‘Circlize’ Package : : **CHEAT SHEET**

Intro

Welcome to the ‘circlize’ library !

Circlize is a useful data visualization tool when we have huge amounts of information.
First, it elegantly represents information with long axes or large amounts of categories.
Second, it intuitively shows data with multiple tracks focusing on the same object.
Third, it easily demonstrates relations between elements.

How might we have encountered a circlize plot?

Circlize has many practical applications. It can not only reflect the interaction relationship between the two variables, but also reflect the intensity of interaction.

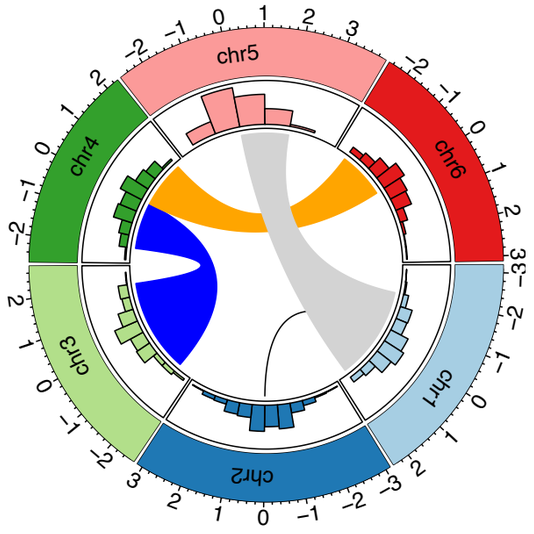
For example, it can be used to plot trade flows between countries, flights between cities, and to visualize cellular and genetic data, etc.

Let’s get started

CODE

```
install.packages('circlize')
library(circlize)
```

A quick glance



Function

Functions to draw plots

circos.points	add points to a plotting region
circos.lines	add lines to a plotting region
circos.boxplot	draw boxplots
circos.text	draw text in a cell
circos.trackPoints	add points to the plotting regions in a same track
circos.heatmap	add points to the plotting regions in a same track
circos.segments	draw segments through pairwise of points
circos.barplot	draw barplots
circos.axis	draw x-axis

Functions to arrange circular layout

circos.track	create plotting regions for a whole track
circos.nested	nested zooming with two circular plots
circos.par	parameters for the circular layout

Arguments

x	data points on x-axis
y	data points on y-axis
pch	point/line type
col	point/line color
cex	point size
bg	background of points
lwd	line width
lty	line style
labels	labels for each point
track.index	index for the track which is going to be updated.
track.height	height of the track

More Circlize Plots

In this part, I’ve created a brief and simple dataset. I hope to show the steps and magic of ‘circlize’ through the example.

Creat a dataset

CODE

```
n <- 1000
df <- data.frame(
  sectors = sample(letters[1:8], n, replace = TRUE),
  x = rnorm(n), y = runif(n)
)
```

Initialize the circular layout(omit)

Add scatterplots

CODE

```
col <- rep(c("#e41a1c", "#4daf4a"), 4)
circos.trackPoints(df$sectors, df$x, df$y, col = col, pch = 16, cex = 0.5)
circos.text(-1, 0.5, "text", sector.index = "a", track.index = 1)
```

Add histograms

CODE

```
bgcol <- rep(c("#fb8072", "#80b1d3"), 4)
circos.trackHist(df$sectors, df$x, bin.size = 0.2, bg.col = bgcol, col = NA)
```

Add line charts

CODE

```
circos.track(df$sectors, x = df$x, y = df$y,panel.fun = function(x, y) {ind = sample(length(x), 12)
x2 = x[ind] y2 = y[ind] od = order(x2)
circos.lines(x2[od], y2[od], col = "#ff7f00")
})
```

Add heatmaps

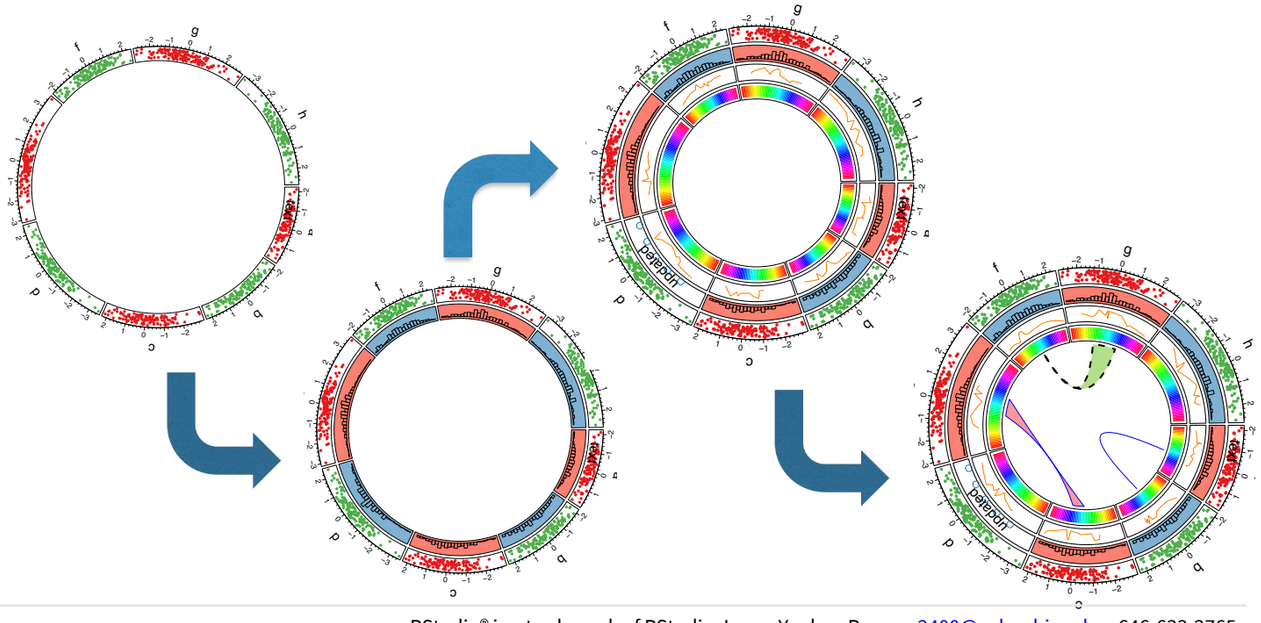
CODE

```
circos.track(ylim = c(0, 1), panel.fun = function(x, y) {xlim = CELL_META$xlim ylim = CELL_META$ylim
breaks = seq(xlim[1], xlim[2], by = 0.1) n_breaks = length(breaks)
circos.rect(breaks[-n_breaks], rep(ylim[1], n_breaks - 1),breaks[-1], rep(ylim[2], n_breaks-1),col=rainbow(n_breaks),border= NA)
})
```

Add links or ribbons

CODE

```
circos.link("a", 0, "b", 0, h = 0.4, col = "blue")
circos.link("c", c(-0.5, 0.5), "e", c(-0.5,1),
            col = "#fb9a99", border = "blue", h = 0.2)
circos.link("f", 0, "g", c(-1,1), col = "#b2df8a",
            border = "black", lwd = 2, lty = 2)
```



RStudio® is a trademark of RStudio, Inc. • Yanhao Ren • yr2400@columbia.edu • 646-633-2765