Package ‘locuszoomr’

July 3, 2024

Title  Gene Locus Plot with Gene Annotations

Version  0.3.1

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)

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  2024-07-03 20:40:02 UTC
eqtl_plot

Description

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

Usage

```r
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.
pct cutoff 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()

description

Plot gene tracks

Description

Plot gene annotation tracks from ensembldb data.
Usage

```r
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,
  highlight = NULL,
  highlight_col = "red",
  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.

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.

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 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

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

Usage

```r
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",
  highlight = NULL,
  highlight_col = "red",
  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.
**genetrack_ly**

Gene tracks using 'plotly'

**Description**

Plot gene annotation tracks from ensembldb data using plotly.

**Usage**

```r
genetrack_ly(
  locus,
  filter_gene_name = NULL,
  filter_gene_biotype = NULL,
  cex.text = 0.7,
  gene_col = ifelse(showExons, "blue4", "skyblue"),
  exon_col = "blue4",
  exon_border = "blue4",
  showExons = TRUE,
  maxrows = 8,
)```

**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**

```r
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)
}
```
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.

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.

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.
gg_addgenes

**Examples**

```r
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**

```r
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**

```r
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", xticks = TRUE, xlab = NULL, highlight = NULL, highlight_col = "red", 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.
gg_genetracks

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>exon_border</td>
<td>Border line colour outlining exons (or genes if showExons is FALSE). Set to NA for no border.</td>
</tr>
<tr>
<td>showExons</td>
<td>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.</td>
</tr>
<tr>
<td>maxrows</td>
<td>Specifies maximum number of rows to display in gene annotation panel.</td>
</tr>
<tr>
<td>text_pos</td>
<td>Character value of either 'top' or 'left' specifying placement of gene name labels.</td>
</tr>
<tr>
<td>xticks</td>
<td>Logical whether x axis ticks and numbers are plotted.</td>
</tr>
<tr>
<td>xlab</td>
<td>Title for x axis. Defaults to chromosome seqname specified in locus.</td>
</tr>
<tr>
<td>highlight</td>
<td>Vector of genes to highlight.</td>
</tr>
<tr>
<td>highlight_col</td>
<td>Single colour or vector of colours for highlighted genes.</td>
</tr>
<tr>
<td>blanks</td>
<td>Controls handling of genes with blank names: &quot;fill&quot; replaces blank gene symbols with ensembl gene ids. &quot;hide&quot; hides genes which are missing gene symbols.</td>
</tr>
</tbody>
</table>

**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**

```r
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)
}
```
Description

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

Usage

```r
gg_scatter(
  loc,
  index_snp = loc$index_snp,
  pcutoff = 5e-08,
  scheme = c("grey", "dodgerblue", "red"),
  size = 2,
  cex.axis = 1,
  cex.lab = 1,
  xlab = NULL,
  ylab = NULL,
  yzero = (loc$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,
  eqtl_gene = NULL,
  beta = NULL,
  ...
)
```

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.
gg_scatter

**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.

**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).

**...** 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**

```r
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)
}
```
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()
**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**

```r
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()`
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

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

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.
- **...**: Optional arguments such as genome_build 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 or chromosome coordinate e.g. "chr7:24966446". 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.
Voltage

See Also

- V.cfr()
locus

Create locus object for plotting

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.
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:

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

```r
## 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**

```r
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,
  highlight = NULL,
  highlight_col = "red",
  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.
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.
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.

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

```r
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)
}
```

Description

Genomic locus plot similar to locuszoom.

Usage

```r
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",
  highlight = NULL,
  highlight_col = "red",
  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 `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 la-
   bels.

highlight
   Vector of genes to highlight.

highlight_col
   Single colour or vector of colours for highlighted genes.

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

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()` and `plot()` to control the scatter
   plot, e.g. `ylab`, `main`, 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 = "EnsDb.Hsapiens.v75")
  ## Add label for index SNP
  locus_plot(loc2, labels = "index")
}

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 = ifelse(showExons, "blue4", "skyblue"),
  exon_col = "blue4",
  exon_border = "blue4",
  showExons = TRUE,
  maxrows = 8,
  width = 600,
  xlab = NULL,
  blanks = "show",
  ...
)
Arguments

- **loc**: Object of class 'locus' to use for plot. See `locus()`.
- **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.
- **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`.
- **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).
- **...**: 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

```r
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

#### Description

Produces pages with multiple locus plots.

#### Usage

```r
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

```r
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**

```r
overlay_plot(
  loc,
  base_col = "black",
  alpha = 0.5,
  scheme = "RdYlBu",
  tissue = "Whole Blood",
  eqtl_gene = loc$gene,
```
Arguments

loc Object of class 'locus' to use for plot. See \texttt{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 \texttt{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 \texttt{legend()}.  
... Other arguments passed to \texttt{locus.plot()} for the locus plot.

Value

No return value. Produces a plot using base graphics.

---

\texttt{quick_peak} \hspace{2cm} \textit{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

\begin{verbatim}
  quick_peak(  
    data,  
    npeaks = NA,  
    p_cutoff = 5e-08,  
    span = 1e+06,  
    min_points = 2,  
    chrom = NULL,  
    pos = NULL,  
    p = NULL  
  )
\end{verbatim}
scatter_plot

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

---

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

```r
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,
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,
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.

cutoff 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.

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.

legend_pos Position of legend. See legend(). Set to NULL to hide legend.
**Description**

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

**Usage**

```r
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$data for eQTL genes.
beta Optional column name for beta coefficient to display upward triangles for posi-
tive beta and downward triangles for negative beta (significant SNPs only).
add_hover Optional vector of column names in loc$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 `layout()` to set up multiple locus plots aligned in a column.

**Usage**

```r
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**

*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**

```r
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, 33

  ensembldb::listGenebiotypes(), 4, 6, 8, 10, 21, 23, 25
eqtl_plot, 2
eqtl_plot(), 15
genetrack_ly, 7
genetrack_ly(), 25
genetracks, 3
genetracks(), 23, 24
genetracks_grob, 6
genetracks_grob(), 11
gg_addgenes, 9
gg_addgenes(), 13
gg_genetracks, 10
gg_genetracks(), 7, 9, 21
gg_scatter, 12
gg_scatter(), 9, 21
graphics::points(), 31
grDevices::hcl.colors, 3, 28

layout, 23
layout(), 3, 33
legend(), 3, 28, 30
line_plot, 14
link_eqtl, 15
link_LD, 16
link_LD(), 23
link_recomb, 17
link_recomb(), 13, 23, 30, 32
locus, 3, 12, 14, 18, 30, 32
locus(), 3, 4, 6, 8–10, 13–17, 21–25, 28, 31, 33
locus_ggplot, 20
locus_ggplot(), 11, 19
locus_plot, 22
locus_plot(), 5, 19, 26, 28, 29
locus_plotly, 24

locus_plotly(), 19, 33
multi_layout, 26
multi_layout(), 23
overlay_plot, 27
overlay_plot(), 15
par(), 3, 5, 14, 31
plot(), 3, 14, 23, 31
quick_peak, 28
scatter_plot, 29
scatter_plot(), 3, 14, 23, 24
scatter_plotly, 31
scatter_plotly(), 25
set_layers, 33
set_layers(), 2, 3, 14, 29, 31
SLE_gwas_sub, 33