

# How to draw relations

Zuguang Gu <z.gu@dkfz.de>  
German Cancer Research Center,  
Heidelberg, Germany

June 2, 2013

One feature of circos layout is the link (or connector) to represent relations between elements ([http://circos.ca/intro/tabular\\_visualization/](http://circos.ca/intro/tabular_visualization/)). Normally, the relationship can be represented as a matrix in which value in  $i^{th}$  row and  $j^{th}$  column is kind of degree for the relation. In this vignette we show how to represent a correlation matrix.

First, we need a correlation matrix.

```
> set.seed(12345)
> n = 9
> m = matrix(rnorm(n^2), n, n)
> colnames(m) = letters[1:n]
> m2 = cor(m)
```

Then the range for each element, which is the summation of correlations with other elements.

```
> xlim = cbind(rep(0, n), apply(m2, 2, function(x) sum(abs(x)) - 1))
```

Initialize the layout and add names of elements around each sector. The size of each sector is determined by the range for each element. Here we create the first track by `circos.trackPlotRegion`. In this track, we only want to draw one rectangle and add the name for the element. This is done by `panel.fun` argument to draw the cell as soon as the corresponding cell has been created.

```
> library(circlize)
> factors = rownames(m2)
> colors = 1:n
> par(mar = c(1, 1, 1, 1))
> circos.initialize(factors = factors, xlim = xlim)
> circos.trackPlotRegion(ylim = c(0, 1), factors = factors, bg.border = NA,
+   panel.fun = function(x, y) {
+     xlim = get.cell.meta.data("xlim")
+     current.sector.index = get.cell.meta.data("sector.index")
+     circos.text(mean(xlim), 0.75, labels = current.sector.index,
```

```

+         direction = "horizontal")
+         i = get.cell.meta.data("sector.numeric.index")
+         circos.rect(min(xlim), 0, max(xlim), 0.25, col = colors[i])
+     })

```

Finally we use `circos.link` to draw links from elements to elements, in which red represents positive correlation and green represents negative correlation.

```

> rn = rownames(m2)
> sector.sum = numeric(length(rn))
> for(i in 2:n) {
+   for(j in 1:(i-1)) {
+     sector.index1 = rn[i]
+     sector.index2 = rn[j]
+     circos.link(sector.index1,
+                 c(sector.sum[i], sector.sum[i] + abs(m2[i, j])),
+                 sector.index2,
+                 c(sector.sum[j], sector.sum[j] + abs(m2[i, j])),
+                 col = ifelse(m2[i, j] > 0, "#FF0000A0", "#00FF00A0"),
+                 border = "grey")
+     sector.sum[i] = sector.sum[i] + abs(m2[i, j])
+     sector.sum[j] = sector.sum[j] + abs(m2[i, j])
+   }
+ }
> circos.clear()

```

The final figure looks like figure 1 (top part) and a more pretty figure is on the bottom part of figure 1 in which correlations are measured by continuous colors.

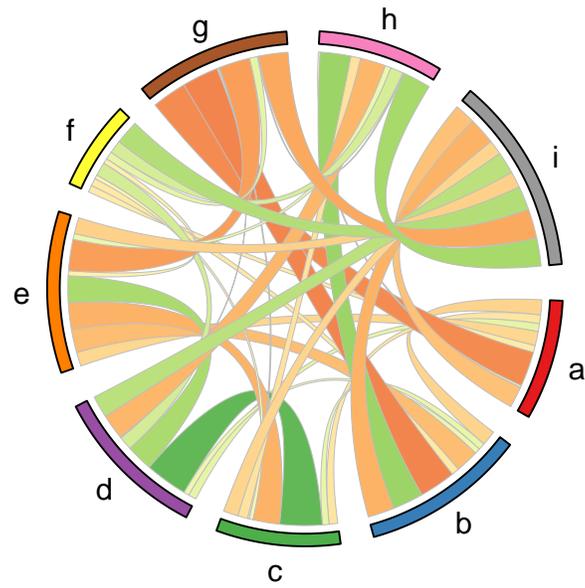
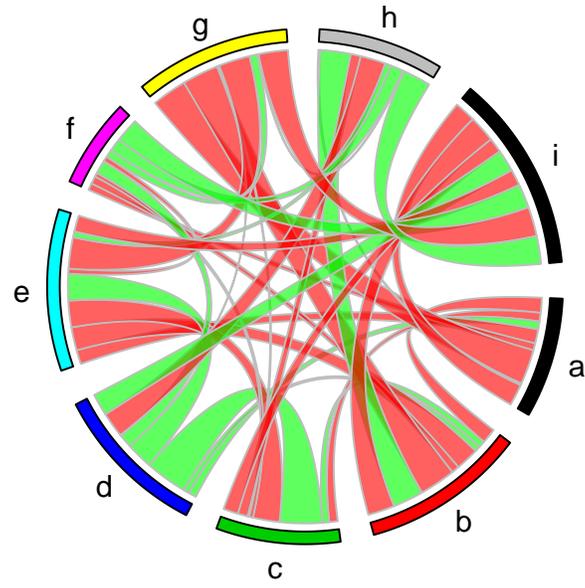


Figure 1: Correlations represented in circos layout