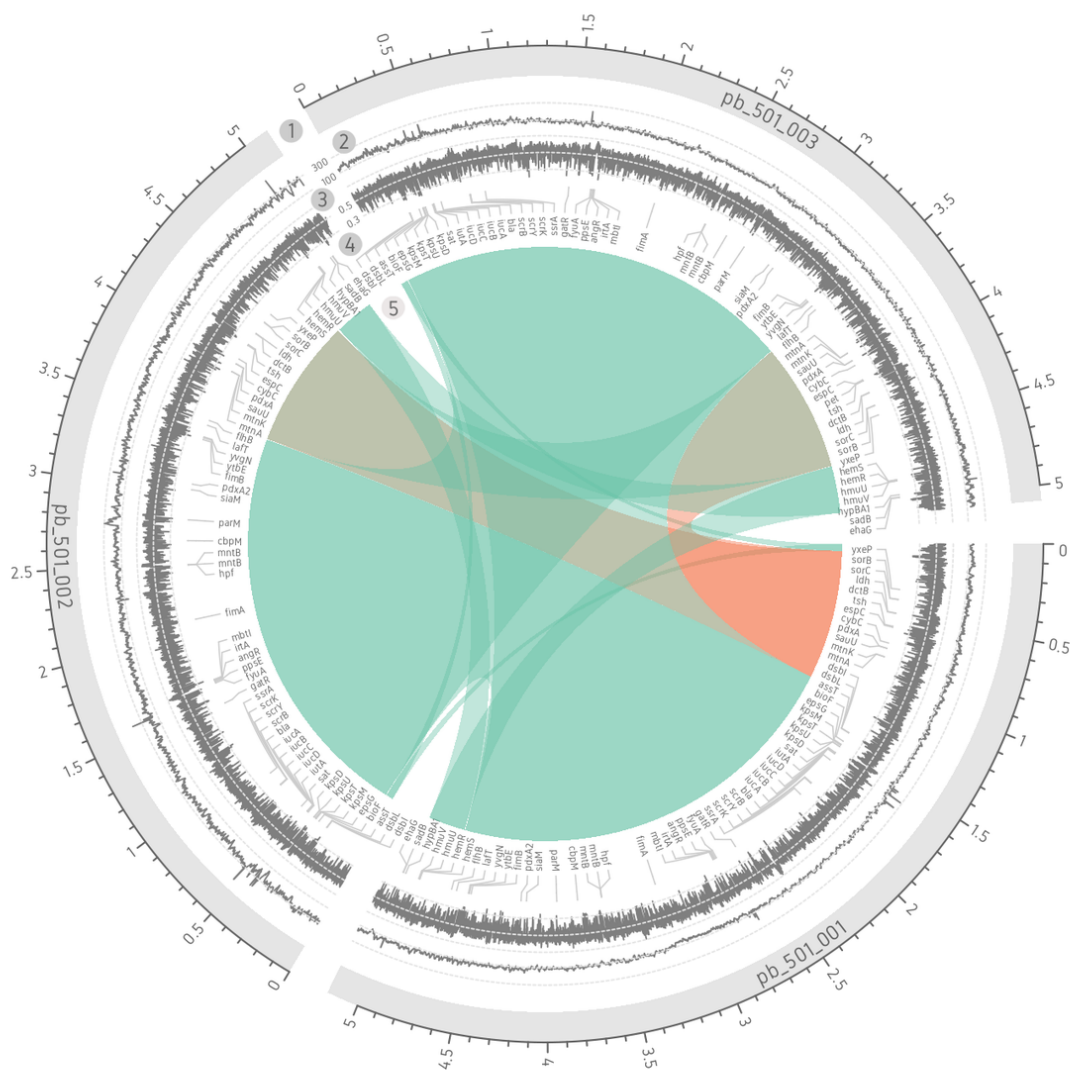


【小刻也能看懂的R数据可视化】好康的circos图（翻译与修改）

原文链接：<https://www.royfrancis.com/beautiful-circos-plots-in-r/>

效果预览：



Circos图（环形图）常用于在较小的视觉空间内显示大量的数据。本文探讨circos风格的基因组数据图。所用R包为**circlize**。

第一步是做画图计划。有多少基因组，每个基因组有多少条染色体，想包括什么样的信息？本文有三个细菌基因组，想显示测序深度、GC含量、重要基因位置和三个基因组之间的重排。

1. 数据

三个基因组命名为**pb_501_001**、**pb_501_002**和**pb_501_003**。

代码中的**cov**文件格式如下：

```

1 | chr      start end  value1
2 | pb_501_001 1      1000 145
3 | pb_501_001 1001   2000 154
4 | pb_501_001 2001   3000 158
5 | ...

```

代码中的**gc**文件格式如下：

```

1 | chr      start end  value1
2 | pb_501_001 1      1000 145.0
3 | pb_501_001 1001   2000 154.0
4 | pb_501_001 2001   3000 158.0
5 | ...

```

代码中的**ann**文件格式如下：

```

1 | chr      start  end    value1 gene
2 | pb_501_001 325210 326376 1      yxeP
3 | pb_501_001 397301 397795 1      sorB
4 | pb_501_001 399088 400035 1      sorC
5 | ...

```

其中value1是一个随机值，具体为什么不重要，大概是基因那相关的东西。

成对的基因体/染色体数据格式如下：

```

1 | pb_501_001 1      59986  pb_501_002 1      62503
2 | pb_501_001 60196 1183032 pb_501_002 4682296 3560259
3 | ...

```

分割的两个文件：

nuc1

```

1 | pb_501_001 1      59986
2 | pb_501_001 60196 1183032
3 | pb_501_001 1182899 4708778
4 | ...

```

nuc2

```

1 | pb_501_002 1      62503
2 | pb_501_002 4682296 3560259
3 | pb_501_002 67566 3555254
4 | ...

```

2. 绘图

首先加载库并初始化布局。

```

1 library(circlize)
2 circos.clear()
3 circos.initialize(factors=c("pb_501_001","pb_501_002","pb_501_003"),
4 xlim=matrix(c(rep(0,3),ref$V2),ncol=2))

```

其中xlim矩阵定义了每个基因组/chr的起点和重点，本质上是基因组/chr的大小，看起来如下：

```

1 > matrix(c(rep(0,3),ref$V2),ncol=2)
2      [,1] [,2]
3 [1,] 0    5167156
4 [2,] 0    5138029
5 [3,] 0    5080212

```

然后我们创建基因组骨架：

```

1 # genomes
2 circos.track(ylim=c(0,1),panel.fun=function(x,y) {
3 chr=CELL_META$sector.index
4 xlim=CELL_META$xlim
5 ylim=CELL_META$ylim
6 circos.text(mean(xlim),mean(ylim),chr)
7 })

```

现在可以看到三个基因组和它们相对长度的图。用circos.text()命令进行标注，用circos.par()设置一些默认值，用gap.degree增加间隙宽度以便后面y轴标签的自定义，将cell.padding设置为0以减少图周围的空隙。然后调整面板高度、文本大小和背景颜色。

```

1 circos.clear()
2 col_text <- "grey40"
3 circos.par("track.height"=0.8,gap.degree=5,cell.padding=c(0,0,0,0))
4 circos.initialize(factors=ref$V1,xlim=matrix(c(rep(0,3),ref$V2),ncol=2))
5
6 # genomes
7 circos.track(ylim=c(0,1),panel.fun=function(x,y) {
8 chr=CELL_META$sector.index
9 xlim=CELL_META$xlim
10 ylim=CELL_META$ylim
11 circos.text(mean(xlim),mean(ylim),chr,cex=0.5,col=col_text,
12 facing="bending.inside",niceFacing=TRUE)
13 },bg.col="grey90",bg.border=F,track.height=0.06)

```

对比下原图和自定义后的图：



图1. 基因组骨架 默认设置（左）和自定义设置（右）

把x轴加到图像中去（见下图）：

```
1 # genomes x axis
2 circos.track(track.index = get.current.track.index(), panel.fun = function(x,
3 y) {
4   circos.axis(h="top")
5 }
```

这一步x轴文本粘在一起，所以要继续自定义断点，改变标签颜色、大小、方向和轴线的厚度。

```
1 # genomes x axis
2 brk <- c(0,0.5,1,1.5,2,2.5,3,3.5,4,4.5,5)*10^6
3 circos.track(track.index = get.current.track.index(), panel.fun = function(x,
4 y) {
5   circos.axis(h="top",major.at=brk,labels=round(brk/10^6,1),labels.cex=0.4,
6   col=col_text,labels.col=col_text,lwd=0.7,labels.facing="clockwise")
7 },bg.border=F)
```

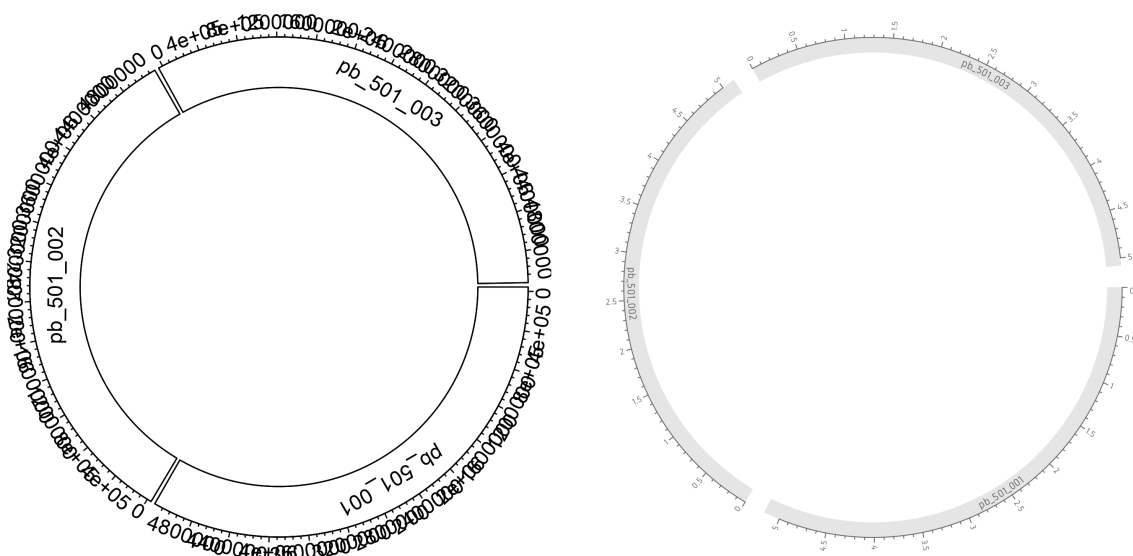


图2. 基因组骨架x轴 默认设置（左）和自定义设置（右）

接着添加覆盖率信息和对应的y轴：

```

1 # coverage
2 circos.genomicTrack(data=cov,panel.fun=function(region,value,...) {
3   circos.genomicLines(region,value)
4 })
5 # coverage y axis
6 circos.yaxis()

```

沿着基因组的覆盖率显示为一条线。y轴用于显示值的参考。接着改变线的属性，例如颜色和厚度。另外添加额外的线段作为网格线。然后删除y轴元素，只保留文本。

```

1 # coverage
2 circos.genomicTrack(data=cov,panel.fun=function(region,value,...) {
3   circos.genomicLines(region,value,type="l",col="grey50",lwd=0.6)
4   circos.segments(x0=0,x1=max(ref$V2),y0=100,y1=100,lwd=0.6,lty="11",col="grey90")
5   circos.segments(x0=0,x1=max(ref$V2),y0=300,y1=300,lwd=0.6,lty="11",col="grey90")
6   #circos.segments(x0=0,x1=max(ref$V2),y0=500,y1=500,lwd=0.6,lty="11",col="grey90")
7 },track.height=0.08,bg.border=F)
8 circos.yaxis(at=c(100,300),labels.cex=0.25,lwd=0,tick.length=0,labels.col=col_text,col="#FFFFFF")

```

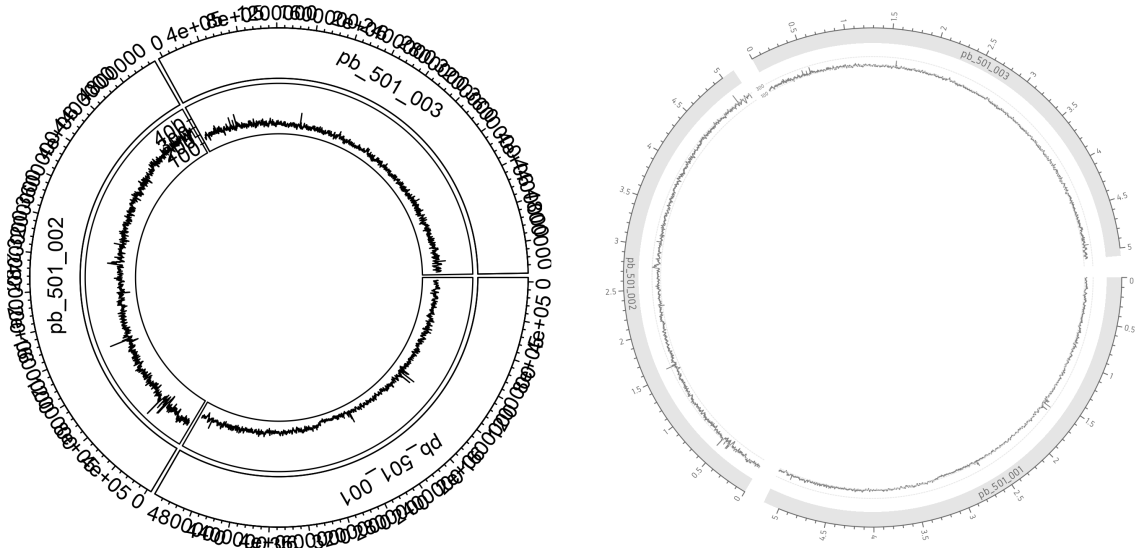


图3. 覆盖率环形图 默认设置（左）和自定义设置（右）

然后添加GC和其y轴：

```

1 # gc content
2 circos.track(factors=gc$chr,x=gc$x,y=gc$value,panel.fun=function(x,y) {
3   circos.lines(x,y)
4 })
5 # gc y axis
6 circos.yaxis()

```

定制步骤与上一步相似：

```

1 # gc content
2 circos.track(factors=gc$chr,x=gc$x,y=gc$value,panel.fun=function(x,y) {
3   circos.lines(x,y,col="grey50",lwd=0.6)
4   circos.segments(x0=0,x1=max(ref$V2),y0=0.3,y1=0.3,lwd=0.6,lty="11",col="grey90")
5   circos.segments(x0=0,x1=max(ref$V2),y0=0.5,y1=0.5,lwd=0.6,lty="11",col="grey90")
6   circos.segments(x0=0,x1=max(ref$V2),y0=0.7,y1=0.7,lwd=0.6,lty="11",col="grey90")
7 },ylim=range(gc$value),track.height=0.08,bg.border=F)
8 # gc y axis
9 circos.yaxis(at=c(0.3,0.5,0.7),labels.cex=0.25,lwd=0,tick.length=0,labels.col=col_text,col="#FFFFFF")

```

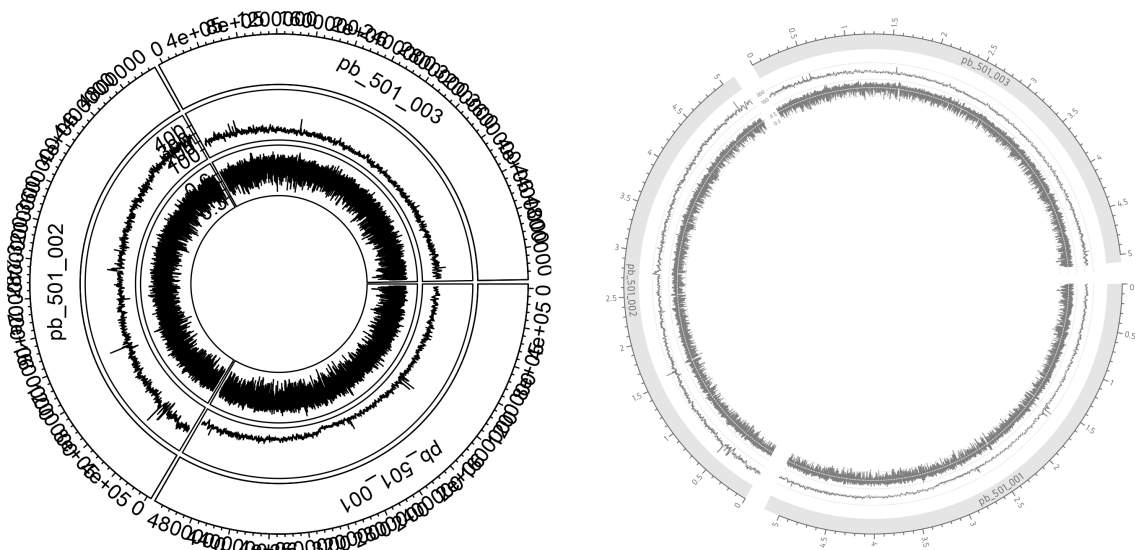


图4. GC环形图 默认设置（左）和自定义设置（右）

接着添加基因标签：

```

1 # gene labels
2 circos.genomicLabels(ann,labels.column=5)

```

基因标签与连接线一起被添加，以显示它们在基因组中的具体位置。如果不自定义会显得太拥挤。所以调整标签大小、颜色和连接线长度、厚度、颜色。轨迹的高设置为一个自定义值。

```

1 # gene labels
2 circos.genomicLabels(ann,labels.column=5,cex=0.25,col=col_text,line_lwd=0.5,
3   line_col="grey80",
4   side="inside",connection_height=0.05,labels_height=0.04)

```

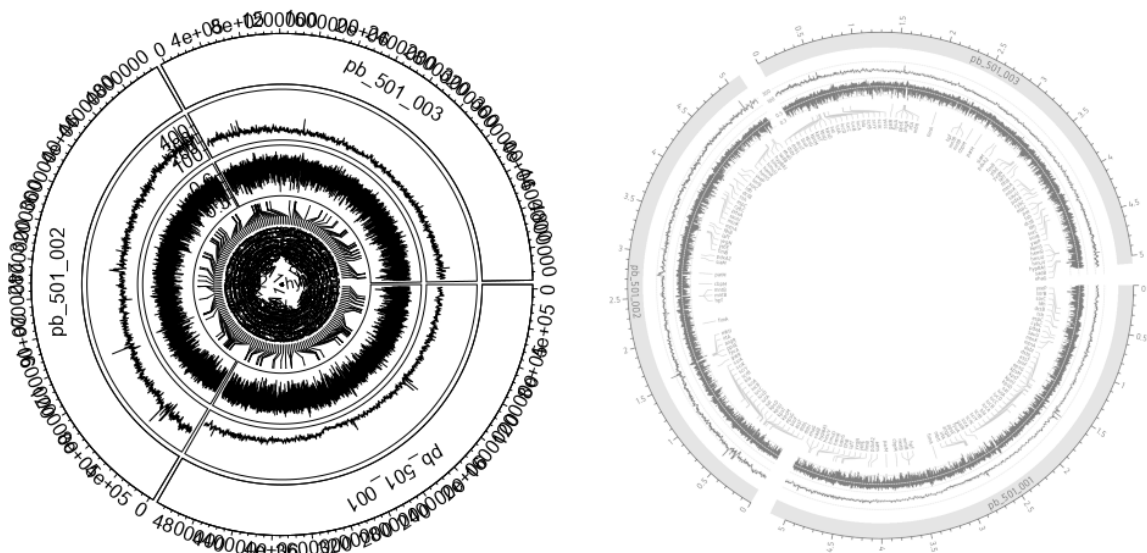


图5. 基因位置和标签的环形图 默认设置（左）和自定义设置（右）

最后用`circos.genomicLink()`函数添加重排：

```
1 # rearrangements
2 circos.genomicLink(nuc1,nuc2)
```

由于没有透明度，重叠的色带会看不到。所以将同向的重排分配绿色，反向分配红色，并设置0.4的不透明度。

```
1 # rearrangements
2 rcols <- scales::alpha(ifelse(sign(nuc1$start-nuc1$end)!=sign(nuc2$start-
  nuc2$end),"#f46d43","#66c2a5"),alpha=0.4)
3 circos.genomicLink(nuc1,nuc2,col=rcols,border=NA)
```

得到完整图像。可以看出轨迹的高被降低了，以便为中间的重排连接图留出空间。

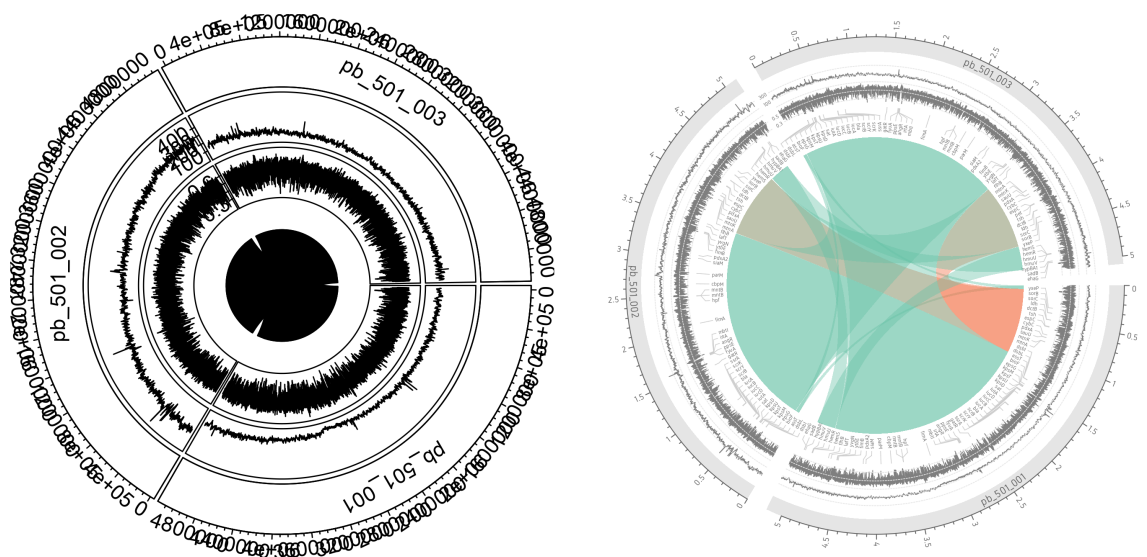


图6. 完整环形图 默认设置（左）和自定义设置（右）

然后AI里进一步添加了一些说明数字，以方便理解。



图7. 成品图