Package 'chameleon'

October 12, 2022

Version 0.2-3

Date 2022-09-27

Title Automatic Colors for Multi-Dimensional Data

Description Assign distinct colors to arbitrary multi-dimensional data, considering its structure.

RoxygenNote 7.2.1

Encoding UTF-8

License MIT + file LICENSE

Imports clue, ggplot2, grDevices, stats, umap

Depends R (>= 3.5.0)

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Oren Ben-Kiki [aut, cre],
Weizmann Institute of Science [cph]

Maintainer Oren Ben-Kiki <oren@ben-kiki.org>

Repository CRAN

Date/Publication 2022-09-27 06:20:02 UTC

R topics documented:

`data_colors` *Compute colors for multi-dimensional data.*

Description

Given a matrix of observation/element rows and variable/measurement columns, compute a color for each row (or group of rows) such that the colors are distinct, and where more-similar colors roughly designate more-similar data rows (or groups of rows).

Usage

```
data_colors(
  data,
  run_umap = TRUE,
  groups = NULL,
  minimal_saturation = 33,
  minimal_lightness = 20,
  maximal_lightness = 80
)
```

Arguments

`data`	A matrix whose rows represent elements/observations and columns represent variables/measurements.
`run_umap`	A boolean specifying whether to run UMAP on the data to convert it to 3D (by default, `TRUE`). If `FALSE`, the data matrix must have exactly 3 columns and will be used as-is.
`groups`	An optional array with an entry per row containing the identifier of the group the row belongs to.
`minimal_saturation`	Exclude colors whose saturation (`hypot(a, b)` in CIELAB color space) is less than this value (by default, 33).
`minimal_lightness`	Exclude colors whose lightnes (`l` in CIELAB color space) is less than this value (by default, 20).
`maximal_lightness`	Exclude colors whose lightnes (`l` in CIELAB color space) is more than this value (by default, 80).

Details

This is intended to provide a "reasonable" set of colors to "arbitrary" data, for use as a convenient default when investigating unknown data sets. It is not meant to replace hand-picked colors tailored for specific data (e.g. using red colors for "bad" rows and green colors for "good" rows).

This ensures all colors are distinct by packing the (visible part) of the CIELAB color space with the needed number of spheres. To assign the colors to the data, it uses UMAP to reduce the data to 3D.

It then uses principal component analysis to represent both the chosen colors (3D sphere centers) and the (3D UMAP) data as point clouds with coordinates in the range 0-1, and finally uses a stable matching algorithm to map these point clouds to each other, thereby assigning a color to each data row. If the data is grouped, then the center of gravity of each group is used to generate a color for each group.

Value

An array with one entry per row, whose names are the matrix `rownames`, containing the color of each row. If `groups` was specified, the array will contain one entry per unique group identifier, whose names are the `as.character` group identifiers, containing the color of each group.

Examples

```
chameleon::data_colors(stackloss)
```

---

distinct_colors *Pick a number of distinct colors.*

Description

This ensures all colors are distinct by packing the (visible part) of the CIELAB color space with the needed number of spheres, and using their centers to generate the colors.

Usage

```
distinct_colors(
  n,
  minimal_saturation = 33,
  minimal_lightness = 20,
  maximal_lightness = 80
)
```

Arguments

`n`	The requested (positive) number of colors.
`minimal_saturation`	Exclude colors whose saturation (`hypot(a, b)` in CIELAB color space) is less than this value (by default, 33).
`minimal_lightness`	Exclude colors whose lightnes (`l` in CIELAB color space) is less than this value (by default, 20).
`maximal_lightness`	Exclude colors whose lightnes (`l` in CIELAB color space) is more than this value (by default, 80).

Value

A list with two elements, `name` containing the color names and `lab` containing a matrix with a row per color and three columns containing the `l`, `a` and `b` coordinates of each color.

Examples

```
chameleon::distinct_colors(8)
```

---

pbmc *Sample scRNA data of PBMC metacells.*

Description

This is a list with the following elements:

Usage

```
data(pbmc)
```

Format

A list with the three elements described above.

Details

'umis' - a matrix, containing ~1.5K metacells (rows), and for each one, the UMI count (# of detected RNA molecules) for each of ~600 different "feature" genes (columns).

'types' - a vector of cell type names assigned to each metacell using a supervised analysis pipeline.

'umap' - a matrix with 2 columns containing 2D UMAP x,y coordinates for each metacell.

Examples

```
data(pbmc)
fractions <- pbmc$umis / rowSums(pbmc$umis)
log_fractions <- log2(fractions + 1e-5)
type_colors <- chameleon::data_colors(log_fractions, group=pbmc$types)
plot(pbmc$umap, col=type_colors[pbmc$types], pch=19, cex=0.6)
legend('topleft', legend=names(type_colors), col=type_colors, lty=1, lwd=3, cex=0.8)
```

`scale_color_chameleon` *Setup a color scale of distinct discrete colors in ggplot2.*

Description

This is a thin wrapper to `ggplot2::discrete_scale('colour', 'chameleon', ...)`, which uses the colors chosen by invoking `distinct_colors`. The order of the colors is arbitrary. If the data has some structure the colors should reflect, use one of the many palettes available in R, or using `data_colors` for automatically matching the colors to the structure of multi-dimensional data.

Usage

```
scale_color_chameleon(
  minimal_saturation = 33,
  minimal_lightness = 20,
  maximal_lightness = 80,
  ...
)
```

Arguments

`minimal_saturation`	Exclude colors whose saturation (`hypot(a, b)` in CIELAB color space) is less than this value (by default, 33).
`minimal_lightness`	Exclude colors whose lightnes (`l` in CIELAB color space) is less than this value (by default, 20).
`maximal_lightness`	Exclude colors whose lightnes (`l` in CIELAB color space) is more than this value (by default, 80).
`...`	Additional parameters for `discrete_scale`.

Examples

```
library(ggplot2)
data(pbmc)
frame <- as.data.frame(pbmc$umap)
frame$type <- pbmc$types
ggplot(frame, aes(x=xs, y=ys, color=type)) +
    geom_point(size=0.75) +
    scale_color_chameleon() +
    theme(legend.text=element_text(size=12), legend.key.height=unit(14, 'pt'))
```

`scale_fill_chameleon` *Setup a fill scale of distinct discrete colors in ggplot2.*

Description

This is a thin wrapper to `ggplot2::discrete_scale('fill', 'chameleon', ...)`, which uses the colors chosen by invoking `distinct_colors`. The order of the colors is arbitrary. If the data has some structure the colors should reflect, use one of the many palettes available in R, or using `data_colors` for automatically matching the colors to the structure of multi-dimensional data.

Usage

```
scale_fill_chameleon(
  minimal_saturation = 33,
  minimal_lightness = 20,
  maximal_lightness = 80,
  ...
)
```

Arguments

`minimal_saturation`
: Exclude colors whose saturation (`hypot(a, b)` in CIELAB color space) is less than this value (by default, 33).

`minimal_lightness`
: Exclude colors whose lightnes (`l` in CIELAB color space) is less than this value (by default, 20).

`maximal_lightness`
: Exclude colors whose lightnes (`l` in CIELAB color space) is more than this value (by default, 80).

`...`
: Additional parameters for `discrete_scale`.

Examples

```
library(ggplot2)
data(pbmc)
frame <- as.data.frame(pbmc$umap)
frame$type <- pbmc$types
ggplot(frame, aes(x=xs, y=ys, fill=type)) +
    geom_point(size=0.75, shape=21, color="black", stroke=0.1) +
    scale_fill_chameleon() +
    theme(legend.text=element_text(size=12), legend.key.height=unit(14, 'pt'))
```

Index