Package ‘SpacoR’

April 9, 2024

Version 0.1

Title Spatially constrained color profiling for visualizing spatial data

Description Spatially constrained color profiling for visualizing spatial data

Depends R (>= 3.2)

Imports colorspace,
FNN,
OpenImageR,
stats

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

URL `https://github.com/BrainStOrmics/SpacoR`

NeedsCompilation no

Author zehua jing [aut],
bolin yang [cre]

Maintainer bolin yang `<yangbolin22@mails.ucas.ac.cn>`

R topics documented:

assign_color 2
colorize 3
map_graph 4
matrix_distance 5
spatial_distance 7

Index **9**

assign_color *Core color mapping function for SpacoR.*

Description

Assign Colors to Clusters Based on Distance Matrix. This function assigns colors to clusters based on a given distance matrix. It supports colorblind-friendly options and can automatically generate a color palette, use a predefined palette, or extract colors from an image. SpacoR provides 3 basic color mapping mode in this function:

1. Optimize the mapping of a pre-defined color palette.
2. Extract colors from image.
3. Automatically generate colors within colorspace.

Usage

```
assign_color(
  cluster_distance_matrix = data.frame(),
 colorblind_type = c("none", "protanopia", "deuteranopia", "tritanopia", "general"),
  palette = NULL,
  image_palette = NULL
)
```

Arguments

`cluster_distance_matrix`	A matrix representing the distances between clusters.A DataFrame with unique cluster names as `index` and `columns`, which contains a distance adjacent matrix for clusters, representing the dissimilarity between clusters.
`colorblind_type`	A character vector specifying the type of colorblindness to accommodate. Options include "none", "protanopia", "deuteranopia", "tritanopia", and "general". Default is "none".
`palette`	An optional vector of color values (in hex format). If provided, this palette will be used and `image_palette` will be ignored.Defaults to None.
`image_palette`	An optional image (in a format compatible with R) used to extract a color palette. Ignored if `palette` is provided.Defaults to None.
`mapping_args`	A list of additional arguments to pass to the `map_graph` function.
`embed_args`	A list of additional arguments to pass to the `embed_graph` function.

Value

A named vector where names are cluster identifiers and values are the assigned hex color codes.

colorize — *Colorize cell clusters based on spatial distribution*

Description

Colorize cell clusters based on spatial distribution, so that spatially interlaced and spatially neighboring clusters are assigned with more perceptually different colors. SpacoR provides 3 basic color mapping mode:

1. Optimize the mapping of a pre-defined color palette.
2. Extract colors from image.
3. Automatically generate colors within colorspace.

Usage

```
colorize(
  cell_coordinates,
  cell_labels,
 colorblind_type = c("none", "protanopia", "deuteranopia", "tritanopia", "general"),
  palette = NULL,
  image_palette = NULL,
  manual_mapping = NULL,
  neighbor_weight = 0.5,
  radius = 90,
  n_neighbors = 16
)
```

Arguments

`cell_coordinates`	a list like object containing spatial coordinates for each cell.
`cell_labels`	a list like object containing cluster labels for each cell.
`colorblind_type`	Optional parameter.
`palette`	a list of colors (in hex). If given, `image_palette`will be ignored. See `Mode 1` above. Defaults to None.
`image_palette`	an image in numpy array format. Should be a typical RGB image of shape (x, y, 3). Ignored if `palette` is given. See `Mode 2`above. Defaults to None.
`manual_mapping`	a data structure for manual color mapping including cluster names and manually assigned colors (in hex).
`neighbor_weight`	Weight for calculating cell neighborhood.Defaults to 0.5.
`radius`	radius used to calculate cell neighborhood.Defaults to 90.
`n_neighbors`	k for KNN neighbor detection.Defaults to 16.
`neighbor_args`	arguments passed to `spatial_distance` function.
`mapping_args`	arguments passed to `map_graph` function.
`embed_args`	arguments passed to `embed_graph` function.

Value

Optimized color mapping for clusters, including cluster names and corresponding hex

---

map_graph *map the vertices between two graph*

Description

Function to embed the cluster distance graph into chosen colorspace, while keeping distance relationship. Currently only supports CIE Lab space. Proper colors are selected within whole colorspace based on the embedding of each cluster

Function to map clusters between different clustering results based on cluster overlap

Usage

```
map_graph(
  cluster_distance,
  color_distance,
  random_seed = 123,
  distance_metric = "mul_1",
  random_max_iter = 5000,
  verbose = FALSE
)

embed_graph(
  cluster_distance,
  transformation = "umap",
  l_range = c(30, 80),
  log_colors = FALSE,
  trim_fraction = 0.0125
)

cluster_mapping_iou(cluster_label_mapping, cluster_label_reference)
```

Arguments

`cluster_distance`	a data.frame representing the dissimilarity between clusters
`color_distance`	a data.frame representing the perceptual difference between colors
`random_seed`	Integer for random seed in heuristic solver
`distance_metric`	Metric used for matrix mapping, default is manhattan
`random_max_iter`	optional parameter
`verbose`	Boolean flag for outputting info, default is FALSE
`cluster_label_mapping`	List of cluster results for cells to be mapped
`cluster_label_reference`	List of cluster results for cells to be mapped to

Value

optimized color mapping for clusters including cluster names and hex colors

A list where keys are cluster names and values are hex colors, representing the optimized color mapping

A list representing the mapping result of cluster_label_mapping

---

`matrix_distance` *Calculate the distance between two matrices*

Description

Calculate the distance between two matrices

Convert hex string to RGB value

Convert RGB value (0~255) to hex string

Convert CIE Lab color value to hex string

Convert RGB image matrix (0~255) to lms image matrix

Convert lms image matrix to RGB image matrix

Calculate the perceptual difference between colors

Revert bin number in `extract_palette` function to Lab values

Calculate the minimal distance within a Lab palette

Score color replacement

Extract palette from image

Usage

```
matrix_distance(matrix_x, matrix_y, metric = "manhattan")

hex_to_rgb(hex_code)

rgb_to_hex(rgb_code)

lab_to_hex(lab_code)

rgb_to_lms_img(img)

lms_to_rgb_img(img)

color_difference_rgb(color_x, color_y)

get_bin_color(bin_number)

palette_min_distance(palette)

color_score(lab_color, color_count, palette, wn)

simulate_cvd(palette_hex, colorblind_type)
```

```
extract_palette(
  reference_image,
  n_colors,
  colorblind_type,
  l_range = c(20, 85),
  trim_percentile = 0.03,
  max_iteration = 20,
  verbose = FALSE
)
```

Arguments

`matrix_x`	matrix x
`matrix_y`	matrix y
`metric`	metric used to calculate distance. Defaults to manhattan
`rgb_code`	RGB channel value
`lab_code`	CIE Lab color value
`img`	lms image
`color_x`	color_x
`color_y`	color_x
`bin_number`	numbered bin color
`palette`	palette
`lab_color`	lab_color
`color_count`	color_count
`wn`	float
`palette_hex`	palette_hex
`colorblind_type`	colorblind type
`reference_image`	reference_image
`n_colors`	colors
`l_range`	intergrate
`trim_percentile`	trim_percentile
`max_iteration`	max_iteration
`verbose`	verbose
`hex_codehex`	string representing a RGB color

Value

the distance between two matrices

integer values for RGB channels

color hex string

color hex string

lms image matrix

RGB image matrix

the perceptual difference between two colors

Lab values for the centroid color of this bin

the min distance

color_score

rgb_to_hex

lab_to_hex

`spatial_distance` *calculate spatial interlacement distance graph for cell clusters*

Description

Function to calculate spatial interlacement distance graph for cell clusters, where we define the interlacement distance as the number of neighboring cells between two cluster

See 'color_difference_rgb' for details

Usage

```
spatial_distance(
  cell_coordinates,
  cell_labels,
  neighbor_weight = 0.5,
  radius = 90,
  n_neighbors = 16,
  n_cells = 3
)

perceptual_distance(
  colors,
 colorblind_type = c("none", "protanopia", "deuteranopia", "tritanopia", "general")
)
```

Arguments

`cell_coordinates`	a list like object containing spatial coordinates for each cell
`cell_labels`	a list like object containing cluster labels for each cell
`neighbor_weight`	cell weight to calculate cell neighborhood. Defaults to 0.5
`radius`	radius used to calculate cell neighborhood. Defaults to 90
`n_neighbors`	k for KNN neighbor detection. Defaults to 16
`n_cells`	nly calculate neighborhood with more than `n_cells`. Defaults to 3
`colors`	a list of colors (in hex)
`colorblind_type`	optional parameter

Value

a Data.Frame with unique cluster names as `index` and `columns`, which contains interlacement distance between clusters

a data.frame with unique colors (in hex) as index and columns,which contains perceptual distance between colors

Index

assign_color, 2

cluster_mapping_iou *(map_graph)*, 4
color_difference_rgb *(matrix_distance)*, 5
color_score *(matrix_distance)*, 5
colorize, 3

embed_graph *(map_graph)*, 4
extract_palette *(matrix_distance)*, 5

get_bin_color *(matrix_distance)*, 5

hex_to_rgb *(matrix_distance)*, 5

lab_to_hex *(matrix_distance)*, 5
lms_to_rgb_img *(matrix_distance)*, 5

map_graph, 4
matrix_distance, 5

palette_min_distance *(matrix_distance)*, 5
perceptual_distance *(spatial_distance)*, 7

rgb_to_hex *(matrix_distance)*, 5
rgb_to_lms_img *(matrix_distance)*, 5

simulate_cvd *(matrix_distance)*, 5
spatial_distance, 7