Package 'locuszoomr'

October 9, 2026

Title Gene Locus Plot with Gene Annotations

Version 1.2.0

BugReports https://github.com/myles-lewis/locuszoomr/issues

URL https://github.com/myles-lewis/locuszoomr

Description Publication-ready regional gene locus plots similar to those produced by the web interface 'LocusZoom' <https://my.locuszoom.org>, but running locally in R. Genetic or genomic data with gene annotation tracks are plotted via R base graphics, 'ggplot2' or 'plotly', allowing flexibility and easy customisation including laying out multiple locus plots on the same page. It uses the 'LDlink' API <https://ldlink.nih.gov/?tab=apiaccess> to query linkage disequilibrium data from the 1000 Genomes Project and can overlay this on plots <doi:10.1093/bioadv/vbaf006>.

Language en-gb

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 3.5)

Imports AnnotationFilter, BiocGenerics, cowplot, ensembldb, GenomeInfoDb, GenomicRanges, gggrid, ggplot2, graphics, grDevices, grid, gtools, htmltools, IRanges, LDlinkR, memoise, plotly, Rcpp, rlang, rtracklayer, shiny

LinkingTo Rcpp

Suggests AnnotationDbi, AnnotationHub, dplyr, DT, EnsDb.Hsapiens.v75, ggrepel, knitr, org.Hs.eg.db, rmarkdown, shinycssloaders, shinyFeedback, shinyWidgets, zoo

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Myles Lewis [aut, cre] (ORCID: <https://orcid.org/0000-0001-9365-5345>), Tom Willis [ctb]

Maintainer Myles Lewis <myles.lewis@qmul.ac.uk>

Repository CRAN

Date/Publication 2026-10-09 13:40:02 UTC

Contents

eqtl_plot . . . 2
eqtl_plotly . . . 4
genetracks . . . 4
genetracks_grob . . . 6
genetrack_ly . . . 8
gg_addgenes . . . 10
gg_genetracks . . . 11
gg_scatter . . . 13
line_plot . . . 15
link_eqtl . . . 16
link_LD . . . 17
link_recomb . . . 18
locus . . . 19
locus_ggplot . . . 21
locus_plot . . . 24
locus_plotly . . . 26
multi_layout . . . 28
overlay_plot . . . 30
overlay_plotly . . . 31
quick_peak . . . 32
sanity_check . . . 33
scatter_plot . . . 34
scatter_plotly . . . 36
set_layers . . . 37
SLE_gwas_sub . . . 38
zoom . . . 38

Index **42**

`eqtl_plot` *Locus eQTL plot*

Description

Produces a plot of eQTL data embedded in a 'locus' class object. Intended for use with `set_layers()`.

Usage

```
eqtl_plot(
  loc,
  tissue = "Whole Blood",
  eqtl_gene = loc$gene,
  scheme = "RdYlBu",
  col = NA,
  pcutoff = NULL,
  xlab = NULL,
```

```
  ylab = expression("-log"[10] ~ "P"),
  cex.axis = 0.9,
  xticks = TRUE,
  border = FALSE,
  add = FALSE,
  align = TRUE,
  legend_pos = "topright",
  ...
)
```

Arguments

loc	Object of class 'locus' to use for plot. See locus.
tissue	GTex tissue in which eQTL has been measured
eqtl_gene	Gene showing eQTL effect
scheme	Character string specifying palette for effect size showing up/downregulation eQTL using grDevices::hcl.colors. Alternatively a vector of 6 colours.
col	Outline point colour. NA for no outlines.
pcutoff	Cut-off for p value significance. Defaults to p = 5e-08. Set to NULL to disable.
xlab	x axis title.
ylab	y axis title.
cex.axis	Specifies font size for axis numbering.
xticks	Logical whether x axis numbers and axis title are plotted.
border	Logical whether a bounding box is plotted around upper and lower plots.
add	Logical whether to add points to an existing plot or generate a new plot.
align	Logical whether set par() to align the plot.
legend_pos	Character value specifying legend position. See legend().
...	Other arguments passed to plot() for the scatter plot.

Value

No return value. Produces a scatter plot using base graphics.

See Also

locus() set_layers() scatter_plot()

eqtl_plotly *Locus eQTL scatter plotly*

Description

Produces a scatter plot using plotly of embedded eQTL data acquired through the LDlink API via `link_eqtl()`.

Usage

```
eqtl_plotly(loc, gene_filter = NULL, tissue_filter = NULL, ...)
```

Arguments

`loc`	Object of class 'locus' to use for plot. See locus.
`gene_filter`	Character vector of genes to filter eQTL results.
`tissue_filter`	Character vector of tissues to filter eQTL results.
`...`	Optional arguments passed to `plot_ly()`.

Value

A `plotly` scatter plot.

See Also

`link_eqtl()` `locus_plotly()`

genetracks *Plot gene tracks*

Description

Plot gene annotation tracks from `ensembldb` data.

Usage

```
genetracks(
  locus,
  filter_gene_name = NULL,
  filter_gene_biotype = NULL,
  border = FALSE,
  cex.axis = 0.9,
  cex.lab = 1,
  cex.text = 0.7,
  gene_col = ifelse(showExons, "blue4", "skyblue"),
```

```
  exon_col = "blue4",
  exon_border = "blue4",
  showExons = TRUE,
  maxrows = NULL,
  text_pos = "top",
  italics = FALSE,
  xticks = TRUE,
  xlab = NULL,
  highlight = NULL,
  highlight_col = "red",
  prioritise = highlight,
  blanks = c("fill", "hide"),
  showRecomb = TRUE,
  align = TRUE
)
```

Arguments

locus	Object of class 'locus' generated by `locus()`.
filter_gene_name	Vector of gene names to display.
filter_gene_biotype	Vector of gene biotypes to be filtered. Use `ensembldb::listGenebiotypes()` to display possible biotypes. For example, `ensembldb::listGenebiotypes(EnsDb.Hsapiens.v75)`
border	Logical whether a bounding box is plotted.
cex.axis	Specifies font size for axis numbering.
cex.lab	Specifies font size for axis titles.
cex.text	Font size for gene text.
gene_col	Colour for gene lines.
exon_col	Fill colour for exons.
exon_border	Border line colour outlining exons (or genes if `showExons` is `FALSE`). Set to `NA` for no border.
showExons	Logical whether to show exons or simply show whole gene as a rectangle. If `showExons = FALSE` colours are specified by `exon_border` for rectangle border and `gene_col` for the fill colour.
maxrows	Specifies maximum number of rows to display in gene annotation panel.
text_pos	Character value of either 'top' or 'left' specifying placement of gene name labels.
italics	Logical whether gene text is in italics.
xticks	Logical whether x axis ticks and numbers are plotted.
xlab	Title for x axis. Defaults to chromosome `seqname` specified in `locus`.
highlight	Vector of genes to highlight.
highlight_col	Single colour or vector of colours for highlighted genes.

prioritise	Vector of genes to be placed first in the gene tracks.
blanks	Controls handling of genes with blank names: "fill" replaces blank gene symbols with ensembl gene ids. "hide" hides genes which are missing gene symbols.
showRecomb	Logical controls alignment of right margin if recombination data present.
align	Logical whether to set par() to align the plot.

Details

This function is called by locus_plot(). It can be used to plot the gene annotation tracks on their own. It uses base graphics, so graphics::layout() can be used to position adjacent plots above or below.

gene_col, exon_col and exon_border set colours for all genes, while highlight and highlight_col can optionally be used together to highlight specific genes of interest. For full control over every single gene, users can add columns gene_col, exon_col and exon_border to the TX object within the 'locus' object. Columns added to TX override their equivalent arguments.

Value

No return value.

Examples

```
if(require(EnsDb.Hsapiens.v75)) {
data(SLE_gwas_sub)
loc <- locus(SLE_gwas_sub, gene = 'UBE2L3', flank = 1e5,
             ens_db = "EnsDb.Hsapiens.v75")
genetracks(loc)

## Limit the number of tracks
genetracks(loc, maxrows = 4)

## Filter by gene biotype
genetracks(loc, filter_gene_biotype = 'protein_coding')

## Customise colours
genetracks(loc, gene_col = 'grey', exon_col = 'orange',
           exon_border = 'darkgrey')
}
```

---

genetracks_grob *Create gene tracks grob*

Description

Plot gene annotation tracks from ensembldb data using the grid package to create a grob.

Usage

```
genetracks_grob(
  locus,
  filter_gene_name = NULL,
  filter_gene_biotype = NULL,
  border = FALSE,
  cex.text = 0.7,
  gene_col = ifelse(showExons, "blue4", "skyblue"),
  exon_col = "blue4",
  exon_border = "blue4",
  showExons = TRUE,
  maxrows = NULL,
  text_pos = "top",
  italics = FALSE,
  highlight = NULL,
  highlight_col = "red",
  prioritise = highlight,
  blanks = c("fill", "hide")
)
```

Arguments

locus	Object of class 'locus' generated by locus().
filter_gene_name	Vector of gene names to display.
filter_gene_biotype	Vector of gene biotypes to be filtered. Use ensembldb::listGenebiotypes() to display possible biotypes. For example, ensembldb::listGenebiotypes(EnsDb.Hsapiens.v75)
border	Logical whether a bounding box is plotted.
cex.text	Font size for gene text.
gene_col	Colour for gene lines.
exon_col	Fill colour for exons.
exon_border	Border line colour outlining exons (or genes if showExons is FALSE). Set to NA for no border.
showExons	Logical whether to show exons or simply show whole gene as a rectangle. If showExons = FALSE colours are specified by exon_border for rectangle border and gene_col for the fill colour.
maxrows	Specifies maximum number of rows to display in gene annotation panel.
text_pos	Character value of either 'top' or 'left' specifying placement of gene name labels.
italics	Logical whether gene text is in italics.
highlight	Vector of genes to highlight.
highlight_col	Single colour or vector of colours for highlighted genes.
prioritise	Vector of genes to be placed first in the gene tracks.

blanks — Controls handling of genes with blank names: "fill" replaces blank gene symbols with ensembl gene ids. "hide" hides genes which are missing gene symbols.

Details

This function is called by gg_genetracks(). It can be used to generate a grob of the gene annotation tracks on their own.

Value

A grob object.

Examples

```
if(require(EnsDb.Hsapiens.v75)) {
data(SLE_gwas_sub)
loc <- locus(SLE_gwas_sub, gene = 'IRF5', flank = c(7e4, 2e5), LD = "r2",
             ens_db = "EnsDb.Hsapiens.v75")
g <- genetracks_grob(loc)
grid::grid.newpage()
grid::grid.draw(g)
}
```

---

genetrack_ly *Gene tracks using 'plotly'*

Description

Plot gene annotation tracks from ensembldb data using plotly.

Usage

```
genetrack_ly(
  locus,
  filter_gene_name = NULL,
  filter_gene_biotype = NULL,
  cex.text = 0.7,
  italics = FALSE,
  gene_col = ifelse(showExons, "blue4", "skyblue"),
  exon_col = "blue4",
  exon_border = "blue4",
  showExons = TRUE,
  maxrows = 8,
  width = 600,
  xlab = NULL,
  highlight = NULL,
  highlight_col = "red",
```

```
  prioritise = NULL,
  blanks = c("fill", "hide", "show"),
  height = NULL,
  plot = TRUE
)
```

Arguments

locus	Object of class 'locus' generated by locus().
filter_gene_name	Vector of gene names to display.
filter_gene_biotype	Vector of gene biotypes to be filtered. Use ensembldb::listGenebiotypes() to display possible biotypes. For example, ensembldb::listGenebiotypes(EnsDb.Hsapiens.v75)
cex.text	Font size for gene text.
italics	Logical whether gene text is in italics.
gene_col	Colour for gene lines.
exon_col	Fill colour for exons.
exon_border	Border line colour outlining exons (or genes if showExons is FALSE). Set to NA for no border.
showExons	Logical whether to show exons or simply show whole gene as a rectangle. If showExons = FALSE colours are specified by exon_border for rectangle border and gene_col for the fill colour.
maxrows	Specifies maximum number of rows to display in gene annotation panel.
width	Width of plotly plot in pixels which is purely used to prevent overlapping text for gene names.
xlab	Title for x axis. Defaults to chromosome seqname specified in locus.
highlight	Vector of genes to highlight.
highlight_col	Colour for highlighted genes.
prioritise	Vector of genes to be placed first in the gene tracks.
blanks	Controls handling of genes with blank names: "fill" replaces blank gene symbols with ensembl gene ids. "hide" completely hides genes which are missing gene symbols. "show" shows gene lines but no label (hovertext is still available).
height	Height in pixels (optional, defaults to automatic sizing).
plot	Logical whether to produce plotly object or return plot coordinates.

Details

This function can used to plot gene annotation tracks on their own.

Value

Either a 'plotly' plotting object showing gene tracks, or if plot = FALSE a list containing TX, a dataframe of coordinates for gene transcripts, and EX, a dataframe of coordinates for exons.

Examples

```
if(require(EnsDb.Hsapiens.v75)) {
data(SLE_gwas_sub)
loc <- locus(SLE_gwas_sub, gene = 'UBE2L3', flank = 1e5,
             ens_db = "EnsDb.Hsapiens.v75")
genetrack_ly(loc)
}
```

---

gg_addgenes *Add gene tracks to a ggplot2 plot*

Description

Adds gene tracks to an existing ggplot2 plot.

Usage

```
gg_addgenes(p, loc, heights = c(3, 2), ...)
```

Arguments

p	ggplot2 plot object. This can be generated by gg_scatter() and then modified.
loc	Object of class 'locus' to use for plot. See locus().
heights	Vector specifying ratio of heights of upper plot and lower gene track.
...	Additional arguments passed to gg_genetracks() to control colours of gene tracks etc.

Value

A ggplot2 plotting object.

See Also

gg_scatter() gg_genetracks()

Examples

```
if(require(EnsDb.Hsapiens.v75)) {
data(SLE_gwas_sub)
loc <- locus(SLE_gwas_sub, gene = 'IRF5', flank = c(7e4, 2e5), LD = "r2",
             ens_db = "EnsDb.Hsapiens.v75")
p <- gg_scatter(loc)
gg_addgenes(p, loc)
}
```

gg_genetracks *Plot gene tracks*

Description

Plot gene annotation tracks from ensembldb data using ggplot2 and grid.

Usage

```
gg_genetracks(
  loc,
  filter_gene_name = NULL,
  filter_gene_biotype = NULL,
  border = FALSE,
  cex.axis = 1,
  cex.lab = 1,
  cex.text = 0.7,
  gene_col = ifelse(showExons, "blue4", "skyblue"),
  exon_col = "blue4",
  exon_border = "blue4",
  showExons = TRUE,
  maxrows = NULL,
  text_pos = "top",
  italics = FALSE,
  xticks = TRUE,
  xlab = NULL,
  highlight = NULL,
  highlight_col = "red",
  prioritise = highlight,
  blanks = c("fill", "hide")
)
```

Arguments

loc	Object of class 'locus' generated by locus().
filter_gene_name	Vector of gene names to display.
filter_gene_biotype	Vector of gene biotypes to be filtered. Use ensembldb::listGenebiotypes() to display possible biotypes. For example, ensembldb::listGenebiotypes(EnsDb.Hsapiens.v75)
border	Logical whether a bounding box is plotted.
cex.axis	Specifies font size for axis numbering.
cex.lab	Specifies font size for axis titles.
cex.text	Font size for gene text.
gene_col	Colour for gene lines.

exon_col	Fill colour for exons.
exon_border	Border line colour outlining exons (or genes if showExons is FALSE). Set to NA for no border.
showExons	Logical whether to show exons or simply show whole gene as a rectangle. If showExons = FALSE colours are specified by exon_border for rectangle border and gene_col for the fill colour.
maxrows	Specifies maximum number of rows to display in gene annotation panel.
text_pos	Character value of either 'top' or 'left' specifying placement of gene name labels.
italics	Logical whether gene text is in italics.
xticks	Logical whether x axis ticks and numbers are plotted.
xlab	Title for x axis. Defaults to chromosome seqname specified in locus.
highlight	Vector of genes to highlight.
highlight_col	Single colour or vector of colours for highlighted genes.
prioritise	Vector of genes to be placed first in the gene tracks.
blanks	Controls handling of genes with blank names: "fill" replaces blank gene symbols with ensembl gene ids. "hide" hides genes which are missing gene symbols.

Details

This function is called by locus_ggplot(), and in turn it calls genetracks_grob(). It can be used to plot the gene annotation tracks on their own as a ggplot2 object.

gene_col, exon_col and exon_border set colours for all genes, while highlight and highlight_col can optionally be used together to highlight specific genes of interest. For full control over every single gene, users can add columns gene_col, exon_col and exon_border to the TX object within the 'locus' object. Columns added to TX override their equivalent arguments.

Value

A ggplot2 object.

See Also

locus_ggplot() genetracks_grob()

Examples

```
if(require(EnsDb.Hsapiens.v75)) {
data(SLE_gwas_sub)
loc <- locus(SLE_gwas_sub, gene = 'IRF5', flank = c(7e4, 2e5), LD = "r2",
             ens_db = "EnsDb.Hsapiens.v75")
gg_genetracks(loc)
}
```

gg_scatter *Locus scatter plot using ggplot2*

Description

Produces a scatter plot from a 'locus' class object (without gene tracks).

Usage

```
gg_scatter(
  loc,
  index_snp = loc$index_snp,
  pcutoff = 5e-08,
  scheme = c("grey", "dodgerblue", "red"),
  size = 2,
  cex.axis = 1,
  cex.lab = 1,
  xlab = NULL,
  ylab = NULL,
  ylim = NULL,
  ylim2 = c(0, 100),
  yzero = (loc$yvar == "logP"),
  xticks = TRUE,
  border = FALSE,
  showLD = TRUE,
 LD_scheme = c("grey", "royalblue", "cyan2", "green3", "orange", "red", "purple"),
  recomb_col = "blue",
  recomb_linewidth = 0.35,
  recomb_offset = 0,
  legend_pos = "topleft",
  labels = NULL,
  eqtl_gene = NULL,
  beta = NULL,
  shape = NULL,
  shape_values = c(21, 24, 25),
  ...
)
```

Arguments

loc	Object of class 'locus' to use for plot. See locus.
index_snp	Specifies index SNP to be shown in a different colour and symbol. Defaults to the SNP with the lowest p-value. Set to NULL to not show this.
pcutoff	Cut-off for p value significance. Defaults to p = 5e-08. Set to NULL to disable.
scheme	Vector of 3 colours if LD is not shown: 1st = normal points, 2nd = colour for significant points, 3rd = index SNP.

size	Specifies size for points.
cex.axis	Specifies font size for axis numbering.
cex.lab	Specifies font size for axis titles.
xlab	x axis title.
ylab	y axis title.
ylim	y axis limits (y1, y2).
ylim2	Secondary y axis limits for recombination line.
yzero	Logical whether to force y axis limit to include y=0.
xticks	Logical whether x axis numbers and axis title are plotted.
border	Logical whether a bounding box is plotted around the plot.
showLD	Logical whether to show LD with colours
LD_scheme	Vector of colours for plotting LD. The first colour is for SNPs which lack LD information. The next 5 colours are for r2 or D' LD results ranging from 0 to 1 in intervals of 0.2. The final colour is for the index SNP.
recomb_col	Colour for recombination rate line if recombination rate data is present. Set to NA to hide the line. See link_recomb() to add recombination rate data.
recomb_linewidth	Line width for recombination rate line.
recomb_offset	Offset from 0-1 which shifts the scatter plot up and recombination line plot down. Recommended value 0.1.
legend_pos	Position of legend. Set to NULL to hide legend.
labels	Character vector of SNP or genomic feature IDs to label. The value "index" selects the highest point or index SNP as defined when locus() is called. Set to NULL to remove all labels.
eqtl_gene	Optional column name in loc$data for colouring eQTL genes.
beta	Optional column name for beta coefficient to display upward triangles for positive beta and downward triangles for negative beta (significant SNPs only).
shape	Optional column name in loc$data for controlling shapes. beta and shape cannot both be set. This column is expected to be a factor.
shape_values	Vector of shape values which match levels of the column specified by shape. This vector is passed to ggplot2::scale_shape_manual() as the argument values. See points() for a list of shapes and the numbers they map to.
...	Optional arguments passed to geom_text_repel() to configure label drawing.

Details

If recombination rate data is included in the locus object following a call to link_recomb(), this is plotted as an additional line with a secondary y axis. In the base graphics version the line is placed under the scatter points, but this is not possible with ggplot2 as the secondary y axis data must be plotted on top of the primary scatter point data.

Value

Returns a ggplot2 plot.

See Also

locus() gg_addgenes()

Examples

```
if (require(EnsDb.Hsapiens.v75)) {
    data(SLE_gwas_sub)
    loc <- locus(SLE_gwas_sub,
        gene = "IRF5", flank = c(7e4, 2e5), LD = "r2",
        ens_db = "EnsDb.Hsapiens.v75"
    )
    gg_scatter(loc)
}
```

---

line_plot *Locus line plot*

Description

Produces a line plot from a 'locus' class object. Intended for use with set_layers().

Usage

```
line_plot(
  loc,
  pcutoff = 5e-08,
  xlab = NULL,
  ylab = expression("-log"[10] ~ "P"),
  cex.axis = 1,
  xticks = FALSE,
  border = FALSE,
  align = TRUE,
  ...
)
```

Arguments

loc	Object of class 'locus' to use for plot. See locus.
pcutoff	Cut-off for p value significance. Defaults to p = 5e-08. Set to NULL to disable.
xlab	x axis title.
ylab	y axis title.
cex.axis	Specifies font size for axis numbering.
xticks	Logical whether x axis numbers and axis title are plotted.
border	Logical whether a bounding box is plotted around upper and lower plots.
align	Logical whether set par() to align the plot.
...	Other arguments passed to plot() for the scatter plot.

Value

No return value. Produces a scatter plot using base graphics.

See Also

`locus()` `set_layers()` `scatter_plot()`

`link_eqtl` *Obtain GTEx eQTL data via LDlinkR*

Description

Adds eQTL (expression quantitative trait loci) information from GTEx (https://gtexportal.org/) to a 'locus' class object. It queries LDlink (https://ldlink.nci.nih.gov/) via the `LDlinkR` package to retrieve GTEx eQTL information on a reference SNP.

Usage

```
link_eqtl(
  loc,
  pop = "CEU",
  r2d = "r2",
  token = "",
  genome_build = tolower(loc$genome),
  ...
)
```

Arguments

`loc`	Object of class 'locus' generated by `locus()`
`pop`	A 1000 Genomes Project population, (e.g. YRI or CEU), multiple allowed, default = "CEU". Passed to `LDlinkR::LDexpress()`.
`r2d`	Either "r2" for LD r^2 or "d" for LD D', default = "r2". Passed to `LDlinkR::LDexpress()`.
`token`	Personal access token for accessing 1000 Genomes LD data via LDlink API. See `LDlinkR` package documentation.
`genome_build`	Choose between one of the three options: 'grch37' for genome build GRCh37 (hg19), 'grch38' for GRCh38 (hg38), or 'grch38_high_coverage' for GRCh38 High Coverage (hg38) 1000 Genome Project data sets.
`...`	Optional arguments passed to `LDlinkR::LDexpress()`

Details

The additional eQTL information obtained from LDlink web server can be displayed using `eqtl_plot()` which generates a scatter plot with gene tracks similar to a locus plot, or with `overlay_plot()` which tries to overlay the EQTL analysis over the original locus results (e.g. GWAS).

Value

Returns an object of class 'locus' with an extra list element 'LDexp' containing a dataframe of information obtained via `LDexpress()`.

See Also

`locus()` `eqtl_plot()` `overlay_plot()`

`link_LD` *Obtain LD at a locus from LDlink*

Description

Adds LD information to a 'locus' class object. It queries LDlink (https://ldlink.nci.nih.gov/) via the `LDlinkR` package to retrieve linkage disequilibrium (LD) information on a reference SNP.

Usage

```
link_LD(
  loc,
  pop = "CEU",
  r2d = "r2",
  token = "",
  method = c("proxy", "matrix"),
  genome_build = loc$genome,
  ...
)
```

Arguments

`loc`	Object of class 'locus' generated by `locus()`
`pop`	A 1000 Genomes Project population, (e.g. YRI or CEU), multiple allowed, default = "CEU". Passed to `LDlinkR::LDmatrix()`.
`r2d`	Either "r2" for LD r^2 or "d" for LD D', default = "r2". Passed to `LDlinkR::LDmatrix()` or `LDproxy()`.
`token`	Personal access token for accessing 1000 Genomes LD data via LDlink API. See `LDlinkR` package documentation.
`method`	Either `"proxy"` or `"matrix"`. Controls whether to use `LDproxy()` or `LDmatrix()` to obtain LD data.
`genome_build`	Choose between one of the three options: 'grch37' for genome build GRCh37 (hg19), 'grch38' for GRCh38 (hg38), or 'grch38_high_coverage' for GRCh38 High Coverage (hg38) 1000 Genome Project data sets.
`...`	Optional arguments which are passed on to `LDlinkR::LDmatrix()` or `LDlinkR::LDproxy()`

Details

The argument method controls which LDlinkR function is used to retrieve LD data. LDmatrix() is slower but usually more complete for small queries (<1000 SNPs). However, it has a limit of 1000 SNPs which can be queried. LDproxy() is faster but data on some SNPs may be absent.

Note, SNPs have to be correctly formatted as required by LDlinkR, either as rsID (works with either method) or chromosome coordinate e.g. "chr7:24966446" (works with LDproxy only). Default genome build is grch37, see LDproxy() or LDmatrix().

Value

Returns a list object of class 'locus'. LD information is added as a column ld in list element data.

See Also

locus()

---

link_recomb *Query UCSC for Recombination data*

Description

Adds recombination data to a 'locus' object by querying UCSC genome browser.

Usage

```
link_recomb(loc, genome = loc$genome, table = NULL, recomb = NULL)
```

Arguments

loc	Object of class 'locus' generated by locus()
genome	Either "hg38" or "hg19"
table	Optional character value specifying which recombination table to use.
recomb	Optional GRanges class object of recombination data.

Details

Uses the rtracklayer package to query UCSC genome browser for recombination rate data.

Possible options for table for hg19 are "hapMapRelease24YRIRecombMap", "hapMapRelease24CEURecombMap", "hapMapRelease24CombinedRecombMap" (the default). The only option for table for hg38 is "recomb1000GAvg" (the default).

If you are doing many queries, it may be much faster to download the entire recombination track data (around 30 MB for hg38) from the Recombination Rate Tracks page at UCSC genome browser. The link to the hg38 download folder is http://hgdownload.soe.ucsc.edu/gbdb/hg38/recombRate/ and for hg19 is http://hgdownload.soe.ucsc.edu/gbdb/hg19/decode/. These .bw files can be converted to useable GRanges objects using rtracklayer::import.bw() (see the vignette).

Sometimes rtracklayer generates intermittent API errors or warnings: try calling link_recomb() again. If warnings persist restart your R session. Errors are handled gracefully using try() to allow users to wrap link_recomb() in a loop without quitting halfway. Error messages are still shown. Successful API calls are cached using memoise to reduce API requests.

Value

A list object of class 'locus'. Recombination data is added as list element recomb.

---

locus *Create locus object for plotting*

Description

Creates object of class 'locus' for genomic locus plot similar to locuszoom.

Usage

```
locus(
  data = NULL,
  gene = NULL,
  xrange = NULL,
  seqname = NULL,
  flank = NULL,
  fix_window = NULL,
  ens_db,
  chrom = NULL,
  pos = NULL,
  p = NULL,
  yvar = NULL,
  labs = NULL,
  index_snp = NULL,
  LD = NULL,
  std_filter = TRUE,
  tx = TRUE
)
```

Arguments

data	Dataset (data.frame or data.table) to use for plot. We recommend that tibbles are converted to a normal data.frame. If unspecified or NULL, gene track information alone is returned.
gene	Optional character value specifying which gene to view. Either gene, or xrange plus seqname, or index_snp must be specified.
xrange	Optional vector of genomic position range for the x axis.
seqname	Optional, specifies which chromosome to plot.

flank	Single value or vector with 2 values for how much flanking region left and right of the gene to show. Defaults to 100kb.
fix_window	Optional alternative to flank, which allows users to specify a fixed genomic window centred on the specified gene. Both flank and fix_window cannot be specified simultaneously.
ens_db	Either a character string which specifies which Ensembl database package (version 86 and earlier for Homo sapiens) to query for gene and exon positions (see ensembldb Bioconductor package). Or an ensembldb object which can be obtained from the AnnotationHub database. See the vignette and the AnnotationHub Bioconductor package for how to create this object.
chrom	Determines which column in data contains chromosome information. If NULL tries to autodetect the column.
pos	Determines which column in data contains position information. If NULL tries to autodetect the column.
p	Determines which column in data contains SNP p-values. If NULL tries to autodetect the column.
yvar	Specifies column in data for plotting on the y axis as an alternative to specifying p-values. Both p and yvar cannot be specified simultaneously.
labs	Determines which column in data contains SNP rs IDs. If NULL tries to autodetect the column.
index_snp	Specifies the index SNP. If not specified, the SNP with the lowest P value is selected. Can be used to specify locus region instead of specifying gene, or seqname and xrange.
LD	Optional character value to specify which column in data contains LD information.
std_filter	Logical, whether standard filters on chromosomes 1-22, X & Y, and filtering of genes to only those whose transcript ids start with "ENS" are applied. For users with novel genome assemblies, this probably needs to be set to FALSE.
tx	Logical, whether to include transcript/exon annotations.

Details

This is an R version of locuszoom (http://locuszoom.org) for generating publication ready Manhattan plots of gene loci. It references Ensembl databases using the ensembldb Bioconductor package framework for annotating genes and exons in the locus.

Value

Returns a list object of class 'locus' ready for plotting, containing:

seqname	chromosome value
xrange	vector of genomic position range
gene	gene name
ens_db	Ensembl or AnnotationHub database
ens_version	Ensembl database version

organism	Ensembl database organism
genome	Ensembl data genome build
chrom	column name in data containing chromosome information
pos	column name in data containing position
p	column name in data containing p-value
yvar	column name in data to be plotted on y axis as alternative to p
labs	column name in data containing SNP IDs
index_snp	id of the most significant SNP
data	the subset of GWAS data to be plotted
TX	dataframe of transcript annotations
EX	GRanges object of exon annotations

If data is NULL when locus() is called then gene track information alone is returned.

See Also

locus_plot() locus_ggplot() locus_plotly()

Examples

```
## Bioconductor package EnsDb.Hsapiens.v75 is needed for these examples
if(require(EnsDb.Hsapiens.v75)) {
data(SLE_gwas_sub)
loc <- locus(SLE_gwas_sub, gene = 'UBE2L3', flank = 1e5,
             ens_db = "EnsDb.Hsapiens.v75")
summary(loc)
locus_plot(loc)
loc2 <- locus(SLE_gwas_sub, gene = 'STAT4', flank = 1e5,
              ens_db = "EnsDb.Hsapiens.v75")
locus_plot(loc2)
}
```

---

locus_ggplot *Locus plot using ggplot2*

Description

Genomic locus plot similar to locuszoom.

Usage

```
locus_ggplot(
  loc,
  heights = c(3, 2),
  filter_gene_name = NULL,
  filter_gene_biotype = NULL,
  border = FALSE,
  cex.axis = 1,
  cex.lab = 1,
  cex.text = 0.7,
  gene_col = ifelse(showExons, "blue4", "skyblue"),
  exon_col = "blue4",
  exon_border = "blue4",
  showExons = TRUE,
  maxrows = 12,
  text_pos = "top",
  italics = FALSE,
  xticks = "top",
  xlab = NULL,
  highlight = NULL,
  highlight_col = "red",
  prioritise = highlight,
  blanks = "fill",
  ...
)
```

Arguments

loc	Object of class 'locus' to use for plot. See `locus()`.
heights	Vector supplying the ratio of top to bottom plot.
filter_gene_name	Vector of gene names to display.
filter_gene_biotype	Vector of gene biotypes to be filtered. Use `ensembldb::listGenebiotypes()` to display possible biotypes. For example, `ensembldb::listGenebiotypes(EnsDb.Hsapiens.v75)`
border	Logical whether a bounding box is plotted.
cex.axis	Specifies font size for axis numbering.
cex.lab	Specifies font size for axis titles.
cex.text	Font size for gene text.
gene_col	Colour for gene lines.
exon_col	Fill colour for exons.
exon_border	Border line colour outlining exons (or genes if `showExons` is `FALSE`). Set to `NA` for no border.
showExons	Logical whether to show exons or simply show whole gene as a rectangle. If `showExons = FALSE` colours are specified by `exon_border` for rectangle border and `gene_col` for the fill colour.

maxrows	Specifies maximum number of rows to display in gene annotation panel.
text_pos	Character value of either 'top' or 'left' specifying placement of gene name labels.
italics	Logical whether gene text is in italics.
xticks	Logical whether x axis ticks and numbers are plotted.
xlab	Title for x axis. Defaults to chromosome seqname specified in locus.
highlight	Vector of genes to highlight.
highlight_col	Single colour or vector of colours for highlighted genes.
prioritise	Vector of genes to be placed first in the gene tracks.
blanks	Controls handling of genes with blank names: "fill" replaces blank gene symbols with ensembl gene ids. "hide" hides genes which are missing gene symbols.
...	Additional arguments passed to gg_scatter() to control the scatter plot, e.g. pcutoff, scheme, recomb_offset etc.

Details

Arguments to control plotting of the gene tracks are passed onto gg_genetracks() and for the scatter plot are passed via ... to gg_scatter(). See the documentation for each of these functions for details.

Value

Returns a ggplot2 plot containing a scatter plot with genetracks underneath.

See Also

gg_scatter() gg_genetracks()

Examples

```
if(require(EnsDb.Hsapiens.v75)) {
data(SLE_gwas_sub)
loc <- locus(SLE_gwas_sub, gene = 'IRF5', flank = c(7e4, 2e5), LD = "r2",
             ens_db = "EnsDb.Hsapiens.v75")
locus_ggplot(loc)
}
```

locus_plot *Locus plot*

Description

Genomic locus plot similar to locuszoom.

Usage

```
locus_plot(
  loc,
  filter_gene_name = NULL,
  filter_gene_biotype = NULL,
  xlab = NULL,
  cex = 1,
  cex.axis = 0.9,
  cex.lab = 1,
  cex.text = 0.7,
  use_layout = TRUE,
  heights = c(3, 2),
  showExons = TRUE,
  maxrows = 7,
  xticks = "bottom",
  border = FALSE,
  gene_col = ifelse(showExons, "blue4", "skyblue"),
  exon_col = "blue4",
  exon_border = "blue4",
  text_pos = "top",
  italics = FALSE,
  highlight = NULL,
  highlight_col = "red",
  prioritise = highlight,
  blanks = "fill",
  recomb_col = "blue",
  ...
)
```

Arguments

loc	Object of class 'locus' to use for plot. See locus().
filter_gene_name	Vector of gene names to display.
filter_gene_biotype	Vector of gene biotypes to be filtered. Use ensembldb::listGenebiotypes() to display possible biotypes. For example, ensembldb::listGenebiotypes(EnsDb.Hsapiens.v75)
xlab	x axis title.

cex	Specifies size for points.
cex.axis	Specifies font size for axis numbering.
cex.lab	Specifies font size for axis titles.
cex.text	Font size for gene text.
use_layout	Logical whether graphics::layout is called. Default TRUE is for a standard single plot. Set to FALSE if a more complex layout with multiple plots is required e.g. using multi_layout().
heights	Ratio of top to bottom plot. See graphics::layout.
showExons	Logical whether to show exons or simply show whole gene as a rectangle
maxrows	Specifies maximum number of rows to display in gene annotation panel.
xticks	Character value of either 'top' or 'bottom' specifying whether x axis ticks and numbers are plotted on top or bottom plot window.
border	Logical whether a bounding box is plotted around upper and lower plots.
gene_col	Colour for gene lines.
exon_col	Fill colour for exons.
exon_border	Border line colour outlining exons (or genes if showExons is FALSE). Set to NA for no border.
text_pos	Character value of either 'top' or 'left' specifying placement of gene name labels.
italics	Logical whether gene text is in italics.
highlight	Vector of genes to highlight.
highlight_col	Single colour or vector of colours for highlighted genes.
prioritise	Vector of genes to be placed first in the gene tracks.
blanks	Controls handling of genes with blank names: "fill" replaces blank gene symbols with ensembl gene ids. "hide" hides genes which are missing gene symbols.
recomb_col	Colour for recombination rate line if recombination rate data is present. Set to NA to hide the line. See link_recomb() to add recombination rate data.
...	Other arguments passed to scatter_plot() e.g. index_snp, pcutoff, scheme, recomb_offset, etc, and arguments for plot() e.g. ylab, main, etc to control the scatter plot.

Details

This is an R version of locuszoom for generating publication ready Manhattan plots of gene loci. It references Ensembl databases for annotating genes and exons. Use locus() first to generate an object of class 'locus' for plotting. LDlink web server can be queried using function link_LD() to retrieve linkage disequilibrium (LD) information on the index SNP.

Arguments to control plotting of the gene tracks are passed onto genetracks() and for the scatter plot are passed via ... to scatter_plot(). See the documentation for each of these functions for details.

Value

No return value.

See Also

locus() scatter_plot() genetracks()

Examples

```
if(require(EnsDb.Hsapiens.v75)) {
data(SLE_gwas_sub)
loc <- locus(SLE_gwas_sub, gene = 'UBE2L3', flank = 1e5,
             ens_db = "EnsDb.Hsapiens.v75")
locus_plot(loc)

## Use embedded LD information in column `r2`
loc2 <- locus(SLE_gwas_sub, gene = 'IRF5', flank = c(7e4, 2e5), LD = "r2",
              ens_db = "EnsDb.Hsapiens.v75")
## Add label for index SNP
locus_plot(loc2, labels = "index")
}
```

---

locus_plotly *Locus plotly*

Description

Genomic locus plot similar to locuszoom, using plotly.

Usage

```
locus_plotly(
  loc,
  loc2 = NULL,
  heights = c(0.6, 0.4),
  filter_gene_name = NULL,
  filter_gene_biotype = NULL,
  cex.text = 0.7,
  italics = FALSE,
  gene_col = ifelse(showExons, "blue4", "skyblue"),
  exon_col = "blue4",
  exon_border = "blue4",
  showExons = TRUE,
  maxrows = 8,
  width = 600,
  xlab = NULL,
  ylab = NULL,
  highlight = NULL,
```

```
  highlight_col = "red",
  prioritise = NULL,
  blanks = "show",
  beta = NULL,
  gene_filter = NULL,
  tissue_filter = NULL,
  eqtl_scheme = NULL,
  ...
)
```

Arguments

loc	Object of class 'locus' to use for plot. See `locus()`.
loc2	Optional 2nd 'locus' object to be layered underneath the 1st scatter plot.
heights	Vector controlling relative height of each panel on 0-1 scale. Alternatively a vector of length 2 of height in pixels passed to `scatter_plotly()` and `genetrack_ly()`.
filter_gene_name	Vector of gene names to display.
filter_gene_biotype	Vector of gene biotypes to be filtered. Use `ensembldb::listGenebiotypes()` to display possible biotypes. For example, `ensembldb::listGenebiotypes(EnsDb.Hsapiens.v75)`
cex.text	Font size for gene text.
italics	Logical whether gene text is in italics.
gene_col	Colour for gene lines.
exon_col	Fill colour for exons.
exon_border	Border line colour outlining exons (or genes if `showExons` is `FALSE`). Set to `NA` for no border.
showExons	Logical whether to show exons or simply show whole gene as a rectangle. If `showExons = FALSE` colours are specified by `exon_border` for rectangle border and `gene_col` for the fill colour.
maxrows	Specifies maximum number of rows to display in gene annotation panel.
width	Width of plotly plot in pixels which is purely used to prevent overlapping text for gene names.
xlab	Title for x axis. Defaults to chromosome `seqname` specified in `locus`.
ylab	Title for y axis, or a vector of 2 titles for each y axis if `loc2` is provided.
highlight	Vector of genes to highlight.
highlight_col	Colour for highlighted genes.
prioritise	Vector of genes to be placed first in the gene tracks.
blanks	Controls handling of genes with blank names: `"fill"` replaces blank gene symbols with ensembl gene ids. `"hide"` completely hides genes which are missing gene symbols. `"show"` shows gene lines but no label (hovertext is still available).

beta	Optional column name for beta coefficient to display upward triangles for positive beta and downward triangles for negative beta (significant SNPs only). If loc2 is supplied, then a vector can be used to specify different beta columns in loc and loc2; use NA to indicate no beta.
gene_filter	Character vector of genes to filter LDlink eQTL results. See overlay_plotly().
tissue_filter	Character vector of tissues to filter LDlink eQTL results. See overlay_plotly().
eqtl_scheme	Vector of colours for eQTL genes. See overlay_plotly().
...	Optional arguments passed to scatter_plotly() to control the scatter plot.

Details

This is an R/plotly version of locuszoom for exploring regional Manhattan plots of gene loci. Use locus() first to generate an object of class 'locus' for plotting. This references a selected Ensembl database for annotating genes and exons. Hover over the points or gene tracks to reveal more information.

Value

A 'plotly' plotting object showing a scatter plot above gene tracks.

See Also

locus() genetrack_ly() scatter_plotly()

Examples

```
if(require(EnsDb.Hsapiens.v75)) {
data(SLE_gwas_sub)
loc <- locus(SLE_gwas_sub, gene = "IRF5", flank = c(7e4, 2e5), LD = "r2",
             ens_db = "EnsDb.Hsapiens.v75")
locus_plotly(loc)
}
```

---

multi_layout *Layout multiple locus plots*

Description

Produces pages with multiple locus plots on.

Usage

```
multi_layout(
  plots,
  nrow = 1,
  ncol = 1,
  heights = c(3, 2),
```

```
  legend_pos = "topleft",
  ...
)
```

Arguments

plots	Either an 'expression' to be evaluated which is a series of calls to locus_plot() or similar plotting functions, or a list of 'locus' class objects which are plotted in sequence.
nrow	Number of rows of plots
ncol	Number of columns of plots
heights	Vector of length 2 specifying height for plot and gene tracks
legend_pos	A keyword either "topleft" or "topright" or NULL to hide the legend. Not invoked if plots is an expression. The legend is only shown on one plot on each page.
...	Optional arguments passed to locus_plot() if plots contains a list

Value

No return value.

See Also

locus_plot()

Examples

```
if(require(EnsDb.Hsapiens.v75)) {

data(SLE_gwas_sub)
genes <- c("STAT4", "UBE2L3", "IRF5")
loclist <- lapply(genes, locus,
                  data = SLE_gwas_sub,
                  ens_db = "EnsDb.Hsapiens.v75",
                  LD = "r2")
## produce 3 locus plots, one on each page
multi_layout(loclist)

## place 3 locus plots in a row on a single page
multi_layout(loclist, ncol = 3)

## full control
loc <- locus(SLE_gwas_sub, gene = 'STAT4', flank = 1e5, LD = "r2",
             ens_db = "EnsDb.Hsapiens.v75")
loc2 <- locus(SLE_gwas_sub, gene = 'IRF5', flank = c(7e4, 2e5), LD = "r2",
              ens_db = "EnsDb.Hsapiens.v75")
loc3 <- locus(SLE_gwas_sub, gene = 'UBE2L3', LD = "r2",
              ens_db = "EnsDb.Hsapiens.v75")
multi_layout(ncol = 3,
             plots = {
               locus_plot(loc, use_layout = FALSE, legend_pos = 'topleft')
```

```
  locus_plot(loc2, use_layout = FALSE, legend_pos = NULL)
  locus_plot(loc3, use_layout = FALSE, legend_pos = NULL)
})

}
```

`overlay_plot` *Plot overlaying eQTL and GWAS data*

Description

Experimental plotting function for overlaying eQTL data from GTEx on top of GWAS results. y axis shows the -log10 p-value for the GWAS result. Significant eQTL for the specified gene are overlaid using colours and symbols.

Usage

```
overlay_plot(
  loc,
  base_col = "black",
  alpha = 0.5,
  scheme = "RdYlBu",
  tissue = "Whole Blood",
  eqtl_gene = loc$gene,
  legend_pos = "topright",
  ...
)
```

Arguments

`loc`	Object of class 'locus' to use for plot. See `locus()`.
`base_col`	Colour of points for SNPs which do not have eQTLs.
`alpha`	Alpha opacity for non-eQTL points
`scheme`	Character string specifying palette for effect size showing up/downregulation eQTL using grDevices::hcl.colors. Alternatively a vector of 6 colours.
`tissue`	GTex tissue in which eQTL has been measured
`eqtl_gene`	Gene showing eQTL effect
`legend_pos`	Character value specifying legend position. See `legend()`.
`...`	Other arguments passed to `locus_plot()` for the locus plot.

Value

No return value. Produces a plot using base graphics.

overlay_plotly *Plot overlaying eQTL and GWAS data using plotly*

Description

Produces a scatter plot using plotly of embedded eQTL data acquired through the LDlink API via link_eqtl() overlaid on GWAS data. As each SNP may have eQTLs with multiple genes in multiple tissues, the method used is to select the gene/tissue eQTL with the lowest p-value. SNPs are matched by rsID.

Usage

```
overlay_plotly(
  loc,
  gene_filter = NULL,
  tissue_filter = NULL,
  pcutoff = 5e-08,
  eqtl_scheme = NULL,
  xlab = NULL,
  ylab = NULL,
  marker_outline = "grey",
  marker_size = 7,
  recomb_col = "blue",
  showlegend = TRUE,
  show_annot = TRUE,
  height = NULL,
  webGL = TRUE
)
```

Arguments

loc	Object of class 'locus' to use for plot. See locus.
gene_filter	Character vector of genes to filter eQTL results.
tissue_filter	Character vector of tissues to filter eQTL results.
pcutoff	Cut-off for p value significance. Defaults to p = 5e-08. Set to NULL to disable.
eqtl_scheme	Vector of colours for eQTL genes, which can be named.
xlab	x axis title.
ylab	y axis title.
marker_outline	Specifies colour for outlining points.
marker_size	Value for size of markers in plotly units.
recomb_col	Colour for recombination rate line if recombination rate data is present. Set to NA to hide the line. See link_recomb() to add recombination rate data.
showlegend	Logical whether to show a legend for the scatter points.

show_annot	Logical whether to show an annotation of how many eQTL SNPs were retrieved from LDlink and how many are shown.
height	Height in pixels (optional, defaults to automatic sizing).
webGL	Logical whether to use webGL or SVG for scatter plot.

Value

A plotly scatter plot.

See Also

link_eqtl() locus_plotly()

---

quick_peak *Fast peak finder in GWAS data*

Description

Simple but fast function for finding peaks in genome-wide association study (GWAS) data based on setting a minimum distance between peaks.

Usage

```
quick_peak(
  data,
  npeaks = NA,
  p_cutoff = 5e-08,
  span = 1e+06,
  min_points = 2,
  chrom = NULL,
  pos = NULL,
  p = NULL
)
```

Arguments

data	GWAS dataset (data.frame or data.table)
npeaks	Number of peaks to find. If set to NA, algorithm finds all distinct peaks separated from one another by region size specified by span.
p_cutoff	Specifies cut-off for p-value significance above which p-values are ignored.
span	Minimum genomic distance between peaks (default 1 Mb)
min_points	Minimum number of p-value significant points which must lie within the span of a peak. This removes peaks with single or only a few low p-value SNPs. To disable set min_points to 1 or less.
chrom	Determines which column in data contains chromosome information. If NULL tries to autodetect the column.

pos	Determines which column in data contains position information. If NULL tries to autodetect the column.
p	Determines which column in data contains SNP p-values. If NULL tries to autodetect the column.

Details

This function is designed for speed. SNP p-values are filtered to only those which are significant as specified by p_cutoff. Each peak is identified as the SNP with the lowest p-value and then SNPs in proximity to each peak within the distance specified by span are removed. Regions such as the HLA whose peaks may well be broader than span may produce multiple entries.

Value

Vector of row indices

---

sanity_check *Sanity checks on GWAS dataframes*

Description

Performs sanity checks on GWAS dataframes to check similar chromosome content, overlap in SNP ids and alignment of SNPs by chromosome/position.

Usage

```
sanity_check(data, data2, chrom = NULL, pos = NULL, p = NULL, labs = NULL)
```

Arguments

data	Dataframe of GWAS results with columns for chromosome, position, p value and SNP rs IDs. Data.tables are coerced to dataframe.
data2	Dataframe of 2nd GWAS results for comparison.
chrom	Vector determines which column in data and data2 contains chromosome information. If NULL or NA tries to autodetect the columns. The 1st element refers to data and 2nd element refers to data2.
pos	Determines which column in data and data2 contains position information. See chrom.
p	Determines which column in data and data2 contains SNP p-values. See chrom.
labs	Determines which column in data and data2 contains SNP rs IDs. See chrom.

Value

No return value. Prints diagnostic messages.

scatter_plot *Locus scatter plot*

Description

Produces a base graphics scatter plot from a 'locus' class object. This function is called by `locus_plot()` to generate the scatter plot portion. Can be used manually with `set_layers()`.

Usage

```
scatter_plot(
  loc,
  index_snp = loc$index_snp,
  pcutoff = 5e-08,
  scheme = c("grey", "dodgerblue", "red"),
  cex = 1,
  cex.axis = 0.9,
  cex.lab = 1,
  xlab = NULL,
  ylab = NULL,
  ylim = NULL,
  ylim2 = c(0, 100),
  yzero = (loc$yvar == "logP"),
  xticks = TRUE,
  border = FALSE,
  showLD = TRUE,
 LD_scheme = c("grey", "royalblue", "cyan2", "green3", "orange", "red", "purple"),
  recomb_col = "blue",
  recomb_offset = 0,
  legend_pos = "topleft",
  labels = NULL,
  label_x = 4,
  label_y = 4,
  eqtl_gene = NULL,
  beta = NULL,
  add = FALSE,
  align = TRUE,
  ...
)
```

Arguments

loc	Object of class 'locus' to use for plot. See locus.
index_snp	Specifies index SNP or a vector of SNPs to be shown in a different colour and symbol. Defaults to the SNP with the lowest p-value. Set to NULL to not show this.

pcutoff	Cut-off for p value significance. Defaults to p = 5e-08. Set to NULL to disable.
scheme	Vector of 3 colours if LD is not shown: 1st = normal points, 2nd = colour for significant points, 3rd = index SNP(s).
cex	Specifies size for points.
cex.axis	Specifies font size for axis numbering.
cex.lab	Specifies font size for axis titles.
xlab	x axis title.
ylab	y axis title.
ylim	y axis limits (y1, y2).
ylim2	Secondary y axis limits for recombination line, if present.
yzero	Logical whether to force y axis limit to include y=0.
xticks	Logical whether x axis numbers and axis title are plotted.
border	Logical whether a bounding box is plotted around upper and lower plots.
showLD	Logical whether to show LD with colours
LD_scheme	Vector of colours for plotting LD. The first colour is for SNPs which lack LD information. The next 5 colours are for r2 or D' LD results ranging from 0 to 1 in intervals of 0.2. The final colour is for the index SNP.
recomb_col	Colour for recombination rate line if recombination rate data is present. Set to NA to hide the line. See link_recomb() to add recombination rate data.
recomb_offset	Offset from 0-1 which shifts the scatter plot up and recombination line plot down. Recommended value 0.1.
legend_pos	Position of legend. See legend(). Set to NULL to hide legend.
labels	Character vector of SNP or genomic feature IDs to label. The value "index" selects the highest point or index SNP as defined when locus() is called. Set to NULL to remove all labels.
label_x	Value or vector for position of label as percentage of x axis scale.
label_y	Value or vector for position of label as percentage of y axis scale.
eqtl_gene	Column name in loc$data for colouring eQTL genes.
beta	Optional column name for beta coefficient to display upward triangles for positive beta and downward triangles for negative beta (significant SNPs only).
add	Logical whether to add points to an existing plot or generate a new plot.
align	Logical whether to set par() to align the plot.
...	Other arguments passed to plot() to control the scatter plot e.g. main, ylim etc.

Details

Advanced users familiar with base graphics can customise every single point on the scatter plot, by adding columns named bg, col, pch or cex directly to the dataframe stored in $data element of the 'locus' object. Setting these will overrule any default settings. These columns refer to their respective base graphics arguments, see graphics::points().

Value

No return value. Produces a scatter plot using base graphics.

See Also

locus() set_layers()

---

scatter_plotly *Locus scatter plotly*

Description

Produces a scatter plot from a 'locus' class object using plotly.

Usage

```
scatter_plotly(
  loc,
  index_snp = loc$index_snp,
  pcutoff = 5e-08,
  scheme = c("grey", "dodgerblue", "red"),
  xlab = NULL,
  ylab = NULL,
  yzero = (loc$yvar == "logP"),
  showLD = TRUE,
 LD_scheme = c("grey", "royalblue", "cyan2", "green3", "orange", "red", "purple"),
  marker_outline = "black",
  marker_size = 7,
  recomb_col = "blue",
  eqtl_gene = NULL,
  beta = NULL,
  add_hover = NULL,
  showlegend = TRUE,
  height = NULL,
  webGL = TRUE
)
```

Arguments

loc	Object of class 'locus' to use for plot. See locus.
index_snp	Specifies index SNP or a vector of SNPs to be shown in a different colour and symbol. Defaults to the SNP with the lowest p-value. Set to NULL to not show this.
pcutoff	Cut-off for p value significance. Defaults to p = 5e-08. Set to NULL to disable.
scheme	Vector of 3 colours if LD is not shown: 1st = normal points, 2nd = colour for significant points, 3rd = index SNP(s).

xlab	x axis title.
ylab	y axis title.
yzero	Logical whether to force y axis limit to include y=0.
showLD	Logical whether to show LD with colours
LD_scheme	Vector of colours for plotting LD. The first colour is for SNPs which lack LD information. The next 5 colours are for r^2 or D' LD results ranging from 0 to 1 in intervals of 0.2. The final colour is for the index SNP.
marker_outline	Specifies colour for outlining points.
marker_size	Value for size of markers in plotly units.
recomb_col	Colour for recombination rate line if recombination rate data is present. Set to NA to hide the line. See link_recomb() to add recombination rate data.
eqtl_gene	Column name in loc$data for eQTL genes.
beta	Optional column name for beta coefficient to display upward triangles for positive beta and downward triangles for negative beta (significant SNPs only).
add_hover	Optional vector of column names in loc$data to add to the plotly hover text for scatter points.
showlegend	Logical whether to show a legend for the scatter points.
height	Height in pixels (optional, defaults to automatic sizing).
webGL	Logical whether to use webGL or SVG for scatter plot.

Value

A plotly scatter plot.

See Also

locus() locus_plotly()

---

set_layers *Set up a column of multiple plots*

Description

Uses graphics::layout() to set up multiple locus plots aligned in a column.

Usage

```
set_layers(n = 1, heights = c(rep(3, n), 2), rev = FALSE)
```

Arguments

n	Number of plots (not including gene tracks on bottom)
heights	Vector of length nrow + 1 specifying height for plots with a gene track on the bottom
rev	Logical whether to reverse plotting order and plot from bottom to top

Value

Sets graphics::layout() to enable multiple plots aligned in a column. The gene track is assumed to be positioned on the bottom. Returns par() invisibly so that layout can be reset to default at the end of plotting.

See Also

graphics::layout()

SLE_gwas_sub *SLE GWAS data subset*

Description

Dataset of SNPs at 3 gene loci (UBE2L3, STAT4, IRF5) from GWAS on SLE (Bentham et al, 2015, Nature Genetics 47(12):1457-64, PMID: 26502338).

Usage

```
data(SLE_gwas_sub)
```

Format

Data frame with 1990 rows and 11 variables

Source

https://www.ebi.ac.uk/gwas/studies/GCST003156

---

zoom *Zoom browser to explore GWAS/eQTL results*

Description

Interactive genome browser to explore GWAS/eQTL results using a shiny interface.

Usage

```
zoom(
  data,
  ens_db,
  chrom = NULL,
  pos = NULL,
  p = NULL,
  labs = NULL,
  data2 = NULL,
  traits = NULL,
  scheme = c("royalblue", "skyblue", "red", "#00CD00"),
  scheme2 = c("#33a02c", "#b2df8a", "purple", "red"),
  pcutoff = 5e-08,
  eqtl_gene = NULL,
  beta = NULL,
 eqtl_scheme = c("#FF0000", "#00FFFF", "#FF9000", "#0080FF", "#FFFF00", "#0000FF",
    "#80DD00", "#8000FF", "#009900", "#FF00FF"),
  add_hover = NULL,
  mh_points = 1e+05,
  recomb = NULL,
  ld_token = Sys.getenv("LDLINK_TOKEN"),
  ld_pop = "EUR",
  seq_filter = c(1:22, "X", "Y"),
  AnnotationDb = "org.Hs.eg.db",
  align = TRUE
)
```

Arguments

data	Dataframe of GWAS results with columns for chromosome, position, p value and SNP rs IDs. Data.tables are coerced to dataframe.
ens_db	Either a character string which specifies which Ensembl database package (version 86 and earlier for Homo sapiens) to query for gene and exon positions (see ensembldb Bioconductor package). Or an ensembldb object which can be obtained from the AnnotationHub database. See the vignette and the AnnotationHub Bioconductor package for how to create this object.
chrom	Determines which column in data contains chromosome information. If NULL or NA tries to autodetect the column. If data2 is provided, this is a vector where 1st element refers to data and 2nd element refers to data2.
pos	Determines which column in data (and optionally data2) contains position information. See chrom.
p	Determines which column in data (and optionally data2) contains SNP p-values. See chrom.
labs	Determines which column in data (and optionally data2) contains SNP rs IDs. See chrom.
data2	Optional 2nd dataframe of GWAS results for comparison.

traits	Vector of trait names for identifying data and data2 datasets.
scheme	Vector of 4 colours for main Manhattan plot: 1st, 2nd colours for alternating chromosomes, 3rd colour for significant points, 4th colour for highlighting chromosomes.
scheme2	Vector of colours for 2nd Manhattan plot.
pcutoff	Cut-off for p value significance. Defaults to p = 5e-08. Set to NULL to disable.
eqtl_gene	Determines which column in data contains eQTL genes. Currently data2 cannot be used to show eQTL colour data.
beta	Optional column name for beta coefficient to display upward triangles for positive beta and downward triangles for negative beta (significant SNPs only). If data2 is supplied, then a vector can be used to specify different beta columns in data and data2; use NA to indicate no beta.
eqtl_scheme	Colour scheme for eQTL genes.
add_hover	Optional vector of column names in data to add to the plotly hover text for scatter points. Not available for data2.
mh_points	Number of points to display in manhattan plot. Default is 1e5.
recomb	Optional GRanges class object of recombination data.
ld_token	Personal access token for the LDlink API, available from https://ldlink.nih.gov/?tab=apiaccess. See LDlinkR package documentation and link_LD(). When empty the LD controls are hidden. LD information can only be requested if eqtl_gene is left as NULL. LD is fetched on demand, not automatically, by pressing the "Get LD" button.
ld_pop	1000 Genomes population used for LD. Defaults to "EUR". See LDlinkR::LDproxy() for the available codes.
seq_filter	Vector of acceptable chromosomes. Used to restrict queries to standard chromosome assembly.
AnnotationDb	An AnnotationDb gene annotation database, specified either as a character string or as an AnnotationDb class object, used to obtain expanded gene names. The ensembl database specified in ens_db is queried first. Set to NULL to disable this feature.
align	Logical, whether to align the chromosomes in the main Manhattan plots and chromosome subplots when comparing 2 GWAS.

Details

This launches a shiny app to explore the GWAS/eQTL results through visualising the Manhattan plot and exploring regional Manhattan plots of gene loci through selecting points or searching SNPs/genes.

Linkage disequilibrium data can be pulled from LDlink API within the app using the 'Get LD' button in the Settings dropdown. These buttons only appear once a user provides a LDlink API token via the ld_token argument. Each API request takes around 5-10 secs. 'Get LD' pins the current index SNP as the reference variant and colours points by r^2 with it. The reference stays pinned while you pan and zoom, so the colouring keeps its meaning and repeat queries are served from the memoise cache rather than the API.

Value

No return value. Opens an interactive shiny window.

Index

∗ **datasets**
SLE_gwas_sub, 38

ensembldb::listGenebiotypes(), *5*, *7*, *9*, *11*, *22*, *24*, *27*
eqtl_plot, 2
eqtl_plot(), *16, 17*
eqtl_plotly, 4

genetrack_ly, 8
genetrack_ly(), *28*
genetracks, 4
genetracks(), *25, 26*
genetracks_grob, 6
genetracks_grob(), *12*
gg_addgenes, 10
gg_addgenes(), *15*
gg_genetracks, 11
gg_genetracks(), *8*, *10*, *23*
gg_scatter, 13
gg_scatter(), *10*, *23*
graphics::layout, *25*
graphics::layout(), *6*, *37, 38*
graphics::points(), *35*
grDevices::hcl.colors, *3*, *30*

legend(), *3*, *30*, *35*
line_plot, 15
link_eqtl, 16
link_eqtl(), *4*, *31, 32*
link_LD, 17
link_LD(), *25*, *40*
link_recomb, 18
link_recomb(), *14*, *25*, *31*, *35*, *37*
locus, *3, 4*, *13*, *15*, 19, *31*, *34*, *36*
locus(), *3*, *5*, *7*, *9–11*, *14–18*, *22*, *24–28*, *30*, *35–37*
locus_ggplot, 21
locus_ggplot(), *12*, *21*
locus_plot, 24
locus_plot(), *6*, *21*, *29, 30*, *34*
locus_plotly, 26
locus_plotly(), *4*, *21*, *32*, *37*

multi_layout, 28
multi_layout(), *25*

overlay_plot, 30
overlay_plot(), *16, 17*
overlay_plotly, 31
overlay_plotly(), *28*

par(), *3*, *6*, *15*, *35*
plot(), *3*, *15*, *25*, *35*
points(), *14*

quick_peak, 32

sanity_check, 33
scatter_plot, 34
scatter_plot(), *3*, *16*, *25, 26*
scatter_plotly, 36
scatter_plotly(), *28*
set_layers, 37
set_layers(), *2, 3*, *15, 16*, *34*, *36*
SLE_gwas_sub, 38

zoom, 38