Package ‘colorrepel’

September 17, 2024

Title Repel Visually Similar Colors for Colorblind Users in Various Plots

Version 0.1.0

Description Iterate and repel visually similar colors away in various 'ggplot2' plots. When many groups are plotted at the same time on multiple axes, for instance stacked bars or scatter plots, effectively ordering colors becomes difficult. This tool iterates through color combinations to find the best solution to maximize visual distinctness of nearby groups, so plots are more friendly toward colorblind users. This is achieved by two distance measurements, distance between groups within the plot, and CIELAB color space distances between colors as described in Carter et al., (2018) <doi:10.25039/TR.015.2018>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 3.5.0)

Imports grDevices, matrixStats, Matrix, distances, stats, dqrng, gtools, purrr, dplyr, ggplot2, ggrepel, ggalt

Suggests colorspace

URL `https://github.com/raysinensis/color_repel`

ByteCompile true

NeedsCompilation no

Author Rui Fu [cre, aut, cph] (<`https://orcid.org/0000-0001-8183-4549`>)

Maintainer Rui Fu <`raysinensis@gmail.com`>

Repository CRAN

Date/Publication 2024-09-17 11:50:08 UTC

Contents

`average_clusters` *Average expression values per cluster*

Description

Average expression values per cluster

Usage

```
average_clusters(
  mat,
  metadata,
  cluster_col = "cluster",
  if_log = TRUE,
  cell_col = NULL,
  low_threshold = 0,
  method = "mean",
  output_log = TRUE,
  cut_n = NULL
)
```

Arguments

`mat`	expression matrix
`metadata`	data.frame or vector containing cluster assignments per cell. Order must match column order in supplied matrix. If a data.frame provide the cluster_col parameters.
`cluster_col`	column in metadata with cluster number
`if_log`	input data is natural log, averaging will be done on unlogged data
`cell_col`	if provided, will reorder matrix first
`low_threshold`	option to remove clusters with too few cells
`method`	whether to take mean (default), median, 10% truncated mean, or trimean, max, min
`output_log`	whether to report log results
`cut_n`	set on a limit of genes as expressed, lower ranked genes are set to 0, considered unexpressed

Value

average or other desired calculation by group/cluster matrix

`average_clusters_rowwise` *Rowwise math from matrix/data.frame per cluster based on another vector/metadata, similar to clustifyr::average_clusters but ids as rows*

Description

Rowwise math from matrix/data.frame per cluster based on another vector/metadata, similar to clustifyr::average_clusters but ids as rows

Usage

```
average_clusters_rowwise(
  mat,
  metadata,
  cluster_col = "cluster",
  if_log = FALSE,
  cell_col = NULL,
  low_threshold = 0,
  method = "mean",
  output_log = FALSE,
  cut_n = NULL,
  trim = FALSE
)
```

Arguments

`mat`	expression matrix
`metadata`	data.frame or vector containing cluster assignments per cell. Order must match column order in supplied matrix. If a data.frame provide the cluster_col parameters.
`cluster_col`	column in metadata with cluster number
`if_log`	input data is natural log, averaging will be done on unlogged data
`cell_col`	if provided, will reorder matrix first
`low_threshold`	option to remove clusters with too few cells
`method`	whether to take mean (default), median, 10% truncated mean, or trimean, max, min
`output_log`	whether to report log results
`cut_n`	set on a limit of genes as expressed, lower ranked genes are set to 0, considered unexpressed
`trim`	whether to remove 1 percentile when doing min caluculation

Value

average expression matrix, with genes for row names, and clusters for column names

Examples

```
mat <- average_clusters_rowwise(data.frame(y = c(1, 2, 3, 4, 5, 6),
x = c(1, 2, 3, 4, 5, 6)), metadata = c(1, 2, 1, 2, 1, 2), method = "min")
```

by_cluster_sampling *Balanced downsampling of matrix/data.frame based on cluster assignment vector*

Description

Balanced downsampling of matrix/data.frame based on cluster assignment vector

Usage

```
by_cluster_sampling(df, vec, frac, seed = 34)
```

Arguments

df	expression matrix or data.frame
vec	vector of ids
frac	fraction 0-1 to downsample to
seed	sampling randomization seed

Value

list with new downsampled matrix/data.frame and id vector

Examples

```
res <- by_cluster_sampling(data.frame(y = c(1, 2, 3, 4, 5, 6)),
vec = c(1, 2, 1, 2, 1, 2), frac = 0.5)
```

calc_distance *Distance calculations for spatial coord*

Description

Distance calculations for spatial coord

Usage

```
calc_distance(
  coord,
  metadata,
  cluster_col = "cluster",
  collapse_to_cluster = FALSE
)
```

Arguments

`coord` dataframe or matrix of spatial coordinates, cell barcode as rownames

`metadata` data.frame or vector containing cluster assignments per cell. Order must match column order in supplied matrix. If a data.frame provide the cluster_col parameters.

`cluster_col` column in metadata with cluster number

`collapse_to_cluster`
instead of reporting min distance to cluster per cell, summarize to cluster level

Value

min distance matrix

color_repel *Reorder ggplot colors to maximize color differences in space*

Description

Reorder ggplot colors to maximize color differences in space

Usage

```
color_repel(
  g,
  coord = NULL,
  groups = NULL,
  nsamp = 50000,
  sim = NULL,
  severity = 0.5,
  verbose = FALSE,
  downsample = 5000,
  seed = 34,
  col = "colour",
  autoswitch = TRUE,
  out_worst = FALSE
)
```

Arguments

g	ggplot plot object
coord	coordinates, default is inferred
groups	groups corresponding to color/fill, default is inferred
nsamp	how many random sampling color combinations to test, default 50000
sim	passing a colorbind simulation function if needed
severity	severity of the color vision defect, between 0 and 1
verbose	whether to print messages
downsample	downsample when too many datapoints are present
seed	sampling randomization seed
col	colour or fill in ggplot
autoswitch	try to switch between colour and fill automatically
out_worst	output the worst combination instead of best

Value

vector of reordered colors

Examples

```
a <- ggplot2::ggplot(ggplot2::mpg, ggplot2::aes(displ, hwy)) +
 ggplot2::geom_point(ggplot2::aes(color = as.factor(cyl)))
new_colors <- color_repel(a)
b <- a + ggplot2::scale_color_manual(values = new_colors)
```

get_labs *Extract custom labels from ggplot object*

Description

Extract custom labels from ggplot object

Usage

```
get_labs(g)
```

Arguments

g	ggplot object

Value

named vector of labels

Examples

```
a <- ggplot2::ggplot(ggplot2::mpg, ggplot2::aes(displ, hwy)) +
ggplot2::geom_point(ggplot2::aes(color = as.factor(cyl))) +
ggplot2::geom_text(ggplot2::aes(label = model))
get_labs(a)
```

`gg_color_repel` *Wrapper to reorder ggplot colors to maximize color differences in space*

Description

Wrapper to reorder ggplot colors to maximize color differences in space

Usage

```
gg_color_repel(
  g = ggplot2::last_plot(),
  col = "colour",
  sim = NULL,
  severity = 0.5,
  verbose = FALSE,
  downsample = 5000,
  nsamp = 50000,
  seed = 34,
  autoswitch = TRUE,
  out_worst = FALSE,
  repel_label = FALSE,
  encircle = FALSE,
  encircle_alpha = 0.25,
  encircle_expand = 0.02,
  encircle_shape = 0.5,
  encircle_threshold = 0.1,
  encircle_nmin = 0.1,
  ...
)
```

Arguments

`g`	ggplot plot object
`col`	colour or fill in ggplot
`sim`	passing a colorbind simulation function if needed
`severity`	severity of the color vision defect, between 0 and 1
`verbose`	whether to print messages
`downsample`	downsample when too many datapoints are present
`nsamp`	how many random sampling color combinations to test, default 50000
`seed`	sampling randomization seed
`autoswitch`	try to switch between colour and fill automatically
`out_worst`	output the worst combination instead of best
`repel_label`	whether to add centroid labels with ggrepel
`encircle`	whether to draw geom_encircle by cluster
`encircle_alpha`	alpha argument passed to geom_encircle
`encircle_expand`	expand argument passed to geom_encircle
`encircle_shape`	shape/smoothing argument passed to geom_encircle
`encircle_threshold`	threshold for removing outliers
`encircle_nmin`	number of near neighbors for removing outliers
`...`	passed to repel_label

Value

new ggplot object

Examples

```
a <- ggplot2::ggplot(ggplot2::mpg, ggplot2::aes(displ, hwy)) +
ggplot2::geom_point(ggplot2::aes(color = as.factor(cyl)))
b <- gg_color_repel(a, col = "colour")
```

`label_repel` *ggrepel labeling of clusters*

Description

ggrepel labeling of clusters

Usage

```
label_repel(
  g,
  group_col = "group",
  x = "x",
  y = "y",
  txt_pt = 3,
  remove_current = TRUE,
  layer = "auto"
)
```

Arguments

`g`	ggplot object or data.frame
`group_col`	column name in data.frame, default to "group" in ggplot data
`x`	column name in data.frame for x
`y`	column name in data.frame for y
`txt_pt`	text size
`remove_current`	whether to remove current text
`layer`	text layer to remove, defaults to last

Value

function, if data.frame input, or new ggplot object

Examples

```
g <- label_repel(ggplot2::ggplot(mtcars, ggplot2::aes(x = hp, y = wt, color = as.character(cyl))) +
  ggplot2::geom_point(), remove_current = FALSE)
```

`matrix2_score` *Score matrix distances*

Description

Score matrix distances

Usage

```
matrix2_score(dist1, dist2)
```

Arguments

`dist1`	distanct matrix 1
`dist2`	distanct matrix 2

Value

numeric score

matrix2_score_n — *Score matrix distances in multiple combinations*

Description

Score matrix distances in multiple combinations

Usage

```
matrix2_score_n(
  dist1,
  dist2,
  n = min(factorial(ncol(dist2)) * 10, 20000),
  verbose = FALSE,
  seed = 34,
  out_worst = FALSE
)
```

Arguments

`dist1`	distanct matrix 1
`dist2`	distanct matrix 2
`n`	number of iterations
`verbose`	whether to output more messages
`seed`	random seed
`out_worst`	instead of default output of best combination, output worst instead

Value

reordered vector

Index