Package 'topr'

April 13, 2026

Title Create Custom Plots for Viewing Genetic Association Results

Version 2.0.4

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

BugReports https://github.com/totajuliusd/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 7.3.3

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), scales (>= 1.1.1), ggrepel, egg, gridExtra, magrittr (>= 1.5), utils, tidyr, enshuman, purrr, tibble

Depends R (>= 4.1.0)

NeedsCompilation no

Author Thorhildur Juliusdottir [cph, aut, cre], Andri Stefansson [aut], Kyle Scott [ctb]

Maintainer Thorhildur Juliusdottir <totajuliusd@gmail.com>

Repository CRAN

Date/Publication 2026-04-13 10:10:02 UTC

Contents

`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,
  build = 38,
  .chr_map = NULL
)
```

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)
build	A number representing the genome build. Set to 37 to change to build (GRCh37). The default is build 38 (GRCh38).
.chr_map	An internally used list which maps chromosome names to numbers.

Value

the input dataframe with Gene_Symbol as an additional column

Examples

```
## Not run:
variants <-get_lead_snps(CD_UKBB)
annotate_with_nearest_gene(variants)

## End(Not run)
```

CD_FINNGEN *Finngen r7 Crohn's disease (K11_CROHNS)*

Description

Dataset retrieved from the Finngen database (version 7) including 3147 crohn´s cases (K50) and 296,100 controls. The dataset has been filtered on variants with P <1e-03. FinnGen data are publicly available and were downloaded from https://finngen.fi.

Usage

```
CD_FINNGEN
```

Format

A data frame with 32,303 rows and 8 variables:

CHROM Chromosome, written as for example chr1 or 1

POS genetic position of the variant

REF the reference allele

ALT the alternative allele

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

BETA Variant effect

ID Variant identifier, e.g. rsid

AF Allele frequency

Source

Crohn's K50 (K11_CROHNS), 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 21,717 rows and 8 variables:

CHROM Chromosome, written as for example chr1 or 1

POS genetic position of the variant

REF the reference allele

ALT the alternative allele

ID Variant identifier, e.g. rsid

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

OR Odds Ratio

AF Allele frequency

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()`

This method is deprecated and will be removed in future versions. use `get_snpset` instead.

Usage

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

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

```
## Not run:
create_snpset(CD_UKBB,CD_FINNGEN, thresh=1e-09)

## End(Not run)
```

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

Description

This method is deprecated and will be removed in future versions. use `get_snpset_code` instead.

```
create_snpset_code()
```

Usage

```
create_snpset_code()
```

Value

Dataframe containing the top hit

Examples

```
## Not run:
create_snpset_code()

## End(Not run)
```

effectplot *Create a plot comparing variant effects in two datasets*

Description

```
effectplot()
```

Usage

```
effectplot(
  df,
  pheno_x = "x_pheno",
  pheno_y = "y_pheno",
  annotate_with = "Gene_Symbol",
  thresh = 5e-08,
  ci_thresh = 1,
  gene_label_thresh = 5e-08,
  color = get_topr_colors()[1],
  scale = 1,
  build = 38,
  label_fontface = "italic",
  label_family = "",
```

```
  nudge_y = 0.001,
  nudge_x = 0.001,
  size = 2,
  segment.size = 0.2,
  segment.linetype = "solid",
  segment.color = "transparent",
  angle = 0,
  title = NULL,
  axis_text_size = 10,
  axis_title_size = 12,
  title_text_size = 13,
  subtitle_text_size = 11,
  gene_label_size = 3.2,
  snpset_thresh = 5e-08,
  snpset_region_size = 1e+06,
  max.overlaps = 10,
  annotate = 0,
  label_color = NULL
)
```

Arguments

df	The input dataframe (snpset) containing one row per variant and P values (P1 and P2) and effects (E1 and E2) from two datasets/phenotypes OR a list containing two datasets.
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	Deprecated: A number, 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)
build	A number representing the genome build or a data frame. Set to 37 to change to build (GRCh37). The default is build 38 (GRCh38).
label_fontface	A string or a vector of strings. Label font “plain”, “bold”, “italic”, “bold.italic” (ggrepel argument)
label_family	A string or a vector of strings. Label font name (default ggrepel argument is "")

nudge_y	A number to horizontally adjust the starting position of each gene label (this is a ggrepel parameter)
nudge_x	A number to vertically adjust the starting position of each gene label (this is a ggrepel parameter)
size	A number or a vector of numbers, setting the size of the plot points (default: size=1.2)
segment.size	line segment color (ggrepel argument)
segment.linetype	line segment solid, dashed, etc.(ggrepel argument)
segment.color	line segment thickness (ggrepel argument)
angle	A number, the angle of the text label
title	A string to set the plot title
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)
subtitle_text_size	A number setting the text size of the subtitle (default: 11)
gene_label_size	A number setting the size of the gene labels shown at the bottom of the plot
snpset_thresh	A number representing the threshold used to create the snpset used for plotting (Only applicable if the input dataframe is a list containing two datasets)
snpset_region_size	A number representing the region size to use when creating the snpset used for plotting (Only applicable if the input dataframe is a list containing two datasets)
max.overlaps	Exclude text labels that overlap too many things. Defaults to 10 (ggrepel argument)
annotate	A number, label datapoints with p-value below below this number (in the second df) by their nearest gene
label_color	A string or a vector of strings. To change the color of the gene or variant labels

Value

ggplot object

Examples

```
## Not run:
effectplot(list(CD_UKBB, CD_FINNGEN))

## End(Not run)
```

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

Description

`effect_plot()`

This method is deprecated and will be removed in future versions. use `effectplot` instead.

Usage

```
effect_plot(
  dat,
  pheno_x = "pheno_x",
  pheno_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)

Examples

```
## Not run:
effect_plot(dat)

## End(Not run)
```

`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

```
## Not run:
CD_UKBB_index_snps <- get_lead_snps(CD_UKBB)
snpset <- get_snpset(CD_UKBB_index_snps, CD_FINNGEN)
flip_to_positive_allele_for_dat1(snpset$matched)

## End(Not run)
```

`foresttopr` *Create a forest plot from one or more association result tables*

Description

`foresttopr()` creates a forest plot visualizing effect estimates and confidence intervals across one or more datasets. The function supports odds ratios (OR) and regression coefficients (beta), allows matching rows across datasets by a key column, and optionally displays human-readable labels from a separate annotation column.

Effect estimates are automatically standardized across input datasets, and confidence intervals are derived preferentially from explicit bounds, standard errors, or p-values when necessary.

Usage

```
foresttopr(
  dat = NULL,
  legend_labels = NULL,
  colors = NULL,
  key_col = "ID",
  label_col = NULL,
  effect_type = c("OR", "beta"),
  xlim = NULL,
  xbreaks = NULL,
  xlabel = NULL,
  size = 2.5,
  shape = 16,
  alpha = 1,
  points_dist = 0.6,
  band_color = "grey96",
  band_border_color = "grey96",
  band_border_linewidth = 0.01,
  sign_thresh = NULL,
  ylabel_order = NULL,
  scale = 1,
  title = NULL,
  title_text_size = 15,
  axis_text_size = 12,
  axis_title_size = 14,
  show_shape_legend = TRUE,
  show_color_legend = TRUE,
  legend_position = "right",
  legend_nrow = NULL,
  legend_name = NULL,
  legend_title_size = axis_text_size * 0.95,
  legend_text_size = axis_text_size * 0.85,
  match_on_gene = FALSE
)
```

Arguments

dat	A data frame or a list of data frames containing association results. Each data frame must contain an effect estimate column (e.g. OR or BETA) and a p-value column. If a single data frame is provided, it is internally wrapped into a list.
legend_labels	A character vector of labels corresponding to each dataset in dat. These labels are used in the plot legend. Defaults to "Set1", "Set2", etc.
colors	A character vector of colors to use for each dataset. If NULL, a default color palette is used.
key_col	Character scalar giving the column name used to match rows across datasets (e.g. gene identifier or variant ID). Defaults to "gene".
label_col	Optional character scalar giving the column name in the reference dataset (the first element of dat) to use for labeling rows on the y-axis. If NULL, key_col is used for labeling.
effect_type	Character scalar specifying the effect scale to plot. Either "OR" (odds ratio; default) or "beta" (regression coefficient). Matching is case-insensitive. When required, effect estimates are automatically converted between scales.
xlim	Numeric length-2 vector giving x-axis limits. If NULL, limits are computed automatically from the data.
xbreaks	Numeric vector or function specifying x-axis breaks. If NULL, reasonable defaults are chosen based on effect_type.
xlabel	Character scalar giving the x-axis label. If NULL, a default label is chosen based on effect_type.
size	Numeric scalar or vector controlling point sizes for each dataset.
shape	Integer scalar or vector specifying point shapes for each dataset.
alpha	Numeric scalar or vector specifying point transparency.
points_dist	Numeric scalar controlling horizontal separation of points from different datasets within the same row.
band_color	Background color for alternating row bands.
band_border_color	Color for row band borders.
band_border_linewidth	Numeric scalar giving the line width for row band borders.
sign_thresh	Optional numeric scalar specifying a p-value threshold for highlighting statistically significant points via shape encoding.
ylabel_order	Optional character vector specifying the order of rows on the y-axis. If NULL, rows are ordered as they appear in the reference dataset.
scale	Numeric scalar used to globally scale text and point sizes.
title	Optional character scalar giving the plot title.
title_text_size	Numeric scalar controlling title text size.
axis_text_size	Numeric scalar controlling axis text size.

`axis_title_size`
: Numeric scalar controlling axis title text size.

`show_shape_legend`
: Logical; whether to display the shape legend.

`show_color_legend`
: Logical; whether to display the color legend.

`legend_position`
: Character string specifying legend position. One of `"right"`, `"top"`, or `"bottom"`.

`legend_nrow`
: Optional integer specifying the number of rows in the legend.

`legend_name`
: Optional character scalar giving the legend title.

`legend_title_size`
: Numeric scalar controlling legend title text size.

`legend_text_size`
: Numeric scalar controlling legend text size.

`match_on_gene`
: Logical; if `FALSE` and vdariant-level columns (e.g. REF/ALT) are detected, matching is performed at the variant level. Otherwise, matching is performed using `key_col`.

Value

A `ggplot2` object representing the forest plot.

See Also

`ggplot`

Examples

```
foresttopr(
  dat = list(
    CD_UKBB |>
      dplyr::arrange(P) |>
      head(n = 10) |>
      annotate_with_nearest_gene(),
    CD_FINNGEN
  ),
  key_col = "ID",
  label_col = "Gene_Symbol",
  legend_labels = c("CD_UKBB", "CD_FINNGEN"),
  effect_type = "beta"
)
```

`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

This method is deprecated and will be removed in future versions. use `get_lead_snps` instead.

Usage

```
get_best_snp_per_MB(
  df,
  thresh = 5e-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 (5e-09 by default)
`region_size`	An integer (default = 20000000) (or a string represented as 200kb or 2MB) indicating the window size for variant labeling. 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=5e-09 by default)

Examples

```
## Not run:
  get_best_snp_per_MB(CD_UKBB)

## End(Not run)
```

`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

This method is deprecated and will be removed in future versions. use `get_gene_coords` instead.

Usage

```
get_gene(gene_name, chr = NULL, build = 38)
```

Arguments

`gene_name`	A string representing a gene name (e.g. "FTO")
`chr`	A string, search for the genes on this chromosome only, (e.g chr="chr1")
`build`	A string, genome build, choose between builds 37 (GRCh37) and 38 (GRCh38) (default is 38)

Value

Dataframe with the gene name and its genetic coordinates

Examples

```
## Not run:
get_gene("FTO")

## End(Not run)
```

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, build = 38)
```

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 chromosome only, (e.g chr="chr1")
build	A string, genome build, choose between builds 37 (GRCh37) and 38 (GRCh38) (default is 38)

Value

Dataframe of genes

Examples

```
## Not run:
  get_genes_by_Gene_Symbol(c("FTO","THADA"))

## End(Not run)
```

get_genes_in_region *Get SNPs/variants within region*

Description

get_genes_in_region()

Usage

```
get_genes_in_region(
  chr = chr,
  xmin = xmin,
  xmax = xmax,
  protein_coding_only = F,
  show_exons = F,
  show_genes = T,
  build = 38,
  region = NULL
)
```

Arguments

chr	A string, chromosome (e.g. chr16)
xmin	An integer representing genetic position
xmax	An integer representing genetic position
protein_coding_only	A logical scalar, if TRUE, only protein coding genes are used for annotation
show_exons	Deprecated : A logical scalar, show exons instead of genes (default show_exons=FALSE)
show_genes	A logical scalar, show genes instead of exons (default show_genes=FALSE)
build	A number representing the genome build or a data frame. Set to 37 to change to build (GRCh37). The default is build 38 (GRCh38).
region	A string representing the genetic region (e.g chr16:50693587-50734041)

Value

the genes the requested region

Examples

```
## Not run:
get_genes_in_region(region="chr16:50593587-50834041")

## End(Not run)
```

---

get_gene_coords *Get the genetic position of a gene by gene name*

Description

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

Usage

```
get_gene_coords(gene_name, chr = NULL, build = 38)
```

Arguments

gene_name	A string representing a gene name (e.g. "FTO")
chr	A string, search for the genes on this chromosome only, (e.g chr="chr1")
build	A string, genome build, choose between builds 37 (GRCh37) and 38 (GRCh38) (default is 38)

Value

Dataframe with the gene name and its genetic coordinates

Examples

```
## Not run:
get_gene_coords("FTO")

## End(Not run)
```

---

get_lead_snps *Get the index/lead variants*

Description

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

Usage

```
get_lead_snps(
  df,
  thresh = 5e-08,
  region_size = 1e+06,
  protein_coding_only = FALSE,
  chr = NULL,
  .checked = FALSE,
  verbose = NULL,
  keep_chr = TRUE
)
```

Arguments

df	Dataframe
thresh	A number. P-value threshold, only extract variants with p-values below this threshold (5e-08 by default)
region_size	An integer (default = 20000000) (or a string represented as 200kb or 2MB) indicating the window size for variant labeling. 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

keep_chr
: Logical, set to FALSE to remove the "chr" prefix before each chromosome if present (TRUE by default)

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=5e-08 by default)

Examples

```
## Not run:
get_lead_snps(CD_UKBB)

## End(Not run)
```

---

get_overlapping_snps_by_pos

Get variants that overlap between two datasets

Description

`get_overlapping_snps_by_pos()`

This method is deprecated and will be removed in future versions. use `match_by_pos` instead.

Usage

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

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

```
## Not run:
get_overlapping_snps_by_pos(dat1, dat2)

## End(Not run)
```

`get_sign_and_sugg_loci` *Get the index/lead variants*

Description

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

Usage

```
get_sign_and_sugg_loci(
  df,
  genome_wide_thresh = 5e-08,
  suggestive_thresh = 1e-06,
  flank_size = 1e+06,
  region_size = 1e+06
)
```

Arguments

`df`	Dataframe, GWAS summary statistics
`genome_wide_thresh`	A number. P-value threshold for genome wide significant loci (5e-08 by default)
`suggestive_thresh`	A number. P-value threshold for suggestive loci (1e-06 by default)
`flank_size`	A number (default = 1e6). The size of the flanking region for the significant and suggestitve snps.
`region_size`	A number (default = 1e6). The size of the region for top snp search. Only one snp per region is returned.

Value

List of genome wide and suggestive loci.

Examples

```
## Not run:
get_sign_and_sugg_loci(CD_UKBB)

## End(Not run)
```

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

Description

```
get_snpset()
```

Usage

```
get_snpset(
  df1,
  df2,
  thresh = 5e-08,
  protein_coding_only = TRUE,
  region_size = 1e+06,
  verbose = NULL,
  show_full_output = FALSE,
  build = 38,
  format = "wide"
)
```

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	A number. P-value threshold, only extract variants with p-values below this threshold (5e-08 by default)
protein_coding_only	Logical, set this variable to TRUE to only use protein_coding genes for annotation
region_size	An integer (default = 20000000) (or a string represented as 200kb or 2MB) indicating the window size for variant labeling. Increase this number for sparser annotation and decrease for denser annotation.
verbose	Logical, (default: FALSE). Assign to TRUE to get information on which alleles are matched and which are not.
show_full_output	A logical scalar (default:FALSE). Assign to TRUE to show the full output from this function
build	A string, genome build, choose between builds 37 (GRCh37) and 38 (GRCh38) (default is 38)
format	A string, representing either wide or long format (default : "wide"). By default a snpset created from two dataframes is returned in a wide format.

Value

Dataframe of overlapping snps (snpset)

Examples

```
## Not run:
CD_UKBB_index_snps <-get_lead_snps(CD_UKBB)
get_snpset(CD_UKBB_index_snps, CD_FINNGEN)

## End(Not run)
```

`get_snpset_code` *Show the code/functions used to get a snpset*

Description

```
get_snpset_code()
```

Usage

```
get_snpset_code()
```

Value

Dataframe containing the top hit

Examples

```
## Not run:
get_snpset_code()

## End(Not run)
```

`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,
  keep_chr = 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
keep_chr	Deprecated: Logical, set to FALSE to remove the "chr" prefix before each chromosome if present (TRUE by default)

Value

the variants within the requested region

Examples

```
## Not run:
get_snps_within_region(CD_UKBB, "chr16:50593587-50834041")

## End(Not run)
```

---

get_topr_colors *Get the topr custom colors*

Description

`get_topr_colors()` Get topr custom colors

Usage

```
get_topr_colors()
```

Value

Vector of colors used for plotting

Examples

```
## Not run:
get_topr_colors()

## End(Not run)
```

get_topr_old_colors *Get the topr custom colors from v. 2.0.2*

Description

`get_topr_old_colors()` Get topr custom colors from v. 2.0.2

Usage

```
get_topr_old_colors()
```

Value

Vector of colors used for plotting

Examples

```
## Not run:
get_topr_old_colors()

## End(Not run)
```

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

```
## Not run:
get_top_snp(CD_UKBB, chr="chr1")

## End(Not run)
```

---

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 = NULL,
  show_overview = FALSE,
  show_exons = FALSE,
  max_genes = 200,
  sign_thresh = 5e-08,
  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,
  gene_color = NULL,
  segment.size = 0.2,
  segment.color = "black",
  segment.linetype = "solid",
  show_gene_legend = TRUE,
  max.overlaps = 10,
  extract_plots = FALSE,
  label_fontface = "plain",
  label_family = "",
  gene_label_fontface = "plain",
  gene_label_family = "",
  build = 38,
  verbose = NULL,
  show_legend = TRUE,
  label_alpha = 1,
  gene_label_size = NULL,
  vline_color = "grey",
  vline_linetype = "dashed",
  vline_alpha = 1,
  vline_size = 0.5,
  log_trans_p = TRUE
)
```

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

`size` A number or a vector of numbers, setting the size of the plot points (default: `size=1.2`)

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

`alpha` A number or a 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` Deprecated : 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=5e-8`). 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 significance thresholds (default text size is 3.5)

`ymin, 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 = 20000000) (or a string represented as 200kb or 2MB) indicating the window size for variant labeling. 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 datasets

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

`title` A string to set the plot title

`label_color` A string or a vector of strings. 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)

`gene_color` A string representing a color, can be used to change the color of the genes/exons on the geneplot

`segment.size` line segment color (ggrepel argument)

`segment.color` line segment thickness (ggrepel argument)

segment.linetype
: line segment solid, dashed, etc.(ggrepel argument)

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

max.overlaps
: Exclude text labels that overlap too many things. Defaults to 10 (ggrepel argument)

extract_plots
: Logical, FALSE by default. Set to TRUE to extract the three plots separately in a list

label_fontface
: A string or a vector of strings. Label font "plain", "bold", "italic", "bold.italic" (ggrepel argument)

label_family
: A string or a vector of strings. Label font name (default ggrepel argument is "")

gene_label_fontface
: Gene label font "plain", "bold", "italic", "bold.italic" (ggrepel argument)

gene_label_family
: Gene label font name (default ggrepel argument is "")

build
: A number representing the genome build or a data frame. Set to 37 to change to build (GRCh37). The default is build 38 (GRCh38).

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

show_legend
: A logical scalar, set to FALSE to hide the legend (default : TRUE)

label_alpha
: An number or vector of numbers to set the transparency of the plot labels (default: label_alpha=1)

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

vline_color
: A string. The color of added vertical line/s (default: grey)

vline_linetype
: A string. The linetype of added vertical line/s (default : dashed)

vline_alpha
: A number. The alpha of added vertical line/s (default : 1)

vline_size
: A number.The size of added vertical line/s (default : 0.5)

log_trans_p
: A logical scalar (default: TRUE). By default the p-values in the input datasets are log transformed using -log10. Set this argument to FALSE if the p-values in the datasets have already been log transformed.

Value

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

Examples

```
## Not run:
locuszoom(R2_CD_UKBB)

## End(Not run)
```

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 = 4,
  title = "",
  annotate = NULL,
  color = NULL,
  sign_thresh = 5e-08,
  sign_thresh_color = "darkred",
  sign_thresh_label_size = 3.5,
  label_size = 3.5,
  size = 0.8,
  shape = 19,
  alpha = 1,
  highlight_genes_color = "darkred",
  highlight_genes_ypos = 1.5,
  axis_text_size = 12,
  axis_title_size = 14,
  title_text_size = 15,
  legend_title_size = axis_text_size * 0.95,
  legend_text_size = axis_text_size * 0.85,
  protein_coding_only = TRUE,
  angle = 0,
  legend_labels = NULL,
  chr = NULL,
  annotate_with = "Gene_Symbol",
  region_size = 2e+07,
  legend_name = NULL,
  legend_position = "bottom",
  nudge_x = 0.1,
  nudge_y = 0.7,
  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,
  shades_color = NULL,
  shades_alpha = 0.5,
  segment.size = 0.2,
  segment.color = "black",
  segment.linetype = "dashed",
  max.overlaps = 10,
  label_fontface = "plain",
  label_family = "",
  gene_label_fontface = "plain",
  gene_label_family = "",
  build = 38,
  verbose = NULL,
  label_alpha = 1,
  shades_line_alpha = 1,
  vline = NULL,
  vline_color = "grey",
  vline_linetype = "dashed",
  vline_alpha = 1,
  vline_size = 0.5,
  region = NULL,
  theme_grey = FALSE,
  xaxis_label = "Chromosome",
  use_shades = FALSE,
  even_no_chr_lightness = 0.8,
  get_chr_lengths_from_data = TRUE,
  log_trans_p = TRUE,
  chr_ticknames = NULL,
  show_all_chrticks = FALSE,
  hide_chrticks_from_pos = 17,
  hide_chrticks_to_pos = NULL,
  hide_every_nth_chrtick = 2,
  downsample_cutoff = 0.05,
  downsample_prop = 0.1
)
```

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 to set the plot title
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=5e-8). 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 significance thresholds (default text size is 3.5)
label_size	An number to set the size of the plot labels (default: label_size=3)
size	A number or a vector of numbers, setting the size of the plot points (default: size=1.2)
shape	A number of a vector of numbers setting the shape of the plotted points
alpha	A number or a vector of numbers setting the transparency of the plotted points
highlight_genes_color	A string, color for the highlighted genes (default: darkred)
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 datasets
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 = 20000000) (or a string represented as 200kb or 2MB) indicating the window size for variant labeling. 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 or a vector of strings. 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 : TRUE)

`sign_thresh_linetype`
A string, the line-type 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

`shades_color` The color of the rectangles (shades) representing the different chromosomes on the Manhattan plot

`shades_alpha` The transparency (alpha) of the rectangles (shades)

`segment.size` line segment color (ggrepel argument)

`segment.color` line segment thickness (ggrepel argument)

segment.linetype
: line segment solid, dashed, etc.(ggrepel argument)

max.overlaps
: Exclude text labels that overlap too many things. Defaults to 10 (ggrepel argument)

label_fontface
: A string or a vector of strings. Label font "plain", "bold", "italic", "bold.italic" (ggrepel argument)

label_family
: A string or a vector of strings. Label font name (default ggrepel argument is "")

gene_label_fontface
: Gene label font "plain", "bold", "italic", "bold.italic" (ggrepel argument)

gene_label_family
: Gene label font name (default ggrepel argument is "")

build
: A number representing the genome build or a data frame. Set to 37 to change to build (GRCh37). The default is build 38 (GRCh38).

verbose
: A logical scalar (default: NULL). Set to FALSE to suppress printed messages

label_alpha
: An number or vector of numbers to set the transparency of the plot labels (default: `label_alpha=1`)

shades_line_alpha
: The transparency (alpha) of the lines around the rectangles (shades)

vline
: A number or vector of numbers to add a vertical line to the plot at a specific chromosomal position, e.g `vline="chr1:204000066"`. Multiple values can be provided in a vector, e.g `vline=c("chr1:204000066","chr5:100500188")`

vline_color
: A string. The color of added vertical line/s (default: grey)

vline_linetype
: A string. The linetype of added vertical line/s (default : dashed)

vline_alpha
: A number. The alpha of added vertical line/s (default : 1)

vline_size
: A number.The size of added vertical line/s (default : 0.5)

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

theme_grey
: A logical scalar (default: FALSE). Use gray rectangles (instead of white to distinguish between chromosomes)

xaxis_label
: A string. The label for the x-axis (default: Chromosome)

use_shades
: A logical scalar (default: FALSE). Use shades/rectangles to distinguish between chromosomes

even_no_chr_lightness
: Lightness value for even numbered chromosomes. A number or vector of numbers between 0 and 1 (default: 0.8). If set to 0.5, the same color as shown for odd numbered chromosomes is displayed. A value below 0.5 will result in a darker color displayed for even numbered chromosomes, whereas a value above 0.5 results in a lighter color.

get_chr_lengths_from_data
: A logical scalar (default: TRUE). If set to FALSE, use the inbuilt chromosome lengths (from hg38), instead of chromosome lengths based on the max position for each chromosome in the input dataset/s.

log_trans_p
: A logical scalar (default: TRUE). By default the p-values in the input datasets are log transformed using -log10. Set this argument to FALSE if the p-values in the datasets have already been log transformed.

`chr_ticknames` A vector containing the chromosome names displayed on the x-axis. If NULL, the following format is used: chr_ticknames <- c(1:16, ",18, ",20, ",22, 'X')

`show_all_chrticks`
A logical scalar (default : FALSE). Set to TRUE to show all the chromosome names on the ticks on the x-axis

`hide_chrticks_from_pos`
A number (default: 17). Hide every nth chromosome name on the x-axis FROM this position (chromosome number)

`hide_chrticks_to_pos`
A number (default: NULL). Hide every nth chromosome name on the x-axis TO this position (chromosome number). When NULL this variable will be set to the number of numeric chromosomes in the input dataset.

`hide_every_nth_chrtick`
A number (default: 2). Hide every nth chromosome tick on the x-axis (from the hide_chr_ticks_from_pos to the hide_chr_ticks_to_pos).

`downsample_cutoff`
A number (default: 0.05) used to downsample the input dataset prior to plotting. Sets the fraction of high p-value (default: P>0.05) markers to display on the plot.

`downsample_prop`
A number (default: 0.1) used to downsample the input dataset prior to plotting. Only a proportion of the variants (10% by default) with P-values higher than the downsample_cutoff will be displayed on the plot.

Value

ggplot object

Examples

```
## Not run:
manhattan(CD_UKBB)

## End(Not run)
```

`manhattanExtra` *Create a Manhattan plot highlighting genome-wide significant and suggestive loci*

Description

`manhattanExtra()` displays association results for the entire genome on a Manhattan plot, highlighting genome-wide significant and suggestive loci. 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

```
manhattanExtra(
  df,
  genome_wide_thresh = 5e-08,
  suggestive_thresh = 1e-06,
  flank_size = 1e+06,
  region_size = 1e+06,
  sign_thresh_color = NULL,
  sign_thresh_label_size = NULL,
  show_legend = TRUE,
  label_fontface = NULL,
  nudge_y = NULL,
  ymax = NULL,
  sign_thresh = NULL,
  label_color = NULL,
  color = NULL,
  legend_labels = NULL,
  annotate = NULL,
  ...
)
```

Arguments

`df`	Dataframe, GWAS summary statistics
`genome_wide_thresh`	A number. P-value threshold for genome wide significant loci (5e-08 by default)
`suggestive_thresh`	A number. P-value threshold for suggestive loci (1e-06 by default)
`flank_size`	A number (default = 1e6). The size of the flanking region for the significant and suggestitve snps.
`region_size`	A number (default = 1e6). The size of the region for gene annotation. Increase this number for sparser annotation and decrease for denser annotation.
`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 significance thresholds (default text size is 3.5)
`show_legend`	A logical scalar, set to FALSE to hide the legend (default : TRUE)
`label_fontface`	A string or a vector of strings. Label font "plain", "bold", "italic", "bold.italic" (ggrepel argument)
`nudge_y`	A number to horizontally adjust the starting position of each gene label (this is a ggrepel parameter)
`ymax`	Integer, max of the y-axis, (default value: `ymax=(max(-log10(df$P)) + max(-log10(df$P)) * .2))`
`sign_thresh`	A number or vector of numbers, setting the horizontal significance threshold (default: `sign_thresh=5e-8`). Set to NULL to hide the significance threshold.

label_color	A string or a vector of strings. To change the color of the gene or variant labels
color	A string or a vector of strings, for setting the color of the datapoints on the plot
legend_labels	A string or vector of strings representing legend labels for the input datasets
annotate	A number (p-value). Display annotation for variants with p-values below this threshold
...	Additional arguments passed to other plotting functions.

Value

ggplot object

Examples

```
## Not run:
manhattanExtra(df)

## End(Not run)
```

---

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

Description

match_alleles()

This method is deprecated and will be removed in future versions. use match_by_alleles instead.

Usage

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

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 whith matched alleles in the snpset

Examples

```
## Not run:
match_alleles(df)

## End(Not run)
```

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

Description

match_by_alleles()

Usage

```
match_by_alleles(df, verbose = NULL, show_full_output = 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.
show_full_output	A logical scalar (default:FALSE). Assign to TRUE to show the full output from this function

Value

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

Examples

```
## Not run:
CD_UKBB_lead_snps <- get_lead_snps(CD_UKBB)
snpset <- get_snpset(CD_UKBB_lead_snps, CD_FINNGEN)
match_by_alleles(snpset$found)

## End(Not run)
```

match_by_pos *Get variants that overlap between two datasets*

Description

match_by_pos()

Usage

```
match_by_pos(df1, df2, verbose = NULL, show_full_output = 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.
show_full_output	A logical scalar (default:FALSE). Assign to TRUE to show the full output from this function

Value

A list containing two dataframes, one of overlapping snps and the other snps not found in the second input dataset

Examples

```
## Not run:
CD_UKBB_index_snps <- get_lead_snps(CD_UKBB)
match_by_pos(CD_UKBB_index_snps, CD_FINNGEN)

## End(Not run)
```

---

qqtopr *Create a quantile quantile (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,
  verbose = NULL,
  diagonal_line_color = "#808080"
)
```

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 to set the plot title
color	A string or vector of strings setting the color's for the input datasets
size	A number or a vector of numbers, 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 datasets
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
verbose	A logical scalar (default: NULL). Set to FALSE to suppress printed messages
diagonal_line_color	A string setting the color of the diagonal line on the plot

Value

ggplot

Examples

```
## Not run:
qqtopr(CD_UKBB)

## End(Not run)
```

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 329 rows and 5 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

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 = 10,
  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 = NULL,
  show_overview = TRUE,
  show_exons = NULL,
  max_genes = 200,
  sign_thresh = 5e-08,
  sign_thresh_color = "darkred",
  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 = NULL,
  gene_color = NULL,
  segment.size = 0.2,
  segment.color = "black",
  segment.linetype = "solid",
  max.overlaps = 10,
  unit_ratios = NULL,
  extract_plots = FALSE,
  label_fontface = "plain",
  label_family = "",
  gene_label_fontface = "plain",
  gene_label_family = "",
  build = 38,
  label_alpha = 1,
  vline_color = "grey",
  vline_linetype = "dashed",
  vline_alpha = 1,
  vline_size = 0.5,
  log_trans_p = TRUE,
  lz_color = 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
annotate	A number (p-value). Display annotation for variants with p-values below this threshold
xmin, xmax	Integer, setting the chromosomal range to display on the x-axis
size	A number or a vector of numbers, setting the size of the plot points (default: size=1.2)
shape	A number of a vector of numbers setting the shape of the plotted points
alpha	A number or a 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 Deprecated : 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=5e-8`). 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 significance thresholds (default text size is 3.5)

ymin, 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 = 20000000) (or a string represented as 200kb or 2MB) indicating the window size for variant labeling. 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 datasets

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

`title`
: A string to set the plot title

`label_color`
: A string or a vector of strings. 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 : TRUE)

`sign_thresh_linetype`
: A string, the line-type 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

`gene_color`
: A string representing a color, can be used to change the color of the genes/exons on the geneplot

`segment.size`
: line segment color (ggrepel argument)

`segment.color` line segment thickness (ggrepel argument)

`segment.linetype` line segment solid, dashed, etc.(ggrepel argument)

`max.overlaps` Exclude text labels that overlap too many things. Defaults to 10 (ggrepel argument)

`unit_ratios` A string of three numbers separated by ":", for the overview, main and gene plots height ratios e.g 1.25:7:2

`extract_plots` Logical, FALSE by default. Set to TRUE to extract the three plots separately in a list

`label_fontface` A string or a vector of strings. Label font "plain", "bold", "italic", "bold.italic" (ggrepel argument)

`label_family` A string or a vector of strings. Label font name (default ggrepel argument is "")

`gene_label_fontface` Gene label font "plain", "bold", "italic", "bold.italic" (ggrepel argument)

`gene_label_family` Gene label font name (default ggrepel argument is "")

`build` A number representing the genome build or a data frame. Set to 37 to change to build (GRCh37). The default is build 38 (GRCh38).

`label_alpha` An number or vector of numbers to set the transparency of the plot labels (default: `label_alpha=1`)

`vline_color` A string. The color of added vertical line/s (default: grey)

`vline_linetype` A string. The linetype of added vertical line/s (default : dashed)

`vline_alpha` A number. The alpha of added vertical line/s (default : 1)

`vline_size` A number.The size of added vertical line/s (default : 0.5)

`log_trans_p` A logical scalar (default: TRUE). By default the p-values in the input datasets are log transformed using -log10. Set this argument to FALSE if the p-values in the datasets have already been log transformed.

`lz_color` A string, to set the base color in the locuszoom plot

Value

plots within ggplotGrobs, arranged with egg::gtable_frame

Examples

```
## Not run:
regionplot(CD_UKBB, gene="IL23R")

## End(Not run)
```

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 45,012 rows and 8 variables

CHROM Chromosome, written as for example chr1 or 1

POS genetic position of the variant

REF the reference allele

ALT the alternative allele

ID Variant identifier, e.g. rsid

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

OR Odds Ratio

AF Allele frequency

Source

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

Index