

Package ‘rGenomeTracks’

April 27, 2024

Title Integrated visualization of epigenomic data

Version 1.9.0

Description rGenomeTracks package leverages the power of pyGenomeTracks software with the interactivity of R.

pyGenomeTracks is a python software that offers robust method for visualizing epigenetic data files like narrowPeak, Hic matrix, TADs and arcs, however though, here is no way currently to use it within R interactive session.

rGenomeTracks wrapped the whole functionality of pyGenomeTracks with additional utilities to make to more pleasant for R users.

Config/reticulate list(packages = list(list(package =
 ``pyGenomeTracks", version = ``3.6")))

License GPL-3

Depends R (>= 4.1.0),

Imports imager, reticulate, methods, rGenomeTracksData

SystemRequirements pyGenomeTracks (preferred to use
 install_pyGenomeTracks())

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

biocViews Software, HiC, Visualization

git_url <https://git.bioconductor.org/packages/rGenomeTracks>

git_branch devel

git_last_commit dc5ac2d

git_last_commit_date 2023-10-24

Repository Bioconductor 3.19

Date/Publication 2024-04-26

Author Omar Elashkar [aut, cre] (<<https://orcid.org/0000-0002-5505-778X>>)

Maintainer Omar Elashkar <omar.i.elashkar@gmail.com>

Contents

+,genome_track,genome_track-method	2
epilogos_json	3
install_pyGenomeTracks	4
plot_gtracks	5
track_bed	8
track_bedgraph	11
track_bedgraph_matrix	13
track_bigwig	15
track_domains	17
track_epilogos	19
track_gtf	21
track_hic_matrix	23
track_hlines	25
track_links	27
track_narrow_peak	29
track_scalebar	32
track_spacer	33
track_vlines	35
track_x_axis	36
Index	38

+,genome_track,genome_track-method
Adding genome_track Objects

Description

This method adds two "genome_track" objects together.

Usage

```
## S4 method for signature 'genome_track,genome_track'
e1 + e2
```

Arguments

e1 genome_track object.
e2 genome_track object.

Value

genome_track object

Author(s)

Omar Elashkar

Examples

```

tads_dir <- system.file("extdata", "tad_classification.bed",
  package = "rGenomeTracks"
)
genes_dir <- system.file("extdata", "dm3_genes.bed.gz",
  package = "rGenomeTracks"
)
links_dir <- system.file("extdata", "test.arcs",
  package = "rGenomeTracks"
)
tads <- track_domains(tads_dir, color = "#cccccc", border_color = "red")
links_overlay <- track_links(links_dir,
  color = "red",
  line_width = 3, links_type = "loop",
  overlay_previous = "share-y"
)
links <- track_links(links_dir,
  color = "blue",
  line_width = 3, height = 3
)
genes <- track_bed(genes_dir,
  height = 7, style = "flybase",
  fontsize = 10
)
vlines <- track_vlines(genes_dir)
## Not run:
plot_gtracks(tads + links_overlay + links + genes + vlines, chr = "X", start = 30 * 10^5, end = 35 * 10^5)

## End(Not run)

```

epilogos_json

Generate epilogo json configuration file

Description

A convenience function to generate epilogo json configuration file to be passed for `epi_logos()`

Usage

```
epilogos_json(cat_df)
```

Arguments

`cat_df` Dataframe with 3 columns of categories, names and colors

Details

The only argument passed to this function is `data.frame` or `data.frame` similar object. It should have 3 column: First is the state number of epilogos. The second is the label of the state. Finally, the desired colored of such state. Check the example provided for the structure of this `data.frame`.

Value

Directory

Author(s)

Omar Elashkar

Examples

```

epilog_dir <- system.file("extdata", "epilog.qcat.bgz", package = "rGenomeTracks")
epi_cat <- data.frame(
  category = 1:15,
  label = c(
    "Active TSS",
    "Flanking Active TSS",
    "Transcr at gene 5 and 3",
    "Strong transcription",
    "Weak transcription",
    "Genic enhancers",
    "Enhancers",
    "ZNF genes & repeats",
    "Heterochromatin",
    "Bivalent/Poised TSS",
    "Flanking Bivalent TSS/Enh",
    "Bivalent Enhancer",
    "Repressed PolyComb",
    "Weak Repressed PolyComb",
    "Quiescent/Low"
  ),
  color = c(
    "#ff0000", "#ff4500", "#32cd32", "#008000",
    "#006400", "#c2e105", "#ffff00", "#66cdaa",
    "#8a91d0", "#cd5c5c", "#e9967a", "#bdb76b",
    "#808080", "#c0c0c0", "#ffffff"
  )
)
epilog <- track_epilogos(file = epilog_dir, categories_file = epilogos_json(epi_cat))
## Not run:
plot_gtracks(epilog, chr = "X", start = 3100000, 3150000)

## End(Not run)

```

install_pyGenomeTracks

Install pyGenomeTracks Dependency

Description

Install pyGenomeTracks dependency for plot_gtracks()

Usage

```
install_pyGenomeTracks()
```

Details

The function will install miniconda if does not exists and check pyGenomeTracks installation.

Value

None

Author(s)

Omar Elashkar

Examples

```
## Not run:  
install_pyGenomeTracks()  
  
## End(Not run)
```

plot_gtracks

Plotting genomic tracks

Description

This is a generic function used to plot genome_track objects.

Usage

```
plot_gtracks(  
  obj,  
  chr,  
  start,  
  end,  
  dir = NULL,  
  plot = TRUE,  
  verbose = FALSE,  
  dpi = 100,  
  title = NULL,  
  fontsize = NULL,  
  width = 40,  
  height = NULL,  
  trackLabelFraction = 0.05,  
  trackLabelHAlign = "left",  
  ...  
)
```

```
## S4 method for signature 'genome_track'
plot_gtracks(
  obj,
  chr,
  start,
  end,
  dir = NULL,
  plot = TRUE,
  verbose = FALSE,
  dpi = 100,
  title = NULL,
  fontsize = NULL,
  width = 40,
  height = NULL,
  trackLabelFraction = 0.05,
  trackLabelHAlign = "left",
  ...
)
```

Arguments

obj	genome_track object. Define all tracks to be plotted.
chr	String or numeric value to indicate the chromosome desire.
start	Numeric. Starting position of plotting on the defined chromosome.
end	Numeric. Starting position of plotting on the defined chromosome.
dir	String. Default is NULL. If defined, a string to directory and extension to which image is exported. Extension could be png, svg or pdf.
plot	Boolean. Default if TRUE. If FALSE, plot will not be generated, only exported.
verbose	If TRUE, print command that will be passed to pyGenomeTracks.
dpi	Numeric. Default is 100
title	String. Title of the generated plot. Default is NULL.
fontsize	If set, global fontsize value overrides individual tracks.R . argument of all tracks passed.
width	Numeric. The width of the plot. Default is 40
height	Numeric. Height of the plot. Default is NULL to set is based on tracks height.
trackLabelFraction	Numeric. Default is 0.05.
trackLabelHAlign	String. Position of labels alignment. Options are "left", "right" or "center". Default is "left".
...	Extra arguments to be passed for generic plot().

Value

None
None

Note

For this function to run, you need pyGenomeTracks installed in R's loading environment. If not, please run `install_pyGenomeTracks()`

Author(s)

Omar Elashkar

Omar Elashkar

Examples

```
## Not run:
# Get example data directories
# Download h5 example
ah <- AnnotationHub()
query(ah, "rGenomeTracksData")
h5_dir <- ah[["AH95901"]]
tads_dir <- system.file("extdata", "tad_classification.bed",
  package = "rGenomeTracks"
)
arcs_dir <- system.file("extdata", "links2.links", package = "rGenomeTracks")
bw_dir <- system.file("extdata", "bigwig2_X_2.5e6_3.5e6.bw", package = "rGenomeTracks")
#
# Create HiC track from HiC matrix
h5 <- track_hic_matrix(
  file = h5_dir, depth = 250000, min_value = 5, max_value = 200,
  transform = "log1p", show_masked_bins = FALSE
)

# Create TADS track
tads <- track_domains(
  file = tads_dir, border_color = "black",
  color = "none", height = 5,
  line_width = 5,
  show_data_range = FALSE,
  overlay_previous = "share-y"
)

# Create arcs track
arcs <- track_links(
  file = arcs_dir, links_type = "triangles", line_style = "dashed",
  overlay_previous = "share-y",
  color = "darkred",
  line_width = 3,
  show_data_range = FALSE
)

# Create bigwig track
bw <- track_bigwig(
  file = bw_dir, color = "red",
  max_value = 50,
```

```

    min_value = 0,
    height = 4,
    overlay_previous = "yes",
    show_data_range = FALSE
  )

# Create one object from HiC, arcs and bigwig
tracks <- h5 + arcs + bw

# Plot the tracks
plot_gtracks(tracks, chr = "X", start = 25 * 10^5, end = 31 * 10^5)
# Plot HiC, TADS and bigwig tracks
plot_gtracks(h5 + tads + bw, chr = "X", start = 25 * 10^5, end = 31 * 10^5)

## End(Not run)

```

track_bed

Generate bed track

Description

Generate genome_track object from a bed file.

Usage

```

track_bed(
  file,
  title = NULL,
  height = 2,
  overlay_previous = "no",
  fontsize = 12,
  orientation = NULL,
  line_width = 0.5,
  color = "#1f78b4",
  max_value = NULL,
  min_value = NULL,
  border_color = "black",
  preferred_name = "transcript_name",
  merge_transcripts = FALSE,
  labels = TRUE,
  style = "flybase",
  display = "stacked",
  max_labels = 60,
  global_max_row = FALSE,
  gene_rows = NULL,
  arrow_interval = 2,
  arrowhead_included = FALSE,
  color_utr = 0,

```



```

    height_utr = 1,
    arrow_length = 0,
    all_labels_inside = FALSE,
    labels_in_margin = FALSE
)

```

Arguments

file	String. The location of the track file
title	String. If specified, the title of the track to be displayed.
height	Numeric. The height of the plotted track in cm. Default is 2. See notes.
overlay_previous	String. Options are "no" (default) or "yes" or "share-y".
fontsize	Numeric value to font size of tracks's text.
orientation	String. Set to "inverted" to make the track upside down. Default is NULL.
line_width	Numeric. Default is 0.5.
color	String. Hex color or string color. Default is "#1f78b4".
max_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
min_value	Numeric. Default is NULL. The min value cut-off for the numeric column.
border_color	String. default is "black"
preferred_name	String. Denote which column to get elements names. Default is "transcript_name".
merge_transcripts	Boolean. Default is FALSE.
labels	Boolean. Default is FALSE.
style	String. Options are "flybase" (default), or "UCSV" or "tassarow".
display	String. options are "stacked" (default) or "collapsed", "triangles" or "interleaved".
max_labels	Numeric. Any integer about 1. Default is 60.
global_max_row	Boolean. Default is FALSE.
gene_rows	Numeric. Default is NULL.
arrow_interval	Numeric. Should be above 1. Default is 2
arrowhead_included	Boolean. Default is FALSE
color_utr	String. Hex color or string. Default is "grey"
height_utr	Numeric. Between 0 and 1. Default is 1.
arrow_length	Numeric. Default is NULL.
all_labels_inside	Boolean. Default is FALSE
labels_in_margin	Boolean. Default is FALSE.

Details

track_bed() supports all common bed files with minimal of 3 columns and maximum of 12 columns.

Value

genome_track

Note

fontsize argument can be overridden by the same argument in plot_gtracks()

Author(s)

Omar Elashkar

Examples

```
bed12_dir <- system.file("extdata", "dm3_genes.bed.gz",
  package = "rGenomeTracks"
)
bed4_dir <- system.file("extdata", "dm3_genes.bed4.gz",
  package = "rGenomeTracks"
)
bed6_dir <- system.file("extdata", "dm3_genes.bed6.gz",
  package = "rGenomeTracks"
)

# Create bed track using bed4 file
bed4 <- track_bed(
  file = bed4_dir, height = 3, title = "bed4", color = "cyan", ,
  border_color = "#9ACD32", line_width = 1.5
)

# Create bed track using bed6 file
bed6 <- track_bed(
  file = bed6_dir, height = 3, title = "bed4", fontsize = 8, color = "red",
  border_color = "yellow", arrowhead_included = TRUE
)

# Create bed track using bed12 file
bed12 <- track_bed(
  file = bed12_dir, height = 3, title = "bed12", style = "UCSC",
  arrow_interval = 10, fontsize = 10
)

# Create a spacer track
space <- track_spacer(height = 1)
## Not run:
# Plotting the tracks
plot_gtracks(bed4 + space + bed6 + space + bed12 + space,
  chr = "X", start = 300 * 10^4, end = 330 * 10^4, verbose = TRUE
```

```
)
## End(Not run)
```

track_bedgraph	<i>Generate bedgraph track</i>
----------------	--------------------------------

Description

Generate genome_track object from bedgraph files.

Usage

```
track_bedgraph(
  file,
  title = NULL,
  height = 2,
  overlay_previous = "no",
  orientation = NULL,
  color = "#1f78b4",
  alpha = 1,
  max_value = NULL,
  min_value = NULL,
  use_middle = FALSE,
  show_data_range = TRUE,
  type = "fill",
  negative_color = NULL,
  nans_to_zeros = FALSE,
  summary_method = NULL,
  number_of_bins = 700,
  transform = "no",
  log_pseudocount = 0,
  y_axis_values = "transformed",
  second_file = NULL,
  operation = "file",
  grid = FALSE,
  rasterize = FALSE
)
```

Arguments

file	String. The location of the track file
title	String. If specified, the title of the track to be displayed.
height	Numeric. The height of the plotted track in cm. Default is 2. See notes.
overlay_previous	String. Options are "no" (default) or "yes" or "share-y".

orientation	String. Default is NULL. Other option is "inverted".
color	String. Hex color or string color. Default is "#1f78b4".
alpha	Numeric variable between 0 and 1 to indicate level of transparency. Default is 1.
max_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
min_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
use_middle	Boolean. Default is FALSE.
show_data_range	Boolean. Default is TRUE.
type	String. Options are "fill" (default), "line", "points".
negative_color	Hex color or string to indicate color of negative values. Default is NULL.
nans_to_zeros	Boolean. To convert empty values to zeros, set this to TRUE. Default is FALSE.
summary_method	String. summary_method applied over bin range. This parameter is set to NULL. See details for options.
number_of_bins	Numeric value to indicate summary method used over the bin range. Default is 700
transform	String to indicate type of transformation applied. Default is "no".
log_pseudocount	Numeric. Default is 0.
y_axis_values	String with two options "transformed" (default) or "original".
second_file	Path for another file to be included in operations. This parameter is not set by default.
operation	Default is set to "file". See details.
grid	Boolean. Default is FALSE.
rasterize	Boolean. Default is FALSE.

Details

summary_method parameter can be chosen to be by "mean", "average", "max", "min", "stdev", "dev", "coverage", "cov" or "sum". Transform parameter options are "no" (default) or "log", "log1p", "-log", "log2" or "log10". 'log1p': transformed_values = log(1 + initial_values) 'log': transformed_values = log(log_pseudocount + initial_values) 'log2': transformed_values = log2(log_pseudocount + initial_values) 'log10': transformed_values = log10(log_pseudocount + initial_values) '-log': transformed_values = log(log_pseudocount + initial_values) To compute operations on the fly on the file or between 2 bedgraph files, you can tweak operation parameter, it should contains file or file and second_file. It is adviced to use nans_to_zeros = TRUE to avoid unexpected results. Example value for operation are "0.89 * file", "- file", "file - second_file", "log2((1 + file) / (1 + second_file))" and "max(file, second_file)"

to add the preferred line width or point size : type = "line:lw" where lw (linewidth) is numeric value. Like type = "line:0.5" and type = "points:0.5"

By default the bedgraph is plotted at the base pair resolution. This can lead to very large pdf/svg files. If plotting large regions. If you want to decrease the size of your file. You can either rasterize the bedgraph profile by using: rasterize = TRUE

Value

genome_track

Note

fontsize parameter can be overridden by the same argument in plot_gtracks() height parameter will be ignored if overlay_previous is set.

Author(s)

Omar Elashkar

Examples

```
bg_dir <- system.file("extdata", "GSM3182416_E12DHL_WT_Hoxd11vp.bedgraph.gz",
  package = "rGenomeTracks"
)
bed_genes_dir <- system.file("extdata", "HoxD_cluster_regulatory_regions_mm10.bed",
  package = "rGenomeTracks"
)

bg <- track_bedgraph(bg_dir, color = "green", height = 5, max_value = 10)
bg_middle <- track_bedgraph(bg_dir,
  use_middle = TRUE, color = "blue",
  height = 5, max_value = 10
)
bed_genes <- track_bed(bed_genes_dir,
  title = "Regulatory regions", ,
  color = "red", height = 3
)

tracks <- track_x_axis(when = "top") + bg + bg_middle + bed_genes
## Not run:
plot_gtracks(tracks,
  chr = 2, start = 738 * 10^5, end = 750 * 10^5,
  trackLabelFraction = 0.2
)

## End(Not run)
```

track_bedgraph_matrix *Generate bedgraph matrix track*

Description

A track for file like bedgraph but with more than 4 columns, like the insulation score from hicPlot-TADs

Usage

```

track_bedgraph_matrix(
    file,
    title = NULL,
    height = 2,
    overlay_previous = "no",
    orientation = NULL,
    max_value = NULL,
    min_value = NULL,
    show_data_range = FALSE,
    type = "matrix",
    rasterize = TRUE,
    pos_score_in_bin = "center",
    plot_horizontal_lines = FALSE,
    colormap = "viridis"
)

```

Arguments

file	String. The location of the track file
title	String. If specified, the title of the track to be displayed.
height	Numeric. The height of the plotted track in cm.
overlay_previous	String. Options are "no" (default) or "yes" or
orientation	String. Set to "inverted" to make the track upside down. Default is NULL.
max_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
min_value	Numeric. Default is NULL. The min value cut-off for the numeric column.
show_data_range	Boolean. Default is FALSE.
type	"matrix" (default) or "lines".
rasterize	Boolean. Default is TRUE
pos_score_in_bin	String value to indicate the position of score with respect to bin start and end. Possible values are either "center" (default) or "block".
plot_horizontal_lines	Boolean. Can be used only if type parameter is set to "lines".
colormap	String with matplotlib-compatible colormap. Default is set to "viridis".

Details

The different options for color maps can be found here: <https://matplotlib.org/users/colormaps.html>.

Value

genome_track

Note

fontsize argument can be overridden by the same argument in plot_gtracks()

Author(s)

Omar Elashkar

Examples

```
IS_dir <- system.file("extdata", package = "rGenomeTracks", "tad_separation_score.bm.gz")
IS <- track_bedgraph_matrix(IS_dir)
## Not run:
plot_gtracks(IS, chr = "X", start = 2000000, end = 3500000)

## End(Not run)
```

track_bigwig

Generate bigwig track

Description

Create genome_track object from bigwig file.

Usage

```
track_bigwig(
  file,
  title = NULL,
  height = 2,
  overlay_previous = "no",
  orientation = NULL,
  color = "#1f78b4",
  alpha = 1,
  max_value = NULL,
  min_value = NULL,
  show_data_range = TRUE,
  type = "fill",
  negative_color = NULL,
  nans_to_zeros = FALSE,
  summary_method = "mean",
  number_of_bins = 700,
  transform = "no",
  log_pseudocount = 0,
  y_axis_values = "transformed",
  second_file = NULL,
  operation = "file",
  grid = FALSE
)
```

Arguments

file	String. The location of the track file
title	String. If specified, the title of the track to be displayed.
height	Numeric. The height of the plotted track in cm.
overlay_previous	String. Options are "no" (default) or "yes" or
orientation	String. Set to "inverted" to make the track upside down. Default is NULL.
color	String. Hex color or string color. Default is "#1f78b4".
alpha	Numeric variable between 0 and 1 to indicate level of transparency. Default is 1.
max_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
min_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
show_data_range	Boolean. Default is TRUE.
type	String. Options are "fill" (default), "line", "points".
negative_color	Hex color or string to indicate color of negative values. Default is NULL.
nans_to_zeros	Boolean. To convert empty values to zeros, set this to TRUE. Default is FALSE.
summary_method	String. summary_method applied over bin range. This parameter is set to NULL. See details for options.
number_of_bins	Numeric value to indicate summary method used over the bin range. Default is 700
transform	String to indicate type of transformation applied. Default is "no".
log_pseudocount	Numeric. Default is 0.
y_axis_values	String with two options "transformed" (default) or "original".
second_file	Path for another file to be included in operations. This parameter is not set by default.
operation	Default is set to "file". See details.
grid	Boolean. Default is FALSE.

Details

summary_method parameter can be chosen to be by "mean", "average", "max", "min", "stdev", "dev", "coverage", "cov" or "sum". Transform parameter options are "no" (default) or "log", "log1p", "-log", "log2" or "log10". 'log1p': transformed_values = $\log(1 + \text{initial_values})$ 'log': transformed_values = $\log(\log_pseudocount + \text{initial_values})$ 'log2': transformed_values = $\log_2(\log_pseudocount + \text{initial_values})$ 'log10': transformed_values = $\log_{10}(\log_pseudocount + \text{initial_values})$ 'log': transformed_values = $\log(\log_pseudocount + \text{initial_values})$ To compute operations on the fly on the file or between 2 bedgraph files, you can tweak operation parameter, it should contains file or file and second_file. It is advised to use nans_to_zeros = TRUE to avoid unexpected results. Example value for operation are "0.89 * file", "- file", "file - second_file", "log2((1 + file) / (1 + second_file))" and "max(file, second_file)"

Value

None

to add the preferred line width or point size : type = "line:lw" where lw (linewidth) is numeric value.

Like type = "line:0.5" and type = "points:0.5"

Author(s)

Omar Elashkar

Examples

```

bw_dir <- system.file("extdata", "bigwig2_X_2.5e6_3.5e6.bw",
  package = "rGenomeTracks"
)
mean_bw <- track_bigwig(
  file = bw_dir, color = "gray",
  type = "point:1", summary_method = "mean", number_of_bins = 300, max_value = 200, min_value = -5
)
min_bw <- track_bigwig(
  file = bw_dir, color = "blue", type = "line:1", summary_method = "min", number_of_bins = 300,
  overlay_previous = "share-y", show_data_range = FALSE,
  max_value = 200, min_value = -5
)
max_bw <- track_bigwig(
  file = bw_dir, color = "red", type = "line:1", summary_method = "max", number_of_bins = 300,
  overlay_previous = "share-y", show_data_range = FALSE,
  max_value = 200, min_value = -5
)
hlines <- track_hlines(
  y_values = "10, 150",
  overlay_previous = "share-y",
  color = "blue", line_style = "dotted"
)
## Not run:
plot_gtracks(mean_bw + min_bw + max_bw + hlines, chr = "X", start = 27 * 10^5, end = 31 * 10^5)

## End(Not run)

```

track_domains

Generate domains track

Description

Domain files are bed files represents TADS in the case of HiC analysis.

Usage

```

track_domains(
  file,
  title = NULL,
  height = 2,
  overlay_previous = "no",
  orientation = NULL,
  line_width = 0.5,
  color = "#1f78b4",
  max_value = NULL,
  show_data_range = TRUE,
  min_value = NULL,
  border_color = "black",
  preferred_name = "transcript_name",
  merge_transcripts = FALSE
)

```

Arguments

file	String. The location of the track file
title	String. If specified, the title of the track to be displayed.
height	Numeric. The height of the plotted track in cm. Default is 2. See notes.
overlay_previous	String. Options are "no" (default) or "yes" or "share-y".
orientation	String. Set to "inverted" to make the track upside down. Default is NULL.
line_width	Numeric. Default is 0.5.
color	String. Hex color or string color. Default is "#1f78b4".
max_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
show_data_range	Boolean. Default is TRUE.
min_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
border_color	String. default is "black"
preferred_name	String. Denote which column to get elements names. Default is "transcript_name".
merge_transcripts	Boolean. Default is FALSE.

Details

To remove the border, set 'border_color' parameter to "none".

Value

genome_track

Author(s)

Omar Elashkar

Examples

```
tads_dir <- system.file("extdata", "tad_classification.bed",
  package = "rGenomeTracks"
)
tads <- track_domains(
  file = tads_dir, border_color = "black",
  color = "#11FF34", height = 5
)
tads_i <- track_domains(
  file = tads_dir, border_color = "red",
  color = "#cccccc", height = 3, orientation = "inverted"
)
tracks <- track_x_axis(when = "top") +
  tads + tads_i
## Not run:
plot_gtracks(tracks, chr = "X", start = 30 * 10^5, end = 35 * 10^5)

## End(Not run)
```

track_epilogos	<i>Generate epilogos track</i>
----------------	--------------------------------

Description

Generate epilogos genome_track from qcat file.

Usage

```
track_epilogos(
  file,
  title = NULL,
  height = 2,
  overlay_previous = "no",
  categories_file = NULL,
  orientation = NULL
)
```

Arguments

file	String. The location of the track file
title	String. If specified, the title of the track to be displayed.
height	Numeric. The height of the plotted track in cm. Default is 2. See notes.
overlay_previous	String. Options are "no" (default) or "yes" or "share-y".

categories_file Optionally pass a string of JSON custom colors configuration file directory. Default is NULL.

orientation String. Set to "inverted" to make the track upside down. Default is NULL.

Details

Epilogos is used widely to represent multiple "states" across genome, like ChromHMM states. More details [here](#) qcat file is needed which can be generated using `epilogos track_epilogos` can optionally take `categories_file` parameter which specify the color scheme for the states present in qcat file. Check the example section for demonstration.

Value

None

Note

fontsize argument can be overridden by the same argument in `plot_gtracks()`

Author(s)

Omar Elashkar

Examples

```
epilog_dir <- system.file("extdata", "epilog.qcat.bgz", package = "rGenomeTracks")
epi_cat <- data.frame(
  category = 1:15,
  label = c(
    "Active TSS",
    "Flanking Active TSS",
    "Transcr at gene 5 and 3",
    "Strong transcription",
    "Weak transcription",
    "Genic enhancers",
    "Enhancers",
    "ZNF genes & repeats",
    "Heterochromatin",
    "Bivalent/Poised TSS",
    "Flanking Bivalent TSS/Enh",
    "Bivalent Enhancer",
    "Repressed PolyComb",
    "Weak Repressed PolyComb",
    "Quiescent/Low"
  ),
  color = c(
    "#ff0000", "#ff4500", "#32cd32", "#008000",
    "#006400", "#c2e105", "#ffff00", "#66cdaa",
    "#8a91d0", "#cd5c5c", "#e9967a", "#bdb76b",
    "#808080", "#c0c0c0", "#ffffff"
  )
)
```

```
)
epilog <- track_epilogos(file = epilog_dir, categories_file = epilogos_json(epi_cat))
## Not run:
plot_gtracks(epilog, chr = "X", start = 3100000, 3150000)

## End(Not run)
```

track_gtf

Generate gtf track

Description

Create genome_track object for gtf annotation files.

Usage

```
track_gtf(
  file,
  title = NULL,
  height = 2,
  overlay_previous = "no",
  fontsize = 12,
  orientation = NULL,
  line_width = 0.5,
  color = "#1f78b4",
  border_color = "black",
  preferred_name = "transcript_name",
  merge_transcripts = FALSE,
  labels = FALSE,
  display = "stacked",
  max_labels = 60,
  global_max_row = FALSE,
  gene_rows = NULL,
  arrow_interval = 2,
  arrowhead_included = FALSE,
  color_utr = "grey",
  height_utr = 1,
  arrow_length = NULL,
  all_labels_inside = FALSE,
  labels_in_margin = FALSE
)
```

Arguments

file	String. The location of the track file
title	String. If specified, the title of the track to be displayed.
height	Numeric. The height of the plotted track in cm. Default is 2. See notes.

overlay_previous	String. Options are "no" (default) or "yes" or "share-y".
fontsize	Numeric value to font size of tracks's text.
orientation	String. Set to "inverted" to make the track upside down. Default is NULL.
line_width	Numeric. Default is 0.5.
color	String. Hex color or string color. Default is "#1f78b4".
border_color	String. default is "black"
preferred_name	String. Denote which column to get elements names. Default is "transcript_name".
merge_transcripts	Boolean. Default is FALSE.
labels	Boolean. Default is FALSE.
display	String. options are "stacked" (default) or "collapsed", "triangles" or "interleaved".
max_labels	Numeric. Any integer about 1. Default is 60.
global_max_row	Boolean. Default is FALSE.
gene_rows	Numeric. Default is NULL.
arrow_interval	Numeric. Should be above 1. Default is 2
arrowhead_included	Boolean. Default is FALSE
color_utr	String. Hex color or string. Default is "grey"
height_utr	Numeric. Between 0 and 1. Default is 1.
arrow_length	Numeric. Default is NULL.
all_labels_inside	Boolean. Default is FALSE
labels_in_margin	Boolean. Default is FALSE.

Details

gtf files, unlike bed file, can provide richer annotation regarding levels of annotation where genomic features can be grouped based on the composing entity.

Value

genome_track

Note

fontsize argument can be overridden by the same argument in plot_gtracks()

Author(s)

Omar Elashkar

Examples

```
gtf_dir <- system.file("extdata", "dm3_subset_BDGP5.78.gtf.gz",
  package = "rGenomeTracks"
)
gtf <- track_gtf(
  file = gtf_dir, height = 10,
  preferred_name = "gene_name", merge_transcripts = TRUE, fontsize = 12
)
## Not run:
plot_gtracks(gtf + track_spacer() +
  track_x_axis(), chr = "X", start = 30 * 10^5, end = 33 * 10^5)

## End(Not run)
```

track_hic_matrix	<i>Generate HiC track</i>
------------------	---------------------------

Description

Create a genome_track for matrix files. Currently, only cool format and h5 format.

Usage

```
track_hic_matrix(
  file,
  title = NULL,
  height = NULL,
  overlay_previous = "no",
  orientation = NULL,
  max_value = NULL,
  min_value = NULL,
  transform = "no",
  rasterize = TRUE,
  colormap = "RdYlBu_r",
  depth = 100000,
  show_masked_bins = FALSE,
  scale_factor = 1
)
```

Arguments

file	String. The location of the track file
title	String. If specified, the title of the track to be displayed.
height	Numeric. The height of the plotted track in cm. Default is 2. See notes.
overlay_previous	String. Options are "no" (default) or "yes" or "share-y".
orientation	String. Set to "inverted" to make the track upside down. Default is NULL.

max_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
min_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
transform	String to indicate type of transformation applied. Default is "no".
rasterize	Boolean. Default is FALSE.
colormap	String with matplotlib-compatible colormap. Default is set to "viridis".
depth	Numeric value above 1 to indicate the maximum distance that should be plotted. Default is 100000.
show_masked_bins	Boolean. If TRUE, showing masked bins as white lines. Default is FALSE.
scale_factor	Numeric factor by which matrix is to be scaled.

Details

This function expect cool or h5 format. Format converter like [hicConvertFormat](#) can help converting to supported formats. depth is the maximum distance that should be plotted. If it is more than 125% of the plotted region, it will be adjsted to this maximum value. colormap argument should be compatible with [matplotlib](#). show_masked_bins plots bins not used during the corrections as white lines. Setting this argument to FALSE (default) extends neighboring bins to obtain an aesthetically pleasant output. scale argument scales the matrix by specific factor. This is useful if plotting multiple hic-matrices to be on the same scale.

Value

genom_track

Author(s)

Omar Elashkar

Examples

```
## Not run:
# Get example data directories
# Download h5 example
ah <- AnnotationHub()
query(ah, "rGenomeTracksData")
h5_dir <- ah[["AH95901"]]
tads_dir <- system.file("extdata", "tad_classification.bed",
  package = "rGenomeTracks"
)
arcs_dir <- system.file("extdata", "links2.links", package = "rGenomeTracks")
bw_dir <- system.file("extdata", "bigwig2_X_2.5e6_3.5e6.bw", package = "rGenomeTracks")
#
# Create HiC track from HiC matrix
h5 <- track_hic_matrix(
  file = h5_dir, depth = 250000, min_value = 5, max_value = 200,
  transform = "log1p", show_masked_bins = FALSE
)
```



```
# Create TADS track
tads <- track_domains(
  file = tads_dir, border_color = "black",
  color = "none", height = 5,
  line_width = 5,
  show_data_range = FALSE,
  overlay_previous = "share-y"
)

# Create arcs track
arcs <- track_links(
  file = arcs_dir, links_type = "triangles", line_style = "dashed",
  overlay_previous = "share-y",
  color = "darkred",
  line_width = 3,
  show_data_range = FALSE
)

# Create bigwig track
bw <- track_bigwig(
  file = bw_dir, color = "red",
  max_value = 50,
  min_value = 0,
  height = 4,
  overlay_previous = "yes",
  show_data_range = FALSE
)

# Create one object from HiC, arcs and bigwig
tracks <- h5 + arcs + bw

# Plot the tracks
plot_gtracks(tracks, chr = "X", start = 25 * 10^5, end = 31 * 10^5)
# Plot HiC, TADS and bigwig tracks
plot_gtracks(h5 + tads + bw, chr = "X", start = 25 * 10^5, end = 31 * 10^5)

## End(Not run)
```

track_hlines

Generate a track with horizontal lines

Description

track_hlines() creates a genome_track with horizontal lines that can be overlaid on the previous track or, by default, track the lines in separate track.

Usage

```
track_hlines(
  y_values,
```

```

    title = NULL,
    height = 0.5,
    overlay_previous = NULL,
    orientation = NULL,
    line_width = 0.5,
    line_style = "solid",
    color = "black",
    alpha = 1,
    max_value = NULL,
    min_value = NULL,
    show_data_range = TRUE
)

```

Arguments

<code>y_values</code>	String for y-values where horizontal lines should be plotted separated by comma.
<code>title</code>	String. If specified, the title of the track to be displayed.
<code>height</code>	Numeric. The height of the plotted track in cm. Default is 2. See notes.
<code>overlay_previous</code>	String. Options are "no" (default) or "yes" or "share-y".
<code>orientation</code>	String. Default is NULL. Other option is "inverted".
<code>line_width</code>	Numeric value for line width.
<code>line_style</code>	String with options of either "solid", "dashed", "dotted", and "dashdot".
<code>color</code>	String. Hex color or string color. Default is "#1f78b4".
<code>alpha</code>	Numeric variable between 0 and 1 to indicate level of transparency. Default is 1.
<code>max_value</code>	Numeric. Default is NULL. The max value cut-off for the numeric column.
<code>min_value</code>	Numeric. Default is NULL. The max value cut-off for the numeric column.
<code>show_data_range</code>	Boolean. Default is TRUE.

Details

`y_values` argument specify locations on the genome where where horizontal lines should be plotted separated by comma, like "50, 90"

Value

genome_track

Author(s)

Omar Elashkar

Examples

```

bw_dir <- system.file("extdata", "bigwig2_X_2.5e6_3.5e6.bw",
  package = "rGenomeTracks"
)
mean_bw <- track_bigwig(
  file = bw_dir, color = "gray",
  type = "point:1", summary_method = "mean", number_of_bins = 300, max_value = 200, min_value = -5
)
min_bw <- track_bigwig(
  file = bw_dir, color = "blue", type = "line:1", summary_method = "min", number_of_bins = 300,
  overlay_previous = "share-y", show_data_range = FALSE,
  max_value = 200, min_value = -5
)
max_bw <- track_bigwig(
  file = bw_dir, color = "red", type = "line:1", summary_method = "max", number_of_bins = 300,
  overlay_previous = "share-y", show_data_range = FALSE,
  max_value = 200, min_value = -5
)
hlines <- track_hlines(
  y_values = "10, 150",
  overlay_previous = "share-y",
  color = "blue", line_style = "dotted"
)
## Not run:
plot_gtracks(mean_bw + min_bw + max_bw + hlines, chr = "X", start = 27 * 10^5, end = 31 * 10^5)

## End(Not run)

```

track_links

Generate links track

Description

Generate links track from arc file.

Usage

```

track_links(
  file,
  title = NULL,
  height = 2,
  overlay_previous = "no",
  orientation = NULL,
  links_type = "arcs",
  line_width = NULL,
  line_style = "solid",
  color = "blue",
  alpha = 0.8,
  max_value = NULL,

```

```

    min_value = NULL,
    ylim = NULL,
    show_data_range = FALSE,
    compact_arcs_level = 0,
    use_middle = FALSE
)

```

Arguments

file	String. The location of the track file
title	String. If specified, the title of the track to be displayed.
height	Numeric. The height of the plotted track in cm. Default is 2. See notes.
overlay_previous	String. Options are "no" (default) or "yes" or "share-y".
orientation	String. Default is NULL. Other option is "inverted".
links_type	String value with options "arcs" (default) or "triangles" or "loops".
line_width	Numeric value for line width.
line_style	String with options of either "solid", "dashed", "dotted", and "dashdot".
color	String. Hex color or string color. Default is "#1f78b4".
alpha	Numeric variable between 0 and 1 to indicate level of transparency. Default is 1.
max_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
min_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
ylim	Numeric value above 0 to set arcs' height cutoff. Default is NULL
show_data_range	Boolean. Default is TRUE.
compact_arcs_level	Numeric value of either 0, 1 or 2 to indicate level of arcs' compactness by distance it travels.
use_middle	Boolean. Default is FALSE.

Details

Level of compactness relative to arcs' length can be manipulated using the argument `compact_arcs_level` where:

- `compact_arcs_level = 0`, The default where the height is proportional to distance
- `compact_arcs_level = 1`, the height is proportional to the square root of the distance
- `compact_arcs_level = 2`, the height is the same for all distances

`ylim` argument sets the cutoff for arcs' height. This could be handy if you have small arc overridden by larger arc.

Value

genome_track

Note

ylim argument is incompatible with compact_arcs_level = 2

Author(s)

Omar Elashkar

Examples

```
tads_dir <- system.file("extdata", "tad_classification.bed",
  package = "rGenomeTracks"
)
genes_dir <- system.file("extdata", "dm3_genes.bed.gz",
  package = "rGenomeTracks"
)
links_dir <- system.file("extdata", "test.arcs",
  package = "rGenomeTracks"
)
tads <- track_domains(tads_dir, color = "#cccccc", border_color = "red")
links_overlay <- track_links(links_dir,
  color = "red",
  line_width = 3, links_type = "loop",
  overlay_previous = "share-y"
)
links <- track_links(links_dir,
  color = "blue",
  line_width = 3, height = 3
)
genes <- track_bed(genes_dir,
  height = 7, style = "flybase",
  fontsize = 10
)
vlines <- track_vlines(genes_dir)
## Not run:
plot_gtracks(tads + links_overlay + links + genes + vlines, chr = "X", start = 30 * 10^5, end = 35 * 10^5)

## End(Not run)
```

track_narrow_peak *Generate narrow peaks track*

Description

Create genome_track object from narrow peak bed format.

Usage

```

track_narrow_peak(
  file,
  title = NULL,
  height = 3,
  overlay_previous = "no",
  orientation = NULL,
  line_width = 1,
  color = "#FF000080",
  max_value = NULL,
  show_data_range = TRUE,
  show_labels = TRUE,
  use_summit = TRUE,
  width_adjust = 1.5,
  type = "peak"
)

```

Arguments

file	String. The location of the track file
title	String. If specified, the title of the track to be displayed.
height	Numeric. The height of the plotted track in cm. Default is 2. See notes.
overlay_previous	String. Options are "no" (default) or "yes" or "share-y".
orientation	String. Default is NULL. Other option is "inverted".
line_width	Numeric value for line width.
color	String. Hex color or string color. Default is "#1f78b4".
max_value	Numeric. Default is NULL. The max value cut-off for the numeric column.
show_data_range	Boolean. Default is TRUE.
show_labels	Boolean. If TRUE, display labels on plotting which include peak tag, p-val and q-val.
use_summit	Boolean. If TRUE, peak summit data will be plotted.
width_adjust	Numeric value above 0 to adjust peaks' width. Default is 1.5.
type	String with options either "peak" or "box".

Details

narrowPeak file is bed file (4+3), where the 5th column is peak name, 6th column in p-value and 7th column in q-value. You might increase height it increased font size. narrowPeak format is very common with analysis pipelines involving MACS2. narrowPeak format provides the information of the peak summit. use_summit argument is used to determine if this information should be used. By default this information is used (use_summit = TRUE) although some peaks may look crooked. type argument specify if the plot will be:

- "box" which will plot a rectangle of the peak width

- or "peak" which will plot the shape of the peak, whose height is the narrowPeak file signal value (usually peak coverage)

Value

genome_track

Author(s)

Omar Elashkar

Examples

```
np_bed_dir <- system.file("extdata", "test2.narrowPeak", package = "rGenomeTracks")

tracks <-
  track_scalebar() +
  track_narrow_peak(np_bed_dir,
    title = "peak type with summit",
    height = 3,
    type = "peak",
    color = "green"
  ) +

  track_spacer(height = 2) +
  track_narrow_peak(np_bed_dir,
    title = "peak type without summit",
    height = 3,
    type = "peak",
    color = "green",
    use_summit = FALSE
  ) +

  track_spacer(height = 2) +
  track_narrow_peak(np_bed_dir,
    title = "Box type with summit",
    height = 3,
    type = "box",
    color = "blue"
  ) +

  track_spacer(height = 2) +
  track_narrow_peak(np_bed_dir,
    title = "Box type without summit",
    height = 3,
    type = "box",
    color = "blue",
    use_summit = FALSE
  ) +
  track_x_axis()
## Not run:
plot_gtracks(tracks, chr = "X", start = 276 * 10^4, end = 280 * 10^4, trackLabelFraction = 0.2)

## End(Not run)
```

track_scalebar	<i>Generate scalebar track</i>
----------------	--------------------------------

Description

scalebar track is a track with a stretch that highlights specific distance on the genomic coordinates

Usage

```
track_scalebar(
  title = NULL,
  height = 2,
  overlay_previous = "no",
  where = "left",
  fontsize = 12,
  line_width = 0.5,
  color = "black",
  alpha = 1,
  x_center = NULL,
  size = NULL,
  scalebar_start_position = NULL,
  scalebar_end_position = NULL
)
```

Arguments

title	String. If specified, the title of the track to be displayed.
height	Numeric. The height of the plotted track in cm. Default is 2. See notes.
overlay_previous	String. Options are "no" (default) or "yes" or "share-y".
where	"left" (default), "right", "top" or "bottom".
fontsize	Numeric value to font size of tracks's text.
line_width	0.5 (default) or any float above 0.
color	String. Hex color or string color. Default is "#1f78b4".
alpha	Numeric variable between 0 and 1 to indicate level of transparency. Default is 1.
x_center	Numeric value above 0. Default is NULL.
size	Numeric value above 0. Default is NULL.
scalebar_start_position	Numeric value above 0. Default is NULL.
scalebar_end_position	Numeric value above 0. Default is NULL.

Value

genome_track

Note

fontsize argument can be overridden by the same argument in plot_gtracks()

Author(s)

Omar Elashkar

Examples

```
np_bed_dir <- system.file("extdata", "test2.narrowPeak", package = "rGenomeTracks")

tracks <-
  track_scalebar(
    scalebar_start_position = 2785 * 10^3,
    scalebar_end_position = 2799 * 10^3
  ) +
  track_narrow_peak(np_bed_dir,
    title = "peak type with summit",
    height = 3,
    type = "peak",
    color = "green"
  ) + track_x_axis()
## Not run:
plot_gtracks(tracks, chr = "X", start = 276 * 10^4, end = 280 * 10^4, trackLabelFraction = 0.2)

## End(Not run)
```

 track_spacer

Generate spacing track

Description

Create spacing track with custom height.

Usage

```
track_spacer(title = NULL, height = 2, overlay_previous = "no")
```

Arguments

title String. If specified, the title of the track to be displayed.

height Numeric. The height of the plotted track in cm. Default is 2. See notes.

overlay_previous String. Options are "no" (default) or "yes" or "share-y".

Value

None

Author(s)

Omar Elashkar

Examples

```
bed12_dir <- system.file("extdata", "dm3_genes.bed.gz",
  package = "rGenomeTracks"
)
bed4_dir <- system.file("extdata", "dm3_genes.bed4.gz",
  package = "rGenomeTracks"
)
bed6_dir <- system.file("extdata", "dm3_genes.bed6.gz",
  package = "rGenomeTracks"
)

# Create bed track using bed4 file
bed4 <- track_bed(
  file = bed4_dir, height = 3, title = "bed4", color = "cyan", ,
  border_color = "#9ACD32", line_width = 1.5
)

# Create bed track using bed6 file
bed6 <- track_bed(
  file = bed6_dir, height = 3, title = "bed4", fontsize = 8, color = "red",
  border_color = "yellow", arrowhead_included = TRUE
)

# Create bed track using bed12 file
bed12 <- track_bed(
  file = bed12_dir, height = 3, title = "bed12", style = "UCSC",
  arrow_interval = 10, fontsize = 10
)

# Create a spacer track
space <- track_spacer(height = 1)
## Not run:
# Plotting the tracks
plot_gtracks(bed4 + space + bed6 + space + bed12 + space,
  chr = "X", start = 300 * 10^4, end = 330 * 10^4, verbose = TRUE
)

## End(Not run)
```

track_vlines	<i>Overlay vertical lines from a bed file</i>
--------------	---

Description

track_vlines() overlay vertical lines over the whole plot. The only parameter to be passed is a bed file.

Usage

```
track_vlines(file)
```

Arguments

file String. The location of the track file

Value

genome_track

Author(s)

Omar Elashkar

Examples

```
tads_dir <- system.file("extdata", "tad_classification.bed",
  package = "rGenomeTracks"
)
genes_dir <- system.file("extdata", "dm3_genes.bed.gz",
  package = "rGenomeTracks"
)
links_dir <- system.file("extdata", "test.arcs",
  package = "rGenomeTracks"
)
tads <- track_domains(tads_dir, color = "#cccccc", border_color = "red")
links_overlay <- track_links(links_dir,
  color = "red",
  line_width = 3, links_type = "loop",
  overlay_previous = "share-y"
)
links <- track_links(links_dir,
  color = "blue",
  line_width = 3, height = 3
)
genes <- track_bed(genes_dir,
  height = 7, style = "flybase",
  fontsize = 10
)
```

```

vlines <- track_vlines(genes_dir)
## Not run:
plot_gtracks(tads + links_overlay + links + genes + vlines, chr = "X", start = 30 * 10^5, end = 35 * 10^5)

## End(Not run)

```

track_x_axis	<i>Specify x_axis option for genome_track.</i>
--------------	--

Description

This track will specify the options for x-axis for location, height, font size and whether to overlay previous track.

Usage

```

track_x_axis(
  title = NULL,
  height = 2,
  overlay_previous = "no",
  where = "bottom",
  fontsize = 15
)

```

Arguments

title	String. If specified, the title of the track to be displayed.
height	Numeric. The height of the plotted track in cm. Default is 2. See notes.
overlay_previous	String. Options are "no" (default) or "yes" or "share-y".
where	String. Either "bottom" (default) or "top"
fontsize	Numeric value to font size of tracks's text.

Value

genome_track

Note

fontsize argument can be overridden by the same argument in plot_gtracks()

Author(s)

Omar Elashkar

Examples

```
tads_dir <- system.file("extdata", "tad_classification.bed",
  package = "rGenomeTracks"
)
tads <- track_domains(
  file = tads_dir, border_color = "black",
  color = "#11FF34", height = 5
)
tads_i <- track_domains(
  file = tads_dir, border_color = "red",
  color = "#cccccc", height = 3, orientation = "inverted"
)
tracks <- track_x_axis(where = "top") +
  tads + tads_i
## Not run:
plot_gtracks(tracks, chr = "X", start = 30 * 10^5, end = 35 * 10^5)

## End(Not run)
```

Index

- * **epilogos_json**
 - track_epilogos, [19](#)
- * **install_pyGenomeTracks**
 - plot_gtracks, [5](#)
- * **plot_gtracks**
 - install_pyGenomeTracks, [4](#)
- * **track_epilogos**
 - epilogos_json, [3](#)
- + ,genome_track,genome_track-method, [2](#)

- epilogos_json, [3](#)

- genome_track (plot_gtracks), [5](#)

- install_pyGenomeTracks, [4](#)

- plot_gtracks, [5](#)
- plot_gtracks, (plot_gtracks), [5](#)
- plot_gtracks,genome_track-method
 (plot_gtracks), [5](#)

- track_bed, [8](#)
- track_bedgraph, [11](#)
- track_bedgraph_matrix, [13](#)
- track_bigwig, [15](#)
- track_domains, [17](#)
- track_epilogos, [19](#)
- track_gtf, [21](#)
- track_hic_matrix, [23](#)
- track_hlines, [25](#)
- track_links, [27](#)
- track_narrow_peak, [29](#)
- track_scalebar, [32](#)
- track_spacer, [33](#)
- track_vlines, [35](#)
- track_x_axis, [36](#)