

Package ‘chromoMap’

April 10, 2019

Type Package

Title Interactive Visualization and Mapping of Chromosomes

Version 0.2

Maintainer Lakshay Anand <lakshayanand15@gmail.com>

Description Provides interactive, configurable and elegant graphics visualization of the chromosomes or chromosome regions of any living organism allowing users to map chromosome elements (like genes, SNPs etc.) on the chromosome plot. It introduces a special plot viz. the “chromosome heatmap” that, in addition to mapping elements, can visualize the data associated with chromosome elements (like gene expression) in the form of heat colors which can be highly advantageous in the scientific interpretations and research work. Because of the large size of the chromosomes, it is impractical to visualize each element on the same plot. However, the plot provides a magnified view for each of chromosome locus to render additional information and visualization specific for that location. You can map thousands of genes and can view all mappings easily. Users can investigate the detailed information about the mappings (like gene names or total genes mapped on a location) or can view the magnified single or double stranded view of the chromosome at a location showing each mapped element in sequential order. The package provide multiple features like visualizing multiple sets, chromosome heatmaps, group annotations, adding hyperlinks, and labelling. The plots can be saved as HTML documents that can be customized and shared easily. In addition, you can include them in R Markdown or in R 'Shiny' applications.

Depends R (>= 3.5.0)

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData false

Imports htmltools (>= 0.3.6), htmlwidgets (>= 1.0)

Suggests knitr, rmarkdown, shiny

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author Lakshay Anand [aut, cre]

Repository CRAN

Date/Publication 2019-04-10 19:27:11 UTC

R topics documented:

chromoMap	2
chromoMap-shiny	4

Index	5
--------------	----------

chromoMap	<i>render interactive chromosome plots of any living organism and annotate elements</i>
-----------	---

Description

render an interactive graphics visualization of entire chromosomes or chromosomal regions of any living organism. Chromosomal elements such as genes can be annotated easily using this tool.

required for creating widgets

Usage

```
chromoMap(ch.files, data.files, title = c(), ch_gap = 5, ploidy = 1,
  top_margin = 25, left_margin = 40, chr_width = 6, chr_length = 4,
  chr_color = c("black"), data_based_color_map = FALSE,
  v_align = FALSE, segment_annotation = FALSE, lg_x = 0, lg_y = 0,
  data_type = c("numeric", "categorical"), labels = FALSE,
  canvas_width = 500, canvas_height = 520, data_colors = list(),
  anno_col = c("yellow"), chr_text = c(TRUE), legend = c(FALSE),
  hlinks = FALSE, aggregate_func = c("avg"))
```

Arguments

ch.files	filename(s) containing co-ordinates of the chromosomes to render
data.files	filename(s) containing data to annotate on the chromosomes.
title	a character string to be used as a title in plot
ch_gap	provide spacing between chromosomes.
ploidy	specify the number of sets of chromosomes being passed.
top_margin	specify the margin from top of the plot

left_margin	specify the margin from the left of the plot
chr_width	specify the width of each chromosome
chr_length	specify the length of each chromosome.
chr_color	a vector specifying the color of each chromosome in a set. A color can be assigned to each set by passing a different color values as vector
data_based_color_map	a boolean to tell the plot to use the data provided in file for visualizing annotation
v_align	a boolean for vertical alignment of plot
segment_annotation	a boolean to use segment-annotation algorithm
lg_x	specify the x or horizontal distance of the legend from origin(bottom right corner)
lg_y	specify the y or vertical distance of the legend from the origin
data_type	specifying the data type of the data used. takes value either 'categorical' or 'numeric'
labels	a boolean to include labels in plot
canvas_width	width of the plot
canvas_height	height of the plot
data_colors	specify annotation colors for the data
anno_col	a vector to specify annotation color for each set.
chr_text	a boolean vector to enable or disable chromosome texts for each ploidy.set
legend	a boolean vector to enable or disable legend for each set/ploidy
hlinks	a boolean to use hyperlinks supplied in data
aggregate_func	takes either 'sum' or 'avg' to specify aggregate function for each loci

Examples

```
## Not run:

library(chromoMap)

#simple annotations
chromoMap("chromosome_file.txt", "annotation_file.txt")

#polyploidy example
chromoMap(c("chromosome_set1.txt", "chromosome_set2.txt"),
          c("annotation_set1.txt", "annotation_set2.txt"), ploidy=2)

#plotting group annotation
chromoMap("chromosome_file.txt", "annotation_file.txt",
          data_base_color_map=T, data_type="categorical")

#plotting chromosome heatmaps
chromoMap("chromosome_file.txt", "annotation_file.txt",
          data_based_color_map=T, data_type="numeric")
```

```
#enabling hyperlinks
chromoMap("chromosome_file.txt", "annotation_file.txt", hlinks=T)

#enabling labels
chromoMap("chromosome_file.txt", "annotation_file.txt", labels=T)

#change chromosome color
chromoMap("chromosome_file.txt", "annotation_file.txt", chr_color="red")

## End(Not run)
```

chromoMap-shiny

Shiny bindings for chromoMap

Description

Output and render functions for using chromoMap within Shiny applications and interactive Rmd documents.

Usage

```
chromoMapOutput(outputId, width = "100%", height = "400px")
```

```
renderChromoMap(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a chromoMap
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

Index

`chromoMap`, [2](#)
`chromoMap-shiny`, [4](#)
`chromoMapOutput` (`chromoMap-shiny`), [4](#)
`renderChromoMap` (`chromoMap-shiny`), [4](#)