Package ‘topr’

January 12, 2022

Title Create Custom Plots for Viewing Genetic Association Results

Version 1.0.0

URL https://github.com/GenuityScience/topr

BugReports https://github.com/GenuityScience/topr/issues

Description A collection of functions for visualizing, exploring and annotating genetic association results. Association results from multiple traits can be viewed simultaneously along with gene annotation, over the entire genome (Manhattan plot) or in the more detailed regional view.

License LGPL (>= 3)

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.1.2

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

Config/testthat/edition 3

Imports ggplot2 (>= 3.3.2), dplyr (>= 1.0.2), stringr (>= 1.4.0),
readr (>= 1.4.0), scales (>= 1.1.1), ggrepel, egg, grid,
gridExtra, magrittr (>= 1.5), utils, toprdata

Depends R (>= 3.5.0)

NeedsCompilation no

Author Genuity Science [cph],
Thorhildur Juliusdottir [aut, cre],
Andri Stefansson [aut]

Maintainer Thorhildur Juliusdottir <thorhildur@genuitysci.com>

Repository CRAN

Date/Publication 2022-01-12 11:00:02 UTC

R topics documented:

`annotate_with_nearest_gene`

Get the nearest gene for one or more snps

Description

`annotate_with_nearest_gene()` Annotate the variant/snp with their nearest gene Required parameters is a dataframe of SNPs (with the columns CHROM and POS)

Usage

```
annotate_with_nearest_gene(variants, protein_coding_only = FALSE)
```

Arguments

`variants`	a dataframe of variant positions (CHROM and POS)
`protein_coding_only`	Logical, if set to TRUE only annotate with protein coding genes (the default value is FALSE)

Value

the input dataframe with Gene_Symbol as an additional column

Examples

```
variants <-get_best_snp_per_MB(CD_UKBB)
annotate_with_nearest_gene(variants)
```

`CD_FINNGEN` *Finngen v5 Crohn‘s disease (CHRONSMALL)*

Description

Dataset retrieved from the Finngen database (version 5) including 968 crohn´s cases and 210,100 controls. The dataset has been filtered on variants with P <1e-03.

Usage

```
CD_FINNGEN
```

Format

A data frame with 29,926rows and 9 variables:

CHROM Chromosome, written as for example chr1 or 1

POS genetic position of the variant

ID Variant identifier, e.g. rsid

P P-value from Plink run, additive model, regression model GLM_FIRTH

beta Variant effect

Source

Crohn’s small intestines (CHRONSMALL), only including variants with P<1e-03

CD_UKBB *UKBB Crohns disease (ICD 10 code K50)*

Description

Dataset retrieved from the UK biobank consisting of 2,799 crohn´s cases (K50) and 484,515 controls. The dataset has been filtered on variants with P <1e-03.

Usage

```
CD_UKBB
```

Format

A data frame with 26,824 rows and 10 variables:

CHROM Chromosome, written as for example chr1 or 1

POS genetic position of the variant

ID Variant identifier, e.g. rsid

P P-value from Plink run, additive model, regression model GLM_FIRTH

OR Odds Ratio

Source

Crohn's UKBB ICD10 code K50, only including variants with P<1e-03

create_snpset *Create a dataframe that can be used as input for making effect plots*

Description

```
create_snpset()
```

Usage

```
create_snpset(
  df1,
  df2,
  thresh = 1e-08,
  protein_coding_only = TRUE,
  region_size = 1e+06,
  verbose = FALSE
)
```

Arguments

`df1`	The dataframe to extract the top snps from (with p-value below thresh)
`df2`	The dataframe in which to search for overlapping SNPs from dataframe1
`thresh`	Numeric, the p-value threshold used for extracting the top snps from dataset 1
`protein_coding_only`	Logical, set this variable to TRUE to only use protein_coding genes for the annotation
`region_size`	Integer, the size of the interval which to extract the top snps from
`verbose`	Logical, (default: FALSE). Assign to TRUE to get information on which alleles are matched and which are not.

Value

Dataframe containing the top hit

Examples

```
CD_UKBB_index_snps <-get_best_snp_per_MB(CD_UKBB)
create_snpset(CD_UKBB_index_snps,CD_FINNGEN)
```

`create_snpset_code` *Show the code/functions used to create a snpset*

Description

`create_snpset_code()`

Usage

```
create_snpset_code()
```

Value

Dataframe containing the top hit

Examples

```
create_snpset_code()
```

`dat_column_check_and_set` *dat_column_check_and_set*

Description

This function is used to standardize the column names in the input dataframe

Usage

```
dat_column_check_and_set(dat)
```

Arguments

`dat`	A data frame or a list of data frames

`effect_plot` *Create a plot comparing effects within two datasets*

Description

`effect_plot()`

Usage

```
effect_plot(
  dat,
  pheno_x = "x_pheno",
  pheno_y = "y_pheno",
  annotate_with = "Gene_Symbol",
  thresh = 1e-08,
  ci_thresh = 1,
  gene_label_thresh = 1e-08,
  color = get_topr_colors()[1],
  scale = 1
)
```

Arguments

`dat`	The input dataframe (snpset) containing one row per variant and P values (P1 and P2) and effects (E1 and E2) from two datasets/phenotypes
`pheno_x`	A string representing the name of the phenotype whose effect is plotted on the x axis
`pheno_y`	A string representing the name of the phenotype whose effect is plotted on the y axis

annotate_with A string, The name of the column that contains the label for the datapoints (default value is Gene_Symbol)

thresh A number. Threshold cutoff, datapoints with P2 below this threshold are shown as filled circles whereas datapoints with P2 above this threshold are shown as open circles

ci_thresh A number.Show the confidence intervals if the P-value is below this threshold

gene_label_thresh A string, label datapoints with P2 below this threshold

color A string, default value is the first of the topr colors

scale A number, to change the size of the title and axes labels and ticks at the same time (default = 1)

Value

ggplot object

Examples

```
CD_UKBB_index_snps <- get_best_snp_per_MB(CD_UKBB)
snpset <- create_snpset(CD_UKBB_index_snps,CD_FINNGEN)
effect_plot(snpset)
```

flip_to_positive_allele_for_dat1

Flip to the positive allele for dataset 1

Description

```
flip_to_positive_allele_for_dat1()
```

Usage

```
flip_to_positive_allele_for_dat1(df)
```

Arguments

df A dataframe that is in the snpset format (like returned by the get_snpset() function)

Value

The input dataframe after flipping to the positive effect allele in dataframe 1

Examples

```
CD_UKBB_index_snps <-get_best_snp_per_MB(CD_UKBB)
snpset <- create_snpset(CD_UKBB_index_snps,CD_FINNGEN)
flip_to_positive_allele_for_dat1(snpset)
```

`get_best_snp_per_MB` *Get the index/lead variants*

Description

`get_best_snp_per_MB()` Get the top variants within 1 MB windows of the genome with association p-values below the given threshold

Usage

```
get_best_snp_per_MB(
  df,
  thresh = 1e-09,
  region_size = 1e+06,
  protein_coding_only = FALSE,
  chr = NULL,
  .checked = FALSE,
  verbose = FALSE
)
```

Arguments

`df` Dataframe

`thresh` A number. P-value threshold, only extract variants with p-values below this threshold (1e-09 by default)

`region_size` An integer (default = 1000000) indicating the window size for variant labelling. Increase this number for sparser annotation and decrease for denser annotation.

`protein_coding_only` Logical, set this variable to TRUE to only use protein_coding genes for annotation

`chr` String, get the top variants from one chromosome only, e.g. chr="chr1"

`.checked` Logical, if the input data has already been checked, this can be set to TRUE so it wont be checked again (FALSE by default)

`verbose` Logical, set to TRUE to get printed information on number of SNPs extracted

Value

Dataframe of lead variants. Returns the best variant per MB (by default, change the region size with the region argument) with p-values below the input threshold (thresh=1e-09 by default)

Examples

```
get_best_snp_per_MB(CD_UKBB)
```

`get_gene` *Get the genetic position of a gene by gene name*

Description

`get_gene()` Get the gene coordinates for a gene Required parameter is gene name

Usage

```
get_gene(gene_name, chr = NULL)
```

Arguments

`gene_name` A string representing a gene name (e.g. "FTO")

`chr` A string, search for the genes on this chromsome only, (e.g chr="chr1")

Value

Dataframe of genes

Examples

```
get_gene("FTO")
```

`get_genes_by_Gene_Symbol` *Get the genetic position of a gene by its gene name*

Description

`get_genes_by_Gene_Symbol()` Get genes by their gene symbol/name Required parameters is on gene name or a vector of gene names

Usage

```
get_genes_by_Gene_Symbol(genes, chr = NULL)
```

Arguments

`genes` A string or vector of strings representing gene names, (e.g. "FTO") or (c("FTO","NOD2"))

`chr` A string, search for the genes on this chromsome only, (e.g chr="chr1")

Value

Dataframe of genes

Examples

```
get_genes_by_Gene_Symbol(c("FTO","THADA"))
```

---

get_overlapping_snps_by_pos

Get variants that overlap between two datasets

Description

`get_overlapping_snps_by_pos()`

Usage

```
get_overlapping_snps_by_pos(df1, df2, verbose = FALSE)
```

Arguments

`df1`	A dataframe of variants, has to contain CHROM and POS
`df2`	A dataframe of variants, has to contain CHROM and POS
`verbose`	A logical scalar (default: FALSE). Assign to TRUE to get information on which alleles are matched and which are not.

Value

The input dataframe containing only those variants with matched alleles in the snpset

Examples

```

CD_UKBB_index_snps <- get_best_snp_per_MB(CD_UKBB)
get_overlapping_snps_by_pos(CD_UKBB_index_snps, CD_FINNGEN)
```

get_snps_within_region

Get SNPs/variants within region

Description

`get_snps_within_region()`

Usage

```
get_snps_within_region(df, region, chr = NULL, xmin = NULL, xmax = NULL)
```

Arguments

`df`	data frame of association results with the columns CHR and POS
`region`	A string representing the genetic region (e.g chr16:50693587-50734041)
`chr`	A string, chromosome (e.g. chr16)
`xmin`	An integer, include variants with POS larger than xmin
`xmax`	An integer, include variants with POS smaller than xmax

Value

the variants within the requested region

Examples

```
get_snps_within_region(CD_UKBB, "chr16:50593587-50834041")
```

get_topr_colors

Get the top hit from the dataframe

Description

`get_topr_colors()` Get the top hit from the dataframe All other input parameters are optional

Usage

```
get_topr_colors()
```

Value

Vector of colors used for plotting

Examples

```
get_topr_colors()
```

get_top_snp *Get the top hit from the dataframe*

Description

`get_top_snp()` Get the top hit from the dataframe All other input parameters are optional

Usage

```
get_top_snp(df, chr = NULL)
```

Arguments

`df`	Dataframe containing association results
`chr`	String, get the top hit in the data frame for this chromosome. If chromosome is not provided, the top hit from the entire dataset is returned.

Value

Dataframe containing the top hit

Examples

```
get_top_snp(CD_UKBB, chr="chr1")
```

locuszoom *Create a locuszoom-like plot*

Description

`locuszoom()` displays the association results for a smaller region within one chromosome. Required parameter is at least one dataset (dataframe) containing the association data (with columns `CHROM,POS,P` in upper or lowercase)

Usage

```
locuszoom(
  df,
  annotate = NULL,
  ntop = 3,
  xmin = 0,
  size = 2,
  shape = 19,
  alpha = 1,
  label_size = 4,
  annotate_with = ”ID”,
  color = NULL,
  axis_text_size = 11,
  axis_title_size = 12,
  title_text_size = 13,
  show_genes = FALSE,
  show_overview = FALSE,
  show_exons = FALSE,
  max_genes = 200,
  sign_thresh = 5e-09,
  sign_thresh_color = ”red”,
  sign_thresh_label_size = 3.5,
  xmax = NULL,
  ymin = NULL,
  ymax = NULL,
  protein_coding_only = FALSE,
  region_size = 1e+06,
  gene_padding = 1e+05,
  angle = 0,
  legend_title_size = 12,
  legend_text_size = 12,
  nudge_x = 0.01,
  nudge_y = 0.01,
  rsids = NULL,
  variant = NULL,
  rsids_color = ”gray40”,
  legend_name = ”Data:”,
  legend_position = ”right”,
  chr = NULL,
  vline = NULL,
  show_gene_names = NULL,
  legend_labels = NULL,
  gene = NULL,
  title = NULL,
  label_color = ”gray40”,
  region = NULL,
  scale = 1,
  rsids_with_vline = NULL,
  annotate_with_vline = NULL,
  sign_thresh_size = 0.5,
  unit_main = 7,
  unit_gene = 2
)
```

Arguments

`df` Dataframe or a list of dataframes (required columns are `CHROM,POS,P`), in upper- or lowercase) of association results.

`annotate` A number (p-value). Display annotation for variants with p-values below this threshold

`ntop` An integer, number of datasets (GWAS results) to show on the top plot

`xmin` Integer, setting the chromosomal range to display on the x-axis

`size` An integer setting the size of the plot points (default: `size=1.2`)

`shape` A number of vector of numbers setting the shape of the plotted points

`alpha` A number or vector of numbers setting the transparency of the plotted points

`label_size` An number to set the size of the plot labels (default: `label_size=3`)

`annotate_with` A string. Annotate the variants with either Gene_Symbol or ID (default: "Gene_Symbol")

`color` A string or a vector of strings, for setting the color of the datapoints on the plot

`axis_text_size` A number, size of the x and y axes tick labels (default: 12)

`axis_title_size` A number, size of the x and y title labels (default: 12)

`title_text_size` A number, size of the plot title (default: 13)

`show_genes` A logical scalar, show genes instead of exons (default show_genes=FALSE)

`show_overview` A logical scalar, shows/hides the overview plot (default= TRUE)

`show_exons` A logical scalar, show exons instead of genes (default show_exons=FALSE)

`max_genes` An integer, only label the genes if they are fewer than max_genes (default values is 200).

`sign_thresh` A number or vector of numbers, setting the horizontal significance threshold (default: `sign_thresh=5.1e-9`). Set to NULL to hide the significance threshold.

`sign_thresh_color` A string or vector of strings to set the color/s of the significance threshold/s

`sign_thresh_label_size` A number setting the text size of the label for the signficance thresholds (default text size is 3.5)

`xmax` Integer, setting the chromosomal range to display on the x-axis

`ymin` Integer, min and max of the y-axis, (default values: `ymin=0,ymax=max(-log10(df$P))`)

`ymax` Integer, min and max of the y-axis, (default values: `ymin=0,ymax=max(-log10(df$P))`)

`protein_coding_only` A logical scalar, if TRUE, only protein coding genes are used for annotation

`region_size` An integer (default = 1000000) indicating the window size for variant labelling. Increase this number for sparser annotation and decrease for denser annotation.

`gene_padding` An integer representing size of the region around the gene, if the gene argument was used (default = 100000)

`angle` A number, the angle of the text label

`legend_title_size` A number, size of the legend title

`legend_text_size`
A number, size of the legend text

`nudge_x` A number to vertically adjust the starting position of each gene label (this is a ggrepel parameter)

`nudge_y` A number to horizontally adjust the starting position of each gene label (this is a ggrepel parameter)

`rsids` A string (rsid) or vector of strings to highlight on the plot, e.g. `rsids=c("rs1234,rs45898")`

`variant` A string representing the variant to zoom in on. Can be either an rsid, or a dataframe (with the columns CHROM,POS,P)

`rsids_color` A string, the color of the variants in variants_id (default color is red)

`legend_name` A string, use to change the name of the legend (default: None)

`legend_position`
A string, top,bottom,left or right

`chr` A string or integer, the chromosome to plot (i.e. chr15), only required if the input dataframe contains results from more than one chromosome

`vline` A number or vector of numbers to add a vertical line to the plot at a specific chromosomal position, e.g `vline=204000066`. Multiple values can be provided in a vector, e.g `vline=c(204000066,100500188)`

`show_gene_names`
A logical scalar, if set to TRUE, gene names are shown even though they exceed the max_genes count

`legend_labels` A string or vector of strings representing legend labels for the input dataset/s

`gene` A string representing the gene to zoom in on (e.g. gene=FTO)

`title` A string

`label_color` A string. To change the color of the gene or variant labels

`region` A string representing a genetic region, e.g. chr1:67038906-67359979

`scale` A number, to change the size of the title and axes labels and ticks at the same time (default = 1)

`rsids_with_vline`
A string (rsid) or vector of strings to highlight on the plot with their rsids and vertical lines further highlighting their positions

`annotate_with_vline`
A number (p-value). Display annotation and vertical lines for variants with p-values below this threshold

`sign_thresh_size`
A number, sets the size of the horizontal significance threshold line (default = 1)

`unit_main` the height unit of the main plot (default = 7)

`unit_gene` the height unit of the gene plot (default= 2)

Value

plots using egg (https://cran.r-project.org/web/packages/egg/vignettes/Ecosystem.html)

Examples

```
locuszoom(R2_CD_UKBB)
```

manhattan *Create a Manhattan plot*

Description

manhattan() displays association results for the entire genome on a Manhattan plot. Required parameter is at least one dataset (dataframe) containing the association data (with columns CHROM,POS,P in upper or lowercase)

All other input parameters are optional

Usage

```
manhattan(
  df,
  ntop = 3,
  title = "",
  annotate = NULL,
  color = get_topr_colors(),
  sign_thresh = 5e-09,
  sign_thresh_color = "red",
  sign_thresh_label_size = 3.5,
  label_size = 3.5,
  size = 0.8,
  shape = 19,
  alpha = 1,
  highlight_genes_color = "green",
  highlight_genes_ypos = 1,
  axis_text_size = 12,
  axis_title_size = 14,
  title_text_size = 15,
  legend_title_size = 13,
  legend_text_size = 12,
  protein_coding_only = TRUE,
  angle = 0,
  legend_labels = NULL,
  chr = NULL,
  annotate_with = "Gene_Symbol",
  region_size = 1e+06,
  legend_name = NULL,
  legend_position = "bottom",
  nudge_x = 0.1,
  nudge_y = 0.2,
  xmin = NULL,
  xmax = NULL,
  ymin = NULL,
  ymax = NULL,
  highlight_genes = NULL,
  label_color = NULL,
  legend_nrow = NULL,
  gene_label_size = NULL,
  gene_label_angle = 0,
  scale = 1,
  show_legend = TRUE,
  sign_thresh_linetype = "dashed",
  sign_thresh_size = 0.5,
  rsids = NULL,
  rsids_color = NULL,
  rsids_with_vline = NULL,
  annotate_with_vline = NULL
)
```

Arguments

`df` Dataframe or a list of dataframes (required columns are `CHROM,POS,P`), in upper- or lowercase) of association results.

`ntop` An integer, number of datasets (GWAS results) to show on the top plot

`title` A string

`annotate` A number (p-value). Display annotation for variants with p-values below this threshold

`color` A string or a vector of strings, for setting the color of the datapoints on the plot

`sign_thresh` A number or vector of numbers, setting the horizontal significance threshold (default: `sign_thresh=5.1e-9`). Set to NULL to hide the significance threshold.

`sign_thresh_color`
A string or vector of strings to set the color/s of the significance threshold/s

`sign_thresh_label_size`
A number setting the text size of the label for the signficance thresholds (default text size is 3.5)

`label_size` An number to set the size of the plot labels (default: `label_size=3`)

`size` An integer setting the size of the plot points (default: `size=1.2`)

`shape` A number of vector of numbers setting the shape of the plotted points

`alpha` A number or vector of numbers setting the transparency of the plotted points

`highlight_genes_color`
A string, color for the highlighted genes (default: green)

`highlight_genes_ypos`
An integer, controlling where on the y-axis the highlighted genes are placed (default value is 1)

`axis_text_size` A number, size of the x and y axes tick labels (default: 12)

`axis_title_size`
A number, size of the x and y title labels (default: 12)

`title_text_size`
A number, size of the plot title (default: 13)

`legend_title_size`
A number, size of the legend title

`legend_text_size`
A number, size of the legend text

`protein_coding_only`
A logical scalar, if TRUE, only protein coding genes are used for annotation

`angle` A number, the angle of the text label

`legend_labels` A string or vector of strings representing legend labels for the input dataset/s

`chr` A string or integer, the chromosome to plot (i.e. chr15), only required if the input dataframe contains results from more than one chromosome

`annotate_with` A string. Annotate the variants with either Gene_Symbol or ID (default: "Gene_Symbol")

`region_size` An integer (default = 1000000) indicating the window size for variant labelling. Increase this number for sparser annotation and decrease for denser annotation.

`legend_name` A string, use to change the name of the legend (default: None)

`legend_position`
A string, top,bottom,left or right

`nudge_x` A number to vertically adjust the starting position of each gene label (this is a ggrepel parameter)

`nudge_y` A number to horizontally adjust the starting position of each gene label (this is a ggrepel parameter)

`xmin, xmax` Integer, setting the chromosomal range to display on the x-axis

`ymin, ymax` Integer, min and max of the y-axis, (default values: `ymin=0,ymax=max(-log10(df$P))`)

`highlight_genes`
A string or vector of strings, gene or genes to highlight at the bottom of the plot

`label_color` A string. To change the color of the gene or variant labels

`legend_nrow` An integer, sets the number of rows allowed for the legend labels

`gene_label_size`
A number setting the size of the gene labels shown at the bottom of the plot

`gene_label_angle`
A number setting the angle of the gene label shown at the bottom of the plot (default: 0)

`scale` A number, to change the size of the title and axes labels and ticks at the same time (default = 1)

`show_legend` A logical scalar, set to FALSE to hide the legend (default value is TRUE)

`sign_thresh_linetype`
A string, the linetype of the horizontal significance threshold (default = dashed)

`sign_thresh_size`
A number, sets the size of the horizontal significance threshold line (default = 1)

`rsids` A string (rsid) or vector of strings to highlight on the plot, e.g. `rsids=c("rs1234,rs45898")`

`rsids_color` A string, the color of the variants in variants_id (default color is red)

`rsids_with_vline`
: A string (rsid) or vector of strings to highlight on the plot with their rsids and vertical lines further highlighting their positions

`annotate_with_vline`
: A number (p-value). Display annotation and vertical lines for variants with p-values below this threshold

Value

ggplot object

Examples

```
manhattan(CD_UKBB)
```

match_alleles *Match the variants in the snpset by their alleles*

Description

`match_alleles()`

Usage

```
match_alleles(df, verbose = FALSE)
```

Arguments

`df` A dataframe that is in the snpset format (like returned by the get_snpset() function)

`verbose` A logical scalar (default: FALSE). Assign to TRUE to get information on which alleles are matched and which are not.

Value

The input dataframe containing only those variants with matched alleles in the snpset

Examples

```
CD_UKBB_index_snps <-get_best_snp_per_MB(CD_UKBB)
snpset <- create_snpset(CD_UKBB_index_snps,CD_FINNGEN)
match_alleles(snpset)
```

qqtopr *Create a QQ plot*

Description

qqtopr() displays QQ plots for association data. Required parameter is at least one dataset (dataframe) containing the association data (with columns CHROM,POS,P

Usage

```
qqtopr(
  dat,
  scale = 1,
  n_variants = 0,
  breaks = 15,
  title = NULL,
  color = get_topr_colors(),
  size = 1,
  legend_name = "",
  legend_position = "right",
  legend_labels = NULL,
  axis_text_size = 11,
  axis_title_size = 12,
  title_text_size = 13,
  legend_title_size = 12,
  legend_text_size = 12
)
```

Arguments

dat Dataframe or a list of dataframes (required columns is P)) of association results.

scale An integer, plot elements scale, default: 1

n_variants An integer, total number of variants used in the study

breaks A number setting the breaks for the axes

title A string

color A string or vector of strings setting the color/s for the input dataset/s

size An integer setting the size of the plot points (default: size=1.2)

legend_name A string, use to change the name of the legend (default: None)

legend_position
A string, top,bottom,left or right

legend_labels A string or vector of strings representing legend labels for the input dataset/s

axis_text_size A number, size of the x and y axes tick labels (default: 12)

axis_title_size
A number, size of the x and y title labels (default: 12)

`title_text_size`
: A number, size of the plot title (default: 13)

`legend_title_size`
: A number, size of the legend title

`legend_text_size`
: A number, size of the legend text

Value

ggplot

R2_CD_UKBB *Example dataset including the R2 column for the locuszoom plot function*

Description

The dataset is a subset of CD_UKBB and only includes variants above and near the IL23R gene on chromosome 1

Usage

```
R2_CD_UKBB
```

Format

A data frame with 26,824 rows and 10 variables:

CHROM Chromosome, written as for example chr1 or 1

POS genetic position of the variant

ID Variant identifier, e.g. rsid

P P-value from Plink run, additive model, reggresion model GLM_FIRTH

R2 variant correlation (r^2)

Source

A subset of the CD_UKBB dataset

regionplot *Create a regionplot*

Description

regionplot() displays the association results for a smaller genetic regions within one chromosome. Required parameter is at least one dataset (dataframe) containing the association data (with columns CHROM,POS,P in upper or lowercase) and either a variant ID, gene name or the genetic region represented as a chromosome together with start and stop positions (either as a single string or as three separate arguments).

All other input parameters are optional

Usage

```
regionplot(
  df,
  ntop = 3,
  annotate = NULL,
  xmin = 0,
  size = 2,
  shape = 19,
  alpha = 1,
  label_size = 4,
  annotate_with = "ID",
  color = get_topr_colors(),
  axis_text_size = 11,
  axis_title_size = 12,
  title_text_size = 13,
  show_genes = FALSE,
  show_overview = TRUE,
  show_exons = FALSE,
  max_genes = 200,
  sign_thresh = 5e-09,
  sign_thresh_color = "red",
  sign_thresh_label_size = 3.5,
  xmax = NULL,
  ymin = NULL,
  ymax = NULL,
  protein_coding_only = FALSE,
  region_size = 1e+06,
  gene_padding = 1e+05,
  angle = 0,
  legend_title_size = 12,
  legend_text_size = 11,
  nudge_x = 0.01,
  nudge_y = 0.01,
  rsids = NULL,
  variant = NULL,
  rsids_color = NULL,
  legend_name = "",
  legend_position = "right",
  chr = NULL,
  vline = NULL,
  show_gene_names = NULL,
  legend_labels = NULL,
  gene = NULL,
  title = NULL,
  label_color = NULL,
  locuszoomplot = FALSE,
  region = NULL,
  legend_nrow = NULL,
  gene_label_size = NULL,
  scale = 1,
  show_legend = TRUE,
  sign_thresh_linetype = "dashed",
  sign_thresh_size = 0.5,
  rsids_with_vline = NULL,
  annotate_with_vline = NULL,
  show_gene_legend = TRUE,
  unit_main = 7,
  unit_gene = 2,
  unit_overview = 1.25,
  verbose = TRUE
)
```

Arguments

`df` Dataframe or a list of dataframes (required columns are `CHROM,POS,P`), in upper- or lowercase) of association results.

`ntop` An integer, number of datasets (GWAS results) to show on the top plot

`annotate` A number (p-value). Display annotation for variants with p-values below this threshold

`xmin` Integer, setting the chromosomal range to display on the x-axis

`size` An integer setting the size of the plot points (default: `size=1.2`)

`shape` A number of vector of numbers setting the shape of the plotted points

`alpha` A number or vector of numbers setting the transparency of the plotted points

`label_size` An number to set the size of the plot labels (default: `label_size=3`)

`annotate_with` A string. Annotate the variants with either Gene_Symbol or ID (default: "Gene_Symbol")

`color` A string or a vector of strings, for setting the color of the datapoints on the plot

`axis_text_size` A number, size of the x and y axes tick labels (default: 12)

`axis_title_size`

A number, size of the x and y title labels (default: 12)

`title_text_size`
A number, size of the plot title (default: 13)

`show_genes` A logical scalar, show genes instead of exons (default show_genes=FALSE)

`show_overview` A logical scalar, shows/hides the overview plot (default= TRUE)

`show_exons` A logical scalar, show exons instead of genes (default show_exons=FALSE)

`max_genes` An integer, only label the genes if they are fewer than max_genes (default values is 200).

`sign_thresh` A number or vector of numbers, setting the horizontal significance threshold (default: `sign_thresh=5.1e-9`). Set to NULL to hide the significance threshold.

`sign_thresh_color`
A string or vector of strings to set the color/s of the significance threshold/s

`sign_thresh_label_size`
A number setting the text size of the label for the signficance thresholds (default text size is 3.5)

`xmax` Integer, setting the chromosomal range to display on the x-axis

`ymin` Integer, min and max of the y-axis, (default values: `ymin=0,ymax=max(-log10(df$P))`)

`ymax` Integer, min and max of the y-axis, (default values: `ymin=0,ymax=max(-log10(df$P))`)

`protein_coding_only`
A logical scalar, if TRUE, only protein coding genes are used for annotation

`region_size` An integer (default = 1000000) indicating the window size for variant labelling. Increase this number for sparser annotation and decrease for denser annotation.

`gene_padding` An integer representing size of the region around the gene, if the gene argument was used (default = 100000)

`angle` A number, the angle of the text label

`legend_title_size`
A number, size of the legend title

`legend_text_size`
A number, size of the legend text

`nudge_x` A number to vertically adjust the starting position of each gene label (this is a ggrepel parameter)

`nudge_y` A number to horizontally adjust the starting position of each gene label (this is a ggrepel parameter)

`rsids` A string (rsid) or vector of strings to highlight on the plot, e.g. `rsids=c("rs1234,rs45898")`

`variant` A string representing the variant to zoom in on. Can be either an rsid, or a dataframe (with the columns CHROM,POS,P)

`rsids_color` A string, the color of the variants in variants_id (default color is red)

`legend_name` A string, use to change the name of the legend (default: None)

`legend_position`
A string, top,bottom,left or right

`chr` A string or integer, the chromosome to plot (i.e. chr15), only required if the input dataframe contains results from more than one chromosome `vline` A number or vector of numbers to add a vertical line to the plot at a specific chromosomal position, e.g `vline=204000066`. Multiple values can be provided in a vector, e.g `vline=c(204000066,100500188)`

`show_gene_names` A logical scalar, if set to TRUE, gene names are shown even though they exceed the max_genes count

`legend_labels` A string or vector of strings representing legend labels for the input dataset/s

`gene` A string representing the gene to zoom in on (e.g. gene=FTO)

`title` A string

`label_color` A string. To change the color of the gene or variant labels

`locuszoomplot` A logical scalar set to FALSE. Only set to TRUE by calling the locuszoom function

`region` A string representing a genetic region, e.g. chr1:67038906-67359979

`legend_nrow` An integer, sets the number of rows allowed for the legend labels

`gene_label_size` A number setting the size of the gene labels shown at the bottom of the plot

`scale` A number, to change the size of the title and axes labels and ticks at the same time (default = 1)

`show_legend` A logical scalar, set to FALSE to hide the legend (default value is TRUE)

`sign_thresh_linetype` A string, the linetype of the horizontal significance threshold (default = dashed)

`sign_thresh_size` A number, sets the size of the horizontal significance threshold line (default = 1)

`rsids_with_vline` A string (rsid) or vector of strings to highlight on the plot with their rsids and vertical lines further highlighting their positions

`annotate_with_vline` A number (p-value). Display annotation and vertical lines for variants with p-values below this threshold

`show_gene_legend` A logical scalar, set to FALSE to hide the gene legend (default value is TRUE)

`unit_main` the height unit of the main plot (default = 7)

`unit_gene` the height unit of the gene plot (default= 2)

`unit_overview` the height unit of the overview plot (default = 1.25)

`verbose` Logical, set to FALSE to get suppress printed information

Value

plots within ggplotGrobs, arranged with egg::gtable_frame

Examples

```
regionplot(CD_UKBB, gene="IL23R")
```

topr *topr*

Description

A package for viewing and annotating genetic association data

topr functions

The main plotting functions are:

- manhattan to create Manhattan plot of association results
- regionplot to create regional plots of association results for smaller genetic regions

Examples

```
library(topr)
# Create a manhattan plot using
manhattan(CD_UKBB)

# Create a regional plot
regionplot(CD_UKBB, gene="IL23R")

# Get the lead/index snps (the top snp per MB window)
get_best_snp_per_MB(CD_UKBB)

# Annotate the index snps with their nearest gene
index_snps <- get_best_snp_per_MB(CD_UKBB)
annotate_with_nearest_gene(index_snps)
```

UC_UKBB *UKBB Ulcerative colitis (ICD 10 code K51)*

Description

Dataset retrieved from the UK biobank including of 5,452 UC cases (K51) and 481,862 controls. The dataset has been filtered on variants with P<1e-03.

Usage

```
UC_UKBB
```

Format

A data frame with 57,383 rows and 10 variables

CHROM Chromosome, written as for example chr1 or 1

POS genetic position of the variant

ID Variant identifier, e.g. rsid

P P-value from Plink run, additive model, reggresion model GLM_FIRTH

OR Odds Ratio

Source

Ulcerative Colitis UKBB ICD10 code K51, only including variants with P<1e-03

Index