Package ‘BioCartaImage’

April 3, 2024

Type Package
Title BioCarta Pathway Images
Version 1.0.0
Date 2023-10-15
Depends R (>= 4.3.0)
Imports magick, grid, stats, grDevices, utils
Suggests testthat, knitr, BiocStyle, ragg
biocViews Software, Pathways, BioCarta, Visualization
Description The core functionality of the package is to provide coordinates of genes on the BioCarta pathway images and to provide methods to add self-defined graphics to the genes of interest.
VignetteBuilder knitr
URL https://github.com/jokergoo/BioCartaImage
BugReports https://github.com/jokergoo/BioCartaImage/issues
License MIT + file LICENSE
NeedsCompilation no
RoxygenNote 7.2.3
Encoding UTF-8
Roxygen list(markdown = TRUE)
git_url https://git.bioconductor.org/packages/BioCartaImage
git_branch RELEASE_3_18
git_last_commit 1de1112
git_last_commit_date 2023-10-24
Repository Bioconductor 3.18
Date/Publication 2024-04-03
Author Zuguang Gu [aut, cre] (<https://orcid.org/0000-0002-7395-8709>)
Maintainer Zuguang Gu <z.gu@dkfz.de>
**BioCartaImage-package**

**R topics documented:**

BioCartaImage-package ..................................................  2
all_pathways .............................................................  3
BIOCARTA_PATHWAYS .....................................................  4
genes_in_pathway ........................................................  5
get_pathway ...............................................................  5
get_pathway_image .......................................................  6
grid.biocarta .............................................................  7
makeContext.biocarta_pathway_grob ..................................  9
mark_gene .................................................................  9
pos_by_polygon ........................................................... 11
print.biocarta_pathway .................................................. 12

Index 13

BioCartaImage-package  The BioCartaImage package

**Description**

BioCarta is a valuable source of biological pathways which not only provides well manually curated pathways, but also remarkable and intuitive pathway images. One useful feature of pathway analysis which is to highlight genes of interest on the pathway images is lost. Since the original source of BioCarta (biocarte.com) is lost from the internet, we dug out the data from the internet archive and formatted it into a package.

**Details**

The core functionality of this package is to highlight certain genes on the pathway image. The BioCartaImage package wraps the pathway image as well as gene locations into a graphic object.

A simple use is as follows:

```r
library(BioCartaImage)
library(grid)
grid.newpage()
grid.biocarta("h_RELAPathway", color = c("1387" = "yellow"))
```

where "h_RELAPathway" is a BioCarta pathway ID, "1387" (in the EntreZ ID type) is the gene to be highlighted. `grid.biocarta()` is a low-level grid graphical function which adds the pathway graphic to a certain position in the plot.

More advanced use is first to create a graphic object (a grob), later to add more complex graphics to it:

```r
grid.newpage()
grob = biocartaGrob("h_RELAPathway")
grob2 = mark_gene(grob, "1387", function(x, y) {
```
Here `biocartaGrob()` creates a grob for the pathway image and `mark_gene()` adds more graphics which are defined by the self-defined function.

For more details, please go to the vignette of this package.

---

### all_pathways

**All BioCarta pathways**

#### Description

All BioCarta pathways

#### Usage

```r
all_pathways()
```

#### Details


#### Value

A vector of pathway IDs (the primary pathway IDs on BioCarta).

#### Examples

```r
all_pathways()
```
Pre-computed data objects

Description

Pre-computed data objects

Usage

BIOCARTA_PATHWAYS
PATHWAY2BC
PATHWAY2ENTREZ
PATHWAY2MSIGDB
BC2ENTREZ

Format

An object of class list of length 314.
An object of class data.frame with 4428 rows and 2 columns.
An object of class data.frame with 5196 rows and 2 columns.
An object of class data.frame with 292 rows and 2 columns.
An object of class data.frame with 1739 rows and 2 columns.

Details

BIOCARTA_PATHWAYS, PATHWAY2BC, PATHWAY2ENTREZ and BC2ENTREZ are collected from web.archive.org (https://web.archive.org/web/20170122225118/https://cgap.nci.nih.gov/Pathways/BioCarta_Pathways). PATHWAY2MSIGDB is collected from MSigDB database (https://www.gsea-msigdb.org/gsea/msigdb/human/genesets.jsp?collection=CP:BIOCARTA). The script for generating these datasets can be found at:

system.file("script", "process.R", package = "BioCartaImage")

Value

- BIOCARTA_PATHWAYS: A list of pathway objects. The pathway object is explained in get_pathway().
- PATHWAY2BC: A two-column data frame of pathway IDs and BC IDs.
- PATHWAY2ENTREZ: A two-column data frame of pathway IDs and gene Entrez IDs.
- PATHWAY2MSIGDB: A two-column data frame of pathway IDs and MSigDB IDs.
genes_in_pathway

- BC2ENTREZ: A two-column data frame of BC IDs and gene Entrez IDs.

The nodes in the original BioCarta pathways are proteins and some of them do not have one-to-one mapping to genes, such as protein families or complex. Here BC_ID is the primary ID of proteins/single nodes in BioCarta Pathways and this package provides mapping to gene Entrez IDs.

<table>
<thead>
<tr>
<th>genes_in_pathway</th>
<th>Genes in a pathway</th>
</tr>
</thead>
</table>

**Description**

Genes in a pathway

**Usage**

`genes_in_pathway(pathway)`

**Arguments**

- `pathway`  A BioCarta pathway ID, a MSigDB ID or a biocarta_pathway object. All valid BioCarta pathway IDs are in `all_pathways()`. The list of MSigDB IDs for BioCarta pathways can be found at [https://www.gsea-msigdb.org/gsea/msigdb/human/genesets.jsp?collection=CP:BIOCARTA](https://www.gsea-msigdb.org/gsea/msigdb/human/genesets.jsp?collection=CP:BIOCARTA).

**Value**

A character vector of Entrez IDs.

**Examples**

`genes_in_pathway("h_RELAPathway")`

<table>
<thead>
<tr>
<th>get_pathway</th>
<th>Get a single pathway</th>
</tr>
</thead>
</table>

**Description**

Get a single pathway

**Usage**

`get_pathway(pathway_id)`

**Arguments**

- `pathway_id`  A BioCarta pathway ID. All valid BioCarta pathway IDs are in `all_pathways()`. To make it more convenient to use, the value can also be a MSigDB pathway ID in the BioCarta catalogue. The format should look like: "BIOCARTA_RELA_PATHWAY".
get_pathway_image

Value

A biocarta_pathway object. The object is a simple list and contains the following elements:

- id: The pathway ID.
- name: The pathway name.
- bc: The nodes in the original BioCarta pathways are proteins and some of them do not have one-to-one mapping to genes, such as protein families or complex. Here bc contains the primary IDs of proteins/single nodes in the pathway. The mapping to genes can be obtained by genes_in_pathway().
- shape: The shape of the corresponding protein/node in the pathway image.
- coords: It is a list of integer vectors, which contains coordinates of the corresponding shapes, in the unit of pixels. This information is retrieved from the HTML source code (in the <area> tag), so the the coordinates start from the top left of the image. The format of the coordinate vectors is c(x1, y1, x2, y2, ...).
- image_file: The file name of the pathway image.

The bc, shape and coords elements have the same length and in the same order.

See Also

The BioCarta pathways on MSigDB: https://www.gsea-msigdb.org/gsea/msigdb/human/genesets.jsp?collection=CP:BIOCARTA.

Examples

get_pathway("h_RELAPathway")
get_pathway("BIOCARTA_RELA_PATHWAY")

download the pathway image

Description

Download the pathway image

Usage

get_pathway_image(pathway)

dimension(pathway)

Arguments

pathway A BioCarta pathway ID, a MSigDB ID or a biocarta_pathway object. All valid BioCarta pathway IDs are in all_pathways(). The list of MSigDB IDs for BioCarta pathways can be found at https://www.gsea-msigdb.org/gsea/msigdb/human/genesets.jsp?collection=CP:BIOCARTA.
grid.biocarta

Details

The images are downloaded from https://data.broadinstitute.org/gsea-msigdb/msigdb/biocarta/human/.

Value

get_pathway_image() returns a raster object. image_dimension() returns an integer vector of the height and width of the image.

Examples

```r
ing = get_pathway_image("h_RELAPathway")
class(img)
# you can directly plot the raster object
plot(img)

image_dimension("h_RELAPathway")
```

grid.biocarta Draw a BioCarta pathway

Description

Draw a BioCarta pathway

Usage

```r
gird.biocarta(
    pathway,
    color = NULL,
    x = unit(0.5, "npc"),
    y = unit(0.5, "npc"),
    width = NULL,
    height = NULL,
    just = "centre",
    default.units = "npc",
    name = NULL
)
```

biocartaGrob(
    pathway,
    color = NULL,
    x = unit(0.5, "npc"),
    y = unit(0.5, "npc"),
    width = NULL,
    height = NULL,
    just = "centre",
```
grid.biocarta

```
  default.units = "npc",
  name = NULL
)
```

Arguments

- **pathway**: A BioCarta pathway ID, a MSigDB ID or a biocarta_pathway object. All valid BioCarta pathway IDs are in `all_pathways()`. The list of MSigDB IDs for BioCarta pathways can be found at https://www.gsea-msigdb.org/gsea/msigdb/human/genesets.jsp?collection=CP:BIOCARTA.

- **color**: A named vector where names should correspond to Entrez IDs.

- **x**: A numeric vector or unit object specifying x-location.

- **y**: A numeric vector or unit object specifying y-location.

- **width**: A numeric vector or unit object specifying width.

- **height**: A numeric vector or unit object specifying width.

- **just**: The same as in `grid::viewport()`.

- **default.units**: The same as in `grid::viewport()`.

- **name**: The same as in `grid::viewport()`.

Details

The graphics object contains a pathway image and genes highlighted on the image.

The aspect ratio of the image is kept. If one of width and height is set, the other dimension is calculated by the aspect ratio. If both of width and height is set or inherit from parent viewport, the width and height are automatically adjust to let one dimension completely fill the viewport.

Value

`biocartaGrob()` returns a `gTree` object.

Examples

```
library(grid)
grid.newpage()
grid.biocarta("h_RELAPathway")

grob = biocartaGrob("h_RELAPathway")
```
Description
Internal functions for drawing the pathway grob

Usage

```r
## S3 method for class 'biocarta_pathway_grob'
makeContext(x)

## S3 method for class 'biocarta_pathway_grob'
grobWidth(x)

## S3 method for class 'biocarta_pathway_grob'
grobHeight(x)
```

Arguments

- `x`: A grob returned by `biocartaGrob()`.

Value

- `makeContext()` returns a grob object.
- `grobWidth()` returns a unit object.
- `grobHeight()` returns a unit object.

mark_gene

Mark a gene on the pathway image

Description
Mark a gene on the pathway image

Usage

```r
mark_gene(grob, entrez_id, fun, min_area = 0, capture = FALSE)
```
Arguments

- **grob**: A grob returned by `biocartaGrob()`.
- **entrez_id**: A single Entrez ID.
- **fun**: A self-defined function to add graphics to the selected gene.
- **min_area**: Multiple polygons may be used for one single gene in the image. It can be used to select the largest polygon. The unit for calculating the area is the pixel in the image (or more properly, square pixels).
- **capture**: It is suggested to let `fun()` directly return `grob`/`gTree` objects. But you can also directly use functions such as `grid.points()` or `grid.lines()` in `fun()`. In this case, `capture` must be set to `TRUE` to capture these graphics.

Details

`fun()` should be applied to each gene. It is possible an Entrez gene is mapped to multiple nodes in the image, so more precisely, `fun()` is applied to every node that contains the input gene.

`fun()` only accepts two arguments, `x` and `y` which are two vectors of xy-coordinates that define the polygon. The helper function `pos_by_polygon()` can be used to get positions around the polygon.

There are two ways to use `fun()`. First, `fun()` directly returns a grob. It can be a simple grob, such as by `grid::pointsGrob()` or complex grob by `grid::gTree()` and `grid::gList()`. Second, `fun()` can directly include plotting functions such as `grid::grid.points()`, in this case, `capture` argument must be set to `TRUE` to capture these graphics.

Value

If `capture = FALSE`, it must return a grob where new graphics are already added.

Examples

```r
library(grid)
grid.newpage()
grob = biocartaGrob("h.RELAPathway")
# gene 1387 is a gene in the pathway
grob2 = mark_gene(grob, "1387", function(x, y) {
  pos = pos_by_polygon(x, y)
  pointsGrob(pos[1], pos[2], default.units = "native", pch = 16,
             gp = gpar(col = "yellow"))
})
grid.draw(grob2)

grid.newpage()
grob3 = mark_gene(grob, "1387", function(x, y) {
  pos = pos_by_polygon(x, y)
  grid.points(pos[1], pos[2], default.units = "native", pch = 16,
              gp = gpar(col = "yellow")), capture = TRUE)
}]
grid.draw(grob3)
grid.newpage()
```
pos_by_polygon

pos = pos_by_polygon(x, y)
pushViewport(viewport(x = pos[1] - 10, y = pos[2],
width = unit(4, "cm"), height = unit(4, "cm"),
default.units = "native", just = "right"))
grid.rect(gp = gpar(fill = "red"))
popViewport()
}
grid.draw(grob4)

---

pos_by_polygon

Position around a polygon

Description

Position around a polygon

Usage

pos_by_polygon(
  x,
  y,
  where = c("left", "right", "top", "bottom", "topleft", "topright", "bottomleft",
    "bottomright")
)

Arguments

x x-coordinate of a polygon.
y y-coordinate of a polygon.
where Which side of the polygon? It should take value in c("left", "right", "top", "bottom", "topleft", "topright", "bottomleft", "bottomright").

Value

A numeric scalar of length two, which is the xy-coordinate of the point.

Examples

y = c(418, 409, 402, 397, 394, 395, 396, 404, 411, 416, 417, 416, 415, 422, 429, 434, 437, 436, 432, 426, 418)
pos_by_polygon(x, y, "left")
pos_by_polygon(x, y, "bottomleft")
print.biocarta_pathway

Print the biocarta_pathway object

Description

Print the biocarta_pathway object

Usage

## S3 method for class 'biocarta_pathway'
print(x, ...)

Arguments

x        A biocarta_pathway object.
...

Other arguments.

Details

It prints two numbers:

• The number of nodes without removing duplicated ones.
• The number of unique genes that are mapped to the pathway.

Value

Nothing.

Examples

p = get_pathway("h_RELAPathway")
p
Index

* datasets
  - BIOCARTA_PATHWAYS, 4
    - all_pathways, 3
    - all_pathways(), 5, 6, 8
  - BC2ENTREZ (BIOCARTA_PATHWAYS), 4
  - BIOCARTA_PATHWAYS, 4
  - biocartaGrob (grid.biocarta), 7
  - biocartaGrob(), 3, 9, 10
  - BioCartaImage (BioCartaImage-package), 2
  - BioCartaImage-package, 2
  - genes_in_pathway, 5
  - genes_in_pathway(), 6
  - get_pathway, 5
  - get_pathway(), 4
  - get_pathway_image, 6
  - grid.biocarta, 7
  - grid.biocarta(), 2
  - grid::gList(), 10
  - grid::grid.points(), 10
  - grid::gTree(), 10
  - grid::pointsGrob(), 10
  - grid::viewport(), 8
    - grobHeight.biocarta_pathway_grob
      - (makeContext.biocarta_pathway_grob), 9
    - grobWidth.biocarta_pathway_grob
      - (makeContext.biocarta_pathway_grob), 9
  - image_dimension (get_pathway_image), 6
  - makeContext.biocarta_pathway_grob, 9
  - mark_gene, 9
  - mark_gene(), 3
  - PATHWAY2BC (BIOCARTA_PATHWAYS), 4
  - PATHWAY2ENTREZ (BIOCARTA_PATHWAYS), 4
  - PATHWAY2MSIGDB (BIOCARTA_PATHWAYS), 4