可以表示丰度/相对丰度，也可以表示相关性

使用circlize包中的chordDiagram来绘制：

用法：

1.形式：

chordDiagram(data, grid.col = col,#扇形颜色

grid.border = NA,#扇形边transparency = 0.5, #设置linker的颜色透明度

col = NULL,

row.col = NULL, column.col = col1,#连线的颜色，与列数一样多，可预设

order = NULL,#控制图上样本顺序

directional = 0, #directional=1或-1 可在矢量(vector)与连线links之间再加一层扇形

xmax = NULL,

symmetric = FALSE,

keep.diagonal = FALSE,

direction.type = "diffHeight",

diffHeight = mm\_h(2),

link.target.prop = TRUE,

target.prop.height = mm\_h(1),

reduce = 1e-5,

self.link = 2,

preAllocateTracks = NULL,

annotationTrack = c("name", "grid", "axis"),

annotationTrackHeight = mm\_h(c(3, 2)),

link.border = NA,

link.lwd = par("lwd"),

link.lty = par("lty"),

link.auto = TRUE,

link.sort = "default",

link.decreasing = TRUE,

link.arr.length = ifelse(link.arr.type == "big.arrow", 0.02, 0.4),

link.arr.width = link.arr.length/2,

link.arr.type = "triangle",

link.arr.lty = par("lty"),

link.arr.lwd = par("lwd"),

link.arr.col = par("col"),

link.largest.ontop = FALSE,

link.visible = TRUE,

link.rank = NULL,

link.zindex = NULL,

link.overlap = FALSE,

scale = FALSE,

group = NULL,

big.gap = 10,

small.gap = 1,

...)

2.重要参数：

data, 作图数据格式：可以是矩阵(matrix)、数据框(dataframe)

link：设置linker的边界线：link.lwd, link.lty, link.border: 线宽、线型、颜色

column.col = col1,#连线的颜色，与列数一样多，可预设

order = c("S2", "S1", "S3",

"E4", "E1", "E5", "E2", "E6", "E3")) *#行列扇区分组*排序

**调整扇区之间的间隙、角度：circos.par()函数**

*#先定义间隙：*

circos.par(gap.after = c(rep(5,nrow(mat)-1),

15,

rep(5,ncol(mat)-1),

15))

*#还可以直接指定各扇区后的间隙：*circos.par(gap.after = c("S1" = 5, "S2" = 5, "S3" = 15, "E1" = 5, "E2" = 5,"E3" = 5, "E4" = 5, "E5" = 5, "E6" = 15))  
*#定义行列扇区之间的big.gap：*chordDiagram(mat, big.gap = 30) small.gap默认为1的宽度，行或列各扇区内部之间的间隙  
*#通过设置start.digree自定义角度：*circos.par(start.degree = 90, clock.wise = F)chordDiagram(mat)

circos.clear() *#养成绘制和弦图后清除前面绘图痕迹的习惯*