

# gTrack

May 18, 2015

**Title** Plotting multiple tracks of complex genomic data across multiple genomic windows.

**Version** 0.0.0.9000

**Description** Object for plotting GRanges, RleList, UCSC file formats, and ffTrack objects in multi-track panels.

**Depends** R (>= 3.1.0), GenomicRanges (>= 1.8), grUtils

**License** GNU GPLv3

**LazyData** true

## R topics documented:

gTrack-class . . . . .	1
karyogram . . . . .	8
track.gencode . . . . .	8
track.refgene . . . . .	9
track.splice . . . . .	10
<b>Index</b>	<b>12</b>

---

gTrack-class	<i>S4 class for gTrack</i>
--------------	----------------------------

---

## Description

Class gTrack defines a subsettable object that wraps formatting information around ranged genomic data (GRanges, GRangesList, RleLists, ucsc formats on file system) and can be displayed in a "genome browser" style plot.

Arguments described as "formatting" are vectors. They are replicated (if necessary) to match the length of the object.

Subsetting gTrack

Accessing submatrices of mdata for length 1 gTrack

Accessing columns of gTrack formatting data.frame

Setting columns of gTrack formatting data.frame

Getting length of gTrack object

Computes the footprint of the track's data on the genome  
 returns Seqinfo of gTrack object  
 sets Seqinfo of gTrack object by fixing its underlying componennts  
 gets lengths of data objects inside trackData  
 Concatenate gTrack objects  
 Get data frame specifying formatting of gTrack objects  
 Get edges list associated with gTrack object  
 Get edges list associated with gTrack object  
 Set edges list associated with gTrack object  
 Clear data from gTrack object (makes into an empty container with the same seqinfo and display features)  
 Extract data list from gTrack object  
 Formatting  
 td is a gTrack object formatting(td)\$height <- 2  
 Get colormap of gTrack object  
 Set colormap of gTrack object  
 Displays gTrack object in a multi-track genome browser view across one or more windows, with signed rearrangement edges optionally plotted over the top.

### Usage

```

gTrack(...)

## S4 method for signature 'gTrack,ANY'
x[i]

mdata(x, ...)

## S4 method for signature 'gTrack'
mdata(x, igr = NULL, jgr = igr)

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

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

## S4 method for signature 'gTrack'
length(x)

## S4 method for signature 'gTrack'
reduce(x, ...)

## S4 method for signature 'gTrack'
seqinfo(x)

seqinfo(.Object) <- value
  
```

```
## S4 replacement method for signature 'gTrack'
seqinfo(.Object) <- value

lengths(x)

## S4 method for signature 'gTrack'
lengths(x)

## S4 method for signature 'gTrack'
c(x, ..., recursive = FALSE)

formatting(.Object, ...)

## S4 method for signature 'gTrack'
formatting(.Object)

edgs(.Object)

## S4 method for signature 'gTrack'
edgs(.Object)

vars(.Object)

## S4 method for signature 'gTrack'
vars(.Object)

edgs(.Object) <- value

## S4 replacement method for signature 'gTrack'
edgs(.Object) <- value

clear(.Object, ...)

## S4 method for signature 'gTrack'
clear(.Object)

dat(.Object, ...)

## S4 method for signature 'gTrack'
dat(.Object)

formatting(.Object) <- value

## S4 replacement method for signature 'gTrack'
formatting(.Object) <- value

colormap(.Object, value)

## S4 method for signature 'gTrack'
colormap(.Object)

colormap(.Object) <- value
```

```
## S4 replacement method for signature 'gTrack'
colormap(.Object) <- value

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

display(.Object, ...)

## S4 method for signature 'gTrack'
display(.Object, windows = seqinfo(.Object),
  win.gap = NULL, y.heights = NULL, y.gaps = NULL, links = NULL,
  cex.xlabel = 1, cex.ylabel = 1, max.ranges = NA, links.feats = NULL,
  pintersect = FALSE, verbose = FALSE, ...)
```

### Arguments

...	additional arguments to GRanges reduce function
name	vector or scalar character specifying name of this track, which will be displayed on label to the left of the track
windows	GRanges specifying windows to view (can also be GRangesList), default is whole genome
win.gap	scalar numeric (or vector if windows is GRangesList) specifying genome-coordinate gap between windows
y.heights	vector specifying relative heights of (stacked) windows, only relevant if windows is GRangesList
y.gaps	vector specifying relative heights of gaps below and above (stacked) windows, only relevant if windows is GRangesList
links	GRangesList of signed locus pairs specifying links to draw above the plot, optional GRangesList values meta data specify formatting of individual links: \$label (text label), \$col (line color), \$lwd (line weight), \$lty (line style), \$arrow (arrow flag), \$col.arrow (arrow color), \$v (vertical spline bulge), \$w (horizontal spline bulge)
cex.xlabel	scalar numeric > 0 specifying character expansion of x axis labels (not yet implemented, use gTrack formatting for now)
cex.ylabel	scalar numeric > 0 specifying character expansion of y tick labels (not yet implemented, use gTrack formatting for now)
max.ranges	vector or scalar numeric specifying what is the max number of ranges to plot in a window (formatting)
links.feats	data.frame specifying links features, must have 1 row or have length(links), will over-ride links meta data (see links)
data	This is an instance of one of the following objects: (1) GRanges (2) GRangesList (3) ffTrack or (4) character representing Bed, Wig, BigWig, or .rds GRanges file. It can also be a list of the above (if specifying multiplying tracks).
y.field	vector or scalar character specifying meta data field of GRanges or GRangesList to use for "y axis" coordinate when plotting numeric tracks, (note: for a RleList or ffTrack this field is automatically set to "score"), default is NA for non-numeric tracks
height	vector or scalar numeric specifying height of track(s) (in relative units)

ygap	vector or scalar numeric specifying gap between tracks
stack.gap	vector or scalar numeric specifying x gap between stacking non-numeric GRanges or GrangesLists items in track(s)
ywid	vector or scalar numeric specifying the y-extent of individual ranges (in local plot coordinates)
col	vector or scalar character specifying static color for track(s), if NA then \$col meta data field of GRanges / GRangesList will override
border	vector or scalar character specifying static border for polygons in track(s), if NA then \$border meta field of GRanges / GRangesList is used
angle	vector of scalar numeric specifying angle of polygons that represent signed ranges
split	whether to split when lifting
lift	vector or scalar logical flag specifying whether to lift this track to other chained-Track items (only relevant if used within chainedTracks object)
split	vector or scalar logical flag specifying whether to split when lifting (only relevant if used within chainedTracks object)
colormaps	length(.Object) length named list of named vectors whose entry i maps unique value of a data field to colors. The data.field is specified by the name of the list entry, and the unique values / colors are specified by the named vector.
edges	Data frame of columns \$from, \$to, and optional fields \$col, \$lwd, and \$lty, specifying edges linking data items. Also can be a list of the above if specifying multiple tracks (and must be compatible in length with data arg)
xaxis.suffix	vector or scalar numeric specifying the suffix that will be used in describing x axis coordinates (TODO: move to display) (formatting)
xaxis.unit	vector or scalar numeric specifying the unit that will be used in describing x axis coordinates (TODO: move to display) (formatting)
xaxis.round	vector or scalar non-neg integer specifying number of decimals to round xaxis coordinate labels (formatting)
xaxis.nticks	vector or scalar positive integer specifying how many xaxis ticks to optimally draw (formatting)
xaxis.label.angle	vector or scalar numeric between 0 and 360 specifying angle with which to draw xaxis coordinate labels (formatting)
xaxis.newline	vector or scalar logical specifying whether to draw a newline in the xaxis coordinate labels (formatting)
y.grid.col	vector or scalar character specifying color of "gridlines" used to specify numeric track data (formatting)
y.grid.cex	vector or scalar non-neg numeric specifying character expansion for y tick / y grid labels (formatting)
y.grid.lty	vector or scalar positive integer specifying line style of y grid lines for numeric tracks (formatting)
y.grid.lwd	vector or scalar positive integer specifying thickness of y grid lines for numeric tracks (formatting)
y.grid.labx	vector or scalar positive integer specifying fraction of xlim left of plot to place y axis labels (formatting)

lwd.border	vector or scalar integer specifying the thickness of the polygon borders (formatting)
cex.label	vector or scalar numeric specifying the expansion factor of the range labels (formatting)
hadj.label	vector or scalar numeric specifying the horizontal adjustment of the range labels (formatting)
vadj.label	vector or scalar numeric specifying the vertical adjustment of the range labels (formatting)
ypad	vector or scalar numeric between 0 and 1 specifying how much whitespace padding to add within panel (formatting)
circles	vector or scalar logical specifying whether to scatter plot range data (formatting)
lines	vector or scalar logical specifying whether to line plot range data (formatting)
bars	vector or scalar logical specifying whether to bar plot range data (formatting)
source.file.chrsub	vector or scalar logical specifying whether or not sub "chr" out of any external files (e.g. UCSC style files) (formatting)
y0.bar	vector or scalar numeric specifying where to draw the lower boundary of a bar in a bar plot (only applicable if bars == T) (formatting)
yaxis	vector or scalar logical specifying whether to print yaxis (formatting)
yaxis.pretty	vector or scalar positive integer specifying how many ticks to optimally draw on yaxis (formatting)
draw.var	vector or scalar logical specifying whether to draw.var for GRanges / GRanges-List specifying reads (GRanges must contain \$cigar +/- \$MD field) (formatting)
draw.paths	vector or scalar logical specifying whether to interpret GRangesLists as "paths" and connect them with a set of spline curves. (formatting)
path.col	vector or scalar character specifying color of path (only applicable for tracks in which draw.paths = T) (formatting)
path.col.arrow	vector or scalar character specifying color of arrow of path (only applicable for tracks in which draw.paths = T) (formatting)
path.cex.arrow	vector or scalar numeric > 0 specifying expansion factor of arrow of path (only applicable for tracks in which draw.paths = T) (formatting)
path.stack.y.gap	vector or scalar numeric > 0 specifying y stack gap of paths (only applicable for tracks in which draw.paths = T) (formatting)
path.stack.x.gap	vector or scalar numeric > 0 specifying x stack gap for paths (only applicable for tracks in which draw.paths = T) (formatting)
path.cex.v	vector or scalar numeric > 0 specifying vertical bulge of spline in paths (only applicable for tracks in which draw.paths = T) (formatting)
path.cex.h	vector or scalar numeric > 0 specifying horizontal bulge of spline in paths (formatting) (only applicable for tracks in which draw.paths = T) (formatting)
draw.backbone	vector or scalar logical specifying whether to draw "backbone" connecting different items in a GRangesList item (formatting)
gr.labelfield	vector or scalar character specifying which GRanges meta data field to use for GRanges label (default "label") (formatting)

<code>gr1.labelfield</code>	vector or scalar character specifying which GRanges meta data field to use for GRangesList label (default "label") (formatting)
<code>legend</code>	vector or scalar logical specifying whether to draw a legend for this track (formatting)
<code>legend.xpos</code>	vector or scalar numeric between 0 and 1 specifying what relative x position in the plot to place the legend for this track (formatting)
<code>legend.ypos</code>	vector or scalar numeric between 0 and 1 specifying what relative y position in the plot to place the legend for this track (formatting)
<code>legend.ncol</code>	vector or scalar positive integer specifying how many columns to put in legend (formatting)
<code>legend.xjust</code>	vector or scalar of 0, 1, 2 specifying x justification of legend (formatting)
<code>legend.yjust</code>	vector or scalar of 0, 1, 2 specifying y justification of legend (formatting)
<code>legend.maxitems</code>	vector or scalar positive integer specifying what is the maximum number of items to include in legend (formatting)
<code>label.suppress</code>	vector or scalar logical flag specifying whether to suppress all GRanges / GRangesList label drawing (formatting)
<code>label.suppress.gr</code>	vector or scalar logical flag specifying whether to suppress GRanges label drawing (formatting)
<code>label.suppress.gr1</code>	vector or scalar logical flag specifying whether to suppress GRangesList label drawing (formatting)
<code>cex.tick</code>	vector or scalar numeric specifying expansion factor for axis tick labels (formatting)
<code>cex.tick.len</code>	vector or scalar numeric specifying lengths for axis ticks (formatting)
<code>gr.cex.label</code>	vector or scalar numeric specifying GRanges label character expansion (default is <code>cex.label</code> ) (formatting)
<code>gr.srt.label</code>	vector or scalar numeric between 0 and 180 specifying rotation of GRanges labels (formatting)
<code>sep.lty</code>	vector or scalar integer specifying line style for window separators (formatting)
<code>sep.lwd</code>	vector or scalar numeric specifying line thickness for window separators (formatting)
<code>bg.col</code>	vector or scalar character specifying window background color (formatting)
<code>seqinfo</code>	Seqinfo object specifying seqinfo of this object (usually not specified since <code>seqinfo</code> is inferred from data argument)
<code>cex.tick</code>	vector specifying relative heights of gaps below and above (stacked) windows, only relevant if windows is GRangesList
<code>max.ranges</code>	scalar numeric > 0 specifying max number of ranges to draw in a window (via sampling). If specified, overrides gTrack max.ranges formatting feature.

## Details

can also just use `$` directly `td$height <- 2`

**Slots**

**data** length(.Object) length list containing genomic data (e.g. GRanges, GrangesLists, RleLists, path to ucsc file or ffTrack file on disk)

**seqinfo** Seqinfo object

**colormap** length(.Object) length named list of named vectors whose entry i specifies the colormap for the meta data field of object entry i (specified by the name) and maps unique value of that data field to colors (specified by the named vector)

**edges** list of data.frames of length length(.Object) which has columns \$from, \$to, and optional fields \$col, \$lwd, and \$lty to specify splined edges joining data items in the corresponding track

---

karyogram	Returns gTrack displaying colormap for ucsc hg18 or hg19 karyotype
-----------	--

---

**Description**

Returns gTrack displaying colormap for ucsc hg18 or hg19 karyotype

**Usage**

```
karyogram(hg19 = T, bands = T, arms = T, tel.width = 2e+06, ...)
```

**Arguments**

hg19	logical scalar flag, if T returns gTrack for hg19
bands	logical scalar, if T returns gTrack with colored giemsa bands
arms	logical scalar, if T and bands F, returns chromosome arms with different colors and centromeres and telomeres marked, if arms is F and bands F returns chromosomes, each with a different color
tel.width	numeric scalar, specifies telomere width in bases (only relevant if arms = T, bands = F)

---

track.gencode	Returns gTrack object representing GENCODE transcripts and their components (utr, cds etc) with assigned colors. Usually built from cached data objects but can also be built from provided GRangesList
---------------	---

---

**Description**

Returns gTrack object representing GENCODE transcripts and their components (utr, cds etc) with assigned colors. Usually built from cached data objects but can also be built from provided GRangesList



## Usage

```
track.gencode(gencode = NULL, gene.collapse = T, genes = NULL,
  grep = NULL, bg.col = alpha("blue", 0.1), cds.col = alpha("blue", 0.6),
  utr.col = alpha("purple", 0.4), st.col = "green", en.col = "red",
  grl.labelfield, gr.labelfield, col, cached = T,
  cached.path = system.file("extdata", "gencode.composite.rds", package =
    "gTrack"), cached.path.collapsed = system.file("extdata",
    "gencode.composite.collapsed.rds", package = "gTrack"), gr.srt.label = 0,
  gr.cex.label = 0.3, cex.label = 0.5, labels.suppress.gr = T,
  drop.rp11 = TRUE, stack.gap = 1e+06, ...)
```

## Arguments

gene.collapse	scalar logical specifying whether to collapse genes by transcript (or used stored version of transcripts)
genes	(optional) character vector specifying genes to limit gTrack object to
bg.col	scalar character representing background color for genespan
cds.col	scalar character representing background color for CDS
st.col	scalar character representing color of CDS start
en.col	scalar character representing color of CDS end
cached	logical scalar whether to use "cached" version provided with package
gr.srt.label	scalar numeric specifying angle on exon label
gr.cex.label	scalar numeric > 0 specifying character expansion on exon label
labels.suppress.gr	scalar logical specifying whether to suppress exon label plotting
stack.gap	stack.gap argument to gTrack
...	additional arguments passed down to gTrack
rg	(optional) GRangesList representing transcript models imported from GENCODE gff3 file using rtracklayer import
cds.utr	scalar character representing background color for UTR

---

track.refgene	<i>Returns gTrack object representing refGene transcripts and their components (utr, cds etc) with assigned colors. Usually built from cahced data objects but can also be built from provided GRangesList</i>
---------------	--

---

## Description

Returns gTrack object representing refGene transcripts and their components (utr, cds etc) with assigned colors. Usually built from cahced data objects but can also be built from provided GRanges-List

**Usage**

```
track.refgene(rg = NULL, gene.collapse = T, genes = NULL,
  bg.col = alpha("blue", 0.1), cds.col = alpha("blue", 0.6),
  utr.col = alpha("purple", 0.4), st.col = "green", en.col = "red",
  genespan = T, utr = T, cds = T, grl.labelfield, gr.labelfield, col,
  cached = T, cached.path = system.file("extdata", "refgene.composite.rds",
  package = "gTrack"), cached.path.collapsed = system.file("extdata",
  "refgene.composite.collapsed.rds", package = "gTrack"), gr.srt.label = 0,
  gr.cex.label = 0.8, labels.suppress.gr = T, stack.gap = 1e+06, ...)
```

**Arguments**

rg	(optional) GRangesList representing transcript models obtained from refgene, with GrangesList meta data fields chr, s1, s2, e1, e2, str, gene_sym, Uniprot,
gene.collapse	scalar logical specifying whether to collapse genes by transcript (or used stored version of transcripts)
genes	(optional) character vector specifying genes to limit gTrack object to
bg.col	scalar character representing background color for genespan
cds.col	scalar character representing background color for CDS
st.col	scalar character representing color of CDS start
en.col	scalar character representing color of CDS end
genespan	logical scalar whether to include genespan range around entire gne
utr	logical scalar whether to include range specifying UTR
cds	logical scalar whether to include range specifying CDS
cached	logical scalar whether to use "cached" version provided with package
gr.srt.label	scalar numeric specifying angle on exon label
gr.cex.label	scalar numeric > 0 specifying character expansion on exon label
labels.suppress.gr	scalar logical specifying whether to suppress exon label plotting
stack.gap	stack.gap argument to gTrack
...	additional arguments passed down to gTrack
cds.utr	scalar character representing background color for UTR

---

track.splice	<i>Given set of exons and rna bam (eg from tophat) determines junction and exon read density and returns a gTrack object of splicing graph</i>
--------------	--

---

**Description**

Given set of exons and rna bam (eg from tophat) determines junction and exon read density and returns a gTrack object of splicing graph

**Usage**

```
track.splice(ex = NULL, region = NULL, bam, verbose = TRUE,
  infer.exons = FALSE, min.reads = 0, min.exon.width = 10,
  max.exon.width = 1000)
```

**Arguments**

ex	GRanges of candidate exons
bam	path to indexed RNA seq bam
verbose	

# Index

[\[,gTrack,ANY-method \(gTrack-class\), 1](#)  
[\[,gTrack-class \(gTrack-class\), 1](#)  
[\\$,gTrack-class \(gTrack-class\), 1](#)  
[\\$,gTrack-method \(gTrack-class\), 1](#)  
[\\$<-,gTrack-class \(gTrack-class\), 1](#)  
[\\$<-,gTrack-method \(gTrack-class\), 1](#)  
  
[c,gTrack-class \(gTrack-class\), 1](#)  
[c,gTrack-method \(gTrack-class\), 1](#)  
[clear \(gTrack-class\), 1](#)  
[clear,gTrack-class \(gTrack-class\), 1](#)  
[clear,gTrack-method \(gTrack-class\), 1](#)  
[colormap \(gTrack-class\), 1](#)  
[colormap,gTrack-class \(gTrack-class\), 1](#)  
[colormap,gTrack-method \(gTrack-class\), 1](#)  
[colormap<- \(gTrack-class\), 1](#)  
[colormap<- ,gTrack-method \(gTrack-class\), 1](#)  
  
[dat \(gTrack-class\), 1](#)  
[dat,gTrack-class \(gTrack-class\), 1](#)  
[dat,gTrack-method \(gTrack-class\), 1](#)  
[display \(gTrack-class\), 1](#)  
[display,gTrack-class \(gTrack-class\), 1](#)  
[display,gTrack-method \(gTrack-class\), 1](#)  
  
[edges \(gTrack-class\), 1](#)  
[edges,gTrack-class \(gTrack-class\), 1](#)  
[edges,gTrack-method \(gTrack-class\), 1](#)  
[edges<- \(gTrack-class\), 1](#)  
[edges<- ,gTrack-class \(gTrack-class\), 1](#)  
[edges<- ,gTrack-method \(gTrack-class\), 1](#)  
  
[formatting \(gTrack-class\), 1](#)  
[formatting,gTrack-class \(gTrack-class\), 1](#)  
[formatting,gTrack-method \(gTrack-class\), 1](#)  
[formatting<- \(gTrack-class\), 1](#)  
[formatting<- ,gTrack-class \(gTrack-class\), 1](#)  
[formatting<- ,gTrack-method \(gTrack-class\), 1](#)  
  
[gTrack \(gTrack-class\), 1](#)

[gTrack-class, 1](#)  
  
[karyogram, 8](#)  
  
[length,gTrack-class \(gTrack-class\), 1](#)  
[length,gTrack-method \(gTrack-class\), 1](#)  
[lengths \(gTrack-class\), 1](#)  
[lengths,gTrack-class \(gTrack-class\), 1](#)  
[lengths,gTrack-class#' \(gTrack-class\), 1](#)  
[lengths,gTrack-method \(gTrack-class\), 1](#)  
  
[mdata \(gTrack-class\), 1](#)  
[mdata,gTrack-method \(gTrack-class\), 1](#)  
  
[reduce,gTrack-class \(gTrack-class\), 1](#)  
[reduce,gTrack-method \(gTrack-class\), 1](#)  
  
[seqinfo,gTrack-class \(gTrack-class\), 1](#)  
[seqinfo,gTrack-method \(gTrack-class\), 1](#)  
[seqinfo<- \(gTrack-class\), 1](#)  
[seqinfo<- ,gTrack-class \(gTrack-class\), 1](#)  
[seqinfo<- ,gTrack-method \(gTrack-class\), 1](#)  
  
[show,gTrack-class \(gTrack-class\), 1](#)  
[show,gTrack-method \(gTrack-class\), 1](#)  
  
[td.karyogram,gTrack-class \(karyogram\), 8](#)  
[track.gencode, 8](#)  
[track.gencode,gTrack-class \(track.gencode\), 8](#)  
[track.refgene, 9](#)  
[track.refgene,gTrack-class \(track.refgene\), 9](#)  
[track.splice, 10](#)  
[track.splice,gTrack-class \(track.splice\), 10](#)  
  
[vars \(gTrack-class\), 1](#)  
[vars,gTrack-class \(gTrack-class\), 1](#)  
[vars,gTrack-method \(gTrack-class\), 1](#)