Package: NetWeaver (via r-universe)

September 10, 2024

Type Package

Title Graphic Presentation of Complex Genomic and Network Data Analysis

Version 0.0.9

Date 2023-07-27

Author Minghui Wang, Bin Zhang

Maintainer Minghui Wang <`m.h.wang@live.com`>

Description Implements various simple function utilities and flexible pipelines to generate circular images for visualizing complex genomic and network data analysis features.

Depends R (>= 3.3.0)

License GPL-3

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL `https://github.com/mw201608/NetWeaver/`

BugReports `https://github.com/mw201608/NetWeaver/issues`

Repository https://mw201608.r-universe.dev

RemoteUrl https://github.com/mw201608/netweaver

RemoteRef HEAD

RemoteSha 461b42bd717db301a9d503103ca189427b9a75c7

Contents

`ensemble_rank` *Ensemble Rank*

Description

Compute composite rank score using evidence from multiple features.

Usage

```
ensemble_rank(x, method=c('ProductOfRank','MeanOfLog','MeanOfLogLog','Mean'),
small=1.0e-320, standardize=TRUE)
```

Arguments

`x`	a matrix of discriminant values (e.g., P values) measuring the strength of the association between objects (eg modules) (in rows) and variables/features (in columns).
`method`	a character string specifyign the ranking metric. See `Details`.
`small`	offset the small p values before taking log transformation.
`standardize`	whether to rescale the final ranking score by dividing the maximum value.

Details

Three ranking metrics are currently implemented:

ProductOfRank is the one used in Zhang et al. (Cell 2013, 153: 707-720) with a slight change to standardize the scores to be between 0 and 1;

MeanOfLog computes the mean of -log(p value) which penalizes insignificant p values;

MeanOfLogLog computes the mean of log(-log(p value)) which penalizes insignificant p values and shrinks the difference in the ranges of p values.

Means computes the row mean of absolute values of `x`.

Value

A vector of ranking scores. The larger the score, the higher the rank.

Author(s)

Minghui Wang <m.h.wang@live.com>

Examples

```
#Rank US cities by the significance test of excessive arrests compared to average
#compute the p value of one-tailed z-test
x=apply(USArrests,2,function(x) pnorm((x-mean(x))/sd(x),lower.tail=FALSE))
#compute ranking score using three different metrics
score1=ensemble_rank(x,method='ProductOfRank')
score2=ensemble_rank(x,method='MeanOfLog')
score3=ensemble_rank(x,method='MeanOfLogLog')
#plot the top 5 worst cities by each metric
par(mfrow=c(1,3))
barplot(sort(score1,decreasing=TRUE)[1:5],las=2)
barplot(sort(score2,decreasing=TRUE)[1:5],las=2)
barplot(sort(score3,decreasing=TRUE)[1:5],las=2)
```

`getrankp` *getrankp*

Description

Compute probability (fraction) of values in a large population more extreme than input series.

Usage

```
getrankp(x, y, truncated.size=0)
```

Arguments

`x`	A vector of input series.
`y`	A vector of population values.
`truncated.size`	See `Details`. Only use when you know this.

Details

This function can be used to compute permutation-based false discovery rate (two-tailed). In such case, `x` contains test statistics from observed data while `y` contains test statistics from permutation. `x` could be a truncated set, with small values discarded to save memory.

Value

A vector of the same size as `x`.

Author(s)

Minghui Wang <m.h.wang@live.com>

Examples

```
x=abs(rnorm(100))
y=abs(rnorm(10000))
getrankp(x,y)[1:4]
```

`is.overlap` *Region overlap test*

Description

Test if two regions are overlapping.

Usage

```
is.overlap(x,y)
```

Arguments

`x`	a vector of two numeric values specifying the start and end positions of the first region.
`y`	a vector of two numeric values specifying the start and end positions of the second region

Value

Logic test output.

Author(s)

Minghui Wang <m.h.wang@live.com>

Examples

```
is.overlap(c(10,100),c(90,120))
is.overlap(c(10,100),c(110,120))
```

`Modules` *Example Module Dataset*

Description

This example dataset and sample code illustrate the use of `NetWeaver` to visualize the complex features of gene coexpression network modules.

Usage

```
data(Modules)
```

Details

This dataset contains a data.frame called `Modules`. Each row is a module, with the module id in the first column. The second column is ranking score. The next 4 columns are coefficients of module-trait correlations. The rest columns are P values of enrichment for various gene signatures. The sample R code shows a pipeline for visualizing the module feature data using a circos style plot.

Author(s)

Minghui Wang <m.h.wang@live.com>, Bin Zhang <bin.zhang@mssm.edu>

See Also

`rc.initialize`

Examples

```
## Not run:
#see a tutorial for how to plot this dataset
vignette("netweaver")

## End(Not run)
```

`NetWeaver` *Graphic Presentation of Complex Genomic and Network Data Analysis*

Description

`NetWeaver` is motivated towards developing a simple and flexible pipeline for visualizing the complex features of enrichment and correlation of gene coexpression network modules. While circos style 2D track plot is one natural choice for such practice, existing packages are designed primarily for handling genome structure and intervals. They are either too complicated to use, requiring certain level of knowledge of scripting, or limited in applications to only genomic structure data. To address these issues, particularly extend beyond applications in genomic structure data, `NetWeaver` offers a lightweight implementation of circular track plot, providing simple and flexible R function utilities and pipelines to generate circular images for visualizing different types of structure/relationship data.

The original version of this package was developed for Figure 7 of Wang et al (2016) *Genome Medicine* 8:104, which illustrates more than 20 properties for 50 coexpression network modules with a circular track plot. Please try to cite the paper when you use results from this software in a publication:

Wang M, Roussos P, McKenzie A, Zhou X, Kajiwara Y, Brennand K, DeLuca GC, Crary JF, Casaccia P, Buxbaum J et al. 2016. Integrative Network Analysis of Nineteen Brain Regions Identifies Molecular Signatures and Networks Underlying Selective Regional Vulnerability to Alzheimer's Disease. *Genome Medicine* 8: 104.

Details

There are two sample pipelines available:

1. Analysis of a real dataset of gene coexpression network modules can be reached through `vignette("netweaver")`.
2. Analysis of a hypothetical data is shown in sample code of `rc.initialize`.

Author(s)

Minghui Wang <m.h.wang@live.com>, Bin Zhang <bin.zhang@mssm.edu>

References

Wang M, Roussos P, McKenzie A, Zhou X, Kajiwara Y, Brennand K, DeLuca GC, Crary JF, Casaccia P, Buxbaum J et al. 2016. Integrative Network Analysis of Nineteen Brain Regions Identifies Molecular Signatures and Networks Underlying Selective Regional Vulnerability to Alzheimer's Disease. *Genome Medicine* 8: 104.

See Also

`rc.initialize`, `Modules`

`rc.get.coordinates` *Get Coordinates*

Description

Retrieve x and y coordinates in a track.

Usage

```
rc.get.trackCoordinates(track.id, Start, End, Chr=NULL, degree=NULL, trackThickness=NULL)
rc.get.coordinates(track.id, Pos, Chr=NULL, degree=NULL, innerSide=TRUE, bottomSide=TRUE,
 adjust.origin=TRUE)
```

Arguments

`track.id`	number of track from outermost.
`Start`, `End`	start/end position on a chromosome or since first chromosome. See `Details`.
`Pos`	a vector of positions on a chromosome or since first chromosome. See `Details`.
`Chr`	a chromosome id. See `Details`.
`degree`	the angle of the arc rotation, overwrites `Start`, `End`, `Pos` and `Chr`.
`trackThickness`	thickness (height) of a track.
`innerSide`	whether to compute coordinates on the inner side of a track. Outer side coordinates will be returned if FALSE. Used for `circular` layout.
`bottomSide`	whether to compute coordinates on the bottom side of a track. Upper side coordinates will be returned if FALSE. Used for `landscape` layout.
`adjust.origin`	whether to relocate origin when plotting only a slice/section of the full 360 degree circle. Used internally for `circular` layout.

Details

If `Chr` is `NULL`, `Start`, `End` and `Pos` will be considered as cumulative positions since first chromosome.

Value

A list of x and y coordinates.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.initialize`, `rc.reset.params`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.get.params` *Get Parameters*

Description

Retrieve parameters of Circos plot after initialization.

Usage

```
rc.get.params()
```

Value

A list.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.initialize`, `rc.reset.params`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.initialize` *Initialize Circos Plot Parameters*

Description

Setup default parameter for Circos plot.

Usage

```
rc.initialize(cyto.info, num.tracks=NULL, chr.order=NULL,
 stepUnit=10^7, Layout=c('circular','landscape'), params=list())
```

Arguments

`cyto.info`	data.frame, see `Details`.
`num.tracks`	integer, number of tracks.
`chr.order`	character vector of chromosome ids specifying the ordering of chromosomes.
`stepUnit`	integer, smoothing factor for faster plotting.
`Layout`	layout for plotting.
`params`	a list of named items. See `Details`.

Details

`cyto.info` is a data.frame of chromosomal position ordered cytobands, with columns: `Chr`, `Start`, `End`, `Stain`, and any additional information (like band color), where `Chr` is chromosome name, `Start` and `End` are the start and end positions on the chromosome, and `Stain` is the cyto stain. The stain is normally one of the "gneg", "acen", "stalk", "gvar", "gpos", "gpos100", "gpos75", "gpos66", "gpos50", "gpos33", and "gpos25", which will be plotted by color "white", "red", "steelblue", "lightgrey", "black", "black", "gray40", "gray50", "gray60", "gray70" and "gray80", accordingly. Customized colors for the cyto bands can be specified in an additional column named "`BandColor`".

Additional plot parameters can be specified through argument `params`, including:

- `color.line`, color for lines and links, default "black".
- `chr.padding`, padding between chromsomes is a fraction of the total chromosome sizes, default 0.1.
- `track.padding`, paddings between tracks is a fraction of the track height, default 0.1
- `track.height`, track height, default 0.15.
- `radius` radius of the circos, default 1.
- `slice.size`, a value between 0 and 360 (degree, default 360) specifying the circular slice size of the circos plot.
- `slice.rotate`, a value between 0 (the default) and 360 specifying the degree to rotate in a clock-wise direction.

After initialization, the parameter settings can be retrieved by `rc.get.params`.

Noted that while `cyto.info` requires input to be in a form of chromosome cytobands, the input is not limited to genomic features. As illustrated in example data `Modules`, complex features of gene coexpression network modules can also be plotted with the current circos visualization technique.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.get.params`, `rc.reset.params`, `rc.plot.ideogram`, `rc.plot.histogram`, `rc.plot.mHistogram`, `rc.plot.barchart`, `rc.plot.link`, `rc.plot.ribbon`, `Modules`

Examples

```
library(NetWeaver)
options(stringsAsFactors=FALSE)
## set fake Cyto data
Cyto=data.frame(Chr=paste0('C',1:20), Start=1, End=100, BandColor=colors()[2:21])

## initialize circos
#firstly set number of tracks, should be larger than the actual number of tracks
#used for plotting in order to leave sufficient sapce in the middle
num.tracks=9
#slice size 135 degree, rotate 22.5 degree
rc.initialize(Cyto, num.tracks=num.tracks, params=list(chr.padding=0.1,slice.size=135,slice.rotate=22.5))

##retrieve parameters
params=rc.get.params()
#make plot area
rc.plot.area(size=0.95)

##plot ideogram on track 1 and 2 (start from the outside to inside)
track.num=1:2
rc.plot.ideogram(track.num, plot.band=TRUE, plot.chromosome.id=TRUE)

##plot histogram/barchart that span multiple chromosomes
MultHistData=data.frame(Chr1=paste0('C',seq(1,18,3)), Start1=50,
 Chr2=paste0('C',seq(3,20,3)), End2=20, Col=1:6)
track.num=3
rc.plot.mHistogram(MultHistData, track.id=track.num, data.col=5,
 color.col=5, fixed.height=FALSE)

##plot histogram that occupies two tracks 4 and 5
HistData=data.frame(Chr=paste0('C',1:20), Start=1, End=50, Data=runif(20))
params$color.hist <- 'black'
rc.reset.params(params)
track.num=5
rc.plot.histogram(HistData, track.num, data.col=4, fixed.height=FALSE,
 track.border=NA, custom.track.height=params$track.height*2)

##plot heatmap on track 6
HeatData=data.frame(Chr=paste0('C',1:20), Start=1,End=100, Data=1:20)
colfuncHeat=function(n) rev(heat.colors(n))
track.num=track.num+1
rc.plot.histogram(HeatData, track.num, data.col=4, color.gradient=colfuncHeat(50),
 fixed.height=TRUE)

##plot stacked barchart on track 7
BarData=data.frame(Chr=paste0('C',1:20), Start=1,
 End=seq(10,86,length.out=20), Data=matrix(runif(20*4),nrow=20))
track.num=track.num+1
rc.plot.barchart(BarData, track.num, data.col=4:7)

##plot links in the middle
LinkData=data.frame(Chr1=sample(Cyto$Chr,40,replace=TRUE), Pos1=20,
```

```
 Chr2=sample(Cyto$Chr,40,replace=TRUE),Pos2=20, Data=runif(20))
LinkData=LinkData[LinkData$Chr1 != LinkData$Chr2,]
params$color.line='blue'
rc.reset.params(params)
track.num=track.num+1
rc.plot.link(LinkData, track.num, data.col=4, arrow.length=0.1)
ribbonData=data.frame(Chr1=c('C1','C3'), Start1=c(10,10), End1=c(40,40),
 Chr2=c('C17','C10'), Start2=20, End2=60, Col=c('red','brown'))
rc.plot.ribbon(ribbonData, track.num, color.col='Col', twist=TRUE)
#label track id
rc.plot.track.id(2:7, col=2)
#add text label
rc.plot.text(data.frame(Chr='C3',Pos=50,Label='GeneX'),
 track.id=3,srt=45,cex=0.8,col='blue')
#add line mark
rc.plot.line(data.frame(Chr='C19',Pos=seq(10,90,by=15),Col='red'),
 track.id=3, color.col=3,arrow.length=0.2)
```

`rc.plot.area` *Setup Plot Area*

Description

Create new graphics frame for circle plot .

Usage

```
rc.plot.area(size=1, oma=rep(0,4), mar=rep(0,4))
```

Arguments

`size`	a value between 0 to 1, specifying the effective size of the circle plot area in the current window. The smaller the `size`, the larger the blank area around the circle plot.
`oma`	A vector of the form `c(bottom, left, top, right)` giving the size of the outer margins in lines of text.
`mar`	A numerical vector of the form `c(bottom, left, top, right)` which gives the number of lines of margin to be specified on the four sides of the plot.

Details

After setting up, the extremes of the user coordinates of the plotting region can be found out by `par('usr')`.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.initialize`, `rc.reset.params`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.barchart` *Bar Plots*

Description

Add a track of barcharts.

Usage

```
rc.plot.barchart(Data, track.id, data.col, color.col=NULL, bar.color=NULL,
 track.color=NA, track.border=NULL, polygon.border=NULL,
 custom.track.height=NULL, ratio=FALSE)
```

Arguments

`Data`	data.frame, data to be plotted. See `Details`.
`track.id`	integer, in which track to plot.
`data.col`	integer, specifying the column(s) that contain the data for plotting. A stacked barchart is created if there are multiple columns of data.
`color.col`	integer, specifying the column that contains the color. Only used when there is only one data column.
`bar.color`	a vector of colors for the bars or bar components. By default, `grey` is used if there is one data column, and a gamma-corrected grey palette if there are multiple columns of data.
`track.color`	NA or color code for the track background.
`track.border`	the color to draw the track border. Use `NA` to omit borders.
`polygon.border`	the color to draw the polygon border. See explanation for `polygon`.
`custom.track.height`	NULL or numeric, specifying customized track height to overwrite the default. See `rc.get.params` for default value.
`ratio`	logical, specifying whether the data values in each row should be converted to ratios. Used only for stacked barchart.

Details

`Data` must have at least three columns. The first three columns must be named as `Chr`, `Start` and `End`, specifying the chromosomes and positions for the start and end points of the links. Additional columns can be used to specify data and color.

`rc.plot.histogram` can also be used for plotting barcharts except stacked barchart.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.plot.link`, `rc.plot.mHistogram`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.grColLegend` *Plot Gradient Color Legend*

Description

Plot gradient color legend at given position.

Usage

```
rc.plot.grColLegend(x, y, cols, at=NULL, legend=at,
 title='Color', width=0.1, height=0.3, gap=0,
 direction=c('vertical', 'horizontal'), cex.text=1, cex.title=1)
```

Arguments

`x, y`	numeric values of coordinates where the legend is plotted. See `details`.
`cols`	a vector of colors.
`at`	integers; index `legend` text in the color vector `cols`. See `details`.
`legend`	character strings of legend text with the same length of vector `at`.
`title`	character string of the legend title.
`width`	numeric, width of the color legend bar.
`height`	numeric, height of the color legend bar.
`gap`	numeric, size of the gap between two color blocks.
`direction`	plot the color legend bar in either vertical or horizontal direction.
`cex.text`	numeric, magnitude of the legend text.
`cex.title`	numeric, magnitude of the legend title.

Details

`x` and `y` set the coordinates of bottom left starting point. The extremes of the user coordinates of the plotting region can be found out by `par('usr')`.

By default `NULL`, parameter `at` will be set as a vector with values 1, `ceiling(length(cols)/2)`, and `length(cols)`.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.plot.link`, `rc.plot.histogram`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.heatmap` *Plot Heat-map*

Description

Add multiple tracks of heat-map on a circos plot.

Usage

```
rc.plot.heatmap(Data, track.id, color.gradient=NULL,
 track.color=NA, track.border=NULL, polygon.border=NULL)
```

Arguments

`Data`	matrix, data to be plotted. See `Details`.
`track.id`	integer, starting track id.
`color.gradient`	a vector of gradient colors. See `details`.
`track.color`	NA or color code for the track background.
`track.border`	the color to draw the track border. Use `NA` to omit borders.
`polygon.border`	the color to draw the polygon border. See explanation for `polygon`.

Details

This function employs `rc.plot.histogram` to plot heat-map. Every row of `Data` will be plotted as heat-map in one track. The column names of `Data` must be the chromosome ids.

If `color.gradient` is not `NULL`, the data will be scaled to positive integers in the range of 1~length(color.gradient) to index the colors in vector `color.gradient`. If `color.gradient` is `NULL`, the input `Data` must be a matrix of colors or anything that can be converted to colors.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.plot.histogram`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.histogram` *Plot Histogram, Barchart or Heat-map*

Description

Add a track of histogram, barchart, or heat-map on a circos plot.

Usage

```
rc.plot.histogram(Data, track.id, data.col=NULL, color.col=NULL,
 color.gradient=NULL, fixed.height=FALSE, track.color=NA, track.border=NULL,
 polygon.border=NULL, custom.track.height=NULL, max.value=NULL)
```

Arguments

`Data`	data.frame, data to be plotted. See `Details`.
`track.id`	integer, in which track to plot.
`data.col`	integer, specifying the column that contains the data for plotting.
`color.col`	integer, specifying the column that contains the color.
`color.gradient`	a vector of gradient colors. See `details`.
`fixed.height`	logical, whether to fix the histogram height which results in heat-map. `TRUE` if `data.col` is `bar.color`.
`track.color`	NA or color code for the track background.
`track.border`	the color to draw the track border. Use `NA` to omit borders.
`polygon.border`	the color to draw the polygon border. See explanation for `polygon`.
`custom.track.height`	NULL or numeric, specifying customized track height to overwrite the default. See `rc.get.params` for default value.
`max.value`	NULL or numeric, specifying the maximum data value for normalization. Default NULL, get max from data column.

Details

`Data` must have at least three columns. The first three columns must be named as `Chr`, `Start` and `End`, specifying the chromosomes and positions for the start and end points of the links. Additional columns can be used to specify data and color.

If `color.gradient` is not `NULL`, `color.col` is ignored and the data will be scaled to positive integers in the range of 1~length(color.gradient) to index the colors in vector `color.gradient`.

For plotting stacked barcharts, use function `rc.plot.barchart` instead.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.plot.link`, `rc.plot.mHistogram`, `rc.plot.barchart`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.ideogram` *Plot Ideogram*

Description

Add chromosome ideogram on circos plot.

Usage

```
rc.plot.ideogram(track.ids, plot.band=TRUE, plot.chromosome.id=TRUE,
 chrom.alias=NULL, color.chromosome.id=NULL, cex.text=1, track.border=NULL,
 polygon.border=NULL, las=0, custom.track.height=NULL)
```

Arguments

`track.ids`	vector, two integers specifying the two tacks for plotting chromosome name and cyto band respectively.
`plot.band`	logic, whether to plot ideogram cyto band.
`plot.chromosome.id`	logic, whether to plot chromosome id.
`chrom.alias`	NULL or a chromosome named vector of alias.
`color.chromosome.id`	NULL or a chromosome named vector of colors.
`cex.text`	numeric, scale of text.
`track.border`	the color to draw the track border. Use NA to omit borders.
`polygon.border`	the color to draw the polygon border. See explanation for `polygon`.
`las`	numeric in 0,1,2,3; the style of chromosome labels. 0, always parallel to the track [default]; 1, always horizontal; 2, always perpendicular to the track; 3, always vertical.
`custom.track.height`	NULL or numeric, specifying customized track height when plotting cyto band.

Author(s)

Minghui Wang <m.h.wang@live.com>

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.line` *Plot Line*

Description

Plot lines at designated position.

Usage

```
rc.plot.line(Data, track.id, color.col=NULL, custom.length=NULL, arrow.length=0,
 arrow.angle=30, arrow.code=2, ...)
```

Arguments

`Data`	a data.frame. See `Details`.
`track.id`	a vector of integers, specifying the tracks for plotting line.
`color.col`	integer, specifying the column that contains the color.
`custom.length`	`NULL` or numeric, specifying line length.
`arrow.length`	length of the edges of the arrow head (in inches). See `arrows`
`arrow.angle`	angle from the shaft of the arrow to the edge of the arrow head. See `arrows`
`arrow.code`	integer code, determining kind of arrows to be drawn. See `arrows`
`...`	further graphical parameters (from `par`), such as lty and lwd.

Details

`Data` is a data.frame, with at least two columns named 'Chr' and 'Pos', specifying the chromosomal positoins for each line in every row. An additional column can be used to specify the line color. Default line length is determined by track height.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.plot.histogram`, `rc.plot.track`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.link` *Plot Link*

Description

Add a track of links on circos plot.

Usage

```
rc.plot.link(Data, track.id, data.col=NULL, color.col = NULL,
 max.lwd=1, sort.links=TRUE, arrow.length=0, arrow.angle=30, ...)
```

Arguments

`Data`	data.frame of at least four columns. See `Details`.
`track.id`	integer, the track number for plotting.
`data.col`	integer or character string, specifies the column of data.
`color.col`	integer or character string, specifies the column of colors.
`max.lwd`	integer, maximum line width.
`sort.links`	logical, whether to re-assign link start and end positions by sorting them. See `details`
`arrow.length`	length of the edges of the arrow head (in inches). See `arrows`
`arrow.angle`	angle from the shaft of the arrow to the edge of the arrow head. See `arrows`
`...`	further graphical parameters (from `par`), such as lty and lwd.

Details

`Data` must have at least four columns: `Chr1`, `Pos1`, `Chr2` and `Pos2`, specifying the chromosomes and positions for the start and end points of the links. Additional columns may be used to specify the data, color and max line width.

If `data.col` is not NULL, the line width will be proportional to the values in column `data.col` with the maximum line width determined by `max.lwd`.

If `sort.links` is true, the links originate from the same chromosome will be sorted and equally spaced to minimize crossing.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.plot.histogram`, `rc.plot.ribbon`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.mHistogram` *Plot Histogram or Barplot Across Chromosomes*

Description

Plot histogram or barplot that overlaps with multiple chromosomes.

Usage

```
rc.plot.mHistogram(Data, track.id, data.col, color.col=NULL, color.gradient=NULL,
 fixed.height=FALSE, track.color=NA, track.border=NULL, polygon.border=NULL,
 custom.track.height=NULL, max.value=NULL)
```

Arguments

`Data`	data.frame of at least four columns. See `Details`.
`track.id`	integer, the track number for plotting.
`data.col`	integer or character string, specifies the column of data.
`color.col`	integer or character string, specifies the column of colors.
`color.gradient`	a vector of gradient colors.
`fixed.height`	logical, whether to fix the histogram height, resulting in heat-map if `TRUE`.
`track.color`	NA or color code for the track background.
`track.border`	the color to draw the track border. Use `NA` to omit borders.
`polygon.border`	the color to draw the polygon border. See explanation for `polygon`.
`custom.track.height`	NULL or numeric, specifying customized track height to overwrite the default. See `rc.get.params` for default value.
`max.value`	NULL or numeric, specifying the maximum data value for normalization. Default NULL, get max from data column `data.col`.

Details

`Data` must have at least four columns: `Chr1`, `Start1`, `Chr2` and `End2`, specifying the chromosomes and positions for the start and end points. Additional columns may be used to specify the data, color, etc.

If `color.gradient` is not `NULL`, it will supersede `color.col`.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.plot.link`, `rc.plot.histogram`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.point` *Plot Point*

Description

Plot point(s) at designated position.

Usage

```
rc.plot.point(Data, track.id, color.col=NULL, custom.track.height=NULL, ...)
```

Arguments

`Data`	a data.frame. See `Details`.
`track.id`	a vector of integers, specifying the tracks for plotting point(s).
`color.col`	column id in `Data` specifying the colors.
`custom.track.height`	NULL or numeric, specifying customized track height.
`...`	further graphical parameters (from `par`).

Details

`Data` is a data.frame with at least two columns named 'Chr', 'Pos', and 'Height', specifying the chrosomal positoin and the height within the track of the points. "`Height`" will be sacled to have a maximum of 1.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.plot.histogram`, `rc.plot.track`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.ribbon` *Plot Ribbon*

Description

Add a track of ribbons on circos plot.

Usage

```
rc.plot.ribbon(Data, track.id, color.col = NULL, twist=FALSE)
```

Arguments

`Data`	data.frame of at least six columns. See `Details`.
`track.id`	integer, the track number for plotting.
`color.col`	integer or character string, specifies the column of colors.
`twist`	logic, whehter to twist the ribbon.

Details

`Data` must have at least six columns: `Chr1`, `Start1`, `End1`, `Chr2`, `Start2` and `End2`, specifying the chromosomes and positions for the start and end points of the ribbons. Additional columns may be used to specify the color.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.plot.histogram`, `rc.plot.link`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.text` *Plot Text*

Description

Plot text labels at designated position.

Usage

```
rc.plot.text(Data, track.id, col='black', custom.track.height=NULL, ...)
```

Arguments

`Data`	a data.frame. See `Details`.
`track.id`	a vector of integers, specifying the tracks for plotting text.
`col`	a vector of color for the text.
`custom.track.height`	NULL or numeric, specifying customized track height.
`...`	further graphical parameters (from `par`), such as srt and family.

Details

`Data` is a data.frame with at least three columns named 'Chr', 'Pos', and 'Label', specifying the chrosomal positoin and text labels.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.plot.histogram`, `rc.plot.track`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.track` *Plot Track*

Description

Add a track to a circos plot.

Usage

```
rc.plot.track(track.id, border='black', col=NA, custom.track.height=NULL)
```

Arguments

`track.id`	integer, track id.
`border`	the color to draw the track border. Use NA to omit borders.
`col`	NA or color code for the track background.
`custom.track.height`	NULL or numeric, specifying customized track height. See `details`.

Details

`custom.track.height` sets customized track height, which will be useful for track(s) requiring bigger/smaller height than default.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.plot.histogram`, `rc.plot.mHistogram`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.plot.track.id` *Plot Track Id*

Description

Plot labels in designated tracks.

Usage

```
rc.plot.track.id(track.id, labels=NULL, degree=0,
 col='black', custom.track.height=NULL, ...)
```

Arguments

`track.id`	a vector of integers, specifying the tracks for plotting id.
`labels`	NULL or a vector of character string, specifying the text to be written.
`degree`	the angle of the arc rotation.
`col`	color for the text.
`custom.track.height`	NULL or numeric, specifying customized track height.
`...`	further graphical parameters (from par), such as srt and family.

Details

If `labels` is NULL, values of `track.id` will be used as text labels.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.plot.histogram`, `rc.plot.track`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.reset.params` *Reset Circos Plot Parameters*

Description

Reset parameters for Circos plot.

Usage

```
rc.reset.params(params)
```

Arguments

`params`	list.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.initialize`, `rc.get.params`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`rc.track.pos` *Get Track Positions*

Description

Calculate inner and outer radius of a track.

Usage

```
rc.track.pos(track.id)
```

Arguments

`track.id`	integer, number of the track from the outermost.

Value

A vector with two radius values: out.pos and in.pos.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.initialize`, `rc.reset.params`

Examples

```
#This is not to be run alone. Please see tutorial vignette("netweaver") for usage.
```

`ucsc.hg19.cytoband` *Human chromosome cytoband*

Description

Human chromosome cytoband information from UCSC (http://genome.ucsc.edu/).

Usage

```
data(ucsc.hg19.cytoband)
data(ucsc.hg38.cytoband)
```

Details

Cytoband is useful for drawing the chromosome ideograms for the genome. Two human cytoband datasets are provided in this package, corresponding to two genome versions hg19 and hg38. Each cytoband dataset is a five-column tab-delimited data.frame. Each row of the data describes the position of a cytogenetic band.

Author(s)

Minghui Wang <m.h.wang@live.com>

See Also

`rc.initialize`

Index