Package: gwaRs (via r-universe)

October 9, 2024

Type Package

Title Manhattan, Q-Q, and PCA Plots using 'ggplot2'

Version 0.3.0

Description Generate Manhattan, Q-Q, and PCA plots from GWAS and PCA results using 'ggplot2'.

License MIT + file LICENCE

Encoding UTF-8

LazyData true

Imports RColorBrewer (>= 1.1.2), ggplot2, ggrepel, dplyr, tidyr, data.table, scales, stats, grDevices

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Depends R (>= 2.10)

Suggests knitr, rmarkdown, markdown

VignetteBuilder knitr

URL `https://github.com/LindoNkambule/gwaRs`

BugReports `https://github.com/LindoNkambule/gwaRs/issues`

Repository https://lindonkambule.r-universe.dev

RemoteUrl https://github.com/lindonkambule/gwars

RemoteRef HEAD

RemoteSha 58f56c7115c33257ab83e4d59f7f5bf7faf3c826

Contents

gwaRs 2
gwasData 2
highlightSNPS 2
karyotype_plot 3
man_plot 4

mirrored_man_plot 5
pcaData 7
pca_plot 8
qq_plot 9

Index **11**

`gwaRs` *Create Manhattan, Q-Q, and PCA plots for GWAS data.*

Description

A package for creating Manhattan, Q-Q, and PCA plots for GWAS data.

Author(s)

Lindokuhle Nkambule <`lindonkambule116@gmail.com`>

See Also

Useful links:

- `https://github.com/LindoNkambule/gwaRs`
- Report bugs at `https://github.com/LindoNkambule/gwaRs/issues`

`gwasData` *GWAS results*

Description

Example GWAS data.

`highlightSNPS` *Highlight SNPs*

Description

A couple of SNPs to be annotated

`karyotype_plot` *Karyotype Plot*

Description

Creates a Karyotype plot

Usage

```
karyotype_plot(
  data,
  density.col = c("darkgreen", "yellow", "red"),
  window.size = 1e+06,
  title = NULL
)
```

Arguments

`data`	A data.frame with "CHR" and "BP"columns.
`density.col`	A character vector with colors to use for gradients.
`window.size`	A double precision numeric value indicating the window size.
`title`	A string denoting the title to use for the plot. Default is 'Manhattan Plot'

Details

Creates a SNP Karyotype or Density plot from an R dataframe with "CHR" and "BP" columns.

Value

A SNP Karyotype plot.

Author(s)

Lindokuhle Nkambule

Examples

```
karyotype_plot(gwasData)
```

`man_plot` *Manhattan Plot*

Description

Creates a Manhattan plot

Usage

```
man_plot(
  data,
  chromCol = c("gray44", "black"),
  genomewideline = -log10(5e-08),
  suggestiveline = -log10(1e-05),
  chromosome = "ALL",
  annotatePval = FALSE,
  annotateSNP = NULL,
  annotateCol = "red",
  highlight = NULL,
  highlightCol = "green3",
  title = NULL
)
```

Arguments

`data`	PLINK assoc output, tab-delimited, or a data.frame with "SNP", "CHR", "BP", and "P" columns.
`chromCol`	A character vector indicating which colors to alternate for the chromosomes.
`genomewideline`	Where to draw the "genome-wide significant" line. Default -log10(5e-8). Set to FALSE or F to disable
`suggestiveline`	Where to draw the "suggestive" line. Default -log10(1e-5). Set to FALSE or F to disable.
`chromosome`	An integer indicating which chromosome to plot. Default is "ALL".
`annotatePval`	If set, SNPs with p-value less than or equal to this p-value will be annotated on the plot.
`annotateSNP`	A character vector of SNPs in your dataset to annotate. If some of the SNPs are not in your dataset, gwaRs will throw a warning message.
`annotateCol`	A string denoting the color to use for the annotations.
`highlight`	A character vector of SNPs in the dataset to highlight. If some of the SNPs are not in your dataset, gwaRs will throw a warning message. Default is NULL.
`highlightCol`	A string denoting the color to use to highlight the SNPs.
`title`	A string denoting the title to use for the plot. Default is ’Manhattan Plot’

Details

Creates a Manhattan plot from PLINK assoc output (or any tab-delimited file or data frame with "SNP", "CHR", "BP", and "P" columns).

Value

A Manhattan plot.

Author(s)

Lindokuhle Nkambule

Examples

```
man_plot(gwasData)
```

`mirrored_man_plot` *Mirrored Manhattan Plot*

Description

Creates a Mirrored Manhattan Plot for two traits

Usage

```
mirrored_man_plot(
  data,
  trait1 = NULL,
  trait2 = NULL,
  trait1_chromCols = c("gray66", "grey36"),
  trait2_chromCols = c("steelblue1", "steelblue4"),
  xlab = "Genomic Position (chromosome)",
  title = "Manhattan Plot",
  annotate_trait1_pval = FALSE,
  annotate_trait1_color = "red",
  annotate_trait2_pval = FALSE,
  annotate_trait2_color = "red",
  annotateSNP = NULL,
  annotateSNPcolor = "red",
  highlight = NULL,
  highlightcolor = "green3",
  genomewideline_trait1 = NULL,
  genomewideline_trait2 = NULL,
  genomewideline_type = "dashed",
  genomewideline_color = "red",
  suggestiveline_trait1 = NULL,
```

```
  suggestiveline_trait2 = NULL,
  suggestiveline_type = "dashed",
  suggestiveline_color = "blue"
)
```

Arguments

`data`	A tab-delimited or data frame with the compulsory columns: "CHR", "SNP", "BP", "P", "Trait".
`trait1`	A character string of the trait1 as it appears in the input data.
`trait2`	A character string of the trait2 as it appears in the input data.
`trait1_chromCols`	A character vector indicating which colors to alternate for trait1 chromosomes.
`trait2_chromCols`	A character vector indicating which colors to alternate for trait2 chromosomes.
`xlab`	A character string to be used as the x-axis label.
`title`	A character string to be used as the plot title
`annotate_trait1_pval`	If set, trait1 SNPs with p-value less than or equal to this p-value will be annotated on the plot.
`annotate_trait1_color`	A character string indicating the color to be used for annotating trait1 SNPs by p-value
`annotate_trait2_pval`	If set, trait2 SNPs with p-value less than or equal to this p-value will be annotated on the plot.
`annotate_trait2_color`	A character string indicating the color to be used for annotating trait2 SNPs by p-value
`annotateSNP`	A character vector of SNPs in your dataset to annotate. If some of the SNPs are not in your dataset, gwaRs will throw a warning message.
`annotateSNPcolor`	A character string denoting the color to use for the annotations.
`highlight`	A character vector of SNPs in the dataset to highlight. If some of the SNPs are not in your dataset, gwaRs will throw a warning message. Default is NULL.
`highlightcolor`	A character string denoting the color to use to highlight the SNPs.
`genomewideline_trait1`	Where to draw the "genome-wide significant" line for trait1
`genomewideline_trait2`	Where to draw the "genome-wide significant" line for trait2
`genomewideline_type`	A character string denoting the type of line to be used for the "genome-wide significant" line. This is the same for both traits. Default is dashed.

`genomewideline_color`
: A character string denoting the color to be used for the "genome-wide significant" line. This is the same for both traits. Default is red.

`suggestiveline_trait1`
: Where to draw the "suggestive" line for trait1.

`suggestiveline_trait2`
: Where to draw the "suggestive" line for trait2.

`suggestiveline_type`
: A character string denoting the type of line to be used for the "suggestive" line. This is the same for both traits. Default is dashed

`suggestiveline_color`
: A character string denoting the color to be used for the "suggestive" line. This is the same for both traits. Default is blue.

Details

Create a Mirrored Manhattan Plot from a tab-delimited file or data frame with the compulsory columns: "CHR", "SNP", "BP", "P", "Trait" .

Value

A Mirrored Manhattan plot for two traits.

Author(s)

Lindokuhle Nkambule

Examples

```
## Not run:
mirrored_man_plot(inputData)

## End(Not run)
```

`pcaData` *PCA results*

Description

Example PCA data.

`pca_plot` *PCA Plot*

Description

Creates a PCA plot

Usage

```
pca_plot(
  data,
  xComponent = "PC1",
  yComponent = "PC2",
  legendPos = "right",
  soft = "PLINK",
  colPalette = "Accent",
  title = NULL
)
```

Arguments

`data`	PLINK pca or EIGENSTRAT smartpca output, (or any tab-delimited file or data.frame with the same format as PLINK pca or EIGENSTRAT smartpca output)
`xComponent`	A character vector indicating the principal component value to use for the x-axis. Default is "PC1"
`yComponent`	A character vector indicating the principal component value to use for the x-axis. Default is "PC2"
`legendPos`	A character vector indicating the legend position. Default is "right".
`soft`	A character vector indicating the software output format. Default is "PLINK". If you have a tab-delimited file or data.frame with the same format as EIGENSTRAT output, use "EIGENSTRAT"
`colPalette`	A character vector indicating the color palette to use. Default is "Accent".
`title`	A string denoting the title to use for the plot. Default is 'PCA Plot'

Details

Creates a PCA plot from PLINK pca output, EIGENSTRAT smartpca, (or any tab-delimited file or data.frame with the same format as PLINK pca or EIGENSTRAT smartpca output).

Value

A PCA plot.

Author(s)

Lindokuhle Nkambule

Examples

```
pca_plot(pcaData)
```

`qq_plot` *QQ Plot*

Description

Creates a Q-Q plot

Usage

```
qq_plot(
  data,
  point_col = "black",
  diag_col = "red",
  diag_line = "solid",
  title = NULL
)
```

Arguments

`data`	PLINK assoc output, tab-delimited, or a data.frame with "P" column.
`point_col`	A character vector indicating the color to use for the SNP p-values. Default is "black".
`diag_col`	A character vector indicating the color to use for the diagonal line. Default is "red".
`diag_line`	A character vector indicating the line type to use for the diagonal line. Default is "solid".
`title`	A string denoting the title to use for the plot. Default is 'Q-Q Plot'

Details

Creates a Q-Q plot from PLINK assoc output (or any tab-delimited file or data frame with "P" column).

Value

A Q-Q plot.

Author(s)

Lindokuhle Nkambule

Examples

```
qq_plot(gwasData)
```

Index

gwaRs, 2
gwaRs-package *(gwaRs)*, 2
gwasData, 2

highlightSNPS, 2

karyotype_plot, 3

man_plot, 4
mirrored_man_plot, 5

pca_plot, 8
pcaData, 7

qq_plot, 9