VENN:

```
library(ggvenn)

compoundName="BaP"
diseaseName="LPD"
setwd("C:\\Users\\Dell\\Desktop\\BaP\\veen R")
geneList=list()

rt=read.table("Compound.txt", header=F, sep="\t", check.names=F)
geneNames=as.vector(rt[,1])
geneList[[compoundName]]=geneNames

rt=read.table("Disease.txt", header=F, sep="\t", check.names=F)
geneNames=as.vector(rt[,1])
geneList[[diseaseName]]=geneNames

pdf(file="venn.pdf", width=6, height=6)
ggvenn(geneList,show_percentage = T,
      stroke_color = "white", stroke_size = 0.5,
      fill_color = c("#E41A1C","#1E90FF"),
      set_name_color =c("#E41A1C","#1E90FF"),
      set_name_size=6, text_size=4.5)
dev.off()

interGenes=Reduce(intersect, geneList)
write.table(file="interGenes.txt", interGenes, sep="\t", quote=F, col.names=F, row.names=F)

networkTab=rbind(cbind(compoundName,interGenes,                    "Compound"),
cbind(diseaseName,interGenes, "Disease"))
colnames(networkTab)=c("Node1", "Node2", "Type")
write.table(file="net.network.txt", networkTab, sep="\t", quote=F, row.names=F)
nodeTab=rbind(cbind(compoundName,"Compound"),          cbind(diseaseName,"Disease"),
cbind(interGenes,"Gene"))
colnames(nodeTab)=c("Node", "Type")
write.table(file="net.node.txt", nodeTab, sep="\t", quote=F, row.names=F)
```

GO：

```
library(clusterProfiler)
library(org.Hs.eg.db)
library(enrichplot)
library(ggplot2)
library(circlize)
library(RColorBrewer)
library(dplyr)
```

```
library("ggpubr")
library(ComplexHeatmap)

pvalueFilter=0.05
p.adjustFilter=0.05

colorSel="p.adjust"
if(p.adjustFilter>0.05){
      colorSel="pvalue"
}

setwd("C:\\Users\\Dell\\Desktop\\BaP\\BaP-LPD\\Go")
rt=read.table("interGenes.txt", header=F, sep="\t", check.names=F)
genes=unique(as.vector(rt[,1]))
entrezIDs=mget(genes, org.Hs.egSYMBOL2EG, ifnotfound=NA)
entrezIDs=as.character(entrezIDs)
gene=entrezIDs[entrezIDs!="NA"]
#gene=gsub("c\\(\"(\\d+)\".*", "\\1", gene)

kk=enrichGO(gene=gene, OrgDb=org.Hs.eg.db, pvalueCutoff=1, qvalueCutoff=1, ont="all", readable=T)
GO=as.data.frame(kk)
GO=GO[(GO$pvalue<pvalueFilter & GO$p.adjust<p.adjustFilter),]
write.table(GO, file="GO.txt", sep="\t", quote=F, row.names = F)

pdf(file="barplot.pdf", width=8, height=7)
bar=barplot(kk, drop=TRUE, showCategory=10, label_format=100, split="ONTOLOGY", color=colorSel) + facet_grid(ONTOLOGY~., scale='free')
print(bar)
dev.off()

pdf(file="bubble.pdf", width=8, height=7)
bub=dotplot(kk, showCategory=10, orderBy="GeneRatio", label_format=100, split="ONTOLOGY", color=colorSel) + facet_grid(ONTOLOGY~., scale='free')
print(bub)
dev.off()

data=GO %>% group_by(ONTOLOGY) %>% slice_head(n=10)
pdf(file="barplot.color.pdf", width=11, height=8)
ggbarplot(data, x="Description", y="Count", fill = "ONTOLOGY", color = "white", xlab="",
          palette = "aaas",
          legend = "right",
          sort.val = "desc",
          sort.by.groups=TRUE)+
```

```
        rotate_x_text(75)+
        theme(panel.background = element_blank(), axis.text.x=element_text(size=10, color="black"))+
        scale_y_continuous(expand=c(0, 0)) + scale_x_discrete(expand=c(0,0))
dev.off()

ontology.col=c("#00CC33FF", "#FFC20AFF", "#CC33FFFF")
data=GO[order(GO$p.adjust),]
datasig=data[data$pvalue<0.05,,drop=F]
BP = datasig[datasig$ONTOLOGY=="BP",,drop=F]
CC = datasig[datasig$ONTOLOGY=="CC",,drop=F]
MF = datasig[datasig$ONTOLOGY=="MF",,drop=F]
BP = head(BP,6)
CC = head(CC,6)
MF = head(MF,6)
data = rbind(BP,CC,MF)
main.col = ontology.col[as.numeric(as.factor(data$ONTOLOGY))]

BgGene = as.numeric(sapply(strsplit(data$BgRatio,"/"),'[',1))
Gene = as.numeric(sapply(strsplit(data$GeneRatio,'/'),'[',1))
ratio = Gene/BgGene
logpvalue = -log(data$pvalue,10)
logpvalue.col = brewer.pal(n = 8, name = "Reds")
f = colorRamp2(breaks = c(0,2,4,6,8,10,15,20), colors = logpvalue.col)
BgGene.col = f(logpvalue)
df = data.frame(GO=data$ID,start=1,end=max(BgGene))
rownames(df) = df$GO
bed2 = data.frame(GO=data$ID,start=1,end=BgGene,BgGene=BgGene,BgGene.col=BgGene.col)
bed3 = data.frame(GO=data$ID,start=1,end=Gene,BgGene=Gene)
bed4 = data.frame(GO=data$ID,start=1,end=max(BgGene),ratio=ratio,col=main.col)
bed4$ratio = bed4$ratio/max(bed4$ratio)*9.5

pdf(file="GO.circlize.pdf", width=10, height=10)
par(omi=c(0.1,0.1,0.1,1.5))
circos.par(track.margin=c(0.01,0.01))
circos.genomicInitialize(df,plotType="none")
circos.trackPlotRegion(ylim = c(0, 1), panel.fun = function(x, y) {
   sector.index = get.cell.meta.data("sector.index")
   xlim = get.cell.meta.data("xlim")
   ylim = get.cell.meta.data("ylim")
   circos.text(mean(xlim), mean(ylim), sector.index, cex = 0.8, facing = "bending.inside", niceFacing = TRUE)
}, track.height = 0.08, bg.border = NA,bg.col = main.col)
```

```
for(si in get.all.sector.index()) {
   circos.axis(h = "top", labels.cex = 0.6, sector.index = si,track.index = 1,
                       major.at=seq(0,max(BgGene),by=100),labels.facing = "clockwise")
}
f = colorRamp2(breaks = c(-1, 0, 1), colors = c("green", "black", "red"))
circos.genomicTrack(bed2, ylim = c(0, 1),track.height = 0.1,bg.border="white",
                            panel.fun = function(region, value, ...) {
                               i = getI(...)
                               circos.genomicRect(region, value, ytop = 0, ybottom = 1, col =
value[,2],
                                                                  border = NA, ...)
                               circos.genomicText(region, value, y = 0.4, labels = value[,1],
adj=0,cex=0.8,...)
                            })
circos.genomicTrack(bed3, ylim = c(0, 1),track.height = 0.1,bg.border="white",
                            panel.fun = function(region, value, ...) {
                               i = getI(...)
                               circos.genomicRect(region, value, ytop = 0, ybottom = 1, col =
'#BA55D3',
                                                                  border = NA, ...)
                               circos.genomicText(region, value, y = 0.4, labels = value[,1],
cex=0.9,adj=0,...)
                            })
circos.genomicTrack(bed4, ylim = c(0, 10),track.height =
0.35,bg.border="white",bg.col="grey90",
                            panel.fun = function(region, value, ...) {
                               cell.xlim = get.cell.meta.data("cell.xlim")
                               cell.ylim = get.cell.meta.data("cell.ylim")
                               for(j in 1:9) {
                                  y = cell.ylim[1] + (cell.ylim[2]-cell.ylim[1])/10*j
                                  circos.lines(cell.xlim, c(y, y), col = "#FFFFFF", lwd = 0.3)
                               }
                               circos.genomicRect(region, value, ytop = 0, ybottom = value[,1], col =
value[,2],
                                                                  border = NA, ...)
                               #circos.genomicText(region, value, y = 0.3, labels = value[,1], ...)
                            })
circos.clear()
middle.legend = Legend(
   labels = c('Number of Genes','Number of Select','Rich Factor(0-1)'),
   type="points",pch=c(15,15,17),legend_gp = gpar(col=c('pink','#BA55D3',ontology.col[1])),
   title="",nrow=3,size= unit(3, "mm")
)
circle_size = unit(1, "snpc")
```

```
draw(middle.legend,x=circle_size*0.42)
main.legend = Legend(
   labels = c("Biological Process","Cellular Component", "Molecular Function"),
type="points",pch=15,
   legend_gp = gpar(col=ontology.col), title_position = "topcenter",
   title = "ONTOLOGY", nrow = 3,size = unit(3, "mm"),grid_height = unit(5, "mm"),
   grid_width = unit(5, "mm")
)
logp.legend = Legend(
   labels=c('(0,2]','(2,4]','(4,6]','(6,8]','(8,10]','(10,15]','(15,20]','>=20'),
   type="points",pch=16,legend_gp=gpar(col=logpvalue.col),title="-log10(Pvalue)",
   title_position = "topcenter",grid_height = unit(5, "mm"),grid_width = unit(5, "mm"),
   size = unit(3, "mm")
)
lgd = packLegend(main.legend,logp.legend)
circle_size = unit(1, "snpc")
print(circle_size)
draw(lgd, x = circle_size*0.85, y=circle_size*0.55,just = "left")
dev.off()
```

KEGG:

```
library(clusterProfiler)
library(org.Hs.eg.db)
library(enrichplot)
library(ggplot2)
library(circlize)
library(RColorBrewer)
library(dplyr)
library(ggpubr)

pvalueFilter=0.05
p.adjustFilter=0.05

colorSel="p.adjust"
if(p.adjustFilter>0.05){
      colorSel="pvalue"
}

setwd("C:\\Users\\Dell\\Desktop\\BaP\\BaP-LPD\\KEGG")
rt=read.table("interGenes.txt", header=F, sep="\t", check.names=F)

genes=unique(as.vector(rt[,1]))
entrezIDs=mget(genes, org.Hs.egSYMBOL2EG, ifnotfound=NA)
entrezIDs=as.character(entrezIDs)
```

```
rt=data.frame(genes, entrezID=entrezIDs)
gene=entrezIDs[entrezIDs!="NA"]
#gene=gsub("c\\(\"(\\d+)\".*", "\\1", gene)

kk <- enrichKEGG(gene=gene, organism="hsa", pvalueCutoff=1, qvalueCutoff=1)
KEGG=as.data.frame(kk)
KEGG$geneID=as.character(sapply(KEGG$geneID,function(x)paste(rt$genes[match(strsplit(x,"/
")[[1]],as.character(rt$entrezID))],collapse="/")))
KEGG=KEGG[(KEGG$pvalue<pvalueFilter & KEGG$p.adjust<p.adjustFilter),]
write.table(KEGG, file="KEGG.txt", sep="\t", quote=F, row.names = F)

showNum=30
if(nrow(KEGG)<showNum){
      showNum=nrow(KEGG)
}

pdf(file="barplot.pdf", width=8, height=7)
barplot(kk, drop=TRUE, showCategory=showNum, label_format=130, color=colorSel)
dev.off()

pdf(file="bubble.pdf", width=8, height=7)
dotplot(kk, showCategory=showNum, orderBy="GeneRatio", label_format=130, color=colorSel)
dev.off()

pdf(file="Lollipop.pdf", width=12.5, height=10)
ggdotchart(KEGG, x="Description", y="Count", color = "category",group = "category", xlab="",
                palette = "aaas",
                legend = "right",
                sorting = "descending",
                add = "segments",
                rotate = TRUE,
                dot.size = 7,
                label = round(KEGG$Count),
                font.label = list(color="white",size=12, vjust=0.5),
                ggtheme = theme_pubr())
dev.off()
```