

Package ‘locuszoomr’

February 17, 2024

Title Gene Locus Plot with Gene Annotations

Version 0.2.1

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.

Language en-gb

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 3.5)

biocViews

Imports AnnotationFilter, BiocGenerics, cowplot, dplyr, ensemblDb, GenomeInfoDb, GenomicRanges, gggrid, ggplot2, ggrepel, graphics, grDevices, grid, IRanges, LDlinkR, memoise, plotly, rlang, rtracklayer, zoo

RoxygenNote 7.3.1

Suggests AnnotationHub, EnsDb.Hsapiens.v75, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2024-02-17 14:30:02 UTC

R topics documented:

eqtl_plot	2
genetracks	3
genetracks_grob	5
genetrack_ly	7
gg_addgenes	8
gg_genetracks	9
gg_scatter	10
line_plot	12
link_eqtl	13
link_LD	14
link_recomb	14
locus	15
locus_ggplot	17
locus_plot	19
locus_plotly	21
multi_layout	22
overlay_plot	24
quick_peak	25
scatter_plot	26
scatter_plotly	27
set_layers	29
SLE_gwas_sub	29
Index	30

eqtl_plot	<i>Locus eQTL plot</i>
-----------	------------------------

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
eqlt_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\(\)](#)

genetracks

Plot gene tracks

Description

Plot gene annotation tracks from ensembl db 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",
  xticks = TRUE,
  xlab = NULL,
  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 ensembladb::listGenebiotypes() to display possible biotypes. For example, <code>ensembladb::listGenebiotypes(EnsDb.Hsapiens.v75)</code>
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.
xticks	Logical whether x axis ticks and numbers are plotted.

xlab	Title for x axis. Defaults to chromosome seqname specified in locus.
showRecomb	Logical controls alignment of right margin if recombination data present.
align	Logical whether to set <code>par()</code> 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 `layout()` can be used to position adjacent plots above or below.

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"
  )

```

Arguments

locus	Object of class 'locus' generated by <code>locus()</code> .
filter_gene_name	Vector of gene names to display.
filter_gene_biotype	Vector of gene biotypes to be filtered. Use <code>ensemblDb::listGenebiotypes()</code> to display possible biotypes. For example, <code>ensemblDb::listGenebiotypes(EnsDb.Hsapiens.v75)</code>
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 <code>showExons</code> is FALSE). Set to NA for no border.
showExons	Logical whether to show exons or simply show whole gene as a rectangle. If <code>showExons = FALSE</code> colours are specified by <code>exon_border</code> for rectangle border and <code>gene_col</code> 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.

Details

This function is called by `locus_ggplot()`. It can be used to plot 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)
}

```

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,  
  gene_col = "blue4",  
  exon_col = "blue4",  
  exon_border = "blue4",  
  maxrows = 8,  
  xlab = NULL  
)
```

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, <code>ensemblDb::listGenebiotypes(EnsDb.Hsapiens.v75)</code>
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. Set to NA for no border.
maxrows	Specifies maximum number of rows to display in gene annotation panel.
xlab	Title for x axis. Defaults to chromosome seqname specified in locus.

Details

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

Value

A 'plotly' plotting object showing gene tracks.

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.

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",
  xticks = TRUE,
  xlab = NULL
)
```

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, <code>ensemblDb::listGenebiotypes(EnsDb.Hsapiens.v75)</code>
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.
xticks	Logical whether x axis ticks and numbers are plotted.
xlab	Title for x axis. Defaults to chromosome seqname specified in locus.

Details

This function is called by `locus_ggplot()`. It can be used to plot the gene annotation tracks on their own.

Value

A ggplot2 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")
  grid::grid.newpage()
  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("royalblue", "red", "purple"),
  size = 2,
  cex.axis = 1,
  cex.lab = 1,
  xlab = NULL,
  ylab = NULL,
```

```

  yzero = (loc$yvar == "logP"),
  xticks = TRUE,
  border = FALSE,
  showLD = TRUE,
  LD_scheme = c("grey", "royalblue", "cyan2", "green3", "orange", "red", "purple"),
  recomb_col = "blue",
  legend_pos = "topleft",
  labels = NULL,
  ...
)

```

Arguments

<code>loc</code>	Object of class 'locus' to use for plot. See locus .
<code>index_snp</code>	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.
<code>pcutoff</code>	Cut-off for p value significance. Defaults to $p = 5e-08$. Set to NULL to disable.
<code>scheme</code>	Vector of 3 colours if LD is not shown: 1st = normal points, 2nd = colour for significant points, 3rd = index SNP.
<code>size</code>	Specifies size for points.
<code>cex.axis</code>	Specifies font size for axis numbering.
<code>cex.lab</code>	Specifies font size for axis titles.
<code>xlab</code>	x axis title.
<code>ylab</code>	y axis title.
<code>yzero</code>	Logical whether to force y axis limit to include $y=0$.
<code>xticks</code>	Logical whether x axis numbers and axis title are plotted.
<code>border</code>	Logical whether a bounding box is plotted around the plot.
<code>showLD</code>	Logical whether to show LD with colours
<code>LD_scheme</code>	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.
<code>recomb_col</code>	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.
<code>legend_pos</code>	Position of legend. Set to NULL to hide legend.
<code>labels</code>	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.
<code>...</code>	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 = "", ...)
```

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.
...	Optional arguments such as genome_build which are passed on 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	<i>Obtain LD at a locus from LDlink</i>
---------	---

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 = "", ...)
```

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().
token	Personal access token for accessing 1000 Genomes LD data via LDlink API. See LDlinkR package documentation.
...	Optional arguments such as genome_build which are passed on to LDlinkR::LDmatrix()

Details

Results are cached using memoise to reduce API requests.

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	<i>Query UCSC for Recombination data</i>
-------------	--

Description

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

Usage

```
link_recomb(loc, genome = "hg38", table = NULL)
```

Arguments

loc	Object of class 'locus' generated by <code>locus()</code>
genome	Either "hg38" or "hg19"
table	Optional character value specifying which recombination table to use.

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).

Sometimes `rtracklayer` generates an intermittent API error: try calling `link_recomb()` again. 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	<i>Create locus object for plotting</i>
-------	---

Description

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

Usage

```
locus(
  gene = NULL,
  data = 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
)
```

Arguments

gene	Optional character value specifying which gene to view. Either gene, or xrange plus seqname, or index_snp must be specified.
data	Dataset (data.frame or data.table) to use for plot. If unspecified or NULL, gene track information alone is returned.
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.

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 an object of class 'locus' ready for plotting, containing the subset of GWAS data to be plotted, chromosome and genomic position range, Ensembl database version number, column names for chromosome, position, SNP ID, p-value or variable for plotting on y axis, locus gene information from Ensembl and locus exon information from Ensembl. If data is NULL then gene track information alone is returned.

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",
  xticks = "top",
  xlab = NULL,
  ...
)
```

Arguments

loc Object of class 'locus' to use for plot. See [locus\(\)](#).

heights Vector supplying the ratio of top to bottom plot.

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

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	<i>Locus plot</i>
------------	-------------------

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 = "blue4",  
  exon_col = "blue4",  
  exon_border = "blue4",  
  text_pos = "top",  
  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 ensemldb::listGenebiotypes() to display possible biotypes. For example, <code>ensemldb::listGenebiotypes(EnsDb.Hsapiens.v75)</code>
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.

<code>use_layout</code>	Logical whether <code>graphics::layout</code> 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 <code>multi_layout()</code> .
<code>heights</code>	Ratio of top to bottom plot. See <code>layout</code> .
<code>showExons</code>	Logical whether to show exons or simply show whole gene as a rectangle
<code>maxrows</code>	Specifies maximum number of rows to display in gene annotation panel.
<code>xticks</code>	Character value of either 'top' or 'bottom' specifying whether x axis ticks and numbers are plotted on top or bottom plot window.
<code>border</code>	Logical whether a bounding box is plotted around upper and lower plots.
<code>gene_col</code>	Colour for gene lines.
<code>exon_col</code>	Fill colour for exons.
<code>exon_border</code>	Border line colour outlining exons (or genes if <code>showExons</code> is FALSE). Set to NA for no border.
<code>text_pos</code>	Character value of either 'top' or 'left' specifying placement of gene name labels.
<code>recomb_col</code>	Colour for recombination rate line if recombination rate data is present. Set to NA to hide the line. See <code>link_recomb()</code> to add recombination rate data.
<code>...</code>	Other arguments passed to <code>scatter_plot()</code> and <code>plot()</code> to control the scatter plot, e.g. <code>ylab</code> , <code>main</code> , etc.

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 = "Ensembl.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,
  heights = c(0.6, 0.4),
  filter_gene_name = NULL,
  filter_gene_biotype = NULL,
  cex.text = 0.7,
  gene_col = "blue4",
  exon_col = "blue4",
  exon_border = "blue4",
  maxrows = 8,
  xlab = NULL,
  ...
)
```

Arguments

loc	Object of class 'locus' to use for plot. See locus() .
heights	Vector controlling relative height of each panel on 0-1 scale.
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, <code>ensemblDb::listGenebiotypes(Ensembl.Hsapiens.v75)</code>
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. Set to NA for no border.
maxrows	Specifies maximum number of rows to display in gene annotation panel.
xlab	Title for x axis. Defaults to chromosome seqname specified in locus.
...	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 <code>locus_plot()</code> 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 <code>locus_plot()</code> 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 $-\log_{10}$ 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

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

Value

No return value. Produces a plot using base graphics.

`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

<code>data</code>	GWAS dataset (data.frame or data.table)
<code>npeaks</code>	Number of peaks to find. If set to NA, algorithm finds all distinct peaks separated from one another by region size specified by span.
<code>p_cutoff</code>	Specifies cut-off for p-value significance above which p-values are ignored.
<code>span</code>	Minimum genomic distance between peaks (default 1 Mb)
<code>min_points</code>	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 <code>min_points</code> to 1 or less.
<code>chrom</code>	Determines which column in data contains chromosome information. If NULL tries to autodetect the column.
<code>pos</code>	Determines which column in data contains position information. If NULL tries to autodetect the column.
<code>p</code>	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

scatter_plot	<i>Locus scatter plot</i>
--------------	---------------------------

Description

Produces a scatter plot from a 'locus' class object. Intended for use with [set_layers\(\)](#).

Usage

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

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.

<code>cex</code>	Specifies size for points.
<code>cex.axis</code>	Specifies font size for axis numbering.
<code>cex.lab</code>	Specifies font size for axis titles.
<code>xlab</code>	x axis title.
<code>ylab</code>	y axis title.
<code>yzero</code>	Logical whether to force y axis limit to include y=0.
<code>xticks</code>	Logical whether x axis numbers and axis title are plotted.
<code>border</code>	Logical whether a bounding box is plotted around upper and lower plots.
<code>showLD</code>	Logical whether to show LD with colours
<code>LD_scheme</code>	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.
<code>recomb_col</code>	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.
<code>legend_pos</code>	Position of legend. See legend() . Set to NULL to hide legend.
<code>labels</code>	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.
<code>label_x</code>	Value or vector for position of label as percentage of x axis scale.
<code>label_y</code>	Value or vector for position of label as percentage of y axis scale.
<code>add</code>	Logical whether to add points to an existing plot or generate a new plot.
<code>align</code>	Logical whether to set par() to align the plot.
<code>...</code>	Other arguments passed to plot() to control the scatter plot e.g. <code>main</code> , <code>ylim</code> etc.

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$index_snp,
  pcutoff = 5e-08,
  chromCol = "royalblue",
  sigCol = "red",
  xlab = NULL,
  ylab = NULL,
  yzero = (loc$yvar == "logP"),
  showLD = TRUE,
  LD_scheme = c("grey", "royalblue", "cyan2", "green3", "orange", "red", "purple"),
  marker_outline = "black",
  marker_size = 7,
  recomb_col = "blue"
)
```

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.
chromCol	Colour for normal points if LD is FALSE when the locus object is made.
sigCol	Colour for significant points if LD is FALSE.
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.

Value

A plotly scatter plot.

See Also

[locus\(\)](#) [locus_plotly\(\)](#)

set_layers	<i>Set up a column of multiple plots</i>
------------	--

Description

Uses `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 `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

[layout\(\)](#)

SLE_gwas_sub	<i>SLE GWAS data subset</i>
--------------	-----------------------------

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>

Index

- * **datasets**
 - SLE_gwas_sub, 29
- ensemldb::listGenebiotypes(), 4, 6, 7, 9, 18, 19, 21
- eqtl_plot, 2
- eqtl_plot(), 13

- genetrack_ly, 7
- genetrack_ly(), 22
- genetracks, 3
- genetracks(), 20
- genetracks_grob, 5
- gg_addgenes, 8
- gg_addgenes(), 12
- gg_genetracks, 9
- gg_genetracks(), 8, 18
- gg_scatter, 10
- gg_scatter(), 8, 18
- grDevices::hcl.colors, 3, 24

- layout, 20
- layout(), 5, 29
- legend(), 3, 24, 27
- line_plot, 12
- link_eqtl, 13
- link_LD, 14
- link_LD(), 20
- link_recomb, 14
- link_recomb(), 11, 20, 27, 28
- locus, 3, 11, 12, 15, 26, 28
- locus(), 3, 4, 6–9, 11–15, 17, 19–22, 24, 27, 28
- locus_ggplot, 17
- locus_ggplot(), 6, 10
- locus_plot, 19
- locus_plot(), 5, 22–24
- locus_plotly, 21
- locus_plotly(), 28

- multi_layout, 22

- multi_layout(), 20

- overlay_plot, 24
- overlay_plot(), 13

- par(), 3, 5, 12, 27
- plot(), 3, 12, 20, 27

- quick_peak, 25

- scatter_plot, 26
- scatter_plot(), 3, 13, 20
- scatter_plotly, 27
- scatter_plotly(), 21, 22
- set_layers, 29
- set_layers(), 2, 3, 12, 13, 26, 27
- SLE_gwas_sub, 29