Package ‘manhattanly’

April 26, 2021

Type Package

Title Interactive Q-Q and Manhattan Plots Using 'plotly.js'

Version 0.3.0

Date 2021-04-06

LazyData true

Description Create interactive manhattan, Q-Q and volcano plots that are usable from the R console, in 'Dash' apps, in the 'RStudio' viewer pane, in 'R Markdown' documents, and in 'Shiny' apps. Hover the mouse pointer over a point to show details or drag a rectangle to zoom. A manhattan plot is a popular graphical method for visualizing results from high-dimensional data analysis such as a (epi)genome wide association study (GWAS or EWAS), in which p-values, Z-scores, test statistics are plotted on a scatter plot against their genomic position. Manhattan plots are used for visualizing potential regions of interest in the genome that are associated with a phenotype. Interactive manhattan plots allow the inspection of specific value (e.g. rs number or gene name) by hovering the mouse over a cell, as well as zooming into a region of the genome (e.g. a chromosome) by dragging a rectangle around the relevant area. This work is based on the 'qqman' package and the 'plotly.js' engine. It produces similar manhattan and Q-Q plots as the 'manhattan' and 'qq' functions in the 'qqman' package, with the advantage of including extra annotation information and interactive web-based visualizations directly from R. Once uploaded to a 'plotly' account, 'plotly' graphs (and the data behind them) can be viewed and modified in a web browser.

Depends R (>= 3.5.0)

Imports stats, magrittr, plotly, ggplot2

Suggests testthat (>= 3.0.0), knitr, rmarkdown, covr

VignetteBuilder knitr

Encoding UTF-8

License MIT + file LICENSE

URL https://github.com/sahirbhatnagar/manhattanly/, https://sahirbhatnagar.com/manhattanly/

BugReports https://github.com/sahirbhatnagar/manhattanly/issues

RoxygenNote 7.1.1

Config/testthat/edition 3

NeedsCompilation no

Author Sahir Bhatnagar [aut, cre] (http://sahirbhatnagar.com/)

Maintainer Sahir Bhatnagar <sahir.bhatnagar@gmail.com>

Repository CRAN

Date/Publication 2021-04-26 06:40:03 UTC

R topics documented:

HapMap 2
manhattanly 3
manhattanr 5
qqly 7
qqr 9
significantSNP 10
volcanoly 11
volcanor 13

Index **15**

HapMap *Subset of HapMap data with simulated GWAS results*

Description

A dataset containing a subset of the draft release 2 for genome-wide SNP genotyping in DNA samples from 11 human populations (sometimes referred to as the "HapMap 3" samples). Only the PLINK .map file was used. Approximately 2.5% of the SNPs in each chromosome were retained. The p-values, zscores, and effectsizes were simulated using random distributions in R. Annotation information (nearest gene and distance to nearest gene) was obtained from the UCSC genome annotation database for the Mar. 2006 GenBank freeze assembled by NCBI (hg18, Build 36.1)

Usage

```
HapMap
```

Format

A data frame with 14412 rows and 8 variables:

CHR chromosome number. Autosomes coded 1 through 22, and 23 is the X chromosome (`integer`)

BP genomic base-pair position (`integer`)

P p-value (`numeric`)

SNP rs# or snp identifier (`character`)

ZSCORE z-score (`numeric`)

EFFECTSIZE effect size (`numeric`)

GENE nearest gene to the SNP (`character`)

DISTANCE distance between the SNP and GENE. if `DISTANCE=0` then the SNP is located in the GENE (`integer`)

Source

ftp://ftp.ncbi.nlm.nih.gov/hapmap/genotypes/2009-01_phaseIII/plink_format/

http://hgdownload.cse.ucsc.edu/goldenPath/hg18/database/

`manhattanly` *Creates a plotly manhattan plot*

Description

Creates an interactive manhattan plot with multiple annotation options

Usage

```
manhattanly(
  x,
  ...,
  col = c("#969696", "#252525"),
  point_size = 5,
  labelChr = NULL,
  suggestiveline = -log10(1e-05),
  suggestiveline_color = "blue",
  suggestiveline_width = 1,
  genomewideline = -log10(5e-08),
  genomewideline_color = "red",
  genomewideline_width = 1,
  highlight = NULL,
  highlight_color = "#00FF00",
  showlegend = FALSE,
  showgrid = FALSE,
  xlab = NULL,
  ylab = "-log10(p)",
  title = "Manhattan Plot"
)
```

Arguments

`x` Can be an object of class `manhattanr` produced by the `manhattanr` function or a `data.frame` which must contain at least the following three columns:

- the chromosome number

- genomic base-pair position
- a numeric quantity to plot such as a p-value or zscore

`...` other parameters passed to `manhattanr`

`col` A character vector indicating the colors of each chromosome. If the number of colors specified is less than the number of unique chromosomes, then the elements will be recycled. Can be Hex Codes as well.

`point_size` A `numeric` indicating the size of the points on the plot. Default is 5

`labelChr` A character vector equal to the number of chromosomes specifying the chromosome labels (e.g., `c(1:22,"X","Y","MT")`). Default is `NULL`, meaning that the actual chromosome numbers will be used.

`suggestiveline` Where to draw a "suggestive" line. Default is `-log10(1e-5)`. Set to `FALSE` to disable.

`suggestiveline_color` color of "suggestive" line. Only used if `suggestiveline` is not set to `FALSE`. Default is `"blue"`.

`suggestiveline_width` Width of `suggestiveline`. Default is 1.

`genomewideline` Where to draw a "genome-wide sigificant" line. Default `-log10(5e-8)`. Set to `FALSE` to disable.

`genomewideline_color` color of "genome-wide sigificant" line. Only used if `genomewideline` is not set to `FALSE`. Default is `"red"`.

`genomewideline_width` Width of `genomewideline`. Default is 1.

`highlight` A character vector of SNPs in your dataset to highlight. These SNPs should all be in your dataset. Default is `NULL` which means that nothing is highlighted.

`highlight_color` Color used to highlight points. Only used if `highlight` argument has been specified

`showlegend` Should a legend be shown. Default is `FALSE`.

`showgrid` Should gridlines be shown. Default is `FALSE`.

`xlab` X-axis label. Default is `NULL` which means that the label is automatically determined by the `manhattanr` function. Specify here to overwrite the default.

`ylab` Y-axis label. Default is `"-log10(p)"`.

`title` Title of the plot. Default is `"Manhattan Plot"`

Value

An interactive manhattan plot.

Note

This package is inspired by the `qqman` package. This package provides additional annotation options and builds on the `plotly` `d3.js` engine. These plots can be included in Dash apps, Shiny apps, Rmarkdown documents or embedded in websites using simple HTML code.

See Also

manhattanr, HapMap, significantSNP

Examples

```
## Not run:
library(manhattanly)
manhattanly(HapMap)

# highlight SNPs of interest
# 'signigicantSNP' is a character vector of SNPs included in this package
manhattanly(HapMap, snp = "SNP", highlight = significantSNP)

## End(Not run)
```

---

manhattanr *Creates a manhattanr object*

Description

An object of class manhattanr includes all the needed information for producing a manhattan plot. The goal is to seperate the pre-processing of the manhattan plot elements from the graphical rendaring of the object, which could be done using any graphical device including plot_ly and plot in base R.

Usage

```
manhattanr(
  x,
  chr = "CHR",
  bp = "BP",
  p = "P",
  snp,
  gene,
  annotation1,
  annotation2,
  logp = TRUE
)
```

Arguments

x — A data.frame which must contain at least the following three columns:

- the chromosome number
- genomic base-pair position
- a numeric quantity to plot such as a p-value or zscore

`chr` A string denoting the column name for the chromosome. Default is `chr = "CHR"`. This column must be `numeric` or `integer`. Minimum number of chromosomes required is 1. If you have X, Y, or MT chromosomes, be sure to renumber these 23, 24, 25, etc.

`bp` A string denoting the column name for the chromosomal position. Default is `bp = "BP"`. This column must be `numeric` or `integer`.

`p` A string denoting the column name for the numeric quantity to be plotted on the y-axis. Default is `p = "P"`. This column must be `numeric` or `integer`. This does not have to be a p-value. It can be any numeric quantity such as peak heights, bayes factors, test statistics. If it is not a p-value, make sure to set `logp = FALSE`.

`snp` A string denoting the column name for the SNP names (e.g. rs number). More generally, this column could be anything that identifies each point being plotted. For example, in an Epigenomewide association study (EWAS) this could be the probe name or cg number. This column should be a `character`. This argument is optional, however it is necessary to specify if you want to highlight points on the plot using the `highlight` argument in the `manhattanly` function

`gene` A string denoting the column name for the GENE names. This column could be a `character` or `numeric`. More generally this could be any annotation information that you want to include in the plot. This argument is optional.

`annotation1` A string denoting the column name for an annotation. This column could be a `character` or `numeric`. This could be any annotation information that you want to include in the plot (e.g. zscore, effect size, minor allele frequency). This argument is optional.

`annotation2` A string denoting the column name for an annotation. This column could be a `character` or `numeric`. This could be any annotation information that you want to include in the plot (e.g. zscore, effect size, minor allele frequency). This argument is optional.

`logp` If TRUE, the -log10 of the p-value is plotted. It isn't very useful to plot raw p-values, but plotting the raw value could be useful for other genome-wide plots, for example, peak heights, bayes factors, test statistics, other "scores" etc.

Value

A `list` object of class `manhattanr` with the following elements

data processed data to be used for plotting

xlabel The label of the x-axis which is determined by the number of chromosomes present in the data

ticks the coordinates on the x-axis of where the tick marks should be placed

labs the labels for each tick. This defaults to the chromosome number but can be changed in the `manhattanly` function

nchr the number of unique chromosomes present in the data

pName, snpName, geneName, annotation1Name, annotation2Name The names of the columns corresponding to the data provided. This information is used for annotating the plot in the `manhattanly` function

Source

The pre-processing is mostly the same as the manhattan function from the qqman package

See Also

manhattanly

Examples

```
# HapMap dataset included in this package already has columns named P, CHR and BP
library(manhattanly)
DT <- manhattanr(HapMap)
class(DT)
head(DT[["data"]])

# include snp and gene information
DT2 <- manhattanr(HapMap, snp = "SNP", gene = "GENE")
head(DT2[["data"]])
```

---

qqly *Creates a plotly Q-Q plot*

Description

Creates an interactive Q-Q plot with multiple annotation options

Usage

```
qqly(
  x,
  ...,
  col = "#252525",
  size = 5,
  type = 20,
  abline_col = "red",
  abline_size = 1,
  abline_type = "solid",
  highlight = NULL,
  highlight_color = "#00FF00",
  xlab = "Expected -log10(p)",
  ylab = "Observed -log10(p)",
  title = "Q-Q Plot"
)
```

Arguments

x	Can be an object of class qqr produced by the qqr function or a data.frame which must contain at least the following column: • a p-value, must be numeric
...	other parameters passed to qqr
col	A character indicating the color of the points. Can be Hex Codes as well.
size	A numeric specifying the size of the points. Default is 1
type	An integer between 0 and 25 specifying the point shape. Default is 20 (filled circle). Deprecated.
abline_col	A character indicating the color of the 45 degree diagonal line. Can be Hex Codes as well. Default is "red".
abline_size	A numeric indicating the size of the 45 degree diagonal line. Default is 0.5.
abline_type	Sets the line type of the 45 degree line. Set to a dash type character among "solid", "dot", "dash", "longdash", "dashdot", or "longdashdot", or a dash length list in px (eg "5px","10px","2px"). Can also be a positive numeric value (e.g 5, 10, 2). Default is "dash". See plotly help page on layouts for complete list and more details
highlight	A character vector of SNPs in your dataset to highlight. These SNPs should all be in your dataset. Default is NULL which means that nothing is highlighted.
highlight_color	Color used to highlight points. Only used if highlight argument has been specified
xlab	X-axis label. Default is "Expected -log10(p)"
ylab	Y-axis label. Default is "Observed -log10(p)"
title	Title of the plot. Default is "Q-Q Plot"

Value

An interactive Q-Q plot.

See Also

qqr, HapMap, significantSNP

Examples

```
## Not run:
library(manhattanly)
qqly(HapMap)

# highlight SNPs of interest
# 'signigicantSNP' is a character vector of SNPs included in this package
qqly(HapMap, snp = "SNP", highlight = significantSNP)

## End(Not run)
```

qqr *Creates a qq object*

Description

An object of class qq includes all the needed information for producing a quantile-quantile plot of p-values. The goal is to seperate the pre-processing of the quantile-quantile plot elements from the graphical rendaring of the object, which could be done using any graphical device including `plot_ly` and `plot` in base `R`.

Usage

```
qqr(x, p = "P", snp, gene, annotation1, annotation2, ...)
```

Arguments

`x`	A `data.frame` which must contain at least the following column: • a p-value, must be numeric
`p`	A string denoting the column name for the p-values. Default is `p = "P"`. This column must be `numeric` or `integer`. Should not have missing, NA, NaN, or NULL values and should be between 0 and 1.
`snp`	A string denoting the column name for the SNP names (e.g. rs number). More generally, this column could be anything that identifies each point being plotted. For example, in an Epigenomewide association study (EWAS) this could be the probe name or cg number. This column should be a `character`. This argument is optional, however it is necessary to specify if you want to highlight points on the plot using the `highlight` argument in the `qqly` function
`gene`	A string denoting the column name for the GENE names. This column could be a `character` or `numeric`. More generally this could be any annotation information that you want to include in the plot. This argument is optional.
`annotation1`	A string denoting the column name for an annotation. This column could be a `character` or `numeric`. This could be any annotation information that you want to include in the plot (e.g. zscore, effect size, minor allele frequency). This argument is optional.
`annotation2`	A string denoting the column name for an annotation. This column could be a `character` or `numeric`. This could be any annotation information that you want to include in the plot (e.g. zscore, effect size, minor allele frequency). This argument is optional.
`...`	currently ignored

Value

An list object of class `qqr` with the following elements

data processed data to be used for plotting the Q-Q plot including the observed and expected p-values on the -log10 scale

pName, snpName, geneName, annotation1Name, annotation2Name The names of the columns corresponding to the data provided. This information is used for annotating the plot in the `qqly` function

Note

This function will return an error if any of the p-values are NA, less than 0 or greater than 1

Source

The calculation of the expected p-value is taken from the qq function from the `qqman` package

See Also

`qqly`

Examples

```
library(manhattanly)
qqrObj <- qqr(HapMap, snp = "SNP", highlight = significantSNP)
class(qqrObj)
head(qqrObj[["data"]])
```

`significantSNP` *Character vector of SNPs to highlight*

Description

SNP rs identifiers from `HapMap` dataset that are significant at `p-value` < `1e-6`

Usage

```
significantSNP
```

Format

A character vector with 20 elements

See Also

`HapMap`

volcanoly *Creates a plotly volcano plot*

Description

Creates an interactive volcano plot with multiple annotation options

Usage

```
volcanoly(
  x,
  ...,
  col = c("#252525"),
  point_size = 5,
  effect_size_line = c(-1, 1),
  effect_size_line_color = "grey",
  effect_size_line_width = 0.5,
  effect_size_line_type = "dash",
  genomewideline = -log10(1e-05),
  genomewideline_color = "grey",
  genomewideline_width = 0.5,
  genomewideline_type = "dash",
  highlight = NULL,
  highlight_color = "red",
  xlab = NULL,
  ylab = "-log10(p)",
  title = "Volcano Plot"
)
```

Arguments

`x` Can be an object of class `volcanor` produced by the `volcanor` function or a `data.frame` which must contain at least the following two columns:

- a p-value, must be numeric
- a measure of the strength of association, typically an odds ratio, regression coefficient or log fold change. Must be numeric

`...` other parameters passed to `volcanor`

`col` A character of length 1 indicating the color of the points. Only the first argument will be used if more than one color is supplied. Can be Hex Codes as well.

`point_size` A `numeric` indicating the size of the points on the plot. Default is 5

`effect_size_line` Where to draw a "suggestive" line on the x-axis. Default is `-1` and `+1`. Must be a vector of length 2. If a longer vector is supplied, only the first two elements will be used. First element must be smaller than second element. Set to `FALSE` to disable.

effect_size_line_color
: color of "suggestive" line. Only used if effect_size_line is not set to FALSE. Default is "blue".

effect_size_line_width
: Width of effect_size_line. Default is 1.

effect_size_line_type
: Sets the line type of the effect_size_line. Set to a dash type character among "solid", "dot", "dash", "longdash", "dashdot", or "longdashdot", or a dash length list in px (eg "5px","10px","2px"). Can also be a positive numeric value (e.g 5, 10, 2). Default is "dash". See plotly help page on layouts for complete list and more details

genomewideline
: Where to draw a "genome-wide sigificant" line. Default -log10(1e-5). Set to FALSE to disable. If more than one element is provided, only the first will be used

genomewideline_color
: color of "genome-wide sigificant" line. Only used if genomewideline is not set to FALSE. Default is "red".

genomewideline_width
: Width of genomewideline. Default is 1.

genomewideline_type
: Sets the line type of the genomewideline. Set to a dash type character among "solid", "dot", "dash", "longdash", "dashdot", or "longdashdot", or a dash length list in px (eg "5px","10px","2px"). Can also be a positive numeric value (e.g 5, 10, 2). Default is "dash". See plotly help page on layouts for complete list and more details

highlight
: A character vector of SNPs in your dataset to highlight. These SNPs should all be in your dataset. Default is NULL which means that all points that are both beyond genomewideline and effect_size_line are highlighted. Set to FALSE if you don't want any points highlighted.

highlight_color
: Color used to highlight points. Only used if highlight argument has been specified

xlab
: X-axis label. Default is NULL which means that the label is automatically determined by the volcanor function. Specify here to overwrite the default.

ylab
: Y-axis label. Default is "-log10(p)".

title
: Title of the plot. Default is "Volcano Plot"

Value

An interactive volcano plot.

Note

This package provides additional annotation options and builds on the plotly d3.js engine. These plots can be included in Shiny apps, Dash apps, Rmarkdown documents or embeded in websites using simple HTML code.

See Also

volcanor, HapMap, significantSNP

Examples

```
volcanorObj <- volcanor(HapMap,
  p = "P",
  effect_size = "EFFECTSIZE",
  snp = "SNP",
  gene = "GENE"
)
class(volcanorObj)
head(volcanorObj$data)
```

---

volcanor *Creates a volcano object*

Description

An object of class volcano includes all the needed information for producing a volcano plot of p-values against effect sizes or fold-changes. The goal is to seperate the pre-processing of the volcano plot elements from the graphical rendaring of the object, which could be done using any graphical device including plot_ly and plot in base R.

Usage

```
volcanor(
  x,
  p = "P",
  effect_size = "EFFECTSIZE",
  snp,
  gene,
  annotation1,
  annotation2,
  ...
)
```

Arguments

x — A data.frame which must contain at least the following columns:

- a p-value, must be numeric
- a measure of the strength of association, typically an odds ratio, regression coefficient or log fold change. Must be numeric

p — A chracter string denoting the column name for the p-values. Default is p = "P". This column must be numeric or integer. Should not have missing, NA, NaN, or NULL values and should be between 0 and 1.

effect_size — A string denoting the column name for the effect size. Default is `effect_size = "EFFECTSIZE"`. This column must be `numeric` or `integer`. Should not have missing, NA, NaN, or NULL values.

snp — A string denoting the column name for the SNP names (e.g. rs number). This argument is optional but required if you want to highlight any points. More generally, this column could be anything that identifies each point being plotted. For example, in an Epigenomewide association study (EWAS) this could be the probe name or cg number. This column should contain `characters`. This argument is necessary. `volcanoly` function

gene — A string denoting the column name for the GENE names. This column could be a `character` or `numeric`. More generally this could be any annotation information that you want to include in the plot. This argument is optional.

annotation1 — A string denoting the column name for an annotation. This column could be a `character` or `numeric`. This could be any annotation information that you want to include in the plot (e.g. zscore, effect size, minor allele frequency). This argument is optional.

annotation2 — A string denoting the column name for an annotation. This column could be a `character` or `numeric`. This could be any annotation information that you want to include in the plot (e.g. zscore, effect size, minor allele frequency). This argument is optional.

... — currently ignored

Value

An list object of class `volcanor` with the following elements

data processed data to be used for plotting the volcano plot including the observed and expected p-values on the -log10 scale

pName, snpName, geneName, annotation1Name, annotation2Name The names of the columns corresponding to the data provided. This information is used for annotating the plot in the `volcanoly` function

Note

This function will return an error if any of the p-values are NA, less than 0 or greater than 1

See Also

`volcanoly`

Examples

```
library(manhattanly)
volcanorObj <- volcanor(HapMap)
class(volcanorObj)
head(volcanorObj)
```

Index

∗ **datasets**
    HapMap, 2
    significantSNP, 10

HapMap, 2, *5*, *8*, *10*, *13*

manhattanly, 3, *6, 7*
manhattanr, *3–5*, 5

plot, *5*, *9*, *13*
plot_ly, *5*, *9*, *13*
plotly, *4*, *12*

qqly, 7, *9, 10*
qqr, *8*, 9

significantSNP, *5*, *8*, 10, *13*

volcanoly, 11, *14*
volcanor, *11–13*, 13