

nbin	number of bins.
xlab	x label.
ylab	y label.
main	title.
indName	when faceted by a RleList, name used for labeling faceted factor. Default is 'sample'.
geom	geometric types.
type	statistical summary method used within bins, shown as bar height or heatmap colors.

Value

a ggplot object.

Author(s)

Tengfei Yin

Examples

```
library(IRanges)
lambda <- c(rep(0.001, 4500), seq(0.001, 10, length = 500),
            seq(10, 0.001, length = 500))
xVector <- rpois(1e4, lambda)
xRle <- Rle(xVector)
xRleList <- RleList(xRle, 2L * xRle)

ggplot() + stat_bin(xRle)
ggplot(xRle) + stat_bin()
ggplot(xRle) + stat_bin(nbin = 100)
ggplot(xRle) + stat_bin(binwidth = 200)

p1 <- ggplot(xRle) + stat_bin(type = "viewMeans")
p2 <- ggplot(xRle) + stat_bin(type = "viewSums")
## y scale are different.
tracks(viewMeans = p1, viewSums = p2)

ggplot(xRle) + stat_bin(geom = "heatmap")
ggplot(xRle) + stat_bin(nbin = 100, geom = "heatmap")
ggplot(xRle) + stat_bin(binwidth = 200, geom = "heatmap")

## for RleList
ggplot(xRleList) + stat_bin()
ggplot(xRleList) + stat_bin(nbin = 100)
ggplot(xRleList) + stat_bin(binwidth = 200)

p1 <- ggplot(xRleList) + stat_bin(type = "viewMeans")
p2 <- ggplot(xRleList) + stat_bin(type = "viewSums")
## y scale are different.
tracks(viewMeans = p1, viewSums = p2)

ggplot(xRleList) + stat_bin(geom = "heatmap")
ggplot(xRleList) + stat_bin(nbin = 100, geom = "heatmap")
ggplot(xRleList) + stat_bin(binwidth = 200, geom = "heatmap")
```

stat_coverage	<i>Calculate coverage</i>
---------------	---------------------------

Description

Calculate coverage.

Usage

```
# for GRanges
## S4 method for signature 'GRanges'
stat_coverage(data, ..., xlim, xlab, ylab, main,
              facets = NULL, geom = NULL)
# for GRangesList
## S4 method for signature 'GRangesList'
stat_coverage(data, ..., xlim, xlab, ylab, main,
              facets = NULL, geom = NULL)

# for Bamfile
## S4 method for signature 'BamFile'
stat_coverage(data, ..., maxBinSize = 2^14,
              xlim, which, xlab, ylab,
              main, facets = NULL, geom = NULL,
              method = c("estimate", "raw"),
              space.skip = 0.1, coord = c("linear", "genome"))
```

Arguments

data	A GRanges or data.frame object.
...	Extra parameters such as aes() passed to geom_rect, geom_alignment, or geom_segment.
xlim	Limits for x.
xlab	Label for x
ylab	Label for y
main	Title for plot.
facets	Faceting formula to use.
geom	The geometric object to use display the data.
maxBinSize	maxBinSize.
method	'estimate' for parsing estimated coverage(fast), 'raw' is slow and parse the accurate coverage.
which	GRanges which defines region to subset the results.
space.skip	used for coordinate genome, skip between chromosomes.
coord	coordinate system.

Value

A 'Layer'.

Author(s)

Tengfei Yin

Examples

```

library(ggbio)
## =====
##  simulated GRanges
## =====
set.seed(1)
N <- 1000
library(GenomicRanges)

gr <- GRanges(seqnames =
  sample(c("chr1", "chr2", "chr3"),
    size = N, replace = TRUE),
  IRanges(
    start = sample(1:300, size = N, replace = TRUE),
    width = sample(70:75, size = N, replace = TRUE)),
  strand = sample(c("+", "-", "*"), size = N,
    replace = TRUE),
  value = rnorm(N, 10, 3), score = rnorm(N, 100, 30),
  sample = sample(c("Normal", "Tumor"),
    size = N, replace = TRUE),
  pair = sample(letters, size = N,
    replace = TRUE))

ggplot(gr) + stat_coverage()
ggplot() + stat_coverage(gr)

ggplot(gr) + stat_coverage(geom = "point")
ggplot(gr) + stat_coverage(geom = "area")
ggplot(gr) + stat_coverage(mapping = aes(y = ..coverage..), geom = "bar")

ggplot(gr) + stat_coverage(mapping = aes(y = ..coverage..)) + geom_point()

## for bam file
## TBD

```

stat_gene

*Calculate gene structure***Description**

Calculate gene structure.

Usage

```

## S4 method for signature 'TxDb'
stat_gene(data, ...)

```

Arguments

data	A GRanges or data.frame object.
...	Extra parameters such as aes() passed to geom_alignment.

Value

A 'Layer'.

Author(s)

Tengfei Yin

See Also

[geom_alignment](#)

Examples

```
## Not run:
## loading package
## Deprecated
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
data(genesymbol, package = "biovizBase")
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene

## made a track comparing full/reduce stat.
p1 <- ggplot() + geom_alignment(txdb, which = genesymbol["RBM17"])
p1 <- ggplot() + stat_gene(txdb, which = genesymbol["RBM17"])
## or
p1 <- ggplot(txdb) + stat_gene(which = genesymbol["RBM17"])

p1 <- ggplot(txdb) + stat_gene(which = genesymbol["RBM17"])
p2 <- ggplot(txdb) + stat_gene(which = genesymbol["RBM17"], stat =
"reduce")
p2 <- ggplot(txdb) + stat_gene(which = genesymbol["RBM17"], stat = "reduce")
## ggplot(txdb) + geom_alignment(which = genesymbol["RBM17"]) + stat_reduce()
## ggplot(txdb) + geom_alignment(which = genesymbol["RBM17"])
tracks(full = p1, reduce = p2, heights = c(3, 1))

## change y labels
ggplot(txdb) + stat_gene(which = genesymbol["RBM17"], names.expr =
"tx_id::gene_id")

## End(Not run)
```

stat_identity

Transform the data to a data.frame and for multiple geoms.

Description

Transform the data to a suitable data.frame and then one could use multiple geom or even stat to re-plot the data.

Usage

```
## S4 method for signature 'ANY'
stat_identity(data, ...)

## S4 method for signature 'GRanges'
stat_identity(data, ..., geom = NULL)

## S4 method for signature 'Rle'
stat_identity(data, ..., xlab, ylab, main, geom = NULL)

## S4 method for signature 'RleList'
stat_identity(data, ..., xlab, ylab, main,
              geom = NULL, indName = "sample")
```

Arguments

<code>data</code>	Typically a <code>GRanges</code> or <code>data.frame</code> object.
<code>...</code>	Extra parameters such as <code>aes()</code> passed to <code>geom_rect</code> , <code>geom_alignment</code> , or <code>geom_segment</code> .
<code>geom</code>	The geometric object to use display the data.
<code>xlab</code>	x label.
<code>ylab</code>	y label.
<code>main</code>	title of graphic..
<code>indName</code>	sample name.

Value

A 'Layer'.

Author(s)

Tengfei Yin

Examples

```
## load
set.seed(1)
N <- 50

require(GenomicRanges)
## simul
## =====
## simulated GRanges
## =====
gr <- GRanges(seqnames =
  sample(c("chr1", "chr2", "chr3"),
    size = N, replace = TRUE),
  IRanges(
    start = sample(1:300, size = N, replace = TRUE),
    width = sample(70:75, size = N, replace = TRUE)),
  strand = sample(c("+", "-", "*"), size = N,
    replace = TRUE),
  value = rnorm(N, 10, 3), score = rnorm(N, 100, 30),
  sample = sample(c("Normal", "Tumor"),
```

```

      size = N, replace = TRUE),
    pair = sample(letters, size = N,
      replace = TRUE))

## geom_point_start
ggplot() + stat_identity(gr, mapping = aes(x = start, y = value), geom = "point")
## or more formal
ggplot(gr) + stat_identity(mapping = aes(x = start, y = value), geom = "point")

## geom_point_midpoint
ggplot(gr) + stat_identity(mapping = aes(x = midpoint, y = value), geom = "point")

## geom_rect_all
ggplot(gr) + stat_identity(mapping = aes(xmin = start, xmax = end,
                                          ymin = value - 0.5, ymax = value + 0.5),
                          geom = "rect")

## geom_rect_y
ggplot(gr) + stat_identity(mapping = aes(y = value), geom = "rect")

## geom_line
ggplot(gr) + stat_identity(mapping = aes(x = start, y = value), geom = "line")

## geom_segment
ggplot(gr) + stat_identity(mapping = aes(y = value), geom = "segment")

## Rle/RleList
library(IRanges)
lambda <- c(rep(0.001, 4500), seq(0.001, 10, length = 500),
            seq(10, 0.001, length = 500))
xVector <- rpois(1e4, lambda)
xRle <- Rle(xVector)
xRleList <- RleList(xRle, 2L * xRle)

ggplot(xRle) + stat_identity(geom = "point")
ggplot(xRleList) + stat_identity(geom = "point")

```

stat_mismatch

Calculate mismatch summary

Description

Calculate mismatch summary

Usage

```

## for GRanges
## S4 method for signature 'GRanges'
stat_mismatch(data, ..., bsgenome,
              xlab, ylab, main,
              geom = c("segment", "bar"),
              show.coverage = TRUE)

## for BamFile
## S4 method for signature 'BamFile'

```

```
stat_mismatch(data, ..., bsgenome, which,
               xlab, ylab, main,
               geom = c("segment", "bar"),
               show.coverage = TRUE)
```

Arguments

data	A GRanges or BamFile object.
...	Extra parameters such as aes() passed to geom_rect, geom_alignment, or geom_segment.
bsgenome	BSgenome object.
which	GRanges object to subset the data.
xlab	Label for x
ylab	Label for y
main	Title for plot.
geom	The geometric object to use display the data.
show.coverage	whether to show coverage as background or not.

Value

A 'Layer'.

Author(s)

Tengfei Yin

stat_reduce	<i>Reduce an object.</i>
-------------	--------------------------

Description

Reduce GRanges, IRanges or TxDb object.

Usage

```
## S4 method for signature 'GRanges'
stat_reduce(data, ...,
            xlab, ylab, main,
            drop.empty.ranges = FALSE,
            min.gapwidth = 1L,
            facets = NULL, geom = NULL)

## S4 method for signature 'IRanges'
stat_reduce(data, ...,
            xlab, ylab, main,
            drop.empty.ranges = FALSE,
            min.gapwidth = 1L,
            with.inframe.attrib=FALSE,
            facets = NULL, geom = NULL)

## S4 method for signature 'TxDbOREnsDb'
stat_reduce(data, ...)
```

Arguments

<code>data</code>	GRanges, IRanges or TxDb object.
<code>...</code>	passed to aesthetics mapping.
<code>xlab</code>	x label.
<code>ylab</code>	y label.
<code>main</code>	title.
<code>drop.empty.ranges</code>	pass to reduce function.
<code>min.gapwidth</code>	pass to reduce function.
<code>with.inframe.attrib</code>	pass to reduce function.
<code>facets</code>	pass to reduce function.
<code>geom</code>	geometric type.

Value

a ggplot object.

Author(s)

Tengfei Yin

See Also

[reduce](#).

Examples

```
set.seed(1)
N <- 1000
library(GenomicRanges)

gr <- GRanges(seqnames =
  sample(c("chr1", "chr2", "chr3"),
    size = N, replace = TRUE),
  IRanges(
    start = sample(1:300, size = N, replace = TRUE),
    width = sample(70:75, size = N, replace = TRUE)),
  strand = sample(c("+", "-", "*"), size = N,
    replace = TRUE),
  value = rnorm(N, 10, 3), score = rnorm(N, 100, 30),
  sample = sample(c("Normal", "Tumor"),
    size = N, replace = TRUE),
  pair = sample(letters, size = N,
    replace = TRUE))

ggplot(gr) + stat_reduce()
autoplot(gr, stat = "reduce")
strand(gr) <- "*"
ggplot(gr) + stat_reduce()

library(TxDb.Hsapiens.UCSC.hg19.knownGene)
```

```
data(genesymbol, package = "biovizBase")
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
## made a track comparing full/reduce stat.
ggplot(txdb) + stat_reduce(which = genesymbol["RBM17"])
```

stat_slice

Slice Rle/RleList to view them as bar or heatmap.

Description

Slice Rle/RleList to different view by set lower or other parameters, then view summary for all those viewed region.

Usage

```
## S4 method for signature 'Rle'
stat_slice(data, ...,
           xlab, ylab, main,
           na.rm = FALSE,
           geom = NULL,
           lower=-Inf, upper=Inf,
           includeLower=TRUE, includeUpper=TRUE,
           rangesOnly = FALSE,
           type = c("viewSums", "viewMins",
                    "viewMaxs", "viewMeans"))

## S4 method for signature 'RleList'
stat_slice(data, ...,
           xlab, ylab, main,
           indName = "sample",
           na.rm = FALSE,
           geom = NULL,
           lower=-Inf, upper=Inf,
           includeLower=TRUE, includeUpper=TRUE,
           rangesOnly = FALSE,
           type = c("viewSums", "viewMins",
                    "viewMaxs", "viewMeans"))
```

Arguments

data	a data.frame or Rle or RleList object.
...	arguments passed to aesthetics mapping.
xlab	x label.
ylab	y label.
main	title.
indName	when faceted by a RleList, name used for labeling faceted factor. Default is 'sample'.
geom	geometric types.

type	statistical summary method used within bins, shown as bar height or heatmap colors.
na.rm	logical value, default FALSE, passed to function like viewMaxs for statistical summary computation.
lower	passed to slice .
upper	passed to slice .
includeLower	passed to slice .
includeUpper	passed to slice .
rangesOnly	passed to slice .

Value

a ggplot object.

Author(s)

Tengfei Yin

See Also

[slice](#)

Examples

```
library(IRanges)
lambda <- c(rep(0.001, 4500), seq(0.001, 10, length = 500),
            seq(10, 0.001, length = 500))
xVector <- rpois(1e4, lambda)
xRle <- Rle(xVector)
xRleList <- RleList(xRle, 2L * xRle)

ggplot(xRle) + stat_slice(lower = 5)
ggplot(xRle) + stat_slice(lower = 5, geom = "bar")
ggplot(xRle) + stat_slice(lower = 5, geom = "heatmap")

p1 <- ggplot(xRle) + stat_slice(type = "viewMeans", lower = 5,
                              geom = "bar")
p2 <- ggplot(xRle) + stat_slice(type = "viewSums", lower = 5,
                              geom = "bar")

## y scale are different.
tracks(viewMeans = p1, viewSums = p2)

ggplot(xRleList) + stat_slice(lower = 5)
ggplot(xRleList) + stat_slice(lower = 5, geom = "bar")
ggplot(xRleList) + stat_slice(lower = 5, geom = "heatmap")

p1 <- ggplot(xRleList) + stat_slice(type = "viewMeans", lower = 5,
                              geom = "bar")
p2 <- ggplot(xRleList) + stat_slice(type = "viewSums", lower = 5,
                              geom = "bar")

## y scale are different.
tracks(viewMeans = p1, viewSums = p2)
```

stat_stepping	<i>Calculate stepping levels</i>
---------------	----------------------------------

Description

Calculate stepping levels.

Usage

```
## S4 method for signature 'GRanges'
stat_stepping(data, ..., xlab, ylab, main,
               facets = NULL,
               geom = c("rect", "alignment", "segment"))
```

Arguments

data	A GRanges or data.frame object.
...	Extra parameters such as aes() passed to geom_rect, geom_alignment, or geom_segment.
xlab	Label for x
ylab	Label for y
main	Title for plot.
facets	Faceting formula to use.
geom	The geometric object used to display the data. For 'stepping', could be one of 'rect', 'alignment', 'segment'.

Value

A 'Layer'.

Author(s)

Tengfei Yin

Examples

```
set.seed(1)
N <- 50

require(GenomicRanges)
## simul
## =====
## simulated GRanges
## =====
gr <- GRanges(seqnames =
               sample(c("chr1", "chr2", "chr3"),
                     size = N, replace = TRUE),
               IRanges(
                 start = sample(1:300, size = N, replace = TRUE),
                 width = sample(70:75, size = N, replace = TRUE)),
               strand = sample(c("+", "-", "*"), size = N,
                              replace = TRUE),
```

```

value = rnorm(N, 10, 3), score = rnorm(N, 100, 30),
sample = sample(c("Normal", "Tumor"),
  size = N, replace = TRUE),
pair = sample(letters, size = N,
  replace = TRUE))

## default
ggplot(gr) + stat_stepping()
## or
ggplot() + stat_stepping(gr)

## facet_aes
ggplot(gr) + stat_stepping(mapping = aes(color = strand, fill = strand),
  facets = sample ~ seqnames)

## geom_segment
ggplot(gr) + stat_stepping(mapping = aes(color = strand),
  geom = "segment", xlab = "Genomic coord", ylab = "y", main = "hello")

## geom_alignment
## ggplot(gr) + stat_stepping(geom = "alignment")

## geom_alignment_group
## ggplot(gr) + stat_stepping(mapping = aes(group = pair),geom = "alignment")

```

stat_table	<i>Tabulate a GRanges object</i>
------------	----------------------------------

Description

Tabulate a GRanges object

Usage

```

## S4 method for signature 'GRanges'
stat_table(data, ..., xlab, ylab, main,
  geom = NULL, stat = NULL)
## S4 method for signature 'GRangesList'
stat_table(data, ..., xlab, ylab, main,
  facets = NULL, geom = NULL)

```

Arguments

data	A GRanges or data.frame object.
...	Extra parameters such as aes() passed to geom_rect, geom_alignment, or geom_segment.
xlab	Label for x
ylab	Label for y
main	Title for plot.
facets	Faceting formula to use.
geom	The geometric object to use display the data.
stat	The geometric object to use display the data.

Value

A 'Layer'.

Author(s)

Tengfei Yin

Examples

```
## load
set.seed(1)
N <- 100
require(ggbio)
require(GenomicRanges)
## simul
## =====
## simulated GRanges
## =====
gr <- GRanges(seqnames =
  sample(c("chr1", "chr2", "chr3"),
    size = N, replace = TRUE),
  IRanges(
    start = sample(1:300, size = N, replace = TRUE),
    width = sample(70:75, size = N, replace = TRUE)),
  strand = sample(c("+", "-", "*"), size = N,
    replace = TRUE),
  value = rnorm(N, 10, 3), score = rnorm(N, 100, 30),
  sample = sample(c("Normal", "Tumor"),
    size = N, replace = TRUE),
  pair = sample(letters, size = N,
    replace = TRUE))

gr <- c(gr[seqnames(gr) == "chr1"][sample(1:10, size = 1e4, replace = TRUE)],gr)

## default
ggplot(gr) + stat_table()
ggplot(gr) + stat_table(geom = "segment", mapping = aes(y = ..score.., color = ..score..))
ggplot(gr) + stat_table(mapping = aes(color = score))
```

theme

theme in ggbio

Description

Theme defined in ggbio for plot or tracks.

Usage

```
theme_null()
theme_noexpand()
theme_alignment(ylabel = FALSE, base_size = 12, base_family = "",
  axis = TRUE, border = TRUE, grid = TRUE)
theme_pack_panels(strip.bg = FALSE, strip.text.y = TRUE)
```

```

theme_clear(grid.y = FALSE, grid.x.minor = FALSE, grid.x.major = FALSE,
            panel.background.fill = "white", panel.border.color = NA,
            axis.ticks.x = FALSE, axis.ticks.y = TRUE, grid.color = "gray95",
            axis.line.color = "gray80")
theme_tracks_sunset(bg = "#fffedb", alpha = 1, ...)
theme_genome()

```

Arguments

<code>alpha</code>	alpha blending from 0(transparent) to 1(solid).
<code>axis</code>	logical value, show axis or not.
<code>axis.line.color</code>	color for axis line .
<code>axis.ticks.x</code>	show x ticks or not.
<code>axis.ticks.y</code>	show y ticks or not.
<code>base_family</code>	family for font.
<code>base_size</code>	size for font.
<code>bg</code>	background color for tracks.
<code>border</code>	logical value, show border or not.
<code>grid</code>	logical value, show background grid or not.
<code>grid.color</code>	grid line color.
<code>grid.x.major</code>	show x major grid line or not.
<code>grid.x.minor</code>	show x minor grid line or not.
<code>grid.y</code>	show y grid or not.
<code>panel.background.fill</code>	panel background fill color.
<code>panel.border.color</code>	panel border color.
<code>strip.bg</code>	if strip background is removed.
<code>strip.text.y</code>	if strip text is removed.
<code>ylabel</code>	logical value. Show labels or not.
<code>...</code>	passed to <code>theme_clear</code> .

Details

Themes speciall designed for tracks, are named following naming schema `theme_tracks_*`

Value

Return a theme.

Author(s)

Tengfei Yin

Examples

```
## load
library(ggbio)
p <- qplot(data = mtcars, x = mpg, y = wt, facets = cyl ~ .)
p + theme_null()
p + theme_clear()
p + theme_pack_panels()
p + theme_alignment()
p1 <- qplot(data = mtcars, x = mpg, y = wt)
tracks(p1 = p, p2 = p1)
tracks(p1 = p, p2 = p1) + theme_tracks_sunset()
```

Tracked	<i>Tracked class</i>
---------	----------------------

Description

Create a tracked object, designed for tracks function.

Usage

```
Tracked(mutable = TRUE, fixed = FALSE, labeled = TRUE,
        hasAxis = FALSE, bgColor = "white", height = unit(1, "null"))
```

Arguments

mutable	logical value, default TRUE. To control whether a track is updatable by applying + on it.
fixed	logical value, default FALSE. To control whether the scale response to a xlim change or not.
labeled	logical value, default TRUE. To control whether to label it all not.
hasAxis	logical value, default FALSE. To control whether to show axis for that track or not.
bgColor	character to control background color of a track.
height	unit, to control track height.

Value

a Tracked object.

Author(s)

Tengfei Yin

Description

`tracks` is a convenient constructor for binding graphics as tracks. You don't have to worry about adjusting different graphics, `tracks` did that for you. It's NOT just limited to bind genomic tracks, you can use this function to bind any tracks with the same definition of x axis, for example, sets of time series plots you made.

Tracks view is most common way to viewing genome features and annotation data and widely used by most genome browsers. Our assumption is that, most graphics you made with `ggbio` or by yourself using `ggplot2`, are almost always sitting on the genomic coordinates or the same x axis. And to compare annotation information along with genome features, we need to align those plots on exactly the same x axis in order to form your hypothesis. This function leaves you the flexibility to construct each tracks separately with worrying your alignments later.

Usage

```
tracks(..., heights, xlim, xlab = NULL, main = NULL,
       title = NULL, theme = NULL,
       track.plot.color = NULL,
       track.bg.color = NULL,
       main.height = unit(1.5, "lines"),
       scale.height = unit(1, "lines"),
       xlab.height = unit(1.5, "lines"),
       padding = unit(-1, "lines"),
       label.bg.color = "white",
       label.bg.fill = "gray80",
       label.text.color = "black",
       label.text.cex = 1,
       label.text.angle = 90,
       label.width = unit(2.5, "lines"))
```

Arguments

<code>...</code>	plots of class <code>ggplot</code> , generated from <code>ggplot2</code> or <code>ggbio</code> .
<code>heights</code>	numeric vector of the same length of passed graphic object to indicate the ratio of each track.
<code>xlim</code>	limits on x. could be IRanges , GRanges , numeric value
<code>xlab</code>	label for x axis.
<code>main</code>	title for the tracks.
<code>title</code>	title for the tracks, alias like <code>main</code> .
<code>theme</code>	theme object used for building tracks, this will set to default, which could be reseted later.
<code>track.plot.color</code>	Vector of characters of length 1 or the same length of passed plots, background color for each track, default is white.
<code>track.bg.color</code>	background color for the whole tracks.

<code>main.height</code>	unit. Height to control the title track height.
<code>scale.height</code>	unit. Height to control the scale track height.
<code>xlab.height</code>	unit. Height to control the xlab track height.
<code>padding</code>	single numeric value or unit, if numeric value, the unit would be "lines" by default.
<code>label.bg.color</code>	track labeling background rectangle border color.
<code>label.bg.fill</code>	track labeling background fill color.
<code>label.text.color</code>	track labeling text color.
<code>label.text.cex</code>	track labeling text size.
<code>label.text.angle</code>	angle to rotate the track labels.
<code>label.width</code>	track labeling size.

Details

tracks did following modification for passed plots.

- remove x-axis, ticks, xlab and title for each track and add scales at bottom. We suppose a new xlab and title would be provided by the tracks function for the whole tracks, but we still keep individual's y axis.
- align x-scale limits to make sure every plots sitting on exactly the same x scale.
- squeezing plots together to some extent.
- labeling tracks if names are provided, please check utilities section about labeled method.
- return a track object. This would allow many features introduced in this manual.

Value

A Tracks object.

Track class

constructor tracks will return a Tracks object, which has following slots.

`grobs` a `ggplotGrobList` object contains a list of `ggplot` object, which is our passed graphics.

`backup` a backup of all the slots for holding the original tracks, so users could edit it and reset it back at any time later, and `backup` method will reset the backedup copy.

`ylim` y limits for each plot.

`labeled` vector of logical value indicates whether a track is labeled or not, for labeled attributes please check utilities section.

`mutable` vector of logical value indicates whether a track is mutable for theme editing or not, for mutable attributes please check utilities section.

`hasAxis` vector of logical value indicates whether a track has axis or not, for `hasAxis` attributes please check utilities section.

`heights`, `xlim`, `xlab`, `main`, `title`, `theme`, `fixed`, `track.plot.color`, `track.bg.color`, `main.height`, `scale`. those slots are described in arguments section for constructor.

Utilities

Please check examples for usage.

`summary(object)` summary information about tracks object.

`fixed(x)`, `fixed(x) <- value` `x` is the ggplot object, this controls if a track has a fixed x scale or not, if the fixed attributes is TRUE, then when you pass this plot to a tracks, this plot won't be re-aligned with other tracks and will keep the original x-axis, this allow you to pass some plot like ideogram. `fixed` function will return a logical value

`labeled(x)`, `labeled(x) <- value` `x` is the ggplot object, if you pass named graphics into tracks, it will create the labels on the left for you. Several ways supported to name it. You can pass a list of graphics with names. Or you can use `tracks('name1' = p1, 'name 2' = p2, ...)` with quotes for complicated words or simply `tracks(part1 = p1, part = p2, ...)`.

`mutable(x)`, `mutable(x) <- value` `x` is the ggplot object, this controls whether a plot in the tracks mutable to theme changing or not, when you use `+` method for Tracks object, add-on edit will only be applied to the the mutable plots.

`bgColor(x)`, `bgColor(x) <- value` `x` is the ggplot object, this change the background color for single plot shown in the tracks.

`xlim(x)`, `xlim(x) <- value` when `x` is the numeric value, it calls `ggplot2::coord_cartesian(xlim = ...)` method, we doesn't use `ggplot2::xlim()` for the reason it will cut data outside the range, and we believe the best behavior would be zoom-in/out like most browser. when `x` is [IRanges](#), [GRanges](#), it get the range and passed to `ggplot2::coord_cartesian` function.

when `x` is Tracks object, `xlim(x)` will return `x` limits for that tracks. `xlim(x) <- value` replace method only works for Tracks object. `value` could be numeric, [IRanges](#), [GRanges](#) object. This will change the `x` limits associated with tracks.

`+ xlim(obj):obj` is the numeric range, or [IRanges](#), [GRanges](#) object.

`+ coord_cartesian()`: please read manual in `ggplot2`, this controls both `xlim` and `ylim`, only accept numerical range.

`+` The most nice features about [Tracks](#) object is the one inherited from `ggplot2`'s components additive features, with `+` method you can use any theme object and utilities in `ggplot2` package, to add them on a [Tracks](#) object, for example, if `x` is our [Tracks](#) object, `x + theme` would apply theme to any plots in the tracks except those are immutable.

`as(x, "grob")` Coerces a Tracks object to a grob for embedding in a larger figure.

Backup and reset

`reset(obj)` `obj` is the Tracks object, this reset the tracks back to original or backuped version.

`backup(obj)` `obj` is the Tracks object, this clear previous backup and use current setting for a new backup.

Author(s)

Tengfei Yin

See Also

[align.plots](#)

Examples

```

## make a simulated time series data set
df1 <- data.frame(time = 1:100, score = sin((1:100)/20)*10)
p1 <- qplot(data = df1, x = time, y = score, geom = "line")
df2 <- data.frame(time = 30:120, score = sin((30:120)/20)*10, value = rnorm(120-30 + 1))
p2 <- ggplot(data = df2, aes(x = time, y = score)) +
  geom_line() + geom_point(size = 4, aes(color = value))
## check p2
p1
## check p2
p2

## binding
tracks(p1, p2)

## or
tk1 <- tracks(p1, p2)
tk1

## combine
c(tk1, tk1)
tk1 + tk1

cbind(tk1, tk1)
rbind(tk1, tk1) ## different with c()!
library(grid)
x <- as(tk1, "grob")
grid.draw(cbind(x, x))

## labeling: default labeling a named graphic
## simply pass a name with it
tracks(time1 = p1, time2 = p2)
## or pass a named list with it
lst <- list(time1 = p1, time2 = p2)
tracks(lst)

## more complicated case please use quotes
tracks(time1 = p1, "second time" = p2)

## set heights
tracks(time1 = p1, time2 = p2, heights = c(1, 3))

## if you want to disable label arbitrarily
## default label is always TRUE
labeled(p2)
labeled(p2) <- FALSE
## set labeled to FALSE, remove label even the plot has a name
tracks(time1 = p1, time2 = p2)
labeled(p2) <- TRUE

## fix a plot, not synchronize with other plots
p3 <- p1
## default is always FALSE
fixed(p3)
## set to TRUE

```

```

fixed(p3) <- TRUE
fixed(p3)

tracks(time1 = p1, time2 = p2, "time3(fixed)" = p3)

fixed(p3) <- FALSE
## otherwise you could run

## control axis
hasAxis(p1)
hasAxis(p1) <- TRUE
# ready for weird looking
tracks(time1 = p1, time2 = p2)
# set it back
hasAxis(p1) <- FALSE

## mutable
mutable(p1)
tracks(time1 = p1, time2 = p2) + theme_bw()
mutable(p1) <- FALSE
# mutable for "+" method
tracks(time1 = p1, time2 = p2) + theme_bw()
mutable(p1) <- TRUE

## bgColor
bgColor(p1)
tracks(time1 = p1, time2 = p2)
bgColor(p1) <- "brown"
# mutable for "+" method
tracks(time1 = p1, time2 = p2)
# set it back
bgColor(p1) <- "white"

## apply a theme to each track
tks <- tracks(time1 = p1, time2 = p2) + theme_bw()
tks
reset(tks)

## store it with tracks
tks <- tracks(time1 = p1, time2 = p2, theme = theme_bw())
tks
tks <- tks + theme_gray()
tks
## reset will be introduced later
reset(tks)

## apply a pre-defined theme for tracks!
tracks(time1 = p1, time2 = p2) + theme_tracks_sunset()
tracks(p1, p2) + theme_tracks_sunset()

## change limits
tracks(time1 = p1, time2 = p2) + xlim(c(1, 40))
tracks(time1 = p1, time2 = p2) + xlim(1, 40)

```

```

tracks(time1 = p1, time2 = p2) + coord_cartesian(xlim = c(1, 40))
# change y
tracks(time1 = p1, time2 = p2) + xlim(1, 40) + ylim(0, 10)
library(GenomicRanges)
gr <- GRanges("chr", IRanges(1, 40))
# GRanges
tracks(time1 = p1, time2 = p2) + xlim(gr)
# IRanges
tracks(time1 = p1, time2 = p2) + xlim(ranges(gr))
tks <- tracks(time1 = p1, time2 = p2)
xlim(tks)
xlim(tks) <- c(1, 35)
xlim(tks) <- gr
xlim(tks) <- ranges(gr)

## xlab, title
tracks(time1 = p1, time2 = p2, xlab = "time")
tracks(time1 = p1, time2 = p2, main = "title")
tracks(time1 = p1, time2 = p2, title = "title")
tracks(time1 = p1, time2 = p2, xlab = "time", title = "title") + theme_tracks_sunset()

## backup and restore
tks <- tracks(time1 = p1, time2 = p2)
tks
tks <- tks + xlim(1, 40)
tks
reset(tks)
tks <- tks + xlim(1, 40)
tks
tks <- backup(tks)
tks <- tks + theme_bw()
tks
reset(tks)

## padding(need to be fixed for more delicate control)
tracks(time1 = p1, time2 = p2, padding = 2)

## track color
tracks(time1 = p1, time2 = p2, track.bg.color = "yellow")
tracks(time1 = p1, time2 = p2, track.plot.color = c("yellow", "brown"))

```

zoom

Simple navigation for ggbio object.

Description

A set of simple navigation API apply to ggbio object, let you move along the genome and zoom in/out.

Usage

```
zoom(fac = 1/2)
```

```
zoom_in(fac = 1/2)
zoom_out(fac = 2)
nextView(unit = c("view", "gene", "exon", "utr"))
prevView(unit = c("view", "gene", "exon", "utr"))
```

Arguments

fac	numeric value to indicate zoom factor, multiple of current view width. If it's smaller than 1, then it's zoom-in operation; if it's bigger than 1, then it's zoom-out operation.
unit	only support 'view' unit now.

Details

zoom_in and zoom_out are just simple wrapper around zoom function.

For more convenient, gene features based jumping we will support it in the future.

Value

A special class of navigation.

Author(s)

Tengfei Yin

Index

- [+, Bioplot, Any-method \(autoplot\), 4](#)
- [+, GGBio, ANY-method \(GGBio\), 36](#)
- [+, Ideogram, ANY-method \(Ideogram\), 42](#)
- [\[, PlotList, numeric, missing, ANY-method \(tracks\), 80](#)
- [\[, Tracks, numeric, missing, ANY-method \(tracks\), 80](#)
- [\\$, GGBio-method \(GGBio\), 36](#)
- [\\$<-, GGBio-method \(GGBio\), 36](#)
- [align.plots, 82](#)
- [align.plots \(tracks\), 80](#)
- [alignPlots \(tracks\), 80](#)
- [AnnotationFilter, 7, 20, 56](#)
- [AnnotationFilterList, 7, 20, 56](#)
- [Arith \(tracks\), 80](#)
- [Arith, Tracks, ANY-method \(tracks\), 80](#)
- [arrangeGrobByParsingLegend, 3](#)
- [autoplot, 4](#)
- [autoplot, BamFile-method \(autoplot\), 4](#)
- [autoplot, BamFileList-method \(autoplot\), 4](#)
- [autoplot, BSgenome-method \(autoplot\), 4](#)
- [autoplot, character-method \(autoplot\), 4](#)
- [autoplot, ExpressionSet-method \(autoplot\), 4](#)
- [autoplot, GAlignments-method \(autoplot\), 4](#)
- [autoplot, GRanges-method \(autoplot\), 4](#)
- [autoplot, GRangesList-method \(autoplot\), 4](#)
- [autoplot, IRanges-method \(autoplot\), 4](#)
- [autoplot, matrix-method \(autoplot\), 4](#)
- [autoplot, OrganismDb-method \(autoplot\), 4](#)
- [autoplot, RangedSummarizedExperiment-method \(autoplot\), 4](#)
- [autoplot, Rle-method \(autoplot\), 4](#)
- [autoplot, RleList-method \(autoplot\), 4](#)
- [autoplot, Seqinfo-method \(autoplot\), 4](#)
- [autoplot, TabixFile-method \(autoplot\), 4](#)
- [autoplot, TxDbOREnsDb-method \(autoplot\), 4](#)
- [autoplot, VCF-method \(autoplot\), 4](#)
- [autoplot, Views-method \(autoplot\), 4](#)
- [autoplot, VRanges-method \(autoplot\), 4](#)
- [backup \(tracks\), 80](#)
- [backup, Tracks-method \(tracks\), 80](#)
- [bgColor \(tracks\), 80](#)
- [bgColor, gg-method \(tracks\), 80](#)
- [bgColor, GGBio-method \(tracks\), 80](#)
- [bgColor, gtable-method \(tracks\), 80](#)
- [bgColor, Tracked-method \(tracks\), 80](#)
- [bgColor<- \(tracks\), 80](#)
- [bgColor<-, gg, character-method \(tracks\), 80](#)
- [bgColor<-, GGBio, character-method \(tracks\), 80](#)
- [bgColor<-, gtable, character-method \(tracks\), 80](#)
- [bgColor<-, Tracked, character-method \(tracks\), 80](#)
- [c, PlotList-method \(tracks\), 80](#)
- [c, Tracks-method \(tracks\), 80](#)
- [cbind, Tracks-method \(tracks\), 80](#)
- [circle \(layout_circle\), 43](#)
- [coerce, Tracks, grob-method \(tracks\), 80](#)
- [EnsDb, 8, 9, 56](#)
- [findOverlaps, 63](#)
- [fixed, gg-method \(tracks\), 80](#)
- [fixed, GGBio-method \(tracks\), 80](#)
- [fixed, Tracked-method \(tracks\), 80](#)
- [fixed<-, gg, logical-method \(tracks\), 80](#)
- [fixed<-, GGBio, logical-method \(tracks\), 80](#)
- [fixed<-, Tracked, logical-method \(tracks\), 80](#)
- [geom_alignment, 19, 68](#)
- [geom_alignment, BamFile-method \(geom_alignment\), 19](#)
- [geom_alignment, GRanges-method \(geom_alignment\), 19](#)
- [geom_alignment, GRangesList-method \(geom_alignment\), 19](#)

- geom_alignment,missing-method
(geom_alignment), 19
- geom_alignment,OrganismDb-method
(geom_alignment), 19
- geom_alignment,TxDbOREnsDb-method
(geom_alignment), 19
- geom_alignment,uneval-method
(geom_alignment), 19
- geom_arch, 23
- geom_arch,data.frame-method
(geom_arch), 23
- geom_arch,GRanges-method (geom_arch), 23
- geom_arch,missing-method (geom_arch), 23
- geom_arch,uneval-method (geom_arch), 23
- geom_arrow, 24
- geom_arrow,GRanges-method (geom_arrow),
24
- geom_arrow,missing-method (geom_arrow),
24
- geom_arrow,uneval-method (geom_arrow),
24
- geom_arrowrect, 26
- geom_arrowrect,GRanges-method
(geom_arrowrect), 26
- geom_arrowrect,missing-method
(geom_arrowrect), 26
- geom_arrowrect,uneval-method
(geom_arrowrect), 26
- geom_bar, 28
- geom_bar,ANY-method (geom_bar), 28
- geom_bar,chevron-method (geom_bar), 28
- geom_bar,GRanges-method (geom_bar), 28
- geom_bar,missing-method (geom_bar), 28
- geom_chevron, 29
- geom_chevron,GRanges-method
(geom_chevron), 29
- geom_chevron,missing-method
(geom_chevron), 29
- geom_chevron,uneval-method
(geom_chevron), 29
- geom_rect, 32
- geom_rect,ANY-method (geom_rect), 32
- geom_rect,GRanges-method (geom_rect), 32
- geom_rect,missing-method (geom_rect), 32
- geom_rect,uneval-method (geom_rect), 32
- geom_segment, 34
- geom_segment,ANY-method (geom_segment),
34
- geom_segment,GRanges-method
(geom_segment), 34
- geom_segment,missing-method
(geom_segment), 34
- geom_segment,uneval-method
(geom_segment), 34
- getIdeogram, 43
- GGBio, 36
- ggbio, 38
- ggbio (GGBio), 36
- GGBio-class (GGBio), 36
- ggbio-class (GGBio), 36
- ggbioPlot-class (Plot), 48
- ggplot, 36, 37
- ggplotGrob-class (Grob-class), 41
- ggplotPlot-class (Plot), 48
- ggsave, 40
- GRanges, 7, 8, 21, 80, 82
- Grob (Grob-class), 41
- Grob,gg-method (Grob-class), 41
- Grob,GGBio-method (Grob-class), 41
- Grob,gtable-method (Grob-class), 41
- Grob,lattice-method (Grob-class), 41
- Grob,trellis-method (Grob-class), 41
- Grob-class, 41
- Grob-method (Grob-class), 41
- GrobList (Grob-class), 41
- GrobList-class (Grob-class), 41
- hasAxis (tracks), 80
- hasAxis,gg-method (tracks), 80
- hasAxis,GGBio-method (tracks), 80
- hasAxis,Tracked-method (tracks), 80
- hasAxis<- (tracks), 80
- hasAxis<- ,gg,logical-method (tracks), 80
- hasAxis<- ,GGBio,logical-method
(tracks), 80
- hasAxis<- ,Tracked,logical-method
(tracks), 80
- height (tracks), 80
- height,gg-method (tracks), 80
- height,GGBio-method (tracks), 80
- height,Tracked-method (tracks), 80
- height<- (tracks), 80
- height<- ,gg,numericORunit-method
(tracks), 80
- height<- ,GGBio,numericORunit-method
(tracks), 80
- height<- ,Tracked,numericORunit-method
(tracks), 80
- Hits, 63
- Ideogram, 42
- Ideogram-class (Ideogram), 42
- IntegerList, 63
- IRanges, 80, 82

- labeled (tracks), 80
- labeled, gg-method (tracks), 80
- labeled, GGBio-method (tracks), 80
- labeled, gtable-method (tracks), 80
- labeled, gTree-method (tracks), 80
- labeled, Ideogram-method (tracks), 80
- labeled, text-method (tracks), 80
- labeled, Tracked-method (tracks), 80
- labeled<- (tracks), 80
- labeled<-, gg, logical-method (tracks), 80
- labeled<-, GGBio, logical-method (tracks), 80
- labeled<-, gtable, logical-method (tracks), 80
- labeled<-, Ideogram, logical-method (tracks), 80
- labeled<-, Tracked, logical-method (tracks), 80
- latticeGrob-class (Grob-class), 41
- latticePlot-class (Plot), 48
- layout_circle, 43
- layout_circle, GRanges-method (layout_circle), 43
- layout_circle, missing-method (layout_circle), 43
- layout_circle, uneval-method (layout_circle), 43
- layout_karyogram, 45
- layout_karyogram, GRanges-method (layout_karyogram), 45
- mold, 37, 38
- mutable (tracks), 80
- mutable, gg-method (tracks), 80
- mutable, GGBio-method (tracks), 80
- mutable, Tracked-method (tracks), 80
- mutable<- (tracks), 80
- mutable<-, gg, logical-method (tracks), 80
- mutable<-, GGBio, logical-method (tracks), 80
- mutable<-, Tracked, logical-method (tracks), 80
- nextView (zoom), 85
- Plot, 48
- Plot, gg-method (Plot), 48
- Plot, GGBio-method (Plot), 48
- Plot, Ideogram-method (Plot), 48
- Plot, trellis-method (Plot), 48
- Plot-class (Plot), 48
- plotFragLength, 49
- plotFragLength, character, GRanges-method (plotFragLength), 49
- plotGrandLinear, 50
- plotIdeogram (Ideogram), 42
- plotKaryogram (plotStackedOverview), 57
- plotRangesLinkedToData, 53
- plotRangesLinkedToData, GenomicRanges_OR_GRangesList-method (plotRangesLinkedToData), 53
- plotRangesLinkedToData, RangedSummarizedExperiment-method (plotRangesLinkedToData), 53
- plotSpliceSum, 55
- plotSpliceSum, character, EnsDb-method (plotSpliceSum), 55
- plotSpliceSum, character, GRangesList-method (plotSpliceSum), 55
- plotSpliceSum, character, TxDb-method (plotSpliceSum), 55
- plotStackedOverview, 57
- prevView (zoom), 85
- print (tracks), 80
- print, Tracks-method (tracks), 80
- qplot, 56
- rbind, Tracks-method (tracks), 80
- reduce, 72
- rescale, 59
- rescale, gg-method (rescale), 59
- rescale, ggplot-method (rescale), 59
- rescale, numeric-method (rescale), 59
- reset (tracks), 80
- reset, Tracks-method (tracks), 80
- scale_fill_fold_change, 60
- scale_fill_giensa, 61
- scale_x_sequint, 61
- ScanBamParam, 7
- show (tracks), 80
- show, Tracks-method (tracks), 80
- slice, 74
- stat_aggregate, 62
- stat_aggregate, GRanges-method (stat_aggregate), 62
- stat_aggregate, missing-method (stat_aggregate), 62
- stat_aggregate, uneval-method (stat_aggregate), 62
- stat_bin, 64
- stat_bin, ANY-method (stat_bin), 64
- stat_bin, missing-method (stat_bin), 64
- stat_bin, Rle-method (stat_bin), 64
- stat_bin, RleList-method (stat_bin), 64
- stat_bin, uneval-method (stat_bin), 64

- stat_coverage, 66
- stat_coverage, BamFile-method (stat_coverage), 66
- stat_coverage, GRanges-method (stat_coverage), 66
- stat_coverage, GRangesList-method (stat_coverage), 66
- stat_coverage, missing-method (stat_coverage), 66
- stat_coverage, uneval-method (stat_coverage), 66
- stat_gene, 67
- stat_gene, TxDb-method (stat_gene), 67
- stat_identity, 68
- stat_identity, ANY-method (stat_identity), 68
- stat_identity, GRanges-method (stat_identity), 68
- stat_identity, missing-method (stat_identity), 68
- stat_identity, Rle-method (stat_identity), 68
- stat_identity, RleList-method (stat_identity), 68
- stat_identity, uneval-method (stat_identity), 68
- stat_mismatch, 70
- stat_mismatch, BamFile-method (stat_mismatch), 70
- stat_mismatch, GRanges-method (stat_mismatch), 70
- stat_mismatch, missing-method (stat_mismatch), 70
- stat_mismatch, uneval-method (stat_mismatch), 70
- stat_reduce, 71
- stat_reduce, GRanges-method (stat_reduce), 71
- stat_reduce, IRanges-method (stat_reduce), 71
- stat_reduce, missing-method (stat_reduce), 71
- stat_reduce, TxDbOREnsDb-method (stat_reduce), 71
- stat_reduce, uneval-method (stat_reduce), 71
- stat_slice, 73
- stat_slice, missing-method (stat_slice), 73
- stat_slice, Rle-method (stat_slice), 73
- stat_slice, RleList-method (stat_slice), 73
- stat_slice, uneval-method (stat_slice), 73
- stat_stepping, 75
- stat_stepping, GRanges-method (stat_stepping), 75
- stat_stepping, missing-method (stat_stepping), 75
- stat_stepping, uneval-method (stat_stepping), 75
- stat_table, 76
- stat_table, GRanges-method (stat_table), 76
- stat_table, GRangesList-method (stat_table), 76
- stat_table, missing-method (stat_table), 76
- stat_table, uneval-method (stat_table), 76
- summary (tracks), 80
- summary, Tracks-method (tracks), 80
- theme, 77
- theme_alignment (theme), 77
- theme_clear (theme), 77
- theme_genome (theme), 77
- theme_noexpand (theme), 77
- theme_null (theme), 77
- theme_pack_panels (theme), 77
- theme_tracks_sunset (theme), 77
- Tracked, 79
- Tracked-class (Tracked), 79
- Tracks, 82
- tracks, 80
- Tracks-class (tracks), 80
- TxDb, 21
- xlim (tracks), 80
- xlim, GRanges-method (tracks), 80
- xlim, IRanges-method (tracks), 80
- xlim, numeric-method (tracks), 80
- xlim, Tracks-method (tracks), 80
- xlim<- (tracks), 80
- xlim<-, Tracks, GRanges-method (tracks), 80
- xlim<-, Tracks, IRanges-method (tracks), 80
- xlim<-, Tracks, numeric-method (tracks), 80
- zoom, 85
- zoom_in (zoom), 85
- zoom_out (zoom), 85