Package ‘ggcoverage’

September 6, 2022

Type Package

Title Visualize Genome Coverage with Various Annotations

Version 0.7.1

Maintainer Yabing Song `<songyb0519@gmail.com>`

Description The goal of 'ggcoverage' is to simplify the process of visualizing genome coverage. It contains functions to
load data from BAM, BigWig or BedGraph files, create genome coverage plot, add various annotations to
the coverage plot, including base and amino acid annotation, GC annotation, gene annotation, transcript annotation, ideogram annotation and peak annotation.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

Imports dplyr, GenomicRanges, ggbio, ggh4x, ggplot2, ggrepel, grDevices, IRanges, magrittr, patchwork, RColorBrewer, rlang, Rsamtools, rtracklayer, scales, stats, utils, methods, GenomeInfoDb, S4Vectors, Biostrings, BSgenome, GenomicAlignments, reshape2, seqinr

Suggests rmarkdown, knitr, BiocStyle, htmltools, BSgenome.Hsapiens.UCSC.hg19, graphics

VignetteBuilder knitr

NeedsCompilation no

Author Yabing Song [aut, cre]

Repository CRAN

Date/Publication 2022-09-06 02:50:02 UTC

R topics documented:

FormatTrack *Prepare Input for Creating Coverage Plot.*

Description

Prepare Input for Creating Coverage Plot.

Usage

```
FormatTrack(
  data,
  region = "chr14:21,677,306-21,737,601",
  gtf.gr = NULL,
  gene.name = "HNRNPC",
  gene.name.type = c("gene_name", "gene_id"),
  extend = 2000
)
```

Arguments

`data`	Track dataframe loaded by `LoadTrackFile`.
`region`	Region used to create coverage plot, eg: chr14:21,677,306-21,737,601 or chr14:21,677,306. Default: NULL.
`gtf.gr`	Granges object of GTF, created with `import.gff`. Default: NULL.
`gene.name`	The name of gene. Default: HNRNPC.
`gene.name.type`	Gene name type (filed of `gtf.gr`), chosen from gene_name and gene_id. Default: gene_name.
`extend`	Extend length of `region`. Default: 2000.

Value

A dataframe.

---

geom_base *Add Base and Amino Acid Annotation to Coverage Plot.*

Description

Add Base and Amino Acid Annotation to Coverage Plot.

Usage

```
geom_base(
  bam.file,
  fa.file = NULL,
  bs.fa.seq = NULL,
  chr.split = "[[:space:]]",
  nuc.offset = -0.1,
  nuc.size = 4,
  nuc.padding = 0.05,
  nuc.padding.r = 0,
  nuc.color = c(A = "#ff2b08", C = "#009aff", G = "#ffb507", T = "#00bc0d"),
  guide.line = NULL,
  guide.line.color = "red",
  guide.line.type = "dashed",
  show.aa = TRUE,
  sens = "F",
  numcode = 1,
  NAstring = "X",
  ambiguous = FALSE,
  aa.color = c(D = "#FF0000", S = "#FF2400", T = "#E34234", G = "#FF8000", P =
   "#F28500", C = "#FFFF00", A = "#FDFF00", V = "#E3FF00", I = "#C0FF00", L = "#89318C",
    M = "#00FF00", F = "#50C878", Y = "#30D5C8", W = "#00FFFF", H = "#0F2CB3", R =
    "#0000FF", K = "#4b0082", N = "#800080", Q = "#FF00FF", E = "#8F00FF", `*` =
    "#FFC0CB"),
  aa.size = 4,
  aa.margin = 2,
  aa.height = 0.4,
  plot.space = 2.5,
  plot.height = 0.5
)
```

Arguments

`bam.file` BAM file.

`fa.file` Genome fasta file. Default: NULL.

`bs.fa.seq` BSgenome for species. Default: NULL.

`chr.split` Split between chromosome name and description in `fa.file`. Default: "[[:space:]]".

`nuc.offset` Offset of nucleotide to frequency plot. Default: -0.1.

`nuc.size` The size of nucleotide text. Default: 4.

`nuc.padding` Background padding of nucleotide annotation. Default: 0.05.

`nuc.padding.r` Radius of background padding. Default: 0.

`nuc.color` Color scheme for nucleotides. Default: "A": "#ff2b08", "C": "#009aff", "G": "#ffb507", "T": "#00bc0d".

`guide.line` Nucleotide frequency guide line. Default: NULL (0.5).

`guide.line.color`
The color of guide line. Default: "red".

`guide.line.type`
The line type of guide line. Default: "dashed".

`show.aa` Logical value, whether to show amino acid. Default: TRUE.

`sens` Sense to translate: F for forward sense and R for reverse sense. Parameter of `translate`. Default: F.

`numcode` The ncbi genetic code number for translation. Parameter of `translate`. By default the standard genetic code is used.

`NAstring` How to translate amino-acids when there are ambiguous bases in codons. Parameter of `translate`. Default: X.

`ambiguous` If TRUE, ambiguous bases are taken into account so that for instance GGN is translated to Gly in the standard genetic code. Parameter of `translate`. Default: FALSE.

`aa.color` Color scheme for amino acids.

`aa.size` The size of amino acid text. Default: 4.

`aa.margin` Top and bottom margin of amino acids. Default: 2.

`aa.height` The relative height of amino acid to base frequency plot. Default: 0.4.

`plot.space` Top and bottom margin. Default: 2.5.

`plot.height` The relative height of base and amino acid annotation to coverage plot. Default: 0.5.

Value

Plot.

Examples

```
library(ggcoverage)
library("BSgenome.Hsapiens.UCSC.hg19")
# get sample metadata
sample.meta <- data.frame(
  SampleName = c("tumorA.chr4.selected"),
  Type = c("tumorA"), Group = c("tumorA")
)
```

```
# get bam file
bam.file <- system.file("extdata", "DNA-seq", "tumorA.chr4.selected.bam", package = "ggcoverage")
# load bam file
track.df <- LoadTrackFile(
  track.file = bam.file,
  meta.info = sample.meta, single.nuc = TRUE,
  single.nuc.region = "chr4:62474235-62474295"
)
ggcoverage(
  data = track.df, color = "grey", range.position = "out",
  single.nuc = TRUE, rect.color = "white"
) +
  geom_base(
    bam.file = bam.file,
    bs.fa.seq = BSgenome.Hsapiens.UCSC.hg19
  )
```

geom_coverage *Layer for Coverage Plot.*

Description

Layer for Coverage Plot.

Usage

```
geom_coverage(
  data,
  mapping = NULL,
  color = NULL,
  rect.color = NA,
  facet.key = "Type",
  facet.order = NULL,
  facet.color = NULL,
  group.key = "Group",
  range.size = 3,
  range.position = c("in", "out"),
  mark.region = NULL,
  mark.color = "grey",
  mark.alpha = 0.5,
  show.mark.label = TRUE,
  mark.label.size = 4
)
```

Arguments

`data` Track prepared by `FormatTrack`.

`mapping` Set of aesthetic mappings created by `aes` or `aes_`. Default: NULL.

`color` Track color. Default: NULL (select automatically).

`rect.color` The color of every bin. Default: NA.

`facet.key` Sample type key to create coverage plot. Default: Type.

`facet.order` The order of coverage plot. Default: NULL.

`facet.color` The color of sample text. Default: NULL (select automatically).

`group.key` Group of samples. Default: NULL.

`range.size` The label size of range text, used when `range.position` is in. Default: 3.

`range.position` The position of y axis range, chosen from in (move y axis in the plot) and out (normal y axis). Default: in.

`mark.region` Mark region on the plot. Default: NULL.

`mark.color` The color of marked region. Default: "grey".

`mark.alpha` The alpha of marked region. Default: 0.5.

`show.mark.label` Logical value, whether to show mark label (use label column in `mark.region`). Default: TRUE.

`mark.label.size` The label size of mark label. Default: 4.

Value

Layers of ggplot2.

Examples

```
library(ggcoverage)
library(utils)
library(ggplot2)
meta.file <- system.file("extdata", "RNA-seq", "meta_info.csv", package = "ggcoverage")
sample.meta <- utils::read.csv(meta.file)
# track folder
track.folder <- system.file("extdata", "RNA-seq", package = "ggcoverage")
# load bigwig file
track.df <- LoadTrackFile(
  track.folder = track.folder, format = "bw",
  meta.info = sample.meta
)
ggplot() +
  geom_coverage(data = track.df, color = "auto", mark.region = NULL)
```

geom_gc *Add GC Content Annotation to Coverage Plot.*

Description

Add GC Content Annotation to Coverage Plot.

Usage

```
geom_gc(
  fa.file = NULL,
  bs.fa.seq = NULL,
  chr.split = "[[:space:]]",
  guide.line = NULL,
  line.color = "black",
  guide.line.color = "red",
  guide.line.type = "dashed",
  plot.space = 0.1,
  plot.height = 0.2
)
```

Arguments

`fa.file`	Genome fasta file. Default: NULL.
`bs.fa.seq`	BSgenome for species. Default: NULL.
`chr.split`	Split between chromosome name and description in `fa.file`. Default: "[[:space:]]".
`guide.line`	GC content guide line. Default: NULL (use mean GC content).
`line.color`	GC line color. Default: "black".
`guide.line.color`	The color of guide line. Default: "red".
`guide.line.type`	The line type of guide line. Default: "dashed".
`plot.space`	Top and bottom margin. Default: 0.1.
`plot.height`	The relative height of GC content annotation to coverage plot. Default: 0.2.

Value

Plot.

Examples

```
library(ggcoverage)
library(utils)
library(rtracklayer)
library("BSgenome.Hsapiens.UCSC.hg19")
# track folder
```

```
track.file <- system.file("extdata", "DNA-seq", "CNV_example.txt", package = "ggcoverage")
track.df <- utils::read.table(track.file, header = TRUE)
gtf.file <- system.file("extdata", "used_hg19.gtf", package = "ggcoverage")
gtf.gr <- rtracklayer::import.gff(con = gtf.file, format = "gtf")
basic.coverage <- ggcoverage(
  data = track.df, color = NULL, mark.region = NULL,
  region = "chr4:61750000-62,700,000", range.position = "out"
)
basic.coverage + geom_gc(bs.fa.seq = BSgenome.Hsapiens.UCSC.hg19)
```

geom_gene *Add Gene Annotation to Coverage Plot.*

Description

Add Gene Annotation to Coverage Plot.

Usage

```
geom_gene(
  gtf.gr,
  overlap.gene.gap = 0.1,
  gene.size = 1,
  utr.size = 2,
  exon.size = 4,
  arrow.size = 1,
  color.by = "strand",
  fill.color = c(`-` = "darkblue", `+` = "darkgreen"),
  show.utr = TRUE,
  arrow.gap = NULL,
  arrow.num = 50,
  arrow.length = 0.06,
  label.size = 3,
  label.vjust = 2,
  plot.space = 0.1,
  plot.height = 0.2
)
```

Arguments

`gtf.gr` Granges object of GTF, created with `import.gff`. Default: NULL.

`overlap.gene.gap` The gap between gene groups. Default: 0.1.

`gene.size` The line size of gene. Default: 1.

`utr.size` The line size of UTR. Default: 2.

`exon.size` The line size of exon. Default: 4.

`arrow.size` The line size of arrow. Default: 1.

`color.by` Color the line by. Default: strand.

`fill.color` Color used for `color.by`. Default: darkblue for - (minus strand), darkgreen for + (plus strand).

`show.utr` Logical value, whether to show UTR. Default: TRUE.

`arrow.gap` The gap distance between arrow. Default: NULL.

`arrow.num` Total arrow num of whole region. Default: 50.

`arrow.length` The length of arrow. Default: 0.06.

`label.size` The size of gene label. Default: 3.

`label.vjust` The vjust of gene label. Default: 2.

`plot.space` Top and bottom margin. Default: 0.1.

`plot.height` The relative height of gene annotation to coverage plot. Default: 0.2.

Value

Plot.

Examples

```
library(ggcoverage)
library(utils)
library(rtracklayer)
meta.file <- system.file("extdata", "RNA-seq", "meta_info.csv", package = "ggcoverage")
sample.meta <- utils::read.csv(meta.file)
# track folder
track.folder <- system.file("extdata", "RNA-seq", package = "ggcoverage")
# load bigwig file
track.df <- LoadTrackFile(
  track.folder = track.folder, format = "bw",
  meta.info = sample.meta
)
gtf.file <- system.file("extdata", "used_hg19.gtf", package = "ggcoverage")
gtf.gr <- rtracklayer::import.gff(con = gtf.file, format = "gtf")
basic.coverage <- ggcoverage(data = track.df, color = "auto", range.position = "out")
basic.coverage + geom_gene(gtf.gr = gtf.gr)
```

`geom_ideogram` *Add Ideogram Annotation to Coverage Plot.*

Description

Add Ideogram Annotation to Coverage Plot.

Usage

```
geom_ideogram(
  genome = "hg19",
  mark.color = "red",
  mark.alpha = 0.7,
  mark.line.size = 1,
  add.shadow = TRUE,
  shadow.color = "grey",
  shadow.alpha = 0.7,
  shadow.line.size = 1,
  plot.space = 0.1,
  plot.height = 0.1
)
```

Arguments

`genome` Single specie names, which must be one of the result from `ucscGenomes()$db`. If missing, will invoke a menu for users to choose from. Default: hg19.

`mark.color` The color to mark plot region on ideogram. Default: "red".

`mark.alpha` The alpha to mark plot region on ideogram. Default: 0.7.

`mark.line.size` The line size to mark plot region on ideogram. Default: 1.

`add.shadow` Logical value, whether to add shadow polygon. Default: TRUE.

`shadow.color` The color to fill shadow polygon. Default: grey.

`shadow.alpha` The alpha of shadow polygon. Default: 0.7.

`shadow.line.size` The line size of shadow polygon. Default: 1.

`plot.space` Top and bottom margin. Default: 0.1.

`plot.height` The relative height of ideogram annotation to coverage plot. Default: 0.2.

Value

Plot.

geom_peak *Add Peak Annotation to Coverage Plot.*

Description

Add Peak Annotation to Coverage Plot.

Usage

```
geom_peak(
  bed.file,
  peak.color = "black",
  peak.size = 5,
  plot.space = 0.1,
  plot.height = 0.1
)
```

Arguments

`bed.file`	The path to bed file.
`peak.color`	Peak color. Default: black.
`peak.size`	The line size of peak. Default: 5.
`plot.space`	Top and bottom margin. Default: 0.1.
`plot.height`	The relative height of peak annotation to coverage plot. Default: 0.2.

Value

Plot.

Examples

```
library(ggcoverage)
library(rtracklayer)
sample.meta <- data.frame(
  SampleName = c("Chr18_MCF7_ER_1", "Chr18_MCF7_ER_2", "Chr18_MCF7_ER_3", "Chr18_MCF7_input"),
  Type = c("MCF7_ER_1", "MCF7_ER_2", "MCF7_ER_3", "MCF7_input"),
  Group = c("IP", "IP", "IP", "Input")
)
# track folder
track.folder <- system.file("extdata", "ChIP-seq", package = "ggcoverage")
# load bigwig file
track.df <- LoadTrackFile(
  track.folder = track.folder, format = "bw",
  meta.info = sample.meta
)
gtf.file <- system.file("extdata", "used_hg19.gtf", package = "ggcoverage")
gtf.gr <- rtracklayer::import.gff(con = gtf.file, format = "gtf")
# create mark region
mark.region <- data.frame(start = c(76822533), end = c(76823743), label = c("Promoter"))
basic.coverage <- ggcoverage(
  data = track.df, color = "auto", region = "chr18:76822285-76900000",
  mark.region = mark.region, show.mark.label = FALSE
)
# get consensus peak file
peak.file <- system.file("extdata", "ChIP-seq", "consensus.peak", package = "ggcoverage")
basic.coverage + geom_gene(gtf.gr = gtf.gr) + geom_peak(bed.file = peak.file)
```

geom_transcript *Add Transcript Annotation to Coverage Plot.*

Description

Add Transcript Annotation to Coverage Plot.

Usage

```
geom_transcript(
  gtf.gr,
  gene.name = "HNRNPC",
  overlap.tx.gap = 0.1,
  tx.size = 1,
  utr.size = 2,
  exon.size = 4,
  arrow.size = 1,
  color.by = "strand",
  fill.color = c(`-` = "darkblue", `+` = "darkgreen"),
  arrow.gap = NULL,
  arrow.num = 50,
  arrow.length = 0.06,
  label.size = 3,
  label.vjust = 2,
  plot.space = 0.1,
  plot.height = 1
)
```

Arguments

`gtf.gr` Granges object of GTF, created with `import.gff`. Default: NULL.

`gene.name` Gene name of all transcripts. Default: HNRNPC.

`overlap.tx.gap` The gap between transcript groups. Default: 0.1.

`tx.size` The line size of transcript. Default: 1.

`utr.size` The line size of UTR. Default: 2.

`exon.size` The line size of exon. Default: 4.

`arrow.size` The line size of arrow. Default: 1.

`color.by` Color the line by. Default: strand.

`fill.color` Color used for `color.by`. Default: darkblue for - (minus strand), darkgreen for + (plus strand).

`arrow.gap` The gap distance between arrow. Default: NULL.

`arrow.num` Total arrow num of whole region. Default: 50.

`arrow.length` The length of arrow. Default: 0.06.

`label.size`	The size of transcript label. Default: 3.
`label.vjust`	The vjust of transcript label. Default: 2.
`plot.space`	Top and bottom margin. Default: 0.1.
`plot.height`	The relative height of transcript annotation to coverage plot. Default: 0.2.

Value

Plot.

Examples

```
library(ggcoverage)
library(utils)
library(rtracklayer)
meta.file <- system.file("extdata", "RNA-seq", "meta_info.csv", package = "ggcoverage")
sample.meta <- utils::read.csv(meta.file)
# track folder
track.folder <- system.file("extdata", "RNA-seq", package = "ggcoverage")
# load bigwig file
track.df <- LoadTrackFile(
  track.folder = track.folder, format = "bw",
  meta.info = sample.meta
)
gtf.file <- system.file("extdata", "used_hg19.gtf", package = "ggcoverage")
gtf.gr <- rtracklayer::import.gff(con = gtf.file, format = "gtf")
basic.coverage <- ggcoverage(data = track.df, color = "auto", range.position = "out")
basic.coverage + geom_transcript(gtf.gr = gtf.gr, label.vjust = 1.5)
```

ggcoverage *Create Coverage Plot.*

Description

Create Coverage Plot.

Usage

```
ggcoverage(
  data,
  region = "chr14:21,677,306-21,737,601",
  gtf.gr = NULL,
  extend = 2000,
  gene.name = "HNRNPC",
  gene.name.type = c("gene_name", "gene_id"),
  single.nuc = FALSE,
  mapping = NULL,
  color = NULL,
  rect.color = NA,
  facet.key = "Type",
  facet.order = NULL,
  facet.color = NULL,
  group.key = "Group",
  range.size = 3,
  range.position = c("in", "out"),
  plot.space = 0.2,
  mark.region = NULL,
  mark.color = "grey",
  mark.alpha = 0.5,
  show.mark.label = TRUE,
  mark.label.size = 4
)
```

Arguments

`data` Coverage dataframe loaded by `LoadTrackFile`.

`region` Region used to create coverage plot, eg: chr14:21,677,306-21,737,601 or chr14:21,677,306. Default: NULL.

`gtf.gr` Granges object of GTF, created with `import.gff`. Default: NULL.

`extend` Extend length of `region`. Default: 2000.

`gene.name` The name of gene. Default: HNRNPC.

`gene.name.type` Gene name type (filed of `gtf.gr`), chosen from gene_name and gene_id. Default: gene_name.

`single.nuc` Logical value, whether to visualize at single nucleotide level. Default: FALSE.

`mapping` Set of aesthetic mappings created by `aes` or `aes_`. Default: NULL.

`color` Track color. Default: NULL (select automatically).

`rect.color` The color of every bin. Default: NA.

`facet.key` Sample type key to create coverage plot. Default: Type.

`facet.order` The order of Coverage plot. Default: NULL.

`facet.color` The color of sample text. Default: NULL (select automatically).

`group.key` Group of samples. Default: NULL.

`range.size` The label size of range text, used when `range.position` is in. Default: 3.

`range.position` The position of y axis range, chosen from in (move y axis in the plot) and out (normal y axis). Default: in.

`plot.space` The space between every facet. Default: 0.2.

`mark.region` Mark region on the plot. Default: NULL.

`mark.color` The color of marked region. Default: "grey".

`mark.alpha` The alpha of marked region. Default: 0.5.

`show.mark.label`
Logical value, whether to show mark label (use label column in `mark.region`). Default: TRUE.

`mark.label.size`
The label size of mark label. Default: 4.

Value

A ggplot2 object.

Examples

```
library(ggcoverage)
library(utils)
library(rtracklayer)
meta.file <- system.file("extdata", "RNA-seq", "meta_info.csv", package = "ggcoverage")
sample.meta <- utils::read.csv(meta.file)
# track folder
track.folder <- system.file("extdata", "RNA-seq", package = "ggcoverage")
# load bigwig file
track.df <- LoadTrackFile(
  track.folder = track.folder, format = "bw",
  meta.info = sample.meta
)
gtf.file <- system.file("extdata", "used_hg19.gtf", package = "ggcoverage")
gtf.gr <- rtracklayer::import.gff(con = gtf.file, format = "gtf")
ggcoverage(data = track.df, color = "auto", range.position = "out")
```

LoadTrackFile *Load Track File to Dataframe.*

Description

Load Track File to Dataframe.

Usage

```
LoadTrackFile(
  track.file,
  track.folder = NULL,
  format = c("bam", "wig", "bw", "bedgraph"),
  meta.info = NULL,
  meta.file = "",
  bamcoverage.path = NULL,
  norm.method = c("RPKM", "CPM", "BPM", "RPGC", "None"),
  single.nuc = FALSE,
  single.nuc.region = NULL,
  bin.size = 10,
  bc.extra.para = NULL
)
```

Arguments

`track.file` Track file, when `track.folder` is not NULL, determined by `track.folder`.

`track.folder` Track file folder. Default: NULL.

`format` Track file format, chosen from bam, wig, bw(bigwig), bedgraph(bedGraph).

`meta.info` Track file metadata. The columns should be: SampleName (`track.file` without suffix), Type (sample with replicates information), Group (sample group). when `meta.file` is not NULL, determined by `meta.file`.Default: NULL.

`meta.file` File contains track file metadata. Default: "".

`bamcoverage.path`
The path to `bamCoverage`, used when `format` is bam. Default: NULL (autodetect).

`norm.method` Methods to normalize the number of reads per bin, chosen from "RPKM", "CPM", "BPM", "RPGC", "None". Default: RPKM.

`single.nuc` Logical value, whether to visualize at single nucleotide level. Default: FALSE.

`single.nuc.region`
Region for `single.nuc`. Default: NULL

`bin.size` Size of the bins, in bases. Default: 50.

`bc.extra.para` Extra parameters for `bamCoverage`, eg: "–effectiveGenomeSize 2700000000 –ignoreForNormalization chrX"

Value

A dataframe.

Examples

```
library(ggcoverage)
sample.meta <- data.frame(
 SampleName = c("Chr18_MCF7_ER_1", "Chr18_MCF7_ER_2", "Chr18_MCF7_ER_3", "Chr18_MCF7_input"),
  Type = c("MCF7_ER_1", "MCF7_ER_2", "MCF7_ER_3", "MCF7_input"),
  Group = c("IP", "IP", "IP", "Input")
)
# track folder
track.folder <- system.file("extdata", "ChIP-seq", package = "ggcoverage")
# load bigwig file
track.df <- LoadTrackFile(
  track.folder = track.folder, format = "bw",
  meta.info = sample.meta
)
```

theme_aa *Theme for geom_base with Amino Acid.*

Description

Theme for geom_base with Amino Acid.

Usage

```
theme_aa(margin.len, fill.color)
```

Arguments

`margin.len`	Top and bottom margin.
`fill.color`	Fill color.

Value

List of layers.

theme_base *Theme for geom_base.*

Description

Theme for geom_base.

Usage

```
theme_base(margin.len, fill.color)
```

Arguments

`margin.len`	Top and bottom margin.
`fill.color`	Fill color.

Value

List of layers.

`theme_base2` *Theme for geom_base without margin.*

Description

Theme for geom_base without margin.

Usage

```
theme_base2(fill.color)
```

Arguments

`fill.color`	Fill color.

Value

List of layers.

`theme_coverage` *Theme for geom_coverage.*

Description

Theme for geom_coverage.

Usage

```
theme_coverage(space = 0.2, x.range)
```

Arguments

`space`	The space between facets. Default: 0.2.
`x.range`	X axis ranges.

Value

List of layers.

theme_coverage2 *Theme for geom_coverage.*

Description

Theme for geom_coverage.

Usage

```
theme_coverage2(space = 0.2, x.range)
```

Arguments

`space`	The space between facets. Default: 0.2.
`x.range`	X axis ranges.

Value

List of layers.

theme_gc *Theme for geom_gc.*

Description

Theme for geom_gc.

Usage

```
theme_gc(x.range, margin.len)
```

Arguments

`x.range`	X axis ranges.
`margin.len`	Top and bottom margin.

Value

List of layers.

theme_gene *Theme for geom_gene.*

Description

Theme for geom_gene.

Usage

```
theme_gene(overlap.gene.gap, group.num, fill.color, x.range, margin.len)
```

Arguments

`overlap.gene.gap`	The gap between gene groups.
`group.num`	The number of groups.
`fill.color`	Fill color.
`x.range`	X axis ranges.
`margin.len`	Top and bottom margin.

Value

List of layers.

theme_ideogram *Theme for geom_ideogram.*

Description

Theme for geom_ideogram.

Usage

```
theme_ideogram()
```

Value

List of layers.

theme_peak *Theme for geom_peak.*

Description

Theme for geom_peak.

Usage

```
theme_peak(margin.len, x.range)
```

Arguments

`margin.len`	Top and bottom margin.
`x.range`	X axis ranges.

Value

List of layers.

theme_transcript *Theme for geom_transcript.*

Description

Theme for geom_transcript.

Usage

```
theme_transcript(overlap.tx.gap, group.num, fill.color, x.range, margin.len)
```

Arguments

`overlap.tx.gap`	The gap between transcript groups.
`group.num`	The number of groups.
`fill.color`	Fill color.
`x.range`	X axis ranges.
`margin.len`	Top and bottom margin.

Value

List of layers.

Index