 4*(j-1)], pch=16, cex=4)
  }
}
rm(colorList, i, j)
```

`colorschemes` *Color Schemes for Images and Heat Maps*

Description

Create a vector of `N` contiguous colors.

Usage

```
redscale(N)
greenscale(N)
bluescale(N)
blueyellow(N)
cyanyellow(N)
redgreen(N)
jetColors(N)
```

```
grayscale(N)
greyscale(N)
wheel(N, sat = 1)
```

Arguments

N	an integer; the number of distinct levels in the color map
sat	a real number between 0 and 1; the saturation amount

Details

The color maps that ship with R (see, for example, terrain.colors) do not include the most common color maps used in publications in the microarray literature. This collection of color maps expands the available options. The functions redscale, greenscale, and bluescale each range from pure black for low values to a pure primary color for high values. The synonyms graysale and greysale range from pure black to pure white.

The redgreen color map ranges from pure green at the low end, through black in the middle, to pure red at the high end. Although this is the most common color map used in the microarray literature, it will prove problematic for individuals with red-green color-blindness.

The blueyellow color map ranges from pure blue at the low end, through gray in the middle, to pure yellow at the high end.

The jetColors map tries to reproduce the default "jet" color map from MATLAB.

The cyanyellow color map was added to provide a divergent map that should be usable by the majority of individuals whose vision has a color deficit. It ranges from cyan (a mixture of blue and green) at the low to yellow (a mixture of red and green) at the high end. Since the vast majority of color deficits arise because an individual lacks cones for one of the three primary colors (red, green, or blue), this colormap should stil provide adequate contrasts.

Value

A character vector 'cv' of color names. This can be used to create a user-defined color palette for subsequent graphics by 'palette(cv)' or directly in a 'col=' specification in 'par' or in graphics functions such as 'image' or 'heatmap'.

BUGS

The names redgreen and blueyellow are inconsistent with respect to which color represents low values and which color represents high values. It is too late to fix this.

Author(s)

Kevin R. Coombes <krc@silicovore.com>

See Also

rainbow, topo.colors, terrain.colors, heat.colors, rgb, image, heatmap, palette.

Examples

```
data <- matrix(1:1024, nrow=1024)
image(data, col=bluescale(64))
image(data, col=redgreen(32))
image(data, col=redscale(128))
image(data, col=blueyellow(64))
image(data, col=jetColors(64))
image(data, col=grayscale(64))
image(data, col=cyanyellow(64))
rm(data) # cleanup
```

`matrixT` *Multiple T Tests by Matrix Multiplication*

Description

Utility functions for computing vectors of row-by-row means, variances, and t-statistics.

Usage

```
matrixMean(x,na.rm=FALSE)
matrixVar(x, xmean, na.rm=FALSE)
matrixT(m, v, na.rm=FALSE)
matrixUnequalT(m, v)
matrixPairedT(m, v, pf)
```

Arguments

`x`	a matrix
`xmean`	a numeric vector or single-column matrix
`m`	a matrix
`na.rm`	a logical value indicating whether means, variances, and t-statistics should be computed after omitting NA values from individual rows of the data matrix.
`v`	a logical vector of length equal to the number of columns of `m`
`pf`	a numerical vector of length equal to the number of columns of `m`, indicating which samples should be paired

Value

`matrixMean` returns a single-column matrix containing the row-by-row means of `x`.

`matrixVar` returns a single-column matrix containing the row-by-row means of `x`, assuming that `xmean` contains the corresponding mean values.

`matrixT` returns a single-column matrix of t-statistics from a two-sample t-test comparing the columns for which `v` is true to those for which `v` is false.

`matrixPairedT` returns a single-column matrix of t-statistics from a paired t-test.

`matrixUnequalT` returns a list with two components: `tt` is a single-column matrix of t-statistics from a two-sample unequal variance t-test comparing the columns for which `v` is true to those for which `v` is false, and `df` is a single-column matrix of the degrees of freedom associated with each row..

Author(s)

Kevin R. Coombes <krc@silicovore.com>

Examples

```
ng <- 1000
ns <- 50
dat <- matrix(rnorm(ng*ns), ncol=ns)
clas <- factor(rep(c('Good', 'Bad'), each=25))
myMean <- matrixMean(dat)
myVar  <- matrixVar(dat, myMean)
plot(myMean, myVar)


myT <- matrixT(dat, clas)
hist(myT)

rm(ng, ns, dat, myMean, myVar, myT)
```

---

numeric or NULL-class *Class "numeric or NULL"*

Description

This class is a union that can represent either a numeric vector or a NULL value.

Objects from the Class

A virtual Class: No objects may be created from it.

Methods

No methods defined with class "numeric or NULL" in the signature.

Author(s)

Kevin R. Coombes <krc@silicovore.com>

semantic colors *Pre-defined colors for microarray plots*

Description

A collection of predefined color names to help ensure consistency in multiple graphical displays of microarray data.

`oompaColor`: a list containing named components allowing the user to systematically use colors for different interpretations.

`oompaColor$BORING`: Used to mark uninteresting points in a plot; gray.

`oompaColor$SIGNIFICANT`: Used to mark points that are statistically significant; red

`oompaColor$EXPECTED`: Used to draw curves representing an expected distribution; blue

`oompaColor$OBSERVED`: Used to draw curves indicating the observed distribution; darkgreen

`oompaColor$PERMTEST`: Used to draw curves indicating distributions derived from a permutation test; magenta

`oompaColor$FITTED`: Used to draw curves obtained by some fitting procedure, such as loess; orange

`oompaColor$CENTRAL.LINE`: Used to draw lines through the centers of distributions or expected values; blue

`oompaColor$CONFIDENCE.CURVE`: Used to draw confidence bounds around curves; red3

`oompaColor$BAD.REPLICATE`: Used to indicate highly variable points; purple1

`oompaColor$WORST.REPLICATE`: Used to mark extraordinarily variable points; purple3

`oompaColor$FOLD.DIFFERENCE`: Used to indicate points with large fold difference; skyblue

`oompaColor$BAD.REPLICATE.RATIO`: Used to flag points for which the ratios at replicate spots are highly variable; violetred

`oompaColor$TOP.TEN`: Used to mark points in the "top ten" list; cadetblue

`oompaColor$BOTTOM.TEN`: Used to mark points in "bottom ten" list of most underexpressed genes; pink

`oompaColor$BOTTOM.TEN.SOLO`: Use unknown; palegreen

`oompaColor$TOP.TEN.SOLO`: Use unknown; deeppink

Examples

```
oompaColor
x <- seq(0, 2*pi, by=0.1)
plot(x, sin(x), col=oompaColor$BORING)
```

Index