

# Package: trackViewer (via r-universe)

September 30, 2026

**Type** Package

**Title** A R/Bioconductor package with web interface for drawing elegant interactive tracks or lollipop plot to facilitate integrated analysis of multi-omics data

**Version** 1.49.3

**Maintainer** Jianhong Ou <jou@morgridge.org>

**Description** Visualize mapped reads along with annotation as track layers for NGS dataset such as ChIP-seq, RNA-seq, miRNA-seq, DNA-seq, SNPs and methylation data.

**License** GPL (>= 2)

**Depends** R (>= 3.5.0), grDevices, methods, GenomicRanges, grid

**Imports** Seqinfo, GenomeInfoDb, GenomicAlignments, GenomicFeatures, Rsamtools, S4Vectors, rtracklayer, BiocGenerics, scales, tools, IRanges, AnnotationDbi, grImport, htmlwidgets, InteractionSet, utils, rhdf5, strawr, txdbmaker

**Suggests** biomaRt, TxDb.Hsapiens.UCSC.hg19.knownGene, RUnit, org.Hs.eg.db, BiocStyle, knitr, VariantAnnotation, httr, htmltools, rmarkdown, motifStack, Gviz

**biocViews** Visualization

**VignetteBuilder** knitr

**Encoding** UTF-8

**Config/roxygen2/version** 8.1.0

**Config/pak/sysreqs** cmake make libbz2-dev libicu-dev liblzma-dev libpng-dev libuv1-dev libxml2-dev libssl-dev xz-utils zlib1g-dev

**Repository** <https://bioc.r-universe.dev>

**Date/Publication** 2026-09-29 19:32:11 UTC

**RemoteUrl** <https://github.com/bioc/trackViewer>

**RemoteRef** HEAD

**RemoteSha** 602809b79fda0f3f14b5280340f7fbb2110fc0cf

## Contents

|                             |    |
|-----------------------------|----|
| trackViewer-package         | 3  |
| addArrowMark                | 4  |
| addGuideLine                | 5  |
| addInteractionAnnotation    | 6  |
| ARA                         | 7  |
| browseTracks                | 8  |
| browseTracks-shiny          | 9  |
| coverageGR                  | 9  |
| dandelion.plot              | 10 |
| geneModelFromTxdb           | 11 |
| geneTrack                   | 13 |
| getCurTrackViewport         | 13 |
| getGeneIDsFromTxDb          | 14 |
| getLocation                 | 15 |
| gi2track                    | 15 |
| gieStain                    | 16 |
| Gloperator                  | 16 |
| gridPlot                    | 17 |
| GRoperator                  | 17 |
| ideogramPlot                | 18 |
| importBam                   | 19 |
| importData                  | 20 |
| importGInteractions         | 21 |
| importScore                 | 23 |
| importScSeqScore            | 24 |
| listChromosomes             | 25 |
| listNormalizations          | 26 |
| listResolutions             | 26 |
| loadIdeogram                | 27 |
| lollipop                    | 27 |
| optimizeStyle               | 30 |
| parse2GRanges               | 30 |
| parseWIG                    | 31 |
| plotGRanges                 | 32 |
| plotIdeo                    | 33 |
| plotOneIdeo                 | 33 |
| pos-class                   | 35 |
| reduce,GInteractions-method | 35 |
| trackList-class             | 36 |
| trackStyle-class            | 36 |
| trackViewerStyle-class      | 43 |
| viewGene                    | 46 |
| viewTracks                  | 47 |
| xscale-class                | 48 |
| yaxisStyle-class            | 49 |

---

|                     |                                                        |
|---------------------|--------------------------------------------------------|
| trackViewer-package | <i>Minimal designed plotting tool for genomic data</i> |
|---------------------|--------------------------------------------------------|

---

## Description

A package that plot data and annotation information along genomic coordinates in an elegance style. This tool is based on Gviz but want to draw figures in minimal style for publication.

## Author(s)

**Maintainer:** Jianhong Ou <jou@morgridge.org> ([ORCID](#))

Authors:

- Jianhong Ou <jou@morgridge.org> ([ORCID](#))
- Julie Lihua Zhu <Julie.Zhu@umassmed.edu>

## Examples

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(org.Hs.eg.db)
trs <- geneModelFromTxdb(TxDb.Hsapiens.UCSC.hg19.knownGene,
                        org.Hs.eg.db,
                        chrom="chr11",
                        start=122929275,
                        end=122930122)
extdata <- system.file("extdata", package="trackViewer",
                      mustWork=TRUE)
repA <- importScore(paste(extdata, "cpsf160.repA+.wig", sep="/"),
                  paste(extdata, "cpsf160.repA-.wig", sep="/"),
                  format="WIG")
strand(repA@dat) <- "+"
strand(repA@dat2) <- "-"
fox2 <- importScore(paste(extdata, "fox2.bed", sep="/"), format="BED")
dat <- coverageGR(fox2@dat)
fox2@dat <- dat[strand(dat)=="+"]
fox2@dat2 <- dat[strand(dat)=="-"]
gr <- GRanges("chr11", IRanges(122929275, 122930122), strand="-")
vp <- viewTracks(trackList(repA, fox2, trs), gr=gr, autoOptimizeStyle=TRUE)
addGuideLine(c(122929767, 122929969), vp=vp)
addArrowMark(list(x=unit(.5, "npc"),
                  y=unit(.39, "npc")),
              col="blue")
```

---

addArrowMark

Add arrow mark to the figure at a given position

---

### Description

A function to add arrow mark for emphasizing peaks

### Usage

```
addArrowMark(
  pos = grid.locator(),
  label = NULL,
  angle = 15,
  length = unit(0.25, "inches"),
  col = "red",
  cex = 1,
  quadrant = 4,
  type = "closed",
  vp = NULL
)
```

### Arguments

|          |                                                                                                                                                                                          |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| pos      | A unit object representing the location of arrow mark to be placed at current viewport. Default is the value of grid.locator, which will get the location of the mouse click.            |
| label    | A character or expression vector.                                                                                                                                                        |
| angle    | A parameter passed into grid::arrow function. The angle of arrow head in degrees (smaller numbers produce narrower, pointier arrows). Essentially describes the width of the arrow head. |
| length   | A parameter passed into grid::arrow function. A unit specifying the length of the arrow head.                                                                                            |
| col      | color of the arrow                                                                                                                                                                       |
| cex      | Multiplier applied to fontsize                                                                                                                                                           |
| quadrant | the direction of arrow, 1: to bottomleft, 2: to bottomright, 3: to topright, 4: to topleft                                                                                               |
| type     | A parameter passed into grid::arrow function. One of "open" or "closed" indicating whether the arrow head should be a closed triangle.                                                   |
| vp       | A Grid viewport object. It must be output of <a href="#">viewTracks</a>                                                                                                                  |

### Value

invisible x, y position value.

**See Also**

See Also as [addGuideLine](#), [arrow](#)

**Examples**

```
grid.newpage()
addArrowMark(list(x=unit(.5, "npc"),
                  y=unit(.5, "npc")),
              label="label1",
              col="blue")
## how to get the position by mouse click
if(interactive()){
  pos <- addArrowMark(label="byClick")
  addArrowMark(pos, label="samePosAsAbove")
}
```

---

|              |                                      |
|--------------|--------------------------------------|
| addGuideLine | <i>Add guide lines to the tracks</i> |
|--------------|--------------------------------------|

---

**Description**

A function to add lines for emphasizing the positions

**Usage**

```
addGuideLine(guideline, col = "gray", lty = "dashed", lwd = 1, vp = NULL)
```

**Arguments**

|           |                                                                         |
|-----------|-------------------------------------------------------------------------|
| guideline | The genomic coordinates to draw the lines                               |
| col       | A vector for the line color                                             |
| lty       | A vector for the line type                                              |
| lwd       | A vector for the line width                                             |
| vp        | A Grid viewport object. It must be output of <a href="#">viewTracks</a> |

**See Also**

See Also as [getCurTrackViewport](#), [addArrowMark](#), [viewTracks](#)

**Examples**

```
vp <- getCurTrackViewport(trackViewerStyle(), 10000, 10200)
addGuideLine(c(10010, 10025, 10150), vp=vp)
```

---

addInteractionAnnotation

*Add annotation markers to the figure at a given position*


---

## Description

A function to add annotation markers for emphasizing interactions

## Usage

```
addInteractionAnnotation(
  obj,
  idx,
  FUN = grid.polygon,
  panel = c("top", "bottom"),
  ...
)
```

## Arguments

|       |                                                                                                                                                                                                                                                                                   |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| obj   | A <a href="#">GInteractions</a> object, <a href="#">GRanges</a> object or numeric vector. For numeric vector, the positive value will generate a line with slope 1 and negative value will generate a line at the position with slope -1.                                         |
| idx   | The layer number of track.                                                                                                                                                                                                                                                        |
| FUN   | Function for plot. Available functions are <a href="#">grid.polygon</a> , <a href="#">grid.lines</a> , and <a href="#">grid.text</a> for GInteractions object; <a href="#">grid.lines</a> , and <a href="#">grid.text</a> for GRanges object; FUN is not used for numeric vector. |
| panel | Plot regions. Available values are "top", "bottom".                                                                                                                                                                                                                               |
| ...   | Parameters will be passed to FUN.                                                                                                                                                                                                                                                 |

## Value

invisible viewport for plot region.

## See Also

See Also as [addGuideLine](#), [addArrowMark](#)

## Examples

```
library(trackViewer)
library(InteractionSet)
gi <- readRDS(system.file("extdata", "nij.chr6.51120000.53200000.gi.rds",
  package="trackViewer"))
tads <- GInteractions(
  GRanges("chr6",
    IRanges(c(51130001, 51130001, 51450001, 52210001), width = 20000)),
```

```
GRanges("chr6",
        IRanges(c(51530001, 52170001, 52210001, 53210001), width = 20000)))
range <- GRanges("chr6", IRanges(51120000, 53200000))
tr <- gi2track(gi)
viewTracks(trackList(tr),
           gr=range, autoOptimizeStyle = TRUE)
addInteractionAnnotation(tads, "tr", grid.lines,
                        gp=gpar(col = "#E69F00", lwd=3, lty=3))
```

ARA

*Aggregate Region Analysis*

## Description

Extract the interaction signal means from given coordinates.

## Usage

```
ARA(gr, upstream = 2e+05, downstream = upstream, resolution = 10000, ...)
```

## Arguments

|                                   |                                                                                                                                                       |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>gr</code>                   | A ‘GRanges’ object. The center of the object will be used for alignment for all the given regions.                                                    |
| <code>upstream, downstream</code> | numeric(1L). Upstream and downstream from the center of given ‘gr’ input will be used to extract the signals.                                         |
| <code>resolution</code>           | numeric(1L). The resolution will be passed to <a href="#">importGInteractions</a> function.                                                           |
| <code>...</code>                  | The parameters used by <a href="#">importGInteractions</a> function. Please note that the ranges resolution and out parameter should not be involved. |

## Value

A [GInteractions](#) object with scores which represent the mean values of the interactions.

## Examples

```
hic <- system.file("extdata", "test_chr22.hic", package = "trackViewer",
                  mustWork=TRUE)
gr <- GRanges("22", c(seq(20000001, 50000001, by=10000000), width=1))
gi <- ARA(gr, file=hic, format="hic")
rg <- GRanges("22", IRanges(1, 400000))
op <- optimizeStyle(trackList(gi2track(gi)))
heatmap <- op$tracks
sty <- op$style
setTrackViewerStyleParam(sty, "xat", c(1, 200000, 400000))
setTrackViewerStyleParam(sty, "xlabel", c("-20K", "center", "20K"))
viewTracks(heatmap, viewerStyle=sty, gr=rg)
```

---

|              |                      |
|--------------|----------------------|
| browseTracks | <i>browse tracks</i> |
|--------------|----------------------|

---

## Description

browse tracks by a web browser.

## Usage

```
browseTracks(
  trackList,
  gr = GRanges(),
  ignore.strand = TRUE,
  width = NULL,
  height = NULL,
  ...
)
```

## Arguments

|               |                                                       |
|---------------|-------------------------------------------------------|
| trackList     | an object of <a href="#">trackList</a>                |
| gr            | an object of <a href="#">GRanges</a>                  |
| ignore.strand | ignore the strand or not when do filter. default TRUE |
| width         | width of the figure                                   |
| height        | height of the figure                                  |
| ...           | parameters not used                                   |

## Value

An object of class `htmlwidget` that will intelligently print itself into HTML in a variety of contexts including the R console, within R Markdown documents, and within Shiny output bindings.

## Examples

```
extdata <- system.file("extdata", package="trackViewer", mustWork=TRUE)
files <- dir(extdata, "-.wig")
tracks <- lapply(paste(extdata, files, sep="/"),
  importScore, format="WIG")
tracks <- lapply(tracks, function(.ele) {strand(.ele@dat) <- "-"; .ele})
names(tracks) <- c("trackA", "trackB")
fox2 <- importScore(paste(extdata, "fox2.bed", sep="/"), format="BED")
dat <- coverageGR(fox2@dat)
fox2@dat <- dat[strand(dat)=="+"]
fox2@dat2 <- dat[strand(dat)=="-"]
gr <- GRanges("chr11", IRanges(122929275, 122930122))
browseTracks(trackList(tracks, fox2), gr=gr)
```

---

|                    |                                        |
|--------------------|----------------------------------------|
| browseTracks-shiny | <i>Shiny bindings for browseTracks</i> |
|--------------------|----------------------------------------|

---

**Description**

Output and render functions for using browseTracks within Shiny applications and interactive Rmd documents.

**Usage**

```
browseTracksOutput(outputId, width = "100%", height = "600px")
```

```
renderbrowseTracks(expr, env = parent.frame(), quoted = FALSE)
```

**Arguments**

|               |                                                                                                                                |
|---------------|--------------------------------------------------------------------------------------------------------------------------------|
| outputId      | output variable to read from                                                                                                   |
| width, height | Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended. |
| expr          | An expression that generates a browseTracks                                                                                    |
| env           | The environment in which to evaluate expr.                                                                                     |
| quoted        | Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.                    |

---

|            |                           |
|------------|---------------------------|
| coverageGR | <i>calculate coverage</i> |
|------------|---------------------------|

---

**Description**

calculate coverage for [GRanges](#), [GAlignments](#) or [GAlignmentPairs](#)

**Usage**

```
coverageGR(gr)
```

**Arguments**

|    |                                                                                                       |
|----|-------------------------------------------------------------------------------------------------------|
| gr | an object of <a href="#">RGanges</a> , <a href="#">GAlignments</a> or <a href="#">GAlignmentPairs</a> |
|----|-------------------------------------------------------------------------------------------------------|

**Value**

an object of [GRanges](#)

**See Also**

See Also as [coverage](#), [coverage-methods](#)

**Examples**

```
bed <- system.file("extdata", "fox2.bed", package="trackViewer",
                  mustWork=TRUE)
fox2 <- importScore(bed)
fox2$dat <- coverageGR(fox2$dat)
```

---

dandelion.plot

*dandelion.plots*


---

**Description**

Plot variants and somatic mutations

**Usage**

```
dandelion.plot(
  SNP.gr,
  features = NULL,
  ranges = NULL,
  type = c("fan", "circle", "pie", "pin"),
  newpage = TRUE,
  ylab = TRUE,
  ylab.gp = gpar(col = "black"),
  xaxis = TRUE,
  xaxis.gp = gpar(col = "black"),
  yaxis = FALSE,
  yaxis.gp = gpar(col = "black"),
  legend = NULL,
  cex = 1,
  maxgaps = 1/50,
  heightMethod = NULL,
  label_on_feature = FALSE,
  ...
)
```

**Arguments**

|          |                                                                                                          |
|----------|----------------------------------------------------------------------------------------------------------|
| SNP.gr   | A object of <a href="#">GRanges</a> or <a href="#">GRangesList</a> . All the width of GRanges must be 1. |
| features | A object of <a href="#">GRanges</a> or <a href="#">GRangesList</a> .                                     |
| ranges   | A object of <a href="#">GRanges</a> or <a href="#">GRangesList</a> .                                     |
| type     | Character. Could be fan, circle, pie or pin.                                                             |

|                             |                                                                                                                                                                                                                                                                  |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| newpage                     | plot in the new page or not.                                                                                                                                                                                                                                     |
| ylab                        | plot ylab or not. If it is a character vector, the vector will be used as ylab.                                                                                                                                                                                  |
| ylab.gp, xaxis.gp, yaxis.gp | An object of class gpar for ylab, xaxis or yaxis.                                                                                                                                                                                                                |
| xaxis, yaxis                | plot xaxis/yaxis or not. If it is a numeric vector with length greater than 1, the vector will be used as the points at which tick-marks are to be drawn. And the names of the vector will be used to as labels to be placed at the tick points if it has names. |
| legend                      | If it is a list with named color vectors, a legend will be added.                                                                                                                                                                                                |
| cex                         | cex will control the size of circle.                                                                                                                                                                                                                             |
| maxgaps                     | maxgaps between the stem of dandelions. It is calculated by the width of plot region divided by maxgaps. If a GRanges object is set, the dandelions stem will be clustered in each genomic range.                                                                |
| heightMethod                | A function used to determine the height of stem of dandelion. eg. Mean. Default is length.                                                                                                                                                                       |
| label_on_feature            | Labels of the feature directly on them. Default FALSE.                                                                                                                                                                                                           |
| ...                         | not used.                                                                                                                                                                                                                                                        |

## Details

In SNP.gr and features, metadata of the GRanges object will be used to control the color, fill, border, height, data source of pie if the type is pie.

## Examples

```
SNP <- c(10, 100, 105, 108, 400, 410, 420, 600, 700, 805, 840, 1400, 1402)
SNP.gr <- GRanges("chr1", IRanges(SNP, width=1, names=paste0("snp", SNP)),
  score=sample.int(100, length(SNP))/100)
features <- GRanges("chr1", IRanges(c(1, 501, 1001),
  width=c(120, 500, 405),
  names=paste0("block", 1:3)),
  color="black",
  fill=c("#FF8833", "#51C6E6", "#DFA32D"),
  height=c(0.1, 0.05, 0.08))
dandelion.plot(SNP.gr, features, type="fan")
```

---

geneModelFromTxdb

*Prepare gene model from an object of TxDb*

---

## Description

Generate an object of [track](#) for [viewTracks](#) by given parameters.

**Usage**

```
geneModelFromTxdb(
  txdb,
  orgDb,
  gr,
  chrom,
  start,
  end,
  strand = c("*", "+", "-"),
  txdump = NULL
)
```

**Arguments**

|                     |                                                                                                                                 |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <code>txdb</code>   | An object of <a href="#">TxDb</a>                                                                                               |
| <code>orgDb</code>  | An object of "OrgDb"                                                                                                            |
| <code>gr</code>     | An object of GRanges.                                                                                                           |
| <code>chrom</code>  | chromosome name, must be a seqname of txdb                                                                                      |
| <code>start</code>  | start position                                                                                                                  |
| <code>end</code>    | end position                                                                                                                    |
| <code>strand</code> | strand                                                                                                                          |
| <code>txdump</code> | output of <code>as.list(txdb)</code> , a list of data frames that can be used to make the db again with no loss of information. |

**Value**

Generate a list of [track](#) from a TxDb object.

**See Also**

See Also as [importScore](#), [importBam](#), [viewTracks](#)

**Examples**

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(org.Hs.eg.db)
trs <- geneModelFromTxdb(TxDb.Hsapiens.UCSC.hg19.knownGene,
  org.Hs.eg.db,
  chrom="chr20",
  start=22560000,
  end=22565000,
  strand="-")
```

---

|           |                        |
|-----------|------------------------|
| geneTrack | <i>track from TxDb</i> |
|-----------|------------------------|

---

**Description**

Generate a track object from TxDb by given gene ids

**Usage**

```
geneTrack(ids, txdb, symbols, type = c("gene", "transcript"), asList = TRUE)
```

**Arguments**

|         |                                                             |
|---------|-------------------------------------------------------------|
| ids     | Gene IDs. A vector of character. It should be keys in txdb. |
| txdb    | An object of <a href="#">TxDb</a> .                         |
| symbols | symbol of genes.                                            |
| type    | Output type of track, "gene" or "transcript".               |
| asList  | Output a list of tracks or not. Default TRUE.               |

**Value**

An object of [track](#)

**Examples**

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(org.Hs.eg.db)
ids <- c("3312", "390259", "341056", "79827")
symbols <- mget(ids, org.Hs.egSYMBOL)
geneTrack(ids, TxDb.Hsapiens.UCSC.hg19.knownGene, symbols)
```

---

|                     |                                   |
|---------------------|-----------------------------------|
| getCurTrackViewport | <i>Get current track viewport</i> |
|---------------------|-----------------------------------|

---

**Description**

Get current track viewport for addGuideLine

**Usage**

```
getCurTrackViewport(curViewerStyle, start, end)
```

**Arguments**

curViewerStyle an object of [trackViewerStyle](#)  
start start position of current track  
end end position of current track

**Value**

an object of [viewport](#)

**See Also**

See Also as [addGuideLine](#)

**Examples**

```
vp <- getCurTrackViewport(trackViewerStyle(), 10000, 10200)
addGuideLine(c(10010, 10025, 10150), vp=vp)
```

---

|                    |                                         |
|--------------------|-----------------------------------------|
| getGeneIDsFromTxDb | <i>get gene ids by genomic location</i> |
|--------------------|-----------------------------------------|

---

**Description**

retrieve gene ids from txdb object by genomic location.

**Usage**

```
getGeneIDsFromTxDb(gr, txdb)
```

**Arguments**

gr GRanges object.  
txdb An object of [TxDb](#).

**Value**

A character vector of gene ids

**Examples**

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
gr <- parse2GRanges("chr11:122,830,799-123,116,707")
ids <- getGeneIDsFromTxDb(gr, TxDb.Hsapiens.UCSC.hg19.knownGene)
```

---

|             |                                            |
|-------------|--------------------------------------------|
| getLocation | <i>get genomic location by gene symbol</i> |
|-------------|--------------------------------------------|

---

**Description**

given a gene name, get the genomic coordinates.

**Usage**

```
getLocation(symbol, txdb, org)
```

**Arguments**

|        |                                        |
|--------|----------------------------------------|
| symbol | Gene symbol                            |
| txdb   | txdb will be used to extract the genes |
| org    | org package name                       |

**Examples**

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(org.Hs.eg.db)
getLocation("HSPA8", TxDb.Hsapiens.UCSC.hg19.knownGene, "org.Hs.eg.db")
```

---

|          |                                              |
|----------|----------------------------------------------|
| gi2track | <i>convert GInteractions to track object</i> |
|----------|----------------------------------------------|

---

**Description**

Convert GInteractions object to track object

**Usage**

```
gi2track(gi, gi2)
```

**Arguments**

|     |                            |
|-----|----------------------------|
| gi  | an object of GInteractions |
| gi2 | an object of GInteractions |

**Value**

an track object

**Examples**

```
gi <- readRDS(system.file("extdata", "nij.chr6.51120000.53200000.gi.rds", package="trackViewer"))
gi2track(gi)
```

---

|                       |                                                                   |
|-----------------------|-------------------------------------------------------------------|
| <code>gieStain</code> | <i>color scheme for the schema for Chromosome Band (Ideogram)</i> |
|-----------------------|-------------------------------------------------------------------|

---

**Description**

Describe the colors of giemsa stain results

**Usage**

`gieStain()`

**Value**

A character vector of colors

**Examples**

`gieStain()`

---

|                        |                               |
|------------------------|-------------------------------|
| <code>Goperator</code> | <i>GInteractions operator</i> |
|------------------------|-------------------------------|

---

**Description**

GInteractions operations (add, aubtract, multiply, divide)

**Usage**

`Goperator(gi_list, col = "score", operator = c("+", "-", "*", "/"))`

**Arguments**

- `gi_list`            a list of GInteractions objects
- `col`                colname of metadata to be calculated
- `operator`           operator, "+" means A + B, and so on. User-defined function also could be used.

**Value**

an object of GInteractions

**Examples**

```
library(InteractionSet)
gr2 <- GRanges(seqnames=c("chr1", "chr1"),
               ranges=IRanges(c(7,13), width=3))
gr3 <- GRanges(seqnames=c("chr1", "chr1"),
               ranges=IRanges(c(1, 4), c(3, 9)))
gi <- GInteractions(gr2, gr3, score=c(1, 2))
gi2 <- GInteractions(gr2, gr3, score=c(3, 4))
GIOperator(list(gi, gi2), col="score", operator="+")
GIOperator(list(gi, gi2), col="score", operator="-")
```

---

|          |                              |
|----------|------------------------------|
| gridPlot | <i>plot GRanges metadata</i> |
|----------|------------------------------|

---

**Description**

plot GRanges metadata for different types

**Usage**

```
gridPlot(gr, gp, type, xscale)
```

**Arguments**

|        |                                                                                   |
|--------|-----------------------------------------------------------------------------------|
| gr     | an object of <a href="#">GRanges</a> with metadata. All metadata must be numeric. |
| gp     | an object of <a href="#">gpar</a>                                                 |
| type   | type of the figure, could be barplot, line, point and heatmap                     |
| xscale | x scale of the viewport                                                           |

---

|            |                         |
|------------|-------------------------|
| GRoperator | <i>GRanges operator</i> |
|------------|-------------------------|

---

**Description**

GRanges operations (add, subtract, multiply, divide)

**Usage**

```
GRoperator(
  A,
  B,
  col = "score",
  operator = c("+", "-", "*", "/", "^", "%"),
  ignore.strand = TRUE
)
```

**Arguments**

|               |                                                                                  |
|---------------|----------------------------------------------------------------------------------|
| A             | an object of GRanges                                                             |
| B             | an object of GRanges                                                             |
| col           | colname of A and B to be calculated                                              |
| operator      | operator, "+" means A + B, and so on. User-defined function also could be used.  |
| ignore.strand | When set to TRUE, the strand information is ignored in the overlap calculations. |

**Value**

an object of GRanges

**Examples**

```
gr2 <- GRanges(seqnames=c("chr1", "chr1"),
  ranges=IRanges(c(7,13), width=3),
  strand=c("-", "-"), score=3:4)
gr3 <- GRanges(seqnames=c("chr1", "chr1"),
  ranges=IRanges(c(1, 4), c(3, 9)),
  strand=c("-", "-"), score=c(6L, 2L))
GRoperator(gr2, gr3, col="score", operator="+")
GRoperator(gr2, gr3, col="score", operator="-")
GRoperator(gr2, gr3, col="score", operator="*")
GRoperator(gr2, gr3, col="score", operator="/")
GRoperator(gr2, gr3, col="score", operator=mean)
```

---

ideogramPlot

*plot ideogram with data*


---

**Description**

plot ideogram with data for multiple chromosomes

**Usage**

```
ideogramPlot(
  ideo,
  dataList,
  layout = NULL,
  horiz = TRUE,
  parameterList = list(vp = plotViewport(margins = c(0.1, 4.1, 0.3, 0.1)), ideoHeight =
    unit(1/(1 + length(dataList)), "npc"), vgap = unit(0.3, "lines"), ylabs = "auto",
    ylabsRot = ifelse(horiz, 0, 90), ylabsPos = unit(2.5, "lines"), xaxis = FALSE, yaxis
    = FALSE, xlab = "", types = "barplot", heights = NULL, dataColumn = "score", gps =
    gpar(col = "black", fill = "gray")),
  colorScheme = gieStain(),
  gp = gpar(fill = NA, lwd = 2),
  ...
)
```

**Arguments**

|               |                                                                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ideo          | output of <a href="#">loadIdeogram</a> .                                                                                                                       |
| dataList      | a <a href="#">GRangesList</a> of data to plot.                                                                                                                 |
| layout        | The layout of chromosomes. Could be a list with chromosome names as its elements.                                                                              |
| horiz         | a logical value. If FALSE, the ideograms are drawn vertically to the left. If TRUE, the ideograms are drawn horizontally at the bottom.                        |
| parameterList | a list of parameters for each dataset in the dataList. The elements of the parameters could be xlabs, ylabs, etc. type could be barplot, line, point, heatmap. |
| colorScheme   | A character vector of giemsa stain colors.                                                                                                                     |
| gp            | parameters used for <a href="#">grid.roundrect</a> .                                                                                                           |
| ...           | parameters not used.                                                                                                                                           |

**Examples**

```
## Not run:
ideo <- loadIdeogram("hg38")
library(rtracklayer)
library(grid)
dataList <- ideo
dataList$score <- as.numeric(dataList$gieStain)
dataList <- dataList[dataList$gieStain!="gneg"]
dataList <- GRangesList(dataList)
grid.newpage()
ideogramPlot(ideo, dataList,
             layout=list("chr1", "chr2", c("chr3", "chr22"),
                          c("chr4", "chr21"), c("chr5", "chr20"),
                          c("chr6", "chr19"), c("chr7", "chr18"),
                          c("chr8", "chr17"), c("chr9", "chr16"),
                          c("chr10", "chr15"), c("chr11", "chr14"),
                          c("chr12", "chr13"), c("chrX", "chrY")),
             parameterList = list(types="heatmap", colorKeyTitle="sample1"))

## End(Not run)
```

importBam

*Reading data from a BAM file***Description**

Read a [track](#) object from a BAM file

**Usage**

```
importBam(file, file2, ranges = GRanges(), pairs = FALSE)
```

**Arguments**

|        |                                                                                          |
|--------|------------------------------------------------------------------------------------------|
| file   | The path to the BAM file to read.                                                        |
| file2  | The path to the second BAM file to read.                                                 |
| ranges | An object of <a href="#">GRanges</a> to indicate the range to be imported                |
| pairs  | logical object to indicate the BAM is paired or not. See <a href="#">readGAlignments</a> |

**Value**

a [track](#) object

**See Also**

See Also as [importScore](#), [track](#), [viewTracks](#)

**Examples**

```
bamfile <- system.file("extdata", "ex1.bam", package="Rsamtools",
mustWork=TRUE)
dat <- importBam(file=bamfile, ranges=GRanges("seq1", IRanges(1, 50), strand="+"))
```

---

|            |                                                       |
|------------|-------------------------------------------------------|
| importData | <i>Reading data from a BED or WIG file to RleList</i> |
|------------|-------------------------------------------------------|

---

**Description**

Read a [track](#) object from a BED, bedGraph, WIG or BigWig file to RleList

**Usage**

```
importData(files, format = NA, ranges = GRanges())
```

**Arguments**

|        |                                                                           |
|--------|---------------------------------------------------------------------------|
| files  | The path to the files to read.                                            |
| format | The format of import file. Could be BAM, BED, bedGraph, WIG or BigWig     |
| ranges | An object of <a href="#">GRanges</a> to indicate the range to be imported |

**Value**

a list of [RleList](#).

## Examples

```
#import a BED file
bedfile <- system.file("tests", "test.bed", package="rtracklayer",
                        mustWork=TRUE)
dat <- importData(files=bedfile, format="BED",
                  ranges=GRanges("chr7", IRanges(127471197, 127474697)))

##import a WIG file
wigfile <- system.file("tests", "step.wig", package = "rtracklayer",
                        mustWork=TRUE)
dat <- importData(files=wigfile, format="WIG",
                  ranges=GRanges("chr19",
                                IRanges(59104701, 59110920)))

##import a BigWig file
if(.Platform$OS.type!="windows"){
  ##this is because we are using rtracklayer::import
  bwfile <- system.file("tests", "test.bw", package = "rtracklayer",
                        mustWork=TRUE)
  dat <- importData(files=bwfile, format="BigWig",
                    ranges=GRanges("chr19", IRanges(1500, 2700)))
}
```

---

importGInteractions     *Reading data from a ginteractions, hic, cool, or validPairs file*

---

## Description

Read a [track](#) object from a ginteractions, hic, mcool, or validPairs file

## Usage

```
importGInteractions(
  file,
  format = c("ginteractions", "hic", "cool", "validPairs"),
  ranges = GRanges(),
  ignore.strand = TRUE,
  out = c("track", "GInteractions"),
  resolution = 1e+05,
  unit = c("BP", "FRAG"),
  normalization = c("NONE", "VC", "VC_SQRT", "KR", "SCALE", "GW_KR", "GW_SCALE", "GW_VC",
                    "INTER_KR", "INTER_SCALE", "INTER_VC", "balanced"),
  matrixType = c("observed", "oe", "expected"),
  ...
)
```

**Arguments**

|               |                                                                                                                                                                                                                 |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| file          | The path to the file to read.                                                                                                                                                                                   |
| format        | The format of import file. Could be ginteractions, hic, cool or validPairs                                                                                                                                      |
| ranges        | An object of <a href="#">GRanges</a> to indicate the range to be imported. For .hic file, if the length of ranges is 2, the first range will be used as anchor 1 and the second range will be used as anchor 2. |
| ignore.strand | ignore the strand or not when do filter. default TRUE                                                                                                                                                           |
| out           | output format. Default is track. Possible values: track, GInteractions.                                                                                                                                         |
| resolution    | Resolutions for the interaction data.                                                                                                                                                                           |
| unit          | BP (base pair) or FRAG (fragment) (.hic file only).                                                                                                                                                             |
| normalization | Type of normalization, NONE, VC, VC_SORT or KR for .hic and NONE, balanced for .cool.                                                                                                                           |
| matrixType    | Type of matrix for .hic file. Available choices are "observed", "oe", and "expected". default is "observed".                                                                                                    |
| ...           | NOT used.                                                                                                                                                                                                       |

**Value**

a [track](#) object

**See Also**

See Also as [listResolutions](#), [listChromosomes](#), [readHicNormTypes](#)

**Examples**

```
#import a ginteractions file
#gi <- system.file("extdata", "test.ginteractions.tsv", package="trackViewer",
#                  mustWork=TRUE)
#dat <- importGInteractions(file=gi, format="ginteractions",
#                           ranges=GRanges("chr7", IRanges(127471197, 127474697)))

##import a hic file
if(!.Platform$OS.type!="windows"){
  hic <- system.file("extdata", "test_chr22.hic", package = "trackViewer",
                    mustWork=TRUE)
  dat <- importGInteractions(file=hic, format="hic",
                           ranges=GRanges("22", IRanges(1500000, 100000000)))
}

##import a cool file
cool <- system.file("extdata", "test.mcool", package = "trackViewer",
                   mustWork=TRUE)
dat <- importGInteractions(file=cool, format="cool",
                          resolution = 2,
                          ranges=GRanges("chr1", IRanges(10, 28)))

##import a validPairs file
```

```
#validPairs <- system.file("extdata", "test.validPairs", package = "trackViewer",
#                           mustWork=TRUE)
#dat <- importGInteractions(file=validPairs, format="validPairs")
```

---

importScore

*Reading data from a BED or WIG file*


---

## Description

Read a [track](#) object from a BED, bedGraph, WIG or BigWig file

## Usage

```
importScore(
  file,
  file2,
  format = c("BED", "bedGraph", "WIG", "BigWig"),
  ranges = GRanges(),
  ignore.strand = TRUE
)
```

## Arguments

|               |                                                                           |
|---------------|---------------------------------------------------------------------------|
| file          | The path to the file to read.                                             |
| file2         | The path to the second file to read.                                      |
| format        | The format of import file. Could be BED, bedGraph, WIG or BigWig          |
| ranges        | An object of <a href="#">GRanges</a> to indicate the range to be imported |
| ignore.strand | ignore the strand or not when do filter. default TRUE                     |

## Value

a [track](#) object

## See Also

See Also as [importBam](#), [track](#), [viewTracks](#)

## Examples

```
#import a BED file
bedfile <- system.file("tests", "test.bed", package="rtracklayer",
                       mustWork=TRUE)
dat <- importScore(file=bedfile, format="BED",
                  ranges=GRanges("chr7", IRanges(127471197, 127474697)))

##import a WIG file
```

```

wigfile <- system.file("tests", "step.wig", package = "rtracklayer",
                      mustWork=TRUE)
dat <- importScore(file=wigfile, format="WIG")

##import a BigWig file
if(.Platform$OS.type!="windows"){##this is because we are using rtracklayer::import
  bwfile <- system.file("tests", "test.bw", package = "rtracklayer",
                      mustWork=TRUE)
  dat <- importScore(file=bwfile, format="BigWig")
}

##import 2 file
wigfile1 <- system.file("extdata", "cpsf160.repA+.wig", package="trackViewer",
                      mustWork=TRUE)
wigfile2 <- system.file("extdata", "cpsf160.repA-.wig", package="trackViewer",
                      mustWork=TRUE)
dat <- importScore(wigfile1, wigfile2, format="WIG",
                  ranges=GRanges("chr11", IRanges(122817703, 122889073)))

```

---

|                  |                                           |
|------------------|-------------------------------------------|
| importScSeqScore | <i>plot tracks for single cell RNAseq</i> |
|------------------|-------------------------------------------|

---

## Description

Plot single cell RNAseq data as heatmap track for Seurat object.

## Usage

```

importScSeqScore(
  object,
  files,
  samplenames,
  ...,
  txdb,
  gene,
  id,
  idents,
  gr,
  color,
  withCoverageTrack = TRUE,
  flag = scanBamFlag(isSecondaryAlignment = FALSE, isUnmappedQuery = FALSE,
                    isNotPassingQualityControls = FALSE, isSupplementaryAlignment = FALSE)
)

```

## Arguments

|        |                         |
|--------|-------------------------|
| object | Seurat object.          |
| files  | bam file to be scanned. |

|                   |                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------|
| sampleNames       | sample names for files.                                                                   |
| ...               | parameters used by <a href="#">readGAlignmentsList</a> or <a href="#">readGAlignments</a> |
| txdb              | TxDb object for gene model.                                                               |
| gene              | Gene name to plot. (row value)                                                            |
| id                | The id of gene used in txdb.                                                              |
| idents            | identity class to define the groups to plot. (column value)                               |
| gr                | GRanges object to define the plotting region.                                             |
| color             | vector of colors used in heatmap.                                                         |
| withCoverageTrack | plot coverage track or not.                                                               |
| flag              | An integer(2) vector used to filter reads based on their 'flag' entry.                    |

### Examples

```
## Not run:
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
test_file <- "https://github.com/10XGenomics/subset-bam/raw/master/test/test.bam"
trs <- importScSeqScore(files=test_file,
                        txdb=TxDb.Hsapiens.UCSC.hg19.knownGene,
                        id="653635", gene = "WASH7P")

## End(Not run)
```

---

|                 |                                      |
|-----------------|--------------------------------------|
| listChromosomes | <i>List the available chromosome</i> |
|-----------------|--------------------------------------|

---

### Description

List the chromosomes available in the file.

### Usage

```
listChromosomes(file, format = c("hic", "cool"))
```

### Arguments

|        |                                                        |
|--------|--------------------------------------------------------|
| file   | character(1). File name of .hic or .cool/.mcool/.scool |
| format | character(1). File format, "hic" or "cool".            |

### Examples

```
hicfile <- system.file("extdata", "test_chr22.hic", package="trackViewer")
listChromosomes(hicfile)
coolfile <- system.file("extdata", "test.mcool", package="trackViewer")
listChromosomes(coolfile, format="cool")
```

---

|                    |                                          |
|--------------------|------------------------------------------|
| listNormalizations | <i>List the available normalizations</i> |
|--------------------|------------------------------------------|

---

**Description**

List the normalizations available in the file.

**Usage**

```
listNormalizations(file, format = c("hic", "cool"))
```

**Arguments**

|        |                                                        |
|--------|--------------------------------------------------------|
| file   | character(1). File name of .hic or .cool/.mcool/.scool |
| format | character(1). File format, "hic" or "cool".            |

**Examples**

```
hicfile <- system.file("extdata", "test_chr22.hic", package="trackViewer")
listNormalizations(hicfile)
coolfile <- system.file("extdata", "test.mcool", package="trackViewer")
listNormalizations(coolfile, format="cool")
```

---

|                 |                                       |
|-----------------|---------------------------------------|
| listResolutions | <i>List the available resolutions</i> |
|-----------------|---------------------------------------|

---

**Description**

List the resolutions available in the file.

**Usage**

```
listResolutions(file, format = c("hic", "cool"))
```

**Arguments**

|        |                                                        |
|--------|--------------------------------------------------------|
| file   | character(1). File name of .hic or .cool/.mcool/.scool |
| format | character(1). File format, "hic" or "cool".            |

**Examples**

```
hicfile <- system.file("extdata", "test_chr22.hic", package="trackViewer")
listResolutions(hicfile)
coolfile <- system.file("extdata", "test.mcool", package="trackViewer")
listResolutions(coolfile, format="cool")
```

---

|              |                                |
|--------------|--------------------------------|
| loadIdeogram | <i>load ideogram from UCSC</i> |
|--------------|--------------------------------|

---

**Description**

Download ideogram table from UCSC

**Usage**

```
loadIdeogram(genome, chrom = NULL, ranges = NULL, ...)
```

**Arguments**

|        |                                                                          |
|--------|--------------------------------------------------------------------------|
| genome | Assembly name assigned by UCSC, such as hg38, mm10.                      |
| chrom  | A character vector of chromosome names, or NULL.                         |
| ranges | A <a href="#">Ranges</a> object with the intervals.                      |
| ...    | Additional arguments to pass to the <a href="#">GRanges</a> constructor. |

**Value**

A [GRanges](#) object.

**See Also**

See Also as [ideogramPlot](#)

**Examples**

```
## Not run:  
head(loadIdeogram("hg38", chrom = "chr1"))  
  
## End(Not run)
```

---

|          |                  |
|----------|------------------|
| lollipop | <i>Lollipops</i> |
|----------|------------------|

---

**Description**

Plot variants and somatic mutations

**Usage**

```

lollipopplot(
  SNP.gr,
  features = NULL,
  ranges = NULL,
  type = "circle",
  newpage = TRUE,
  ylab = TRUE,
  ylab.gp = gpar(col = "black"),
  yaxis = TRUE,
  yaxis.gp = gpar(col = "black"),
  xaxis = TRUE,
  xaxis.gp = gpar(col = "black"),
  legend = NULL,
  legendPosition = "top",
  cex = 1,
  dashline.col = "gray80",
  jitter = c("node", "label"),
  rescale = FALSE,
  label_on_feature = FALSE,
  lollipop_style_switch_limit = 10,
  ...
)

```

**Arguments**

|                             |                                                                                                                                                                                                                                                            |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SNP.gr                      | A object of <a href="#">GRanges</a> , <a href="#">GRangesList</a> or a list of <a href="#">GRanges</a> . All the width of <a href="#">GRanges</a> must be 1.                                                                                               |
| features                    | A object of <a href="#">GRanges</a> , <a href="#">GRangesList</a> or a list of <a href="#">GRanges</a> . The metadata 'featureLayerID' are used for drawing features in different layers. See details in vignette.                                         |
| ranges                      | A object of <a href="#">GRanges</a> or <a href="#">GRangesList</a> .                                                                                                                                                                                       |
| type                        | character. Could be circle, pie, pin, pie.stack or flag.                                                                                                                                                                                                   |
| newpage                     | Plot in the new page or not.                                                                                                                                                                                                                               |
| ylab                        | Plot ylab or not. If it is a character vector, the vector will be used as ylab.                                                                                                                                                                            |
| ylab.gp, xaxis.gp, yaxis.gp | An object of class gpar for ylab, xaxis or yaxis.                                                                                                                                                                                                          |
| yaxis                       | Plot yaxis or not.                                                                                                                                                                                                                                         |
| xaxis                       | Plot xaxis or not. If it is a numeric vector with length greater than 1, the vector will be used as the points at which tick-marks are to be drawn. And the names of the vector will be used to as labels to be placed at the tick points if it has names. |
| legend                      | If it is a list with named color vectors, a legend will be added.                                                                                                                                                                                          |
| legendPosition              | The position of legend. Possible positions are 'top', 'right', and 'left'.                                                                                                                                                                                 |
| cex                         | cex will control the size of circle.                                                                                                                                                                                                                       |
| dashline.col                | color for the dashed line.                                                                                                                                                                                                                                 |

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| jitter                      | jitter the position of nodes or labels.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| rescale                     | logical(1), character(1), numeric vector, or a dataframe with rescale from and to. Rescale the x-axis or not. if dataframe is used, colnames must be from.start, from.end, to.start, to.end. And the from scale must cover the whole plot region. The rescale parameter can be set as "exon" or "intron" to emphasize "exon" or "intron" region. The "exon" or "intron" can be followed with an integer e.g. "exon_80", or "intron_99". The integer indicates the total percentage of "exon" or "intron" region. Here "exon" indicates all regions in features. And "intron" indicates all flank regions of the features. |
| label_on_feature            | Labels of the feature directly on them. Default FALSE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| lollipop_style_switch_limit | The cutoff value for lollipop style for the 'circle' type. If the max score is greater than this cutoff value, trackViewer will only plot one shape at the highest score. Otherwise trackViewer will draw the shapes like 'Tanghulu'.                                                                                                                                                                                                                                                                                                                                                                                     |
| ...                         | not used.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## Details

In SNP.gr and features, metadata of the GRanges object will be used to control the color, fill, border, alpha, shape, height, cex, dashline.col, data source of pie if the type is pie. And also the controls for labels by name the metadata start as label.parameter.<properties>, and for node labels by name the metadata start as node.label.<properties>, such as label.parameter.rot, label.parameter.gp. The parameter is used for [grid.text](#) or [plotMotifLogoA](#). The metadata 'featureLayerID' for features are used for drawing features in different layers. The metadata 'SNPsideID' for SNP.gr are used for determining the side of lollipops. And the 'SNPsideID' could only be 'top' or 'bottom'.

## Examples

```
SNP <- c(10, 100, 105, 108, 400, 410, 420, 600, 700, 805, 840, 1400, 1402)
x <- sample.int(100, length(SNP))
SNP.gr <- GRanges("chr1", IRanges(SNP, width=1, names=paste0("snp", SNP)),
                  value1=x, value2=100-x)
SNP.gr$color <- rep(list(c("red", 'blue')), length(SNP))
SNP.gr$border <- sample.int(7, length(SNP), replace=TRUE)
features <- GRanges("chr1", IRanges(c(1, 501, 1001),
                                   width=c(120, 500, 405),
                                   names=paste0("block", 1:3)),
                  color="black",
                  fill=c("#FF8833", "#51C6E6", "#DFA32D"),
                  height=c(0.1, 0.05, 0.08),
                  label.parameter.rot=45)
lollipop(SNP.gr, features, type="pie")
```

---

|               |                                   |
|---------------|-----------------------------------|
| optimizeStyle | <i>Optimize the style of plot</i> |
|---------------|-----------------------------------|

---

**Description**

Automatic optimize the stlye of trackViewer

**Usage**

```
optimizeStyle(trackList, viewerStyle = trackViewerStyle(), theme = NULL)
```

**Arguments**

|             |                                                     |
|-------------|-----------------------------------------------------|
| trackList   | An object of <a href="#">trackList</a>              |
| viewerStyle | An object of <a href="#">trackViewerStyle</a>       |
| theme       | A character string. Could be "bw", "col" or "safe". |

**Value**

a list of a [trackList](#) and a [trackViewerStyle](#)

**See Also**

See Also as [viewTracks](#)

**Examples**

```
extdata <- system.file("extdata", package="trackViewer",
                        mustWork=TRUE)
files <- dir(extdata, ".wig")
tracks <- lapply(paste(extdata, files, sep="/"),
                importScore, format="WIG")
re <- optimizeStyle(trackList(tracks))
trackList <- re$tracks
viewerStyle <- re$style
```

---

|               |                                |
|---------------|--------------------------------|
| parse2GRanges | <i>parse text into GRanges</i> |
|---------------|--------------------------------|

---

**Description**

parse text like "chr13:99,443,451-99,848,821:-" into GRanges

**Usage**

```
parse2GRanges(text)
```

**Arguments**

text                      character vector like "chr13:99,443,451-99,848,821:-" or "chr13:99,443,451-99,848,821"

**Value**

an object of [GRanges](#)

**Examples**

```
parse2GRanges("chr13:99,443,451-99,848,821:-")
```

---

|          |                                                     |
|----------|-----------------------------------------------------|
| parseWIG | <i>convert WIG format track to BED format track</i> |
|----------|-----------------------------------------------------|

---

**Description**

convert WIG format track to BED format track for a given range

**Usage**

```
parseWIG(trackScore, chrom, from, to)
```

**Arguments**

trackScore            an object of track with WIG format

chrom                sequence name of the chromosome

from                 start coordinate

to                    end coordinate

**Value**

an object of [track](#)

**Examples**

```
extdata <- system.file("extdata", package="trackViewer", mustWork=TRUE)
repA <- importScore(file.path(extdata, "cpsf160.repA_-.wig"),
                    file.path(extdata, "cpsf160.repA_+.wig"),
                    format="WIG")
strand(repA$dat) <- "-"
strand(repA$dat2) <- "+"
parseWIG(repA, chrom="chr11", from=122929275, to=122930122)
```

---

plotGRanges

*plot GRanges data*


---

## Description

A function to plot GRanges data for given range

## Usage

```
plotGRanges(
  ...,
  range = GRanges(),
  viewerStyle = trackViewerStyle(),
  autoOptimizeStyle = FALSE,
  newpage = TRUE
)
```

## Arguments

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| ...               | one or more objects of <a href="#">GRanges</a>             |
| range             | an object of <a href="#">GRanges</a>                       |
| viewerStyle       | an object of <a href="#">trackViewerStyle</a>              |
| autoOptimizeStyle | should use <a href="#">optimizeStyle</a> to optimize style |
| newpage           | should be draw on a new page?                              |

## Value

An object of [viewport](#) for [addGuideLine](#)

## See Also

See Also as [addGuideLine](#), [addArrowMark](#)

## Examples

```
gr1 <- GRanges("chr1", IRanges(1:50, 51:100))
gr2 <- GRanges("chr1", IRanges(seq(from=10, to=80, by=5),
                                seq(from=20, to=90, by=5)))
vp <- plotGRanges(gr1, gr2, range=GRanges("chr1", IRanges(1, 100)))
addGuideLine(guideLine=c(5, 10, 50, 90), col=2:5, vp=vp)

gr <- GRanges("chr1", IRanges(c(1, 11, 21, 31), width=9),
              score=c(5, 10, 5, 1))
plotGRanges(gr, range=GRanges("chr1", IRanges(1, 50)))
```

---

|          |                      |
|----------|----------------------|
| plotIdeo | <i>plot ideogram</i> |
|----------|----------------------|

---

**Description**

plot ideogram for one chromosome

**Usage**

```
plotIdeo(  
  ideo,  
  chrom = seqlevels(ideo)[1],  
  colorSheme = gieStain(),  
  gp = gpar(fill = NA),  
  ...  
)
```

**Arguments**

|            |                                                      |
|------------|------------------------------------------------------|
| ideo       | output of <a href="#">loadIdeogram</a> .             |
| chrom      | A length 1 character vector of chromosome name.      |
| colorSheme | A character vector of giemsa stain colors.           |
| gp         | parameters used for <a href="#">grid.roundrect</a> . |
| ...        | parameters not used.                                 |

**Examples**

```
## Not run:  
ideo <- loadIdeogram("hg38")  
library(grid)  
grid.newpage()  
plotIdeo(ideo)  
  
## End(Not run)
```

---

|             |                                                   |
|-------------|---------------------------------------------------|
| plotOneIdeo | <i>plot ideogram with data for one chromosome</i> |
|-------------|---------------------------------------------------|

---

**Description**

plot ideogram with data for one chromosome

**Usage**

```
plotOneIdeo(
  ideo,
  dataList,
  parameterList = list(vp = plotViewport(margins = c(0.1, 4.1, 1.1, 0.1)), ideoHeight =
    unit(1/(1 + length(dataList)), "npc"), vgap = unit(1, "lines"), ylabs =
    seqlevels(ideo)[1], ylabsRot = 90, ylabsPos = unit(2.5, "lines"), xaxis = FALSE,
    yaxis = FALSE, xlab = "", types = "barplot", heights = NULL, dataColumn = "score",
    gps = gpar(col = "black", fill = "gray")),
  chrom = seqlevels(ideo)[1],
  colorSheme = gieStain(),
  gp = gpar(fill = NA, lwd = 2),
  ...
)
```

**Arguments**

|               |                                                                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ideo          | output of <a href="#">loadIdeogram</a> .                                                                                                                       |
| dataList      | a <a href="#">GRangesList</a> of data to plot.                                                                                                                 |
| parameterList | a list of parameters for each dataset in the dataList. The elements of the parameters could be xlabs, ylabs, etc. type could be barplot, line, point, heatmap. |
| chrom         | A length 1 character vector of chromosome name.                                                                                                                |
| colorSheme    | A character vector of giemsa stain colors.                                                                                                                     |
| gp            | parameters used for <a href="#">grid.roundrect</a> .                                                                                                           |
| ...           | parameters not used.                                                                                                                                           |

**Examples**

```
## Not run:
ideo <- loadIdeogram("hg38")
library(rtracklayer)
library(grid)
dataList <- ideo[seqnames(ideo) %in% "chr1"]
dataList$score <- as.numeric(dataList$gieStain)
dataList <- dataList[dataList$gieStain!="gneg"]
dataList <- GRangesList(dataList, dataList)
grid.newpage()
plotOneIdeo(ideo, dataList, chrom="chr1")

## End(Not run)
```

pos-class

Class "pos"

**Description**

An object of class "pos" represents a point location

**Slots**

x A **numeric** value, indicates the x position

y A **numeric** value, indicates the y position

unit "character" specifying the units for the corresponding numeric values. See [unit](#)

reduce, GInteractions-method

*Reduce method for 'GInteractions'***Description**

Reduce returns an object of the same type as x containing reduced ranges for each distinct (seqname, strand) pairing.

**Usage**

```
## S4 method for signature 'GInteractions'
reduce(x, min.gapwidth = 1L, ignore.strand = TRUE, ...)
```

**Arguments**

|               |                                                                                 |
|---------------|---------------------------------------------------------------------------------|
| x             | GInteractions object.                                                           |
| min.gapwidth  | Ranges separated by a gap of at least min.gapwidth positions are not merged.    |
| ignore.strand | TRUE or FALSE. Whether the strand of the input ranges should be ignored or not. |
| ...           | Not used.                                                                       |

**Examples**

```
## Not run:
library(InteractionSet)
gi <- readRDS(system.file("extdata", "gi.rds", package="trackViewer"))
reduce(head(gi, n=20))

## End(Not run)
```

---

|                 |                       |
|-----------------|-----------------------|
| trackList-class | <i>List of tracks</i> |
|-----------------|-----------------------|

---

### Description

An extension of List that holds only [track](#) objects.

### Usage

```
## S4 replacement method for signature 'trackList'
seqlevelsStyle(x) <- value

trackList(..., heightDist = NA)
```

### Arguments

|            |                                                                                                                                                                                                     |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x          | trackList object.                                                                                                                                                                                   |
| value      | values to be assigned.                                                                                                                                                                              |
| ...        | Each tracks in ... becomes an element in the new trackList, in the same order. This is analogous to the list constructor, except every argument in ... must be derived from <a href="#">track</a> . |
| heightDist | A vector or NA to define the height of each track.                                                                                                                                                  |

### See Also

[track](#).

---

|                  |                           |
|------------------|---------------------------|
| trackStyle-class | <i>Class "trackStyle"</i> |
|------------------|---------------------------|

---

### Description

An object of class "trackStyle" represents track style.

An object of class "track" bundles the genomic data to be plotted together with the style used to draw it. A track is the basic unit accepted by [viewTracks](#); several of them are collected into a [trackList](#) for a multi-panel figure.

**Usage**

```

## S4 method for signature 'track'
seqlevels(x)

## S4 method for signature 'track'
seqlevelsStyle(x)

## S4 replacement method for signature 'track'
seqlevelsStyle(x) <- value

## S4 method for signature 'track'
show(object)

## S4 method for signature 'track'
x$name

## S4 replacement method for signature 'track'
x$name <- value

## S3 method for class 'track'
.DollarNames(x, pattern = "")

## S3 method for class 'trackStyle'
.DollarNames(x, pattern = "")

setTrackStyleParam(ts, attr, value)

## S4 method for signature 'track,character'
setTrackStyleParam(ts, attr, value)

setTrackXscaleParam(ts, attr, value)

## S4 method for signature 'track,character'
setTrackXscaleParam(ts, attr, value)

setTrackYaxisParam(ts, attr, value)

## S4 method for signature 'track,character'
setTrackYaxisParam(ts, attr, value)

```

**Arguments**

|         |                                                         |
|---------|---------------------------------------------------------|
| x       | an object of trackStyle or track                        |
| value   | values to be assigned.                                  |
| object  | an object of trackStyle.                                |
| name    | slot name of trackStyle or track                        |
| pattern | A regular expression. Only matching names are returned. |

ts                    An object of track.  
 attr                the name of slot of [trackStyle](#) object to be changed.

## Details

The validity method enforces the type-specific requirements summarised above; the most common errors are a missing score or feature metadata column, a lollipop range wider than one base, and a dat2 whose length does not match dat for interaction data.

Note that dat and dat2 are plain GRanges objects, so the seqlevel style of the track must match that of the other tracks and of the plotting range. seqlevelsStyle and its replacement method are defined for track and rename both dat and dat2 at once.

setTrackStyleParam changes one styling slot of the [trackStyle](#) object held in ts@style. It accepts only the slots that are plain values; the two compound slots have their own setters, [setTrackXscaleParam](#) for xscale and [setTrackYaxisParam](#) for yaxis. Passing any other name, including "xscale" or "yaxis", raises an error.

Accepted values of attr:

tracktype "character". How the data is drawn. For "data" tracks use "peak" (the default), "line", "histogram" or "annotation", the last marking peak regions rather than plotting their scores. For "interactionData" tracks use "heatmap" or "link".

color "character" vector of colors. Give two values when the track has both dat and dat2, the first for the positive side and the second for the negative side of the baseline.

height "numeric" between 0 and 1, the fraction of the figure given to this track. Note that [trackList](#) rewrites this slot for every element it is given, so set it after building the list, not before.

marginTop, marginBottom "numeric", space left above and below the drawing region of the track, as a fraction of its height. Defaults are 0 and 0.05.

ylim "numeric" of length 2 fixing the y-axis range. Useful for making two tracks directly comparable, since the range is otherwise taken from the data in the current view.

ylabpos "character", one of "left", "right", "topleft", "bottomleft", "topright", "bottomright", "abovebaseline", "underbaseline" or "none"; for gene-type tracks also "upstream" or "downstream".

ylablas "numeric" in {0, 1, 2, 3}, the rotation of the y label. See [par:las](#).

ylabgp "list" of graphical parameters for the y label, converted to [gpar](#), e.g. list(cex=.8, col="gray30").

breaks "numeric" breaks for the color key of interaction data.

NAcolor "character", the color used for missing cells of an interaction heatmap. Default "white".

ysplit "numeric", where to split the y region for back-to-back interaction plots. Default 0.5 splits it evenly; it has no effect on other layouts.

The value is assigned with check = TRUE, so the validity method of [trackStyle](#) runs on every call and an out-of-range value (an unknown tracktype, an ylablas outside 0:3, an unrecognized ylabpos) fails immediately rather than at plotting time.

setTrackXscaleParam changes the x-scale bar of a track, stored as an [xscale](#) object at ts@style@xscale. The x-scale bar is a short horizontal ruler drawn inside the track's own panel, independent of the

shared genomic x-axis of the whole figure, typically used to show a distance such as "1K" next to a zoomed-in feature.

attr can be any slot of `xscale`, or the convenience name "position":

from, to Objects of class `pos`, the two endpoints of the scale bar. Because these are pos objects rather than bare numbers, setting them directly means constructing the object yourself, e.g. `new("pos", x=12345678, y=0.5, unit="native")` for a point given in native (genomic) coordinates, or a fraction with `unit="npc"` for a position relative to the panel. Most callers will find `attr="position"` (below) more convenient than setting from and to separately, since it keeps them centered and in sync with the label.

label "character", the text drawn on the scale bar, e.g. "500 bp". Set directly this is used verbatim.

gp "list" of graphical parameters for the scale bar and its label, converted to `gpar`, e.g. `list(cex=.5, col="black", lwd=2)`.

draw "logical", whether the scale bar is drawn at all. Default FALSE; set to TRUE even from/to (or position) have been set, the scale bar will show.

position A shortcut that sets from, to and label together from a single scale length, instead of requiring two pos objects. value must be a "list" with named elements x, y and label:

- x, y — the native-coordinate of the bar (genomic position and track-relative y, respectively). The bar is drawn symmetrically about this point.
- label — the width the bar should represent.

A label that cannot be coerced to numeric (NA after `as.numeric`) is rejected before any slot is touched.

As with the other slots of `xscale`, updated values are assigned with `check = TRUE`, so the `pos` validity method (checking the unit string) still applies when from/to are set through "position".

`setTrackYaxisParam` changes the y-axis of a track, stored as a `yaxisStyle` object at `ts@style@yaxis`. This is the axis drawn alongside the track's own data panel (score, coverage, etc.), separate from the shared genomic x-axis of the whole figure.

Accepted values of attr, all slots of `yaxisStyle`:

at "numeric" vector of y-values at which tick marks are drawn, e.g. `c(0, 25, 50)`. If left at its default (`numeric(0)`), tick positions are chosen automatically from the plotted range.

label "logical", whether the numeric value of each tick in at is printed next to it. Default FALSE, i.e. ticks are drawn without their values; set to TRUE to show them.

gp "list" of graphical parameters for the axis line, ticks and labels, converted to `gpar`, e.g. `list(cex=.6, col="blue", lwd=1.5)`.

draw "logical", whether the y-axis is drawn at all. Default TRUE.

main "logical", which side of the panel the axis is drawn on: TRUE (the default) draws it on the left of the track, FALSE on the right. Useful for telling apart the axes of two adjacent or back-to-back tracks at a glance.

Any other value of attr, including "xscale" or a slot name belonging to `trackStyle` itself (like "color" or "height"), is rejected, use `setTrackStyleParam` for the plain style slots and `setTrackXscaleParam` for the x-scale bar. Values are assigned with `check = TRUE`, so an ill-typed value (for example a non-logical passed to "draw") fails immediately rather than at plotting time.

**Value**

An object of class [track](#) with the modified style, returned invisibly. The variable supplied as `ts` is updated as a side effect.

An object of class [track](#) with the modified `xscale`, returned invisibly. The variable supplied as `ts` is updated as a side effect.

An object of class [track](#) with the modified `yaxis`, returned invisibly. The variable supplied as `ts` is updated as a side effect.

**Slots**

`tracktype` "character" track type, could be peak, line, histogram, or annotation. Default is "peak". "annotation" is used to mark the peak regions. For interaction data, it could be "heatmap" or "link".

`color` "character" track color. If the track has `dat` and `dat2` slot, it should have two values.

`NAcolor` "character" NA color for `interactionData`.

`breaks` "numeric" breaks for color keys of `interactionData`.

`height` "numeric" track height. It should be a value between 0 and 1

`marginTop` "numeric" track top margin

`marginBottom` "numeric" track bottom margin

`xscale` object of [xscale](#), describe the details of x-scale

`yaxis` object of [yaxisStyle](#), describe the details of y-axis

`ylim` "numeric" y-axis range

`ylabpos` "character", ylable position, `ylabpos` should be 'left', 'right', 'topleft', 'bottomleft', 'topright', 'bottomright', 'abovebaseline', 'underbaseline', or 'none'. For gene type track, it also could be 'upstream' or 'downstream'

`ylablas` "numeric" y label direction. It should be a integer 0-3. See [par:las](#)

`ylabgp` A "list" object, It will convert to an object of class [gpar](#). This is basically a list of graphical parameter settings of y-label.

`ysplit` A "numeric" to split y plot region for interaction data. Default is 0.5, which will split the region half to half. It will only work for back to back plot.

`dat` Object of class [GRanges](#), the primary data of the track. The metadata columns it must carry depend on type (see 'Details'): coverage-like tracks ("data", "scSeq"), "lollipopData" and "interactionData" require a score column, whereas "gene" and "transcript" tracks require a feature column.

`dat2` Object of class [GRanges](#), the optional second data set of the track; leave it as a zero-length [GRanges](#) when unused. Its meaning depends on type:

- for "data", "scSeq" and "lollipopData" tracks it is drawn back-to-back with `dat`: `dat` above the baseline and `dat2` below it as  $-1 * \text{score}$ , which is the usual way to compare two samples or the two strands of one sample in a single panel;
- for "interactionData" it holds the second anchor of each interaction, so it must either be the same length as `dat` (anchors matched element-wise) or carry a target metadata column.

When supplied it must contain a score column, and for lollipop data every range must have width 1, exactly as for dat.

type The type of track, one of:

"data" continuous or interval scores — coverage, signal, peaks. Drawn as a peak, line or histogram depending on style@tracktype.

"gene" gene models. dat must have a feature column describing each range (for example "exon", "CDS", "utr5", "utr3").

"transcript" transcript models; same feature requirement as "gene", but isoforms are drawn on separate rows.

"scSeq" single-cell signal. Validated like "data", i.e. a format and a numeric score are required.

"lollipopData" point features such as variants or modification sites. Every range in dat (and dat2) must have width 1; ranges wider than one base are rejected.

"interactionData" chromatin interactions, for example from Hi-C or 4C. Drawn as a heatmap or as arcs according to style@tracktype.

format The format the data was imported from: "BED", "bedGraph", "WIG", "BigWig" or "BAM". It is required, and must be a single string, for "data" and "scSeq" tracks; other types ignore it. For every format except "WIG" the score column must be numeric or integer. "WIG" is the exception: because a WIG file has no fixed step boundaries until it is plotted, score is stored as a [CompressedCharacterList](#) and is converted on the fly — see [importScore](#).

style Object of class [trackStyle](#) controlling color, height, margins, y-axis, x-scale and the drawing mode. Do not assign to its slots directly; use [setTrackStyleParam](#), [setTrackXscaleParam](#) and [setTrackYaxisParam](#), which validate the value and update the object in the calling frame.

name unused yet

### Assignment in the calling frame

Unlike most R functions, [setTrackStyleParam](#) modifies its first argument in place: it assigns the updated track back to the variable that was passed in, so [setTrackStyleParam](#)(fox2, "color", "red") is enough and fox2 <- [setTrackStyleParam](#)(...) is not needed. This works only when ts is given as a name or a subsetting expression that can be assigned to; calling it on a temporary value, for example [setTrackStyleParam](#)(importScore(f), "color", "red"), updates nothing the caller can see. Inside lapply and friends the assignment targets the loop variable, which is discarded, so use the returned object there instead — it is returned invisibly and carries the same change.

Like [setTrackStyleParam](#), this method assigns the updated track back to the variable passed as ts in the caller's environment, so [setTrackXscaleParam](#)(fox2, "draw", TRUE) is sufficient and the result need not be re-assigned. It is also returned invisibly for use in contexts (such as lapply) where the calling-frame assignment has no visible target.

Like [setTrackStyleParam](#) and [setTrackXscaleParam](#), this method assigns the updated track back to the variable passed as ts in the caller's environment, so [setTrackYaxisParam](#)(fox2, "draw", FALSE) is sufficient on its own. It is also returned invisibly for use in contexts, such as lapply, where that calling-frame assignment has no visible target.

## See Also

Please try to use `importScore` and `importBam` to generate the object.

`setTrackXscaleParam`, `setTrackYaxisParam`, `trackStyle`  
`xscale`, `pos`, `setTrackStyleParam`, `setTrackYaxisParam`  
`yaxisStyle`, `setTrackStyleParam`, `setTrackXscaleParam`

## Examples

```
extdata <- system.file("extdata", package="trackViewer",
mustWork=TRUE)
fox2 <- importScore(file.path(extdata, "fox2.bed"), format="BED")
setTrackStyleParam(fox2, "color", c("red","green"))
setTrackXscaleParam(fox2, "gp", list(cex=.5))
setTrackYaxisParam(fox2, "gp", list(col="blue"))
fox2$dat <- GRanges(score=numeric(0))

## inspect and change the style
fox2$type
fox2$style$tracktype
setTrackStyleParam(fox2, "tracktype", "histogram")
setTrackStyleParam(fox2, "height", .2)

extdata <- system.file("extdata", package="trackViewer", mustWork=TRUE)
fox2 <- importScore(file.path(extdata, "fox2.bed"), format="BED")

## the track is updated in place; no re-assignment needed
setTrackStyleParam(fox2, "tracktype", "histogram")
setTrackStyleParam(fox2, "color", c("#E69F00", "#56B4E9"))
setTrackStyleParam(fox2, "ylim", c(0, 50))
setTrackStyleParam(fox2, "ylabpos", "topleft")
setTrackStyleParam(fox2, "ylabgp", list(cex=.8, col="gray30"))
fox2$style$tracktype

## inside lapply, use the returned value
trs <- lapply(list(a=fox2, b=fox2), function(.ele){
  setTrackStyleParam(.ele, "height", .25)
})

## invalid values are rejected straight away
## Not run:
setTrackStyleParam(fox2, "ylablas", 5)      # must be 0:3
setTrackStyleParam(fox2, "xscale", list())  # use setTrackXscaleParam

## End(Not run)
extdata <- system.file("extdata", package="trackViewer", mustWork=TRUE)
fox2 <- importScore(file.path(extdata, "fox2.bed"), format="BED")

## simplest way: give a centre point and a width in bases
setTrackXscaleParam(fox2, "position",
  list(x=122929675, y=4, label=500))
setTrackXscaleParam(fox2, "draw", TRUE)
```

```

## style the bar itself
setTrackXscaleParam(fox2, "gp", list(cex=.5, col="gray30"))

## equivalent, done by hand with explicit pos objects
setTrackXscaleParam(fox2, "from",
  new("pos", x=122929675-500, y=4, unit="native"))
setTrackXscaleParam(fox2, "to",
  new("pos", x=122929675+500, y=4, unit="native"))
setTrackXscaleParam(fox2, "label", "1K")

## Not run:
## rejected: label can't be parsed as a number
setTrackXscaleParam(fox2, "position",
  list(x=122929675, y=4, label="five hundred"))

## End(Not run)
extdata <- system.file("extdata", package="trackViewer", mustWork=TRUE)
fox2 <- importScore(file.path(extdata, "fox2.bed"), format="BED")

## show tick values at chosen positions
setTrackYaxisParam(fox2, "at", c(0, 25, 50))
setTrackYaxisParam(fox2, "label", TRUE)

## style the axis and move it to the right-hand side
setTrackYaxisParam(fox2, "gp", list(cex=.6, col="blue"))
setTrackYaxisParam(fox2, "main", FALSE)

## hide the axis entirely
setTrackYaxisParam(fox2, "draw", FALSE)

## Not run:
## rejected: "color" belongs to trackStyle, not yaxisStyle
setTrackYaxisParam(fox2, "color", "red")

## End(Not run)

```

---

trackViewerStyle-class

*Class "trackViewerStyle"*


---

## Description

An object of class "trackViewerStyle" represents track viewer style.

## Usage

```
trackViewerStyle(...)
```

```
setTrackViewerStyleParam(tvs, attr, value)
```

```
## S4 method for signature 'trackViewerStyle,character'
setTrackViewerStyleParam(tvs, attr, value)
```

## Arguments

|       |                                                                   |
|-------|-------------------------------------------------------------------|
| ...   | Each argument in ... becomes an slot in the new trackViewerStyle. |
| tvs   | An object of trackViewerStyle.                                    |
| attr  | the name of slot to be changed.                                   |
| value | values to be assigned.                                            |

## Details

setTrackViewerStyleParam changes one slot of a [trackViewerStyle](#) object. Unlike setTrackStyleParam and its siblings, this style is not attached to a single track: it controls the figure-wide x-axis shared by every track in a plot (drawn once, beneath the bottom-most track), so this function is typically called on the trackViewerStyle object passed to viewTracks, not on a track itself.

Accepted values of attr, all slots of [trackViewerStyle](#):

margin "numeric" of length 4, the bottom, left, top and right margins of the whole plot, as fractions of the device (each value must be between 0 and 0.8). Default `c(.01, .05, 0, 0)`.

xlas "numeric" in {0, 1, 2, 3}, the rotation of the x-axis tick labels. See [par](#):las. Ignored when autolas is TRUE.

xgp "list" of graphical parameters for the x-axis, converted to [gpar](#), e.g. `list(cex=.8, col="black")`. For the equivalent y-axis settings, see [yaxisStyle](#) and [setTrackYaxisParam](#), which are per-track rather than figure-wide.

xaxis "logical", whether the shared x-axis is drawn at all. Default FALSE.

xat "numeric" vector of genomic positions at which x-axis tick marks are drawn, passed to `grid.xaxis` as `at`. If left at its default (`numeric(0)`), positions are chosen automatically from the plotted range.

xlabel "character" vector of labels for the ticks in `xat`, passed to `grid.xaxis` as `label`. Should be the same length as `xat` when both are set explicitly; left at its default, labels are generated automatically from the tick positions.

autolas "logical", whether the y-axis label direction is chosen automatically rather than following `xlas`. Default FALSE.

flip "logical", whether the x-axis (and the whole plot) is flipped, e.g. to display a feature on the minus strand 5' to 3' left to right. Default FALSE.

Any other value of attr is rejected. Values are assigned with `check = TRUE`, so the validity method of [trackViewerStyle](#) runs on every call — an `xlas` outside 0:3 or a `margin` outside [0, 0.8] fails immediately with a descriptive message rather than surfacing later as a plotting error.

## Value

An object of class [trackViewerStyle](#) with the modified slot, returned invisibly. The variable supplied as `tvs` is updated as a side effect.

**Slots**

margin "numeric", specify the bottom, left, top and right margin.

xlas "numeric", label direction of x-axis mark. It should be a integer 0-3. See [par:las](#)

xgp A "list", object, It will convert to an object of class [gpar](#). This is basically a list of graphical parameter settings of x-axis. For y-axis, see [yaxisStyle](#)

xaxis "logical", draw x-axis or not

xat "numeric", the values will be passed to grid.xaxis as 'at' parameter.

xlabel "character", the values will be passed to grid.xaxis as 'label' parameter.

autolas "logical" automatic determine y label direction

flip "logical" flip the x-axis or not, default FALSE

**Assignment in the calling frame**

Like [setTrackStyleParam](#), this method assigns the updated object back to the variable passed as tvs in the caller's environment, so `setTrackViewerStyleParam(tvs, "xaxis", TRUE)` is sufficient on its own and the result need not be re-assigned. It is also returned invisibly for use in contexts, such as `lapply`, where that calling-frame assignment has no visible target — see [setTrackStyleParam](#) for a worked example of that case.

**See Also**

[trackViewerStyle](#), [setTrackStyleParam](#), [setTrackXscaleParam](#), [setTrackYaxisParam](#)

**Examples**

```
tvs <- trackViewerStyle()
setTrackViewerStyleParam(tvs, "xaxis", TRUE)

tvs <- trackViewerStyle()

## turn on the shared x-axis and rotate its labels
setTrackViewerStyleParam(tvs, "xaxis", TRUE)
setTrackViewerStyleParam(tvs, "xlas", 2)

## choose explicit tick positions and labels
setTrackViewerStyleParam(tvs, "xat", c(122929000, 122929500, 122930000))
setTrackViewerStyleParam(tvs, "xlabel", c("122,929,000", "122,929,500", "122,930,000"))

## tighten the plot margins
setTrackViewerStyleParam(tvs, "margin", c(.02, .08, .01, .01))

## flip the plot, e.g. for a minus-strand gene
setTrackViewerStyleParam(tvs, "flip", TRUE)

## Not run:
## rejected: xlas must be one of 0:3
setTrackViewerStyleParam(tvs, "xlas", 5)
## rejected: not a slot of trackViewerStyle
```

```
setTrackViewerStyleParam(tvs, "color", "red")

## End(Not run)
```

---

|          |                                       |
|----------|---------------------------------------|
| viewGene | <i>plot tracks based on gene name</i> |
|----------|---------------------------------------|

---

## Description

given a gene name, plot the tracks.

## Usage

```
viewGene(
  symbol,
  filenames,
  format,
  txdb,
  org,
  upstream = 1000,
  downstream = 1000,
  anchor = c("gene", "TSS"),
  plot = FALSE
)
```

## Arguments

|            |                                        |
|------------|----------------------------------------|
| symbol     | Gene symbol                            |
| filenames  | files used to generate tracks          |
| format     | file format used to generate tracks    |
| txdb       | txdb will be used to extract the genes |
| org        | org package name                       |
| upstream   | upstream from anchor                   |
| downstream | downstream from anchor                 |
| anchor     | TSS, or gene                           |
| plot       | plot the tracks or not.                |

## Value

an invisible list of a [trackList](#), a [trackViewerStyle](#) and a [GRanges](#)

**Examples**

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(org.Hs.eg.db)
extdata <- system.file("extdata", package="trackViewer", mustWork=TRUE)
filename = file.path(extdata, "fox2.bed")
optSty <- viewGene("HSPA8", filenames=filename, format="BED",
                  txdb=TxDb.Hsapiens.UCSC.hg19.knownGene,
                  org="org.Hs.eg.db")
```

viewTracks

*plot the tracks***Description**

A function to plot the data for given range

**Usage**

```
viewTracks(
  trackList,
  chromosome,
  start,
  end,
  strand,
  gr = GRanges(),
  ignore.strand = TRUE,
  viewerStyle = trackViewerStyle(),
  autoOptimizeStyle = FALSE,
  newpage = TRUE,
  operator = NULL,
  smooth = FALSE,
  lollipop_style_switch_limit = 10
)
```

**Arguments**

|               |                                                       |
|---------------|-------------------------------------------------------|
| trackList     | an object of <a href="#">trackList</a>                |
| chromosome    | chromosome                                            |
| start         | start position                                        |
| end           | end position                                          |
| strand        | strand                                                |
| gr            | an object of <a href="#">GRanges</a>                  |
| ignore.strand | ignore the strand or not when do filter. default TRUE |
| viewerStyle   | an object of <a href="#">trackViewerStyle</a>         |

|                             |                                                                                                                                                                                                                                          |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| autoOptimizeStyle           | should use <a href="#">optimizeStyle</a> to optimize style                                                                                                                                                                               |
| newpage                     | should be draw on a new page?                                                                                                                                                                                                            |
| operator                    | operator, could be +, -, *, /, ^, %%, and NA. "-" means dat - dat2, and so on. NA means do not apply any operator. Note: if multiple operator is supplied, please make sure the length of operator keep same as the length of trackList. |
| smooth                      | logical(1) or numeric(). Plot smooth curve or not. If it is numeric, eg n, mean of nearby n points will be used for plot. If it is numeric, the second number will be the color. Default color is 2 (red).                               |
| lollipop_style_switch_limit | The cutoff value for lollipop style for the 'circle' type. If the max score is greater than this cutoff value, trackViewer will only plot one shape at the highest score. Otherwise trackViewer will draw the shapes like 'Tanghulu'.    |

### Value

An object of [viewport](#) for [addGuideLine](#)

### See Also

See Also as [addGuideLine](#), [addArrowMark](#)

### Examples

```
extdata <- system.file("extdata", package="trackViewer",
                      mustWork=TRUE)
files <- dir(extdata, "-.wig")
tracks <- lapply(paste(extdata, files, sep="/"),
               importScore, format="WIG")
tracks <- lapply(tracks, function(.ele) {strand(.ele@dat) <- "-"; .ele})
fox2 <- importScore(paste(extdata, "fox2.bed", sep="/"), format="BED")
dat <- coverageGR(fox2@dat)
fox2@dat <- dat[strand(dat)=="+"]
fox2@dat2 <- dat[strand(dat)=="-"]
gr <- GRanges("chr11", IRanges(122929275, 122930122), strand="-")
viewTracks(trackList(track=tracks, fox2=fox2), gr=gr, autoOptimizeStyle=TRUE)
```

---

xscale-class

*Class "xscale"*

---

### Description

An object of class "xscale" represents x-scale style.

**Slots**

from A [pos](#) class, indicates the start point position of x-scale.  
 to A [pos](#) class, indicates the end point position of x-scale.  
 label "character" the label of x-scale  
 gp A "list" object, It will convert to an object of class [gpar](#). This is basically a list of graphical parameter settings of x-scale.  
 draw A "logical" value indicating whether the x-scale should be draw.

---

|                  |                    |
|------------------|--------------------|
| yaxisStyle-class | Class "yaxisStyle" |
|------------------|--------------------|

---

**Description**

An object of class "yaxisStyle" represents y-axis style.

**Slots**

at "numeric" vector of y-value locations for the tick marks  
 label "logical" value indicating whether to draw the labels on the tick marks.  
 gp A "list" object, It will convert to an object of class [gpar](#). This is basically a list of graphical parameter settings of y-axis.  
 draw A "logical" value indicating whether the y-axis should be draw.  
 main A "logical" value indicating whether the y-axis should be draw in left (TRUE) or right (FALSE).

# Index

`.DollarNames.track` (trackStyle-class), 36  
`.DollarNames.trackStyle` (trackStyle-class), 36  
`$, track-method` (trackStyle-class), 36  
`$, trackStyle-method` (trackStyle-class), 36  
`$<-`, track-method (trackStyle-class), 36  
`$<-`, trackStyle-method (trackStyle-class), 36  
  
`addArrowMark`, 4, 5, 6, 32, 48  
`addGuideLine`, 5, 5, 6, 14, 32, 48  
`addInteractionAnnotation`, 6  
`ARA`, 7  
`arrow`, 5  
  
`browseTracks`, 8  
`browseTracks-shiny`, 9  
`browseTracksOutput` (browseTracks-shiny), 9  
  
`CompressedCharacterList`, 41  
`coverage`, 10  
`coverageGR`, 9  
  
`dandelion.plot`, 10  
  
`GAlignmentPairs`, 9  
`GAlignments`, 9  
`geneModelFromTxdb`, 11  
`geneTrack`, 13  
`getCurTrackViewport`, 5, 13  
`getGeneIDsFromTxDb`, 14  
`getLocation`, 15  
`gi2track`, 15  
`gieStain`, 16  
`GInteractions`, 6, 7  
`GIoperator`, 16  
`gpar`, 17, 38–40, 44, 45, 49  
  
`GRanges`, 6, 8–10, 17, 20, 22, 23, 27, 28, 31, 32, 40, 46, 47  
`GRangesList`, 10, 19, 28, 34  
`grid.lines`, 6  
`grid.polygon`, 6  
`grid.roundrect`, 19, 33, 34  
`grid.text`, 6, 29  
`gridPlot`, 17  
`GRoperator`, 17  
  
`ideogramPlot`, 18, 27  
`importBam`, 12, 19, 23, 42  
`importData`, 20  
`importGInteractions`, 7, 21  
`importScore`, 12, 20, 23, 41, 42  
`importScSeqScore`, 24  
  
`listChromosomes`, 22, 25  
`listNormalizations`, 26  
`listResolutions`, 22, 26  
`loadIdeogram`, 19, 27, 33, 34  
`lollipop`, 27  
  
`numeric`, 35  
  
`optimizeStyle`, 30, 32, 48  
  
`par`, 38, 40, 44, 45  
`parse2GRanges`, 30  
`parseWIG`, 31  
`plotGRanges`, 32  
`plotIdeo`, 33  
`plotMotifLogoA`, 29  
`plotOneIdeo`, 33  
`pos`, 39, 42, 49  
`pos` (pos-class), 35  
`pos-class`, 35  
  
`Ranges`, 27  
`readGAlignments`, 20, 25  
`readGAlignmentsList`, 25

- readHicNormTypes, 22
- reduce, GInteractions
  - (reduce, GInteractions-method), 35
- reduce, GInteractions-method, 35
- renderbrowseTracks
  - (browseTracks-shiny), 9
- RleList, 20
- seqlevels, track-method
  - (trackStyle-class), 36
- seqlevelsStyle, track-method
  - (trackStyle-class), 36
- seqlevelsStyle<-, track-method
  - (trackStyle-class), 36
- seqlevelsStyle<-, trackList-method
  - (trackList-class), 36
- setTrackStyleParam, 39, 41, 42, 45
- setTrackStyleParam (trackStyle-class), 36
- setTrackStyleParam, track, character, ANY-method
  - (trackStyle-class), 36
- setTrackStyleParam, track, character-method
  - (trackStyle-class), 36
- setTrackViewerStyleParam
  - (trackViewerStyle-class), 43
- setTrackViewerStyleParam, trackViewerStyle, character, ANY-method
  - (trackViewerStyle-class), 43
- setTrackViewerStyleParam, trackViewerStyle, character-method
  - (trackViewerStyle-class), 43
- setTrackXscaleParam, 38, 39, 41, 42, 45
- setTrackXscaleParam (trackStyle-class), 36
- setTrackXscaleParam, track, character, ANY-method
  - (trackStyle-class), 36
- setTrackXscaleParam, track, character-method
  - (trackStyle-class), 36
- setTrackYaxisParam, 38, 41, 42, 44, 45
- setTrackYaxisParam (trackStyle-class), 36
- setTrackYaxisParam, track, character, ANY-method
  - (trackStyle-class), 36
- setTrackYaxisParam, track, character-method
  - (trackStyle-class), 36
- show, track-method (trackStyle-class), 36
- track, 11–13, 19–23, 31, 36, 40
- track (trackStyle-class), 36
- track-class (trackStyle-class), 36
- trackList, 8, 30, 36, 38, 46, 47
- trackList (trackList-class), 36
- trackList-class, 36
- trackStyle, 38, 39, 41, 42
- trackStyle (trackStyle-class), 36
- trackStyle-class, 36
- trackViewer (trackViewer-package), 3
- trackViewer-package, 3
- trackViewerStyle, 14, 30, 32, 44–47
- trackViewerStyle
  - (trackViewerStyle-class), 43
- trackViewerStyle-class, 43
- TxDb, 12–14
- unit, 35
- viewGene, 46
- viewport, 14, 32, 48
- viewTracks, 4, 5, 11, 12, 20, 23, 30, 36, 47
- xscale, 38–40, 42
- xscale (xscale-class), 48
- xscale-class, 48
- yaxisStyle, 39, 40, 42, 44, 45
- yaxisStyle (yaxisStyle-class), 49
- yaxisStyle-class, 49