

Package ‘trackViewer’

October 9, 2015

Type Package

Title A bioconductor package with minimalist design for plotting elegant track layers

Version 1.4.0

Date 2015-03-26

Author Jianhong Ou, Yong-Xu Wang, Lihua Julie Zhu

Maintainer Jianhong Ou <jianhong.ou@umassmed.edu>

Description

visualize mapped reads along with annotation as track layers for NGS dataset such as ChIP-seq, RNA-seq, miRNA-seq, DNA-seq.

License GPL (>= 2)

Depends R (>= 3.1.0), methods, GenomicRanges, grid, gWidgetstcltk

Imports GenomicAlignments, GenomicFeatures, Gviz, pbapply, Rsamtools, rtracklayer, scales

Suggests biomaRt, TxDb.Hsapiens.UCSC.hg19.knownGene, RUnit, BiocGenerics, BiocStyle

biocViews Visualization

NeedsCompilation no

R topics documented:

| | |
|---------------------|----|
| trackViewer-package | 2 |
| addArrowMark | 3 |
| addGuideLine | 4 |
| coverageGR | 5 |
| geneModelFromTxdb | 6 |
| getCurTrackViewport | 7 |
| importBam | 7 |
| importScore | 8 |
| interactiveViewer | 9 |
| optimizeStyle | 10 |
| plotGRanges | 11 |

| | |
|----------------------------------|----|
| pos-class | 12 |
| track-class | 12 |
| trackList-class | 14 |
| trackStyle-class | 14 |
| trackViewerStyle-class | 15 |
| viewTracks | 16 |
| xscale-class | 17 |
| yaxisStyle-class | 17 |

Index 19

trackViewer-package *Minimal designed plotting tool for genomic data*

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.

Details

```

Package: trackViewer
Type: Package
Version: 1.0
Date: 2013-10-18
License: Artistic-2.0

```

This package is minimal designed to plot figure for publication.

Author(s)

Jianhong Ou, Julie Lihua Zhu

Maintainer: Jianhong Ou <jianhong.ou@umassmed.edu>

Examples

```

library(TxDb.Hsapiens.UCSC.hg19.knownGene)
trs <- geneModelFromTxdb(TxDb.Hsapiens.UCSC.hg19.knownGene,
                        "chr11", 122929275, 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(.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 viewTracks |

Value

NULL

Author(s)

Jianhong Ou

See Also

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

Examples

```
grid.newpage()
addArrowMark(list(x=unit(.5, "npc"),
                  y=unit(.5, "npc")),
             label="label1",
             col="blue")
```

| | |
|--------------|--------------------------------------|
| 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 viewTracks |

Value

NULL

Author(s)

Jianhong Ou

See Also

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

Examples

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

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

Description

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

Usage

```
coverageGR(gr)
```

Arguments

`gr` an object of [RGanges](#), [GAlignments](#) or [GAlignmentPairs](#)

Value

an object of [GRanges](#)

Author(s)

Jianhong Ou

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)
```

getCurTrackViewport *Get current track viewport*

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](#)

Author(s)

Jianhong Ou

See Also

See Also as [addGuideLine](#)

Examples

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

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 GRanges to indicate the range to be imported |
| pairs | logical object to indicate the BAM is paired or not. See readGAlignments |

Value

a [track](#) object

Author(s)

Jianhong Ou

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="+"))
```

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 GRanges to indicate the range to be imported |
| ignore.strand | ignore the strand or not when do filter. default TRUE |

Value

a [track](#) object

Author(s)

Jianhong Ou

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)))
```

interactiveViewer *plot the tracks with GUI*

Description

interactively plot the tracks

Usage

```
interactiveViewer(trackList, chromosome, start, end, strand,
                 viewerStyle = trackViewerStyle(), autoOptimizeStyle = FALSE)
```

Arguments

| | |
|-------------------|--|
| trackList | an object of trackList |
| chromosome | chromosome |
| start | start position |
| end | end position |
| strand | strand |
| viewerStyle | an object of trackViewerStyle |
| autoOptimizeStyle | should use optimizeStyle to optimize style |

Author(s)

Jianhong Ou

See Also

See Also as [viewTracks](#)

Examples

```
if(interactive()){  
  interactiveViewer()  
}
```

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

Description

Automatic optimize the stlye of trackViewer

Usage

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

Arguments

| | |
|-------------|---|
| trackList | An object of trackList |
| viewerStyle | An object of trackViewerStyle |

Value

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

Author(s)

Jianhong Ou

See AlsoSee 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

```

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 GRanges |
| range | an object of GRanges |
| viewerStyle | an object of trackViewerStyle |
| autoOptimizeStyle | should use optimizeStyle to optimize style |
| newpage | should be draw on a new page? |

ValueAn object of [viewport](#) for [addGuideLine](#)**Author(s)**

Jianhong Ou

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)))
```

pos-class

Class "pos"

Description

An object of class "pos" represents a point location

Objects from the Class

Objects can be created by calls of the form `new("pos", x, y, unit)`.

Slots

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

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

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

track-class

Class "track"

Description

An object of class "track" represents scores of a given track.

Usage

```
## S4 method for signature 'track,character,ANY'
setTrackStyleParam(ts, attr, value)
## S4 method for signature 'track,character,ANY'
setTrackXscaleParam(ts, attr, value)
## S4 method for signature 'track,character,ANY'
setTrackYaxisParam(ts, attr, value)
```

Arguments

| | |
|-------|---|
| ts | An object of track. |
| attr | the name of slot of <code>trackStyle</code> object to be changed. |
| value | values to be assigned. |

Objects from the Class

Objects can be created by calls of the form `new("track", dat, dat2, type, format, style, name)`.

Slots

| | |
|--------|--|
| dat | Object of class <code>GRanges</code> the scores of a given track. It should contain score metadata. |
| dat2 | Object of class <code>GRanges</code> the scores of a given track. It should contain score metadata. When dat2 and dat is paired, dat will be drawn as positive value where dat2 will be drawn as negative value (-1 * score) |
| type | The type of track. It could be 'data' or 'gene'. |
| format | The format of the input. It could be "BED", "bedGraph", "WIG", "BigWig" or "BAM" |
| style | Object of class <code>trackStyle</code> |
| name | unused yet |

Methods

| | |
|----------------------------|---|
| setTrackStyleParam | change the slot values of <code>trackStyle</code> object for an object of track |
| setTrackXscaleParam | change the <code>xscale</code> slot values for an object of track |
| setTrackYaxisParam | change the <code>yaxisStyle</code> values for an object of track |
| \$, \$<- | Get or set the slot of <code>track</code> |
| show | show the details of <code>track</code> |

See Also

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

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))
```

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

Description

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

constructor

trackList(..., heightDist=NA): 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 [track](#). The heightDist is 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.

Objects from the Class

Objects can be created by calls of the form new("trackStyle", tracktype, color, height, marginTop, marginBottom)

Slots

tracktype "character" track type, could be peak or cluster. Default is "peak". "cluster" is not supported yet.

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

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' or 'bottomright'.

ylablas "numeric" y lable 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.

```
trackViewerStyle-class
      Class "trackViewerStyle"
```

Description

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

Usage

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

Arguments

| | |
|-------|---------------------------------|
| tvs | An object of trackViewerStyle. |
| attr | the name of slot to be changed. |
| value | values to be assigned. |

Objects from the Class

Objects can be created by calls of the form `new("trackViewerStyle", margin, xlas,`

xgp

constructor

`trackViewerStyle(...)`: Each argument in ... becomes an slot in the new trackViewerStyle.

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
autolas "logical" automatic determine y label direction
flip "logical" flip the x-axis or not, default FALSE

Methods

setTrackViewerStyleParam change the slot values of an object of trackViewerStyle

Examples

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

| | |
|------------|------------------------|
| viewTracks | <i>plot the tracks</i> |
|------------|------------------------|

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)
```

Arguments

| | |
|-------------------|--|
| trackList | an object of trackList |
| chromosome | chromosome |
| start | start position |
| end | end position |
| strand | strand |
| gr | an object of GRanges |
| ignore.strand | ignore the strand or not when do filter. default TRUE |
| viewerStyle | an object of trackViewerStyle |
| autoOptimizeStyle | should use optimizeStyle to optimize style |
| newpage | should be draw on a new page? |

Value

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

Author(s)

Jianhong Ou

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(tracks, fox2), gr=gr, autoOptimizeStyle=TRUE)

```

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

Description

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

Objects from the Class

Objects can be created by calls of the form `new("xscale", from, to, label, gp, draw)`.

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 | <i>Class "yaxisStyle"</i> |
|------------------|---------------------------|

Description

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

Objects from the Class

Objects can be created by calls of the form `new("yaxisStyle", at, label, gp, draw, main)`.

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

*Topic **\textasciitildemisc**

addArrowMark, 3

*Topic **classes**

pos-class, 12

track-class, 12

trackList-class, 14

trackStyle-class, 14

trackViewerStyle-class, 15

xscale-class, 17

yaxisStyle-class, 17

*Topic **importData**

coverageGR, 5

geneModelFromTxdb, 6

importBam, 7

importScore, 8

*Topic **misc**

addGuideLine, 4

getCurTrackViewport, 7

*Topic **package**

trackViewer-package, 2

*Topic **plot**

interactiveViewer, 9

optimizeStyle, 10

plotGRanges, 11

viewTracks, 16

\$, track-method (track-class), 12

\$<-, track-method (track-class), 12

addArrowMark, 3, 4, 12, 16

addGuideLine, 4, 4, 7, 11, 12, 16

arrow, 4

coverage, 5

coverageGR, 5

GAlignmentPairs, 5

GAlignments, 5

geneModelFromTxdb, 6

getCurTrackViewport, 4, 7

gpar, 14, 15, 17, 18

GRanges, 5, 8, 11, 13, 16

importBam, 6, 7, 9, 13

importScore, 6, 8, 8, 13

interactiveViewer, 9

numeric, 12

optimizeStyle, 10, 10, 11, 16

par, 14, 15

plotGRanges, 11

pos, 17

pos (pos-class), 12

pos-class, 12

readGAlignments, 8

setTrackStyleParam (track-class), 12

setTrackStyleParam, track, character, ANY-method
(track-class), 12

setTrackStyleParam, track, character-method
(track-class), 12

setTrackViewerStyleParam
(trackViewerStyle-class), 15

setTrackViewerStyleParam, trackViewerStyle, character, ANY-me
(trackViewerStyle-class), 15

setTrackViewerStyleParam, trackViewerStyle, character-method
(trackViewerStyle-class), 15

setTrackXscaleParam (track-class), 12

setTrackXscaleParam, track, character, ANY-method
(track-class), 12

setTrackXscaleParam, track, character-method
(track-class), 12

setTrackYaxisParam (track-class), 12

setTrackYaxisParam, track, character, ANY-method
(track-class), 12

setTrackYaxisParam, track, character-method
(track-class), 12

show, track-method (track-class), 12

track, [6–9](#), [13](#), [14](#)
track (track-class), [12](#)
track-class, [12](#)
trackList, [10](#), [16](#)
trackList (trackList-class), [14](#)
trackList-class, [14](#)
trackStyle, [13](#)
trackStyle (trackStyle-class), [14](#)
trackStyle-class, [14](#)
trackViewer (trackViewer-package), [2](#)
trackViewer-package, [2](#)
trackViewerStyle, [7](#), [10](#), [11](#), [16](#)
trackViewerStyle
 (trackViewerStyle-class), [15](#)
trackViewerStyle-class, [15](#)
TxDb, [6](#)

unit, [12](#)

viewport, [7](#), [11](#), [16](#)
viewTracks, [3](#), [4](#), [6](#), [8–11](#), [16](#)

xscale, [13](#), [14](#)
xscale (xscale-class), [17](#)
xscale-class, [17](#)

yaxisStyle, [13–15](#)
yaxisStyle (yaxisStyle-class), [17](#)
yaxisStyle-class, [17](#)