Package 'colorplane'

February 2, 2026

Type Package

Title Basic S4 Classes and Methods for Mapping Between Numeric Values and Colors

Version 0.6.0

Date 2026-02-01

Encoding UTF-8

Maintainer Bradley R Buchsbaum <brad.buchsbaum@gmail.com>

Description A simple set of classes and methods for mapping between scalar intensity values and colors. There is also support for layering maps on top of one another using alpha composition.

License MIT + file LICENSE

RoxygenNote 7.3.3

Imports assertthat, methods

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Collate 'all_class.R' 'all_generic.R' 'color_plane.R' 'color_scale.R'

NeedsCompilation no

Author Bradley R Buchsbaum [aut, cre]

Repository CRAN

Date/Publication 2026-02-01 23:50:02 UTC

Contents

`alpha_channel` *alpha_channel*

Description

extract the alpha channel

Usage

```
alpha_channel(x, ...)

## S4 method for signature 'HexColorPlane'
alpha_channel(x, normalize = TRUE)

## S4 method for signature 'ConstantColorPlane'
alpha_channel(x, normalize = TRUE)

## S4 method for signature 'RGBColorPlane'
alpha_channel(x, normalize = TRUE)
```

Arguments

`x`	the object to extract alpha channel from
`...`	extra args
`normalize`	divide by 255

Value

a numeric vector of alpha channel values

Examples

```
cp <- IntensityColorPlane(seq(1,5), cols=rainbow(25))
cl <- map_colors(cp, irange=c(0,50))
stopifnot(length(alpha_channel(cl)) == 5)
```

as_hexcol *convert to hex colors*

Description

convert to hex colors

Usage

```
as_hexcol(x, ...)

## S4 method for signature 'RGBColorPlane'
as_hexcol(x)

## S4 method for signature 'HexColorPlane'
as_hexcol(x)
```

Arguments

x	the object to convert
...	extra args

Value

a character vector of ex colors

See Also

rgb

as_rgb *convert to rgb colors*

Description

convert to rgb colors

Usage

```
as_rgb(x, ...)

## S4 method for signature 'RGBColorPlane'
as_rgb(x)

## S4 method for signature 'HexColorPlane'
as_rgb(x)
```

```
## S4 method for signature 'ConstantColorPlane'
as_rgb(x)
```

Arguments

x	the object to convert
...	extra args

Value

a numeric matrix of rgb components

Examples

```
cp <- IntensityColorPlane(seq(1,100), cols=rainbow(25))
cl <- map_colors(cp, irange=c(0,50))
rgbcols <- as_rgb(cl)
```

---

blend_colors *blend two color planes*

Description

given two color planes, generate a new color plane by blending the colors using the supplied alpha multiplier.

Usage

```
blend_colors(bottom, top, alpha)

## S4 method for signature 'ColorPlane,ColorPlane,numeric'
blend_colors(bottom, top, alpha = 1)

## S4 method for signature 'ColorPlane,ColorPlane,missing'
blend_colors(bottom, top)

## S4 method for signature 'HexColorPlane,RGBColorPlane,numeric'
blend_colors(bottom, top, alpha)

## S4 method for signature 'HexColorPlane,ConstantColorPlane,numeric'
blend_colors(bottom, top, alpha = 1)
```

Arguments

bottom	the bottom color plane
top	the top color plane
alpha	the alpha overlay value.

Details

The functions in this package blend colors based on the "over" operator where ‘top‘ if foreground and ‘bottom‘ is background.

Value

a new ColorPlane instance with ‘top‘ and ‘bottom‘ alpha-blended.

References

https://en.wikipedia.org/wiki/Alpha_compositing

Examples

```
top <- IntensityColorPlane(1:5, cols=rainbow(5))
bottom <- IntensityColorPlane(1:5, cols=rev(rainbow(5)))

top <- map_colors(top)
bottom <- map_colors(bottom)
bc <- blend_colors(bottom, top, .5)
```

---

col2hex *convert color name to hex character string*

Description

convert color name to hex character string

Usage

```
col2hex(cname, alpha = 1)
```

Arguments

cname	one or more color names, e.g. "red"
alpha	the value of the alpha channel, ranging from 0 to 1 (default is 1)

Value

a vector of hex color values, one per color name

`ColorPlane-class` *ColorPlane*

Description

ColorPlane

Slots

`clr` a field of colors

`ConstantColorPlane-class`
ConstantColorPlane

Description

ConstantColorPlane constructor taking a single hex ‘character‘ vector defining a constant color plane.

Usage

```
ConstantColorPlane(clr)
```

Arguments

`clr`	a single hex color as a ‘character‘ vector of length 1 defining the constant color.

Value

a new `ConstantColorPlane` instance

Slots

`clr` the constant color as hex value

Examples

```
cp <- ConstantColorPlane(clr="#FF0000")
```

DiscreteColorPlane-class *DiscreteColorPlane*

Description

DiscreteColorPlane constructor taking list with names mapping to color values in hex representation. This object is used when one has a one to one mapping between discrete set of strings/values to discrete set of colors.

Usage

```
DiscreteColorPlane(lookup)
```

Arguments

lookup	a "lookup table", which is a named list mapping discrete values to hex colors

Value

a new DiscreteColorPlane instance

Slots

lookup a lookup table mapping values to hex colors

Examples

```
lookup <- as.list(col2hex(c("red", "blue", "green")))
names(lookup) <- c("a", "b", "c")
cp <- DiscreteColorPlane(lookup)

values <- c("a", "b", "c", "a", "c")
```

HexColorPlane-class *HexColorPlane*

Description

HexColorPlane constructor taking a ‘character‘ vector of colors to define a color plane.

Usage

```
HexColorPlane(clr)
```

Arguments

clr	a vector of hex colors

Value

a new HexColorPlane instance

Slots

clr a character vector of hex color codes

IntensityColorPlane-class

IntensityColorPlane

Description

An association of intensities and colors

IntensityColorPlane constructor

Usage

```
IntensityColorPlane(intensity, cols = rainbow(255), alpha = 1)
```

Arguments

intensity	a numeric vector of intensity values
cols	a vector of hex character codes
alpha	a vector of alpha values ranging from 0 to 1

Value

a new IntensityColorPlane instance

Slots

intensity a vector of intensity values

alpha a vector of alpha values

colmap a color map containing a vector of hex character codes

map_colors *map data values to a set of colors*

Description

instantiate a vector of colors from a ColorPlane specification.

Usage

```
map_colors(x, ...)

## S4 method for signature 'ConstantColorPlane'
map_colors(x)

## S4 method for signature 'HexColorPlane'
map_colors(x)

## S4 method for signature 'DiscreteColorPlane'
map_colors(x, values, ...)

## S4 method for signature 'IntensityColorPlane'
map_colors(x, alpha = NULL, threshold = NULL, irange = NULL)
```

Arguments

x	the object to map over
...	extra args
values	the values to map to colors via the discrete lookup table
alpha	alpha multiplier from 0 to 1. If NULL, uses the alpha value stored in the object.
threshold	two-sided threshold as a 2-element vector, e.g. 'threshold=c(-3,3)' indicating two-sided transparency thresholds.
irange	the intensity range defining min and max of scale.

Value

a HexColorPlane instance containing the mapped colors

Examples

```
cp <- IntensityColorPlane(seq(1,100), cols=rainbow(25))
cl <- map_colors(cp, irange=c(0,50))
stopifnot(cl@clr[50] == rainbow(25)[25])
```

rgb2hex *convert rgb colors to hex colors*

Description

convert rgb colors to hex colors

Usage

```
rgb2hex(r, g, b, alpha)
```

Arguments

r	the red color component
g	the green color component
b	the blue color component
alpha	the alpha component

Value

a hex color represenation as ‘character‘ vector

RGBColorPlane-class *RGBColorPlane*

Description

RGBColorPlane constructor taking a 3- or 4-column numeric matrix of RGB(A) colors in the 0-255 range.

Usage

```
RGBColorPlane(clr)
```

Arguments

clr	a matrix of colors where the first column is red, second column is green, third column is blue, and optional fourth column is alpha.

Value

a new RGBColorPlane instance

Slots

clr a numeric matrix of RGBA color values

Examples

```
rgba_cmat <- rbind(c(255,0,0,255),
              c(0, 255, 0, 255),
              c(0, 0, 255, 0))

cp <- RGBColorPlane(rgba_cmat)
stopifnot(all(cp@clr[1,] == c(255,0,0,255)))
```

Index