some_GD

xueke

2/21/2022

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

library

```r
rm(list = ls())
library(GDCRNATools)
library(ggplot2)
library(readxl)
library(org.Hs.eg.db)
library(clusterProfiler)
library(pheatmap)
library(ggpubr)
library(digest)
library(GOplot)
library(survival)
library(limma)
library(glmnet)
library(survminer)
library(timeROC)
library(rms)
library(maftools)
library(tidyverse)
library(RCircos)
library(igraph)
library(psych)
```

```
## Warning: package 'psych' was built under R version 4.1.3
```

```r
library(reshape2)
library(RColorBrewer)
library(ConsensusClusterPlus)
```

global vairants

```r
adjpFilter <- 0.05
logFCfilter <- 1
FCfilter <- 2^logFCfilter
hyperPfilter <- 0.05
corPfilter <- 0.05
```

data preparation

```r
# project <- 'TCGA-BRCA'

## data download
###下载转录组数据
# gdcRNADownload(project.id     = project,
#                data.type      = 'RNAseq',
#                write.manifest = FALSE,
#                method = 'gdc-client',
#                directory      = rnadir)

## data load
# rnadir <- 'raw_data/RNAseq'
#
# metaMatrix.RNA <- gdcParseMetadata(project.id = project,
#                                    data.type  = 'RNAseq',
#                                    write.meta = FALSE)
#
# metaMatrix.RNA <- gdcFilterDuplicate(metaMatrix.RNA)
# metaMatrix.RNA <- gdcFilterSampleType(metaMatrix.RNA)
# rnaCounts <- gdcRNAMerge(metadata  = metaMatrix.RNA,
#                          path      = rnadir,
#                          organized = FALSE,   ## if target data are in folders
#                          data.type = 'RNAseq')
# GT <- read_excel(path = 'raw_data/GTgenes.xlsx', col_names = c('symbol', 'entrez'))
# GT_ensembel <- bitr(GT$entrez, fromType = 'ENTREZID', toType = c('SYMBOL', 'ENSEMBL'), OrgDb = org.Hs.eg.db)
# t <- intersect(GT_ensembel$ENSEMBL, rownames(rnaCounts))
# GT_Counts <- rnaCounts[t, ]
# GTrnaExpr <- gdcVoomNormalization(counts = GT_Counts, filter = FALSE)
# save(list = c('GTrnaExpr', 'metaMatrix.RNA', 'GT_ensembel', 'GT_Counts'), file = 'raw_data/GTrnaExpr.Rdata')
load('raw_data/GTrnaExpr.Rdata')
```

DEG analysis

```
GT_DEG <-
    gdcDEAnalysis(
        n.cores = 4,
        counts     = GT_Counts,
        group      = metaMatrix.RNA$sample_type,
        comparison = 'PrimaryTumor-SolidTissueNormal',
        method     = 'DESeq2'
    )
```

```
## DE analysis using DESeq2 may takelong time with a single core
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
## Warning in MulticoreParam(n.cores): MulticoreParam() not supported on Windows,
## use SnowParam()
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates: 1 workers
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates, fitting model and testing: 1 workers
```

```
## -- replacing outliers and refitting for 17 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
```

```
## estimating dispersions
```

```
## fitting model and testing
```

```
GT_all <-     gdcDEReport(
        deg = GT_DEG,
        gene.type = 'all',
        fc = 0,
        pval = 1
    )
GT_de <-
    gdcDEReport(
        deg = GT_DEG,
        gene.type = 'all',
        fc = FCfilter,
        pval = adjpFilter
    )
GT_ensembel[GT_ensembel$ENSEMBL %in% rownames(GT_de), ]$SYMBOL
```

```
##  [1] "A4GNT"      "ALG3"       "B3GALT1"    "B3GAT1"     "B4GALNT1"
##  [6] "B4GALNT2"   "B4GALNT3"   "B4GALNT4"   "B4GALT3"    "B4GALT6"
## [11] "CHPF"       "DPM2"       "FUT2"       "FUT3"       "FUT7"
## [16] "GAL"        "GALNT14"    "GALNT5"     "GALNT6"     "GALNT7"
## [21] "GALNT8"     "GALNT15"    "GCNT3"      "GCNT4"      "GGTA1"
## [26] "GLT1D1"     "COLGALT2"   "GYG2"       "GYS2"       "HAS1"
## [31] "HAS3"       "MFNG"       "MGAT3"      "MGAT5B"     "NEU4"
## [36] "PIGQ"       "PYGM"       "ST3GAL4"    "ST6GAL2"    "ST6GALNAC3"
## [41] "ST6GALNAC6" "ST8SIA2"    "UGCG"       "UGT2B17"    "UGT2B28"
## [46] "UGT2B4"     "UGT3A2"
```

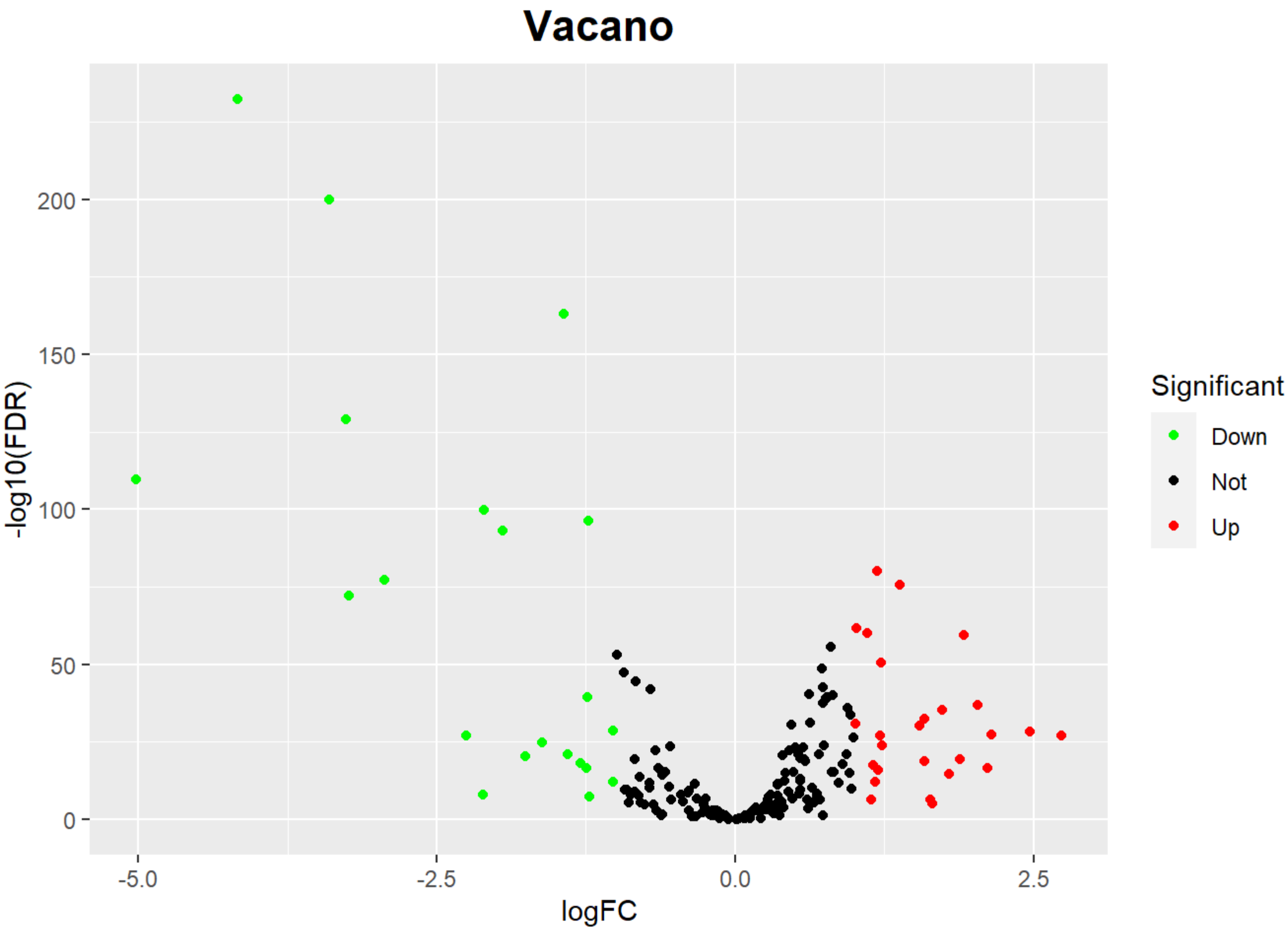
FPM of transciptome, so use wilcox test

bad results

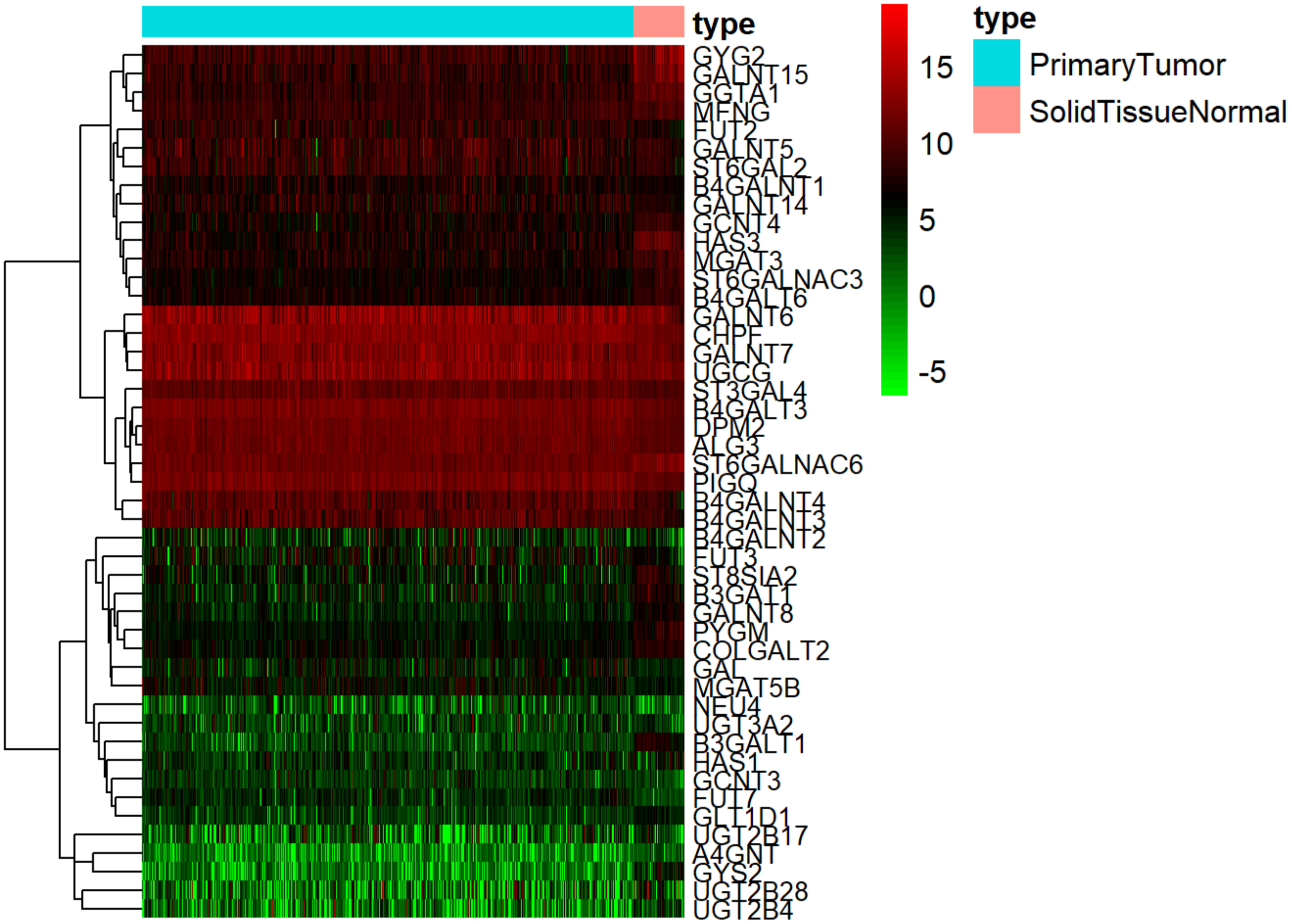
```
tumor_num <- table(metaMatrix.RNA$sample_type)[[1]]
normal_num <- table(metaMatrix.RNA$sample_type)[[2]]
grade <- c(rep(1, normal_num), rep(2, tumor_num))
outTab=data.frame()
for (i in row.names(GT_Counts)) {
    geneName <- unlist(strsplit(i, "\\|",))[1]
    geneName <- gsub("\\/", "-", geneName)
    rt <- rbind(expression <- GT_Counts[i, ], grade <- grade)
    rt <- as.matrix(t(rt))
    wilcoxTest <- wilcox.test(expression ~ grade, GT_Counts = rt)
    normGeneMeans = mean(GT_Counts[i, metaMatrix.RNA[metaMatrix.RNA$sample_type=='SolidTissueNormal', ]$sample])
    tumorGeneMeans = mean(GT_Counts[i, metaMatrix.RNA[metaMatrix.RNA$sample_type=='PrimaryTumor', ]$sample])
    logFC = log2(tumorGeneMeans) - log2(normGeneMeans)
    pvalue = wilcoxTest$p.value
    normMed = median(GT_Counts[i, metaMatrix.RNA[metaMatrix.RNA$sample_type=='SolidTissueNormal', ]$sample])
    tumorMed = median(GT_Counts[i, metaMatrix.RNA[metaMatrix.RNA$sample_type=='PrimaryTumor', ]$sample])
    diffMed = tumorMed - normMed
    if (((logFC > 0) & (diffMed > 0)) | ((logFC < 0) &
                                         (diffMed < 0))) {
        outTab = rbind(
            outTab,
            cbind(
                gene = i,
                normMean = normGeneMeans,
                tumorMean = tumorGeneMeans,
                logFC = logFC,
                pValue = pvalue
            )
        )
    }
}
pValue <- outTab[, "pValue"]
fdr <- p.adjust(as.numeric(as.vector(pValue)), method = "fdr")
outTab <- cbind(outTab, fdr = fdr)
outDiff <-
    outTab[(abs(as.numeric(as.vector(outTab$logFC))) > logFCfilter &
                as.numeric(as.vector(outTab$fdr)) < 0.05),]
```

plots

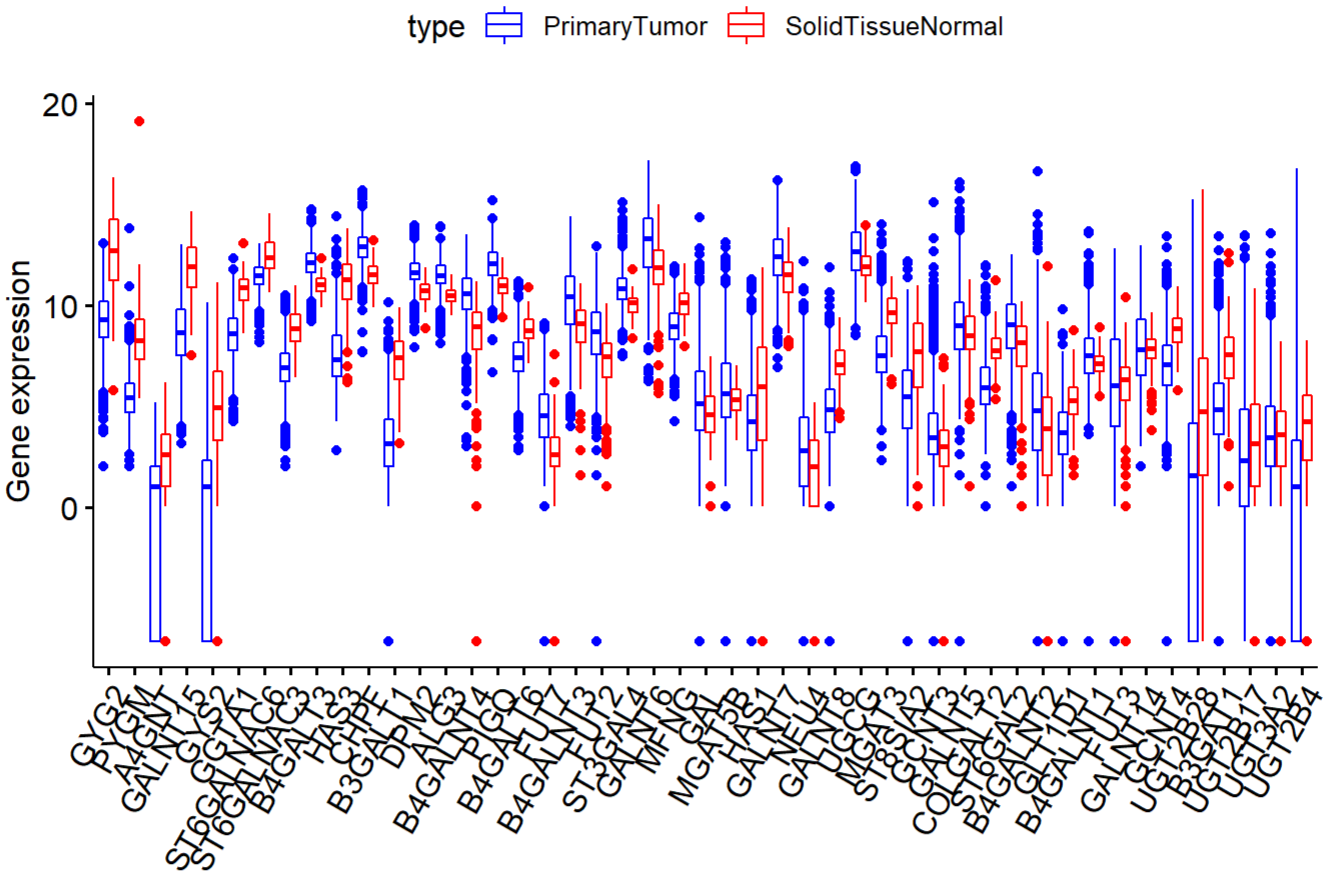
```
Significant <- ifelse((GT_all$FDR < adjpFilter &
                           abs(GT_all$logFC) > logFCfilter),
                      ifelse(GT_all$logFC > logFCfilter, "Up", "Down"),
                      "Not"
)
p <- ggplot(GT_all, aes(logFC, -log10(FDR))) +
    geom_point(aes(col = Significant)) +
    scale_color_manual(values = c("green", "black", "red")) +
    labs(title = "Vacano") +
    theme(plot.title = element_text(size = 16, hjust = 0.5, face = "bold"))
print(p)
```



```
metaMatrix.RNA <-
    metaMatrix.RNA[order(metaMatrix.RNA$sample_type),]
type <- metaMatrix.RNA$sample_type
names(type) <- metaMatrix.RNA$sample
type <- as.data.frame(type)
hmExp <- GT_Counts[rownames(GT_de), rownames(type)]
hmExp <- log2(hmExp + 0.01)
temp <- GT_ensembel[rownames(hmExp) %in% GT_ensembel$ENSEMBL, ]
temp <- temp[match(rownames(hmExp), temp$ENSEMBL), ]
rownames(hmExp) <- temp$SYMBOL

pheatmap(
    mat = hmExp,
    annotation = type,
    color = colorRampPalette(c("green", "black", "red"))(50),
    cluster_cols = F,
    show_colnames = F,
    show_rownames = T,
    fontsize = 12,
    fontsize_col = 10,
    fontsize_row = 10
)
```

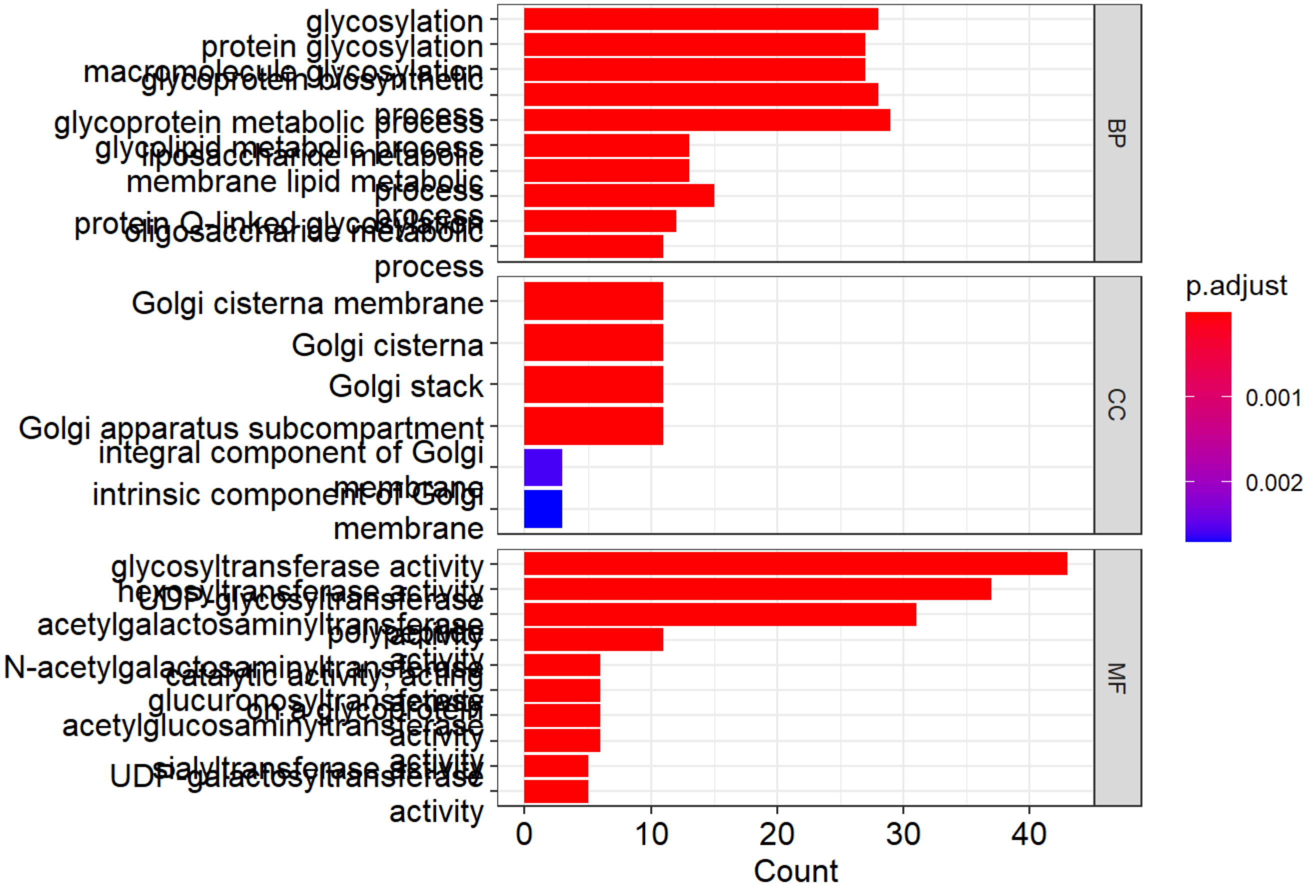


```
data=data.frame()
for(i in rownames(hmExp)){
  data=rbind(data,cbind(expression=hmExp[i, ],gene=i,type))
}
p3 <- ggboxplot(
    data,
    x = "gene",
    y = "expression",
    color = "type",
    ylab = "Gene expression",
    xlab = "",
    palette = c("blue", "red")
) +
    rotate_x_text(60)
print(p3)
```

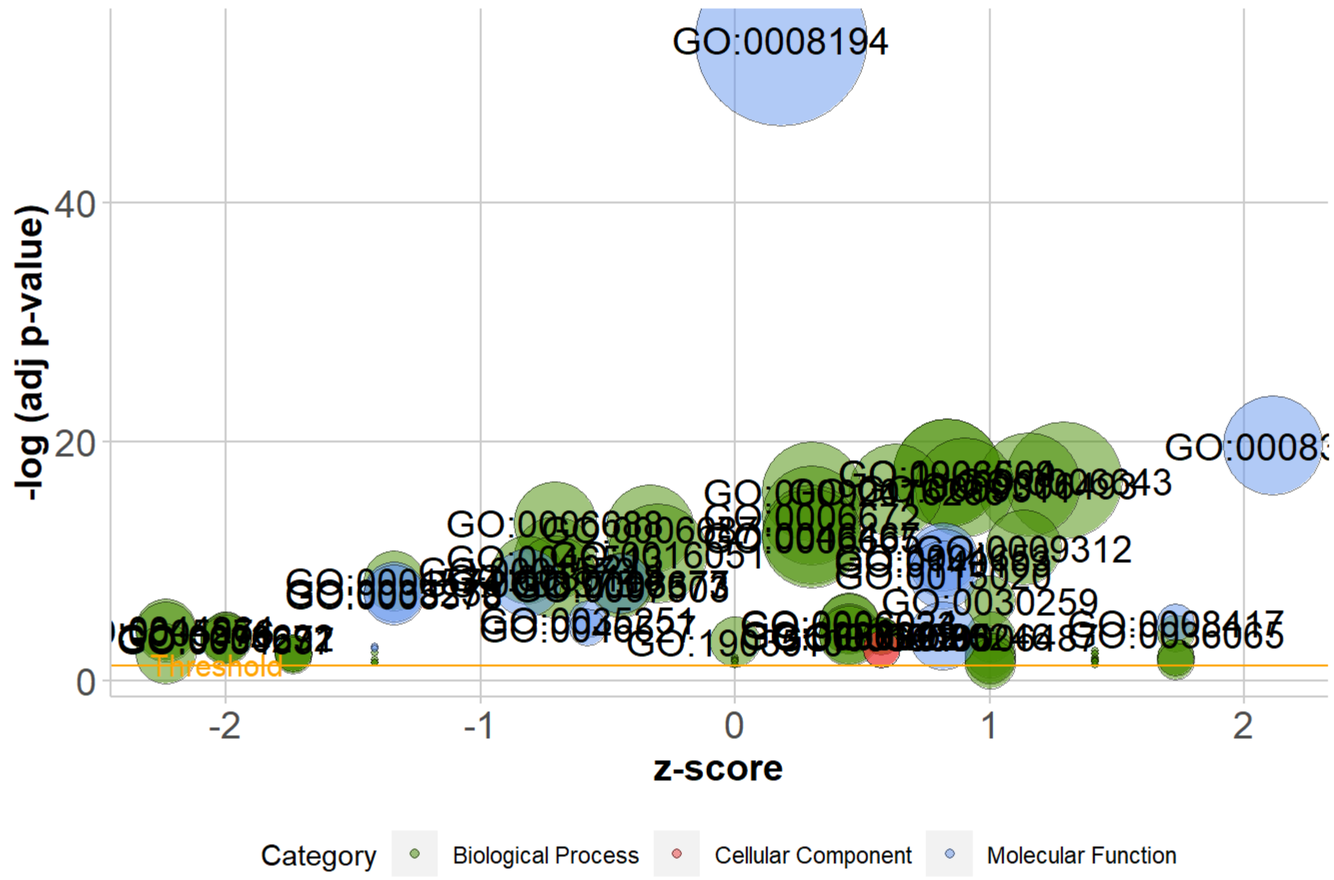


GO & KEGG

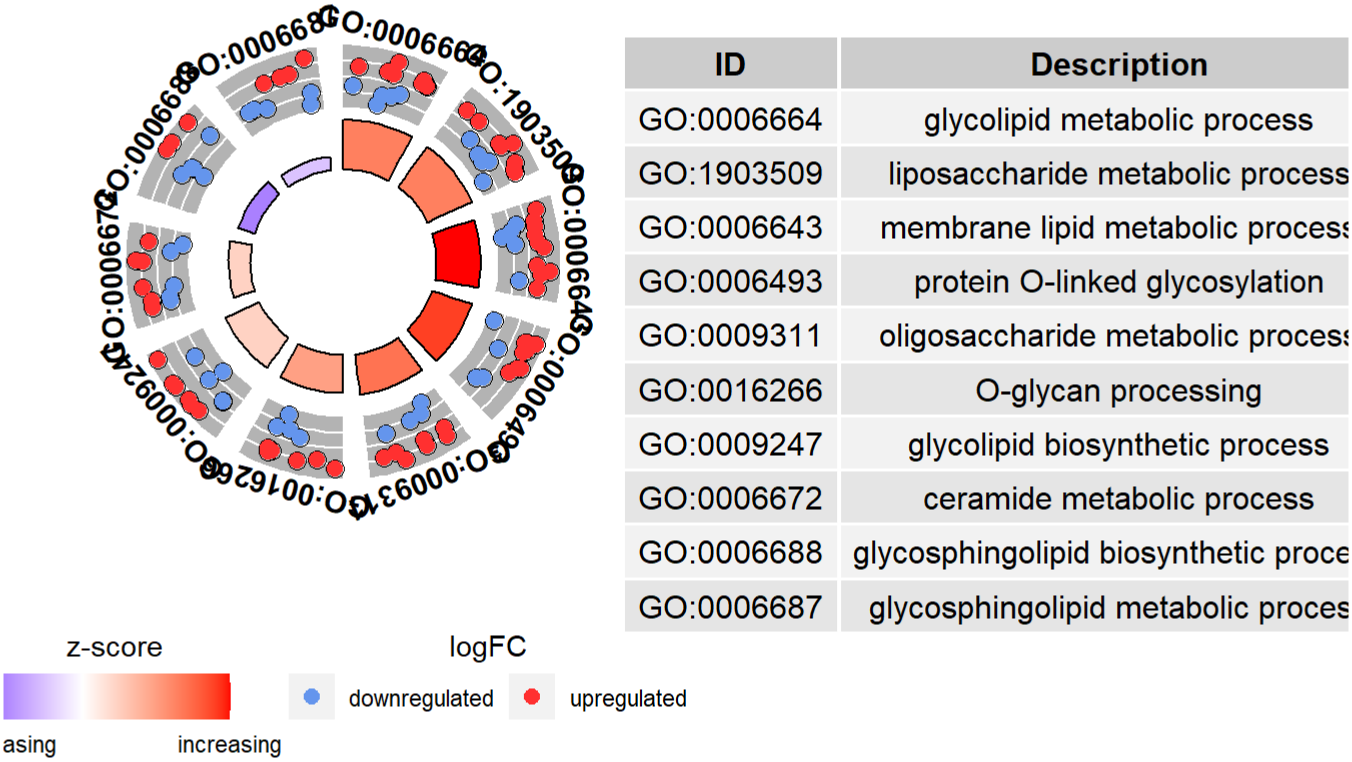
```
genes <- GT_ensembel[GT_ensembel$ENSEMBL %in% rownames(GT_de), ]$ENTREZID
# go <- enrichGO(
#   gene = genes,
#   OrgDb = org.Hs.eg.db,
#   pvalueCutoff = 0.05,
#   qvalueCutoff = 0.05,
#   ont = 'all',
#   readable = T
# )
# save(go, file = "go.Rdata")
load("go.Rdata")
barplot(go,
        drop = TRUE,
        showCategory = 10,
        split = "ONTOLOGY") + facet_grid(ONTOLOGY ~ ., scale = 'free')
```



```
# dotplot(go, showCategory = 10, split = "ONTOLOGY") +
#   facet_grid(ONTOLOGY ~ ., scale = 'free')
go_bubble <-
  data.frame(
    Category = go@result$ONTOLOGY,
    ID = go@result$ID,
    Term = go@result$Description,
    Genes = gsub("/", ", ", go@result$geneID),
    adj_pval = go@result$p.adjust
  )
gene_list <- data.frame(ID = GT_de$symbol, logFC = GT_de$logFC)
circ <- circle_dat(go_bubble, gene_list)
circ <- na.omit(circ)
GOBubble(circ, labels = 3, table.legend = F)
```

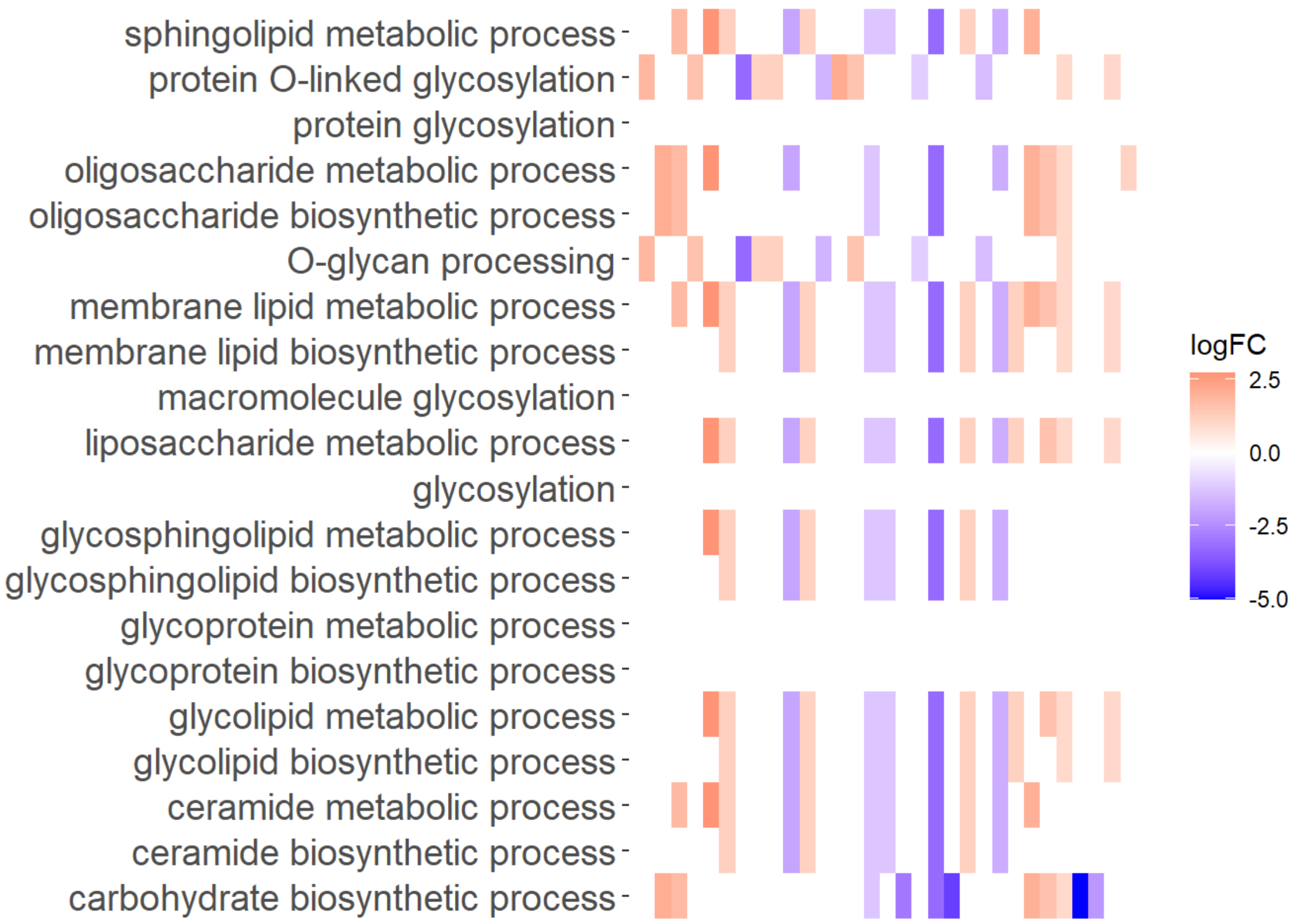


```
GOCircle(
  circ,
  rad1 = 2.5,
  rad2 = 3.5,
  label.size = 4,
  nsub = 10
)
```



ID	Description
GO:0006664	glycolipid metabolic process
GO:1903509	liposaccharide metabolic process
GO:0006643	membrane lipid metabolic process
GO:0006493	protein O-linked glycosylation
GO:0009311	oligosaccharide metabolic process
GO:0016266	O-glycan processing
GO:0009247	glycolipid biosynthetic process
GO:0006672	ceramide metabolic process
GO:0006688	glycosphingolipid biosynthetic proce
GO:0006687	glycosphingolipid metabolic proces

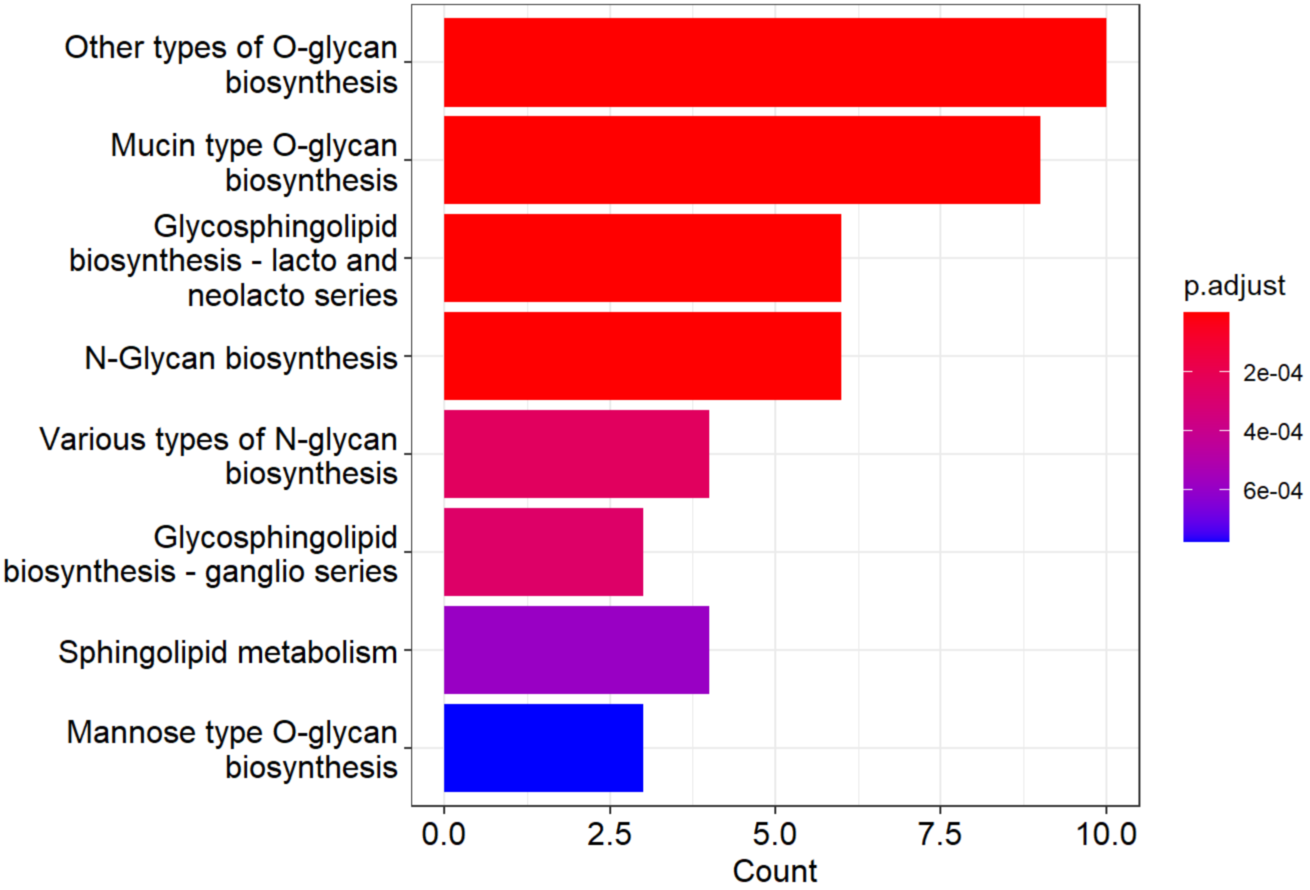
```
termNum = 20
chord <- chord_dat(circ, gene_list, go_bubble$Term[1:termNum])
GOHeat(chord, nlfc =1, fill.col = c('red', 'white', 'blue'))
```



```
kk <- enrichKEGG(
  gene = genes,
  organism = "hsa",
  pvalueCutoff = 0.05,
  qvalueCutoff = 0.05,
)
```

```
## Reading KEGG annotation online:
##
## Reading KEGG annotation online:
```

```
barplot(kk)
```



```
# dotplot(kk)
# kk_bubble <-
#   data.frame(
#     Category = "ALL",
#     ID = kk@result$ID,
#     Term = kk@result$Description,
#     Genes = gsub("/", ", ", kk@result$geneID),
#     adj_pval = kk@result$p.adjust
#   )
# circ <- circle_dat(kk_bubble, gene_list)
# circ <- na.omit(circ)
# GOBubble(circ, labels = 3,table.legend =F)
# GOCircle(circ,rad1=2.5,rad2=3.5,label.size=4,nsub=10)
# chord <- chord_dat(circ, gene_list, kk@result$Term[1:termNum])
# GOHeat(chord, nlfc =1, fill.col = c('red', 'white', 'blue'))
```

Survival Analysis

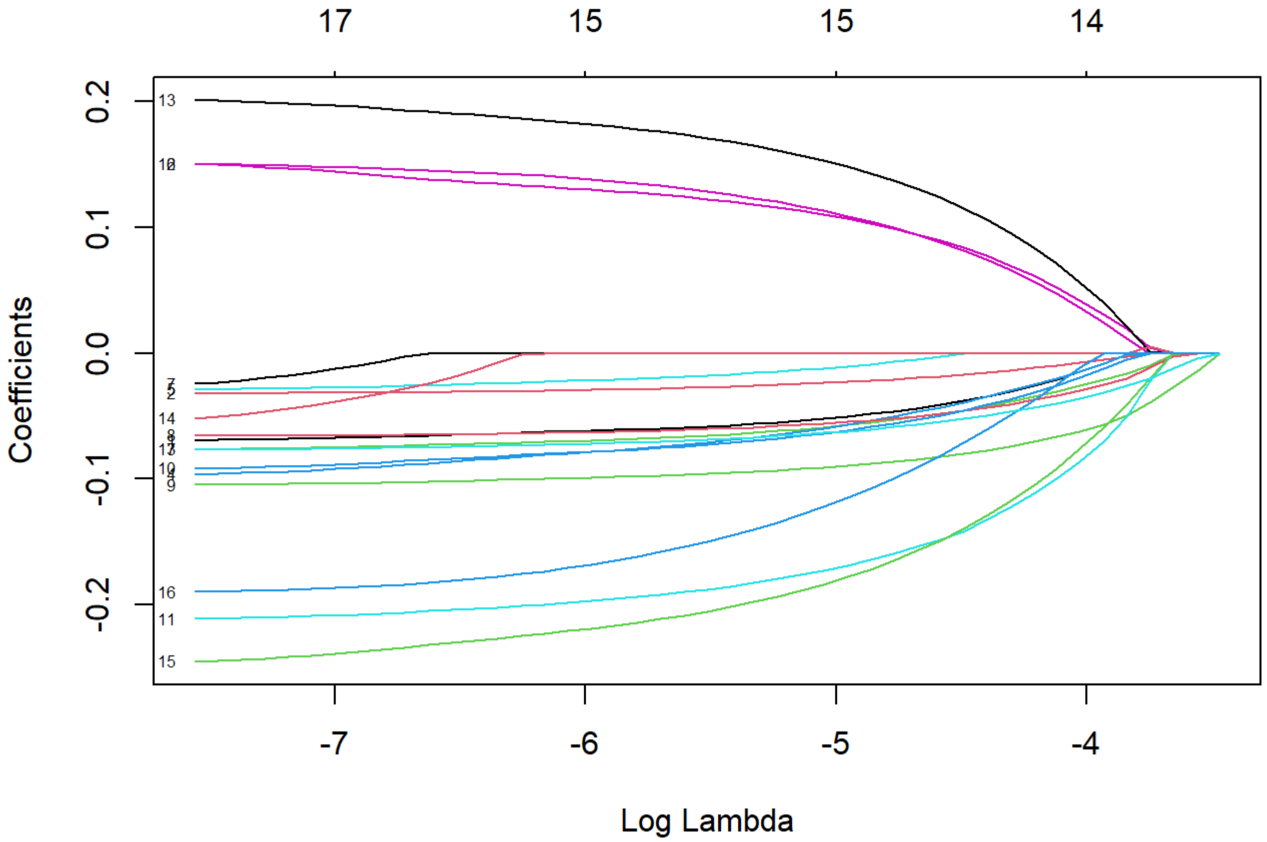
```
survOutput_cox <- gdcSurvivalAnalysis(gene = rownames(GT_de),
                                  method = "coxph",
                                  rna.expr = GTrnaExpr,
                                  metadata = metaMatrix.RNA)
survOutput_cox <- survOutput_cox[survOutput_cox$pValue<0.05, ]
```

```
survOutput_km <- gdcSurvivalAnalysis(gene = rownames(GT_de),
                                  method = "KM",
                                  rna.expr = GTrnaExpr,
                                  metadata = metaMatrix.RNA)
survOutput_km <- survOutput_cox[survOutput_km$pValue<0.05, ]
for (gene in rownames(survOutput_cox)) {
  gdcKMPlot(gene = gene,
            rna.expr = GTrnaExpr,
            metadata = metaMatrix.RNA,
            sep = "median")
}

# shinyKMPlot(gene = rownames(survOutput_km), rna.expr = GTrnaExpr, metadata = metaMatrix.RNA)
```

Regression Model

```
# read clinical
clinical <-
  read.table(
    "raw_data/Survival_SupplementalTable_S1_20171025_xena_sp",
    sep = "\t",
    check.names = F,
    header = T,
    row.names = 1
  )
clinical <- clinical[clinical$`cancer type abbreviation` == "BRCA", c("OS.time", "OS")]

# Filter out tumor samples
exp_time <- t(GTrnaExpr)
group=sapply(strsplit(rownames(exp_time),"\\-"),"[",4)
group=sapply(strsplit(group,""),"[",1)
group=gsub("2","1",group)
exp_time=exp_time[group==0, ]
exp_time <- avereps(exp_time)
exp_time <- exp_time[rowMeans(exp_time) > 0, ]
exp_time <- as.data.frame(exp_time)

# merge OS
temp <- intersect(rownames(exp_time), rownames(clinical))
exp_time <- cbind(clinical[temp, ], exp_time[temp, ])
exp_time <- na.omit(exp_time)

# unicox regression
outTab <- data.frame()
sigGenes <- c("OS.time", "OS")
for (i in colnames(exp_time[, 3:ncol(exp_time)])) {
  if (sd(exp_time[, i]) < 0.001) {
    next
  }
  cox <- coxph(Surv(OS.time, OS) ~ exp_time[, i], data = exp_time)
  cox_summary <- summary(cox)
  coxP <- cox_summary$coefficients[, "Pr(>|z|)"]
```

```
    if (coxP < 0.05) {
      sigGenes = c(sigGenes, i)
      outTab = rbind(
        outTab,
        cbind(
          id = i,
          HR = cox_summary$conf.int[, "exp(coef)"],
          HR.95L = cox_summary$conf.int[, "lower .95"],
          HR.95H = cox_summary$conf.int[, "upper .95"],
          pvalue = cox_summary$coefficients[, "Pr(>|z|)"]
        )
      )
    }
  }
uniSigExp=exp_time[,sigGenes]
uniSigExp$OS.time[uniSigExp$OS.time == 0] <- 1

## LASSO regression
x=as.matrix(uniSigExp[ ,c(3:ncol(uniSigExp))])
y=data.matrix(Surv(uniSigExp$OS.time,uniSigExp$OS))

# Single Train
set.seed(1010)
fit <- glmnet(x, y, family = "cox", maxit = 1000)
plot(fit, xvar = "lambda", label = TRUE)
```



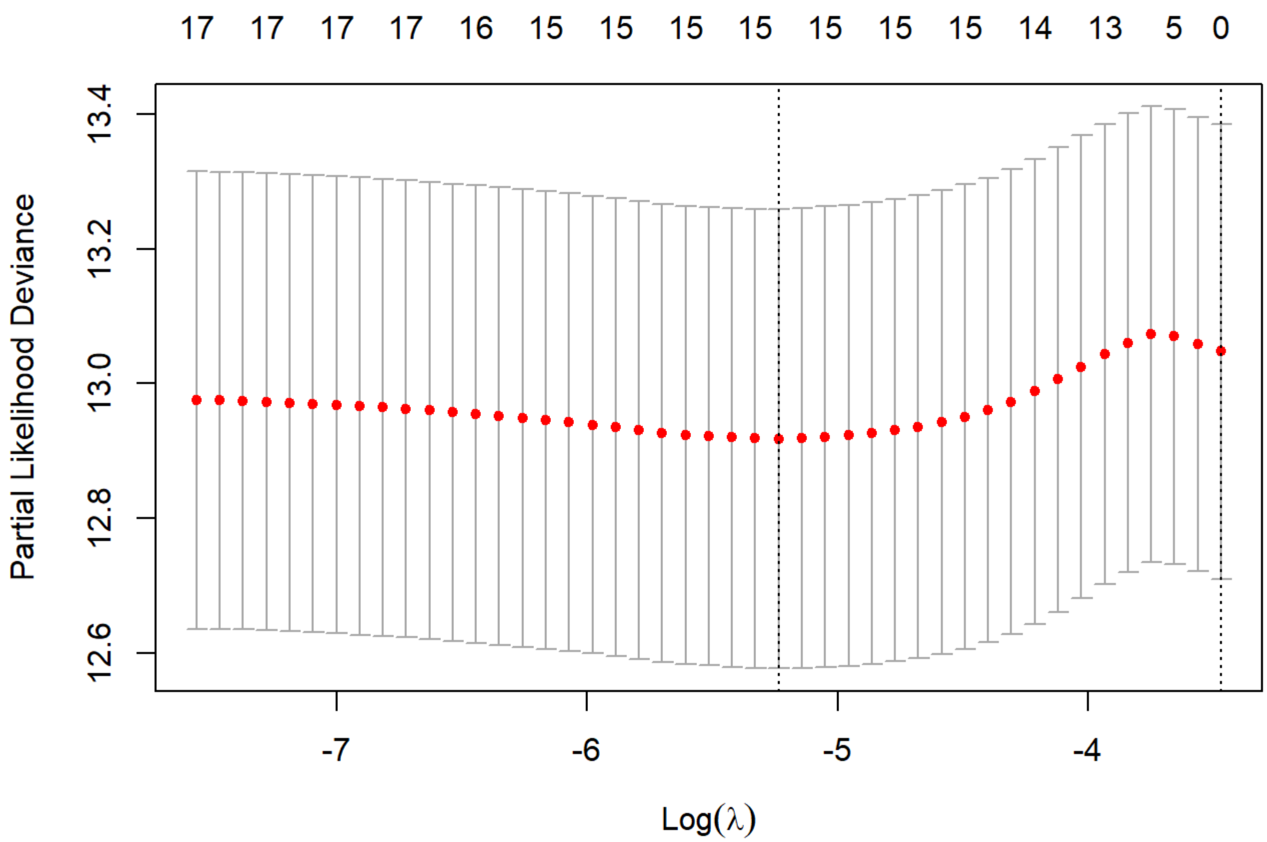
```
# Cross Validation
set.seed(202)
cvfit <- cv.glmnet(x, y, alpha=1, family="cox", maxit = 1000)
plot(cvfit)

# output
coef <- coef(fit, s = cvfit$lambda.min)
index <- which(coef != 0)
actCoef <- coef[index]
lassoGene=row.names(coef)[index]
lassoGene=c("OS.time","OS",lassoGene)
lassoSigExp=uniSigExp[,lassoGene]
temp <- GT_ensembel[GT_ensembel$ENSEMBL %in% colnames(lassoSigExp[, -(1:2)]), ]
temp <- temp[match(colnames(lassoSigExp[, -(1:2)]), temp$ENSEMBL), ]
colnames(lassoSigExp) <- c("OS.time","OS", temp$SYMBOL)


## cox model
multiCox=coxph(Surv(OS.time, OS) ~ ., data = lassoSigExp)
multiCox <- step(multiCox, direction = "both")
```

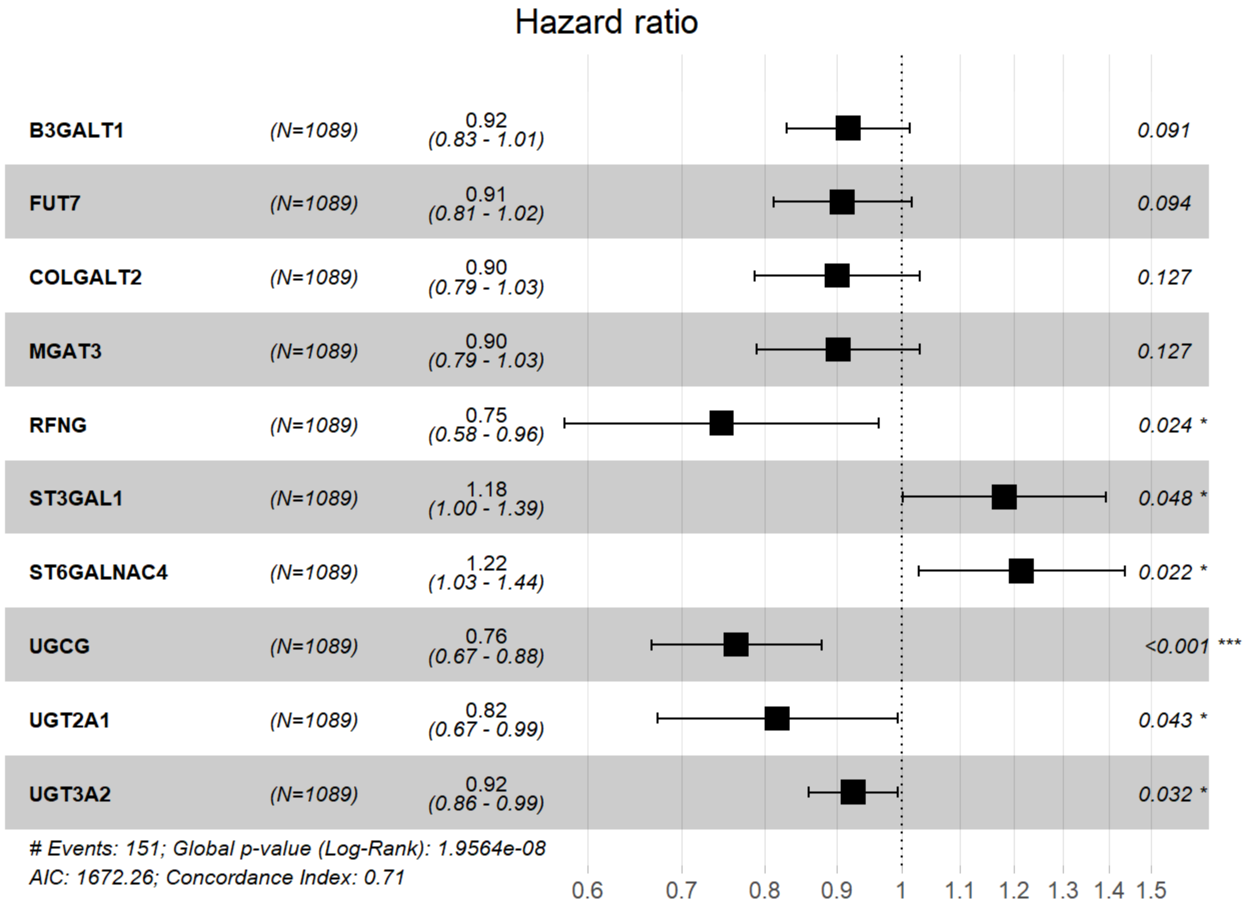
```
## Start:  AIC=1679.02
## Surv(OS.time, OS) ~ A4GALT + ABO + B3GALT1 + B3GALT4 + B4GALNT4 + 
##     C1GALT1C1 + FUT7 + COLGALT2 + MGAT3 + RFNG + ST3GAL1 + ST6GALNAC4 + 
##     UGCG + UGT2A1 + UGT3A2
## 
##              Df    AIC
## - B4GALNT4    1 1677.2
## - ABO         1 1677.5
## - A4GALT      1 1677.6
## - C1GALT1C1   1 1677.7
## - B3GALT4     1 1677.9
## - FUT7        1 1678.2
## - MGAT3       1 1678.8
## - RFNG        1 1679.0
## <none>          1679.0
## - B3GALT1     1 1679.1
## - COLGALT2    1 1679.4
## - ST3GAL1     1 1680.3
## - UGT3A2      1 1681.5
## - UGT2A1      1 1681.5
## - ST6GALNAC4  1 1682.4
## - UGCG        1 1688.5
## 
## Step:  AIC=1677.16
## Surv(OS.time, OS) ~ A4GALT + ABO + B3GALT1 + B3GALT4 + C1GALT1C1 + 
##     FUT7 + COLGALT2 + MGAT3 + RFNG + ST3GAL1 + ST6GALNAC4 + UGCG + 
##     UGT2A1 + UGT3A2
## 
##              Df    AIC
## - ABO         1 1675.6
## - A4GALT      1 1675.8
## - B3GALT4     1 1676.0
## - C1GALT1C1   1 1676.0
## - FUT7        1 1676.4
## - MGAT3       1 1676.9
## <none>          1677.2
## - B3GALT1     1 1677.3
## - COLGALT2    1 1677.6
## - RFNG        1 1678.0
## - ST3GAL1     1 1678.5
## + B4GALNT4    1 1679.0
## - UGT2A1      1 1679.6
## - UGT3A2      1 1679.6
## - ST6GALNAC4  1 1680.7
## - UGCG        1 1687.1
##
## Step:  AIC=1675.62
## Surv(OS.time, OS) ~ A4GALT + B3GALT1 + B3GALT4 + C1GALT1C1 +
##     FUT7 + COLGALT2 + MGAT3 + RFNG + ST3GAL1 + ST6GALNAC4 + UGCG +
##     UGT2A1 + UGT3A2
##
##               Df    AIC
## - A4GALT      1 1674.3
## - C1GALT1C1   1 1674.4
## - B3GALT4     1 1674.5
## - FUT7        1 1674.9
## <none>          1675.6
## - MGAT3       1 1675.8
## - B3GALT1     1 1675.9
## - COLGALT2    1 1676.0
## - RFNG        1 1676.2
## + ABO         1 1677.2
## - ST3GAL1     1 1677.4
## + B4GALNT4    1 1677.5
## - UGT2A1      1 1678.0
## - UGT3A2      1 1678.3
## - ST6GALNAC4  1 1679.2
## - UGCG        1 1687.5
##
## Step:  AIC=1674.3
## Surv(OS.time, OS) ~ B3GALT1 + B3GALT4 + C1GALT1C1 + FUT7 + COLGALT2 +
##     MGAT3 + RFNG + ST3GAL1 + ST6GALNAC4 + UGCG + UGT2A1 + UGT3A2
##
##               Df    AIC
## - C1GALT1C1   1 1673.3
## - B3GALT4     1 1673.6
## - FUT7        1 1674.0
## <none>          1674.3
## - MGAT3       1 1674.7
## - B3GALT1     1 1674.7
## - COLGALT2    1 1674.7
## - RFNG        1 1675.5
## + A4GALT      1 1675.6
## + ABO         1 1675.8
## - ST3GAL1     1 1675.8
## + B4GALNT4    1 1676.2
## - UGT2A1       1 1676.8
## - UGT3A2       1 1677.0
## - ST6GALNAC4  1 1677.5
## - UGCG        1 1687.0
##
## Step:  AIC=1673.29
## Surv(OS.time, OS) ~ B3GALT1 + B3GALT4 + FUT7 + COLGALT2 + MGAT3 +
##     RFNG + ST3GAL1 + ST6GALNAC4 + UGCG + UGT2A1 + UGT3A2
##
##              Df    AIC
## - B3GALT4     1 1672.3
## <none>          1673.3
## - COLGALT2    1 1673.6
## - FUT7        1 1673.7
## - MGAT3       1 1673.7
## + C1GALT1C1   1 1674.3
## - B3GALT1     1 1674.4
## + A4GALT      1 1674.4
## + ABO         1 1674.7
## + B4GALNT4    1 1675.1
## - ST3GAL1     1 1675.1
## - RFNG        1 1675.4
## - UGT2A1      1 1675.7
## - UGT3A2      1 1676.1
## - ST6GALNAC4  1 1676.8
## - UGCG        1 1686.0
##
## Step:  AIC=1672.26
## Surv(OS.time, OS) ~ B3GALT1 + FUT7 + COLGALT2 + MGAT3 + RFNG +
##     ST3GAL1 + ST6GALNAC4 + UGCG + UGT2A1 + UGT3A2
##
##              Df    AIC
## <none>          1672.3
## - COLGALT2    1 1672.7
## - MGAT3       1 1672.7
## - FUT7        1 1673.0
## + A4GALT      1 1673.0
## - B3GALT1     1 1673.1
## + B3GALT4     1 1673.3
## + C1GALT1C1   1 1673.6
## + ABO         1 1673.6
## + B4GALNT4    1 1674.1
## - ST3GAL1     1 1674.2
## - UGT2A1      1 1675.0
## - UGT3A2      1 1675.0
## - RFNG        1 1675.4
## - ST6GALNAC4  1 1675.5
## - UGCG        1 1685.3
```

```
multiCox_sum <- summary(multiCox)



# output parameters
```

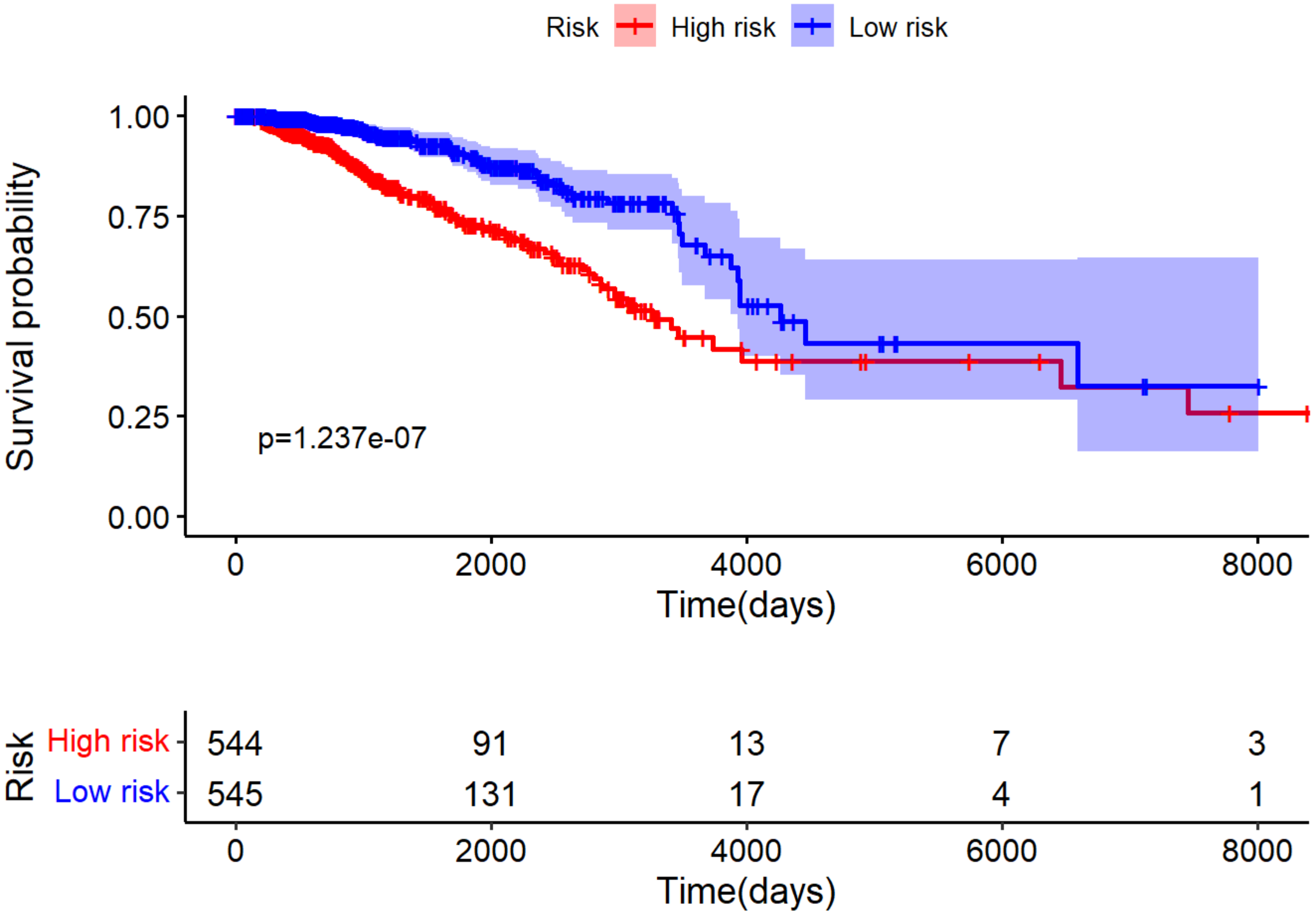
```
outTab=data.frame()
outTab=cbind(
    coef=multiCox_sum$coefficients[,"coef"],
    HR=multiCox_sum$conf.int[,"exp(coef)"],
    HR.95L=multiCox_sum$conf.int[,"lower .95"],
    HR.95H=multiCox_sum$conf.int[,"upper .95"],
    pvalue=multiCox_sum$coefficients[,"Pr(>|z|)"])

## predict patients
riskScore=predict(multiCox,type="risk",newdata=lassoSigExp)
coxGene=rownames(multiCox_sum$coefficients)
coxGene=gsub("`","",coxGene)
outCol=c("OS.time","OS",coxGene)
risk=as.vector(ifelse(riskScore>median(riskScore),"high","low"))
riskOut=cbind(lassoSigExp[,outCol],riskScore,risk)
#

## plot forest
ggforest(model = multiCox,
         main = "Hazard ratio",
         cpositions = c(0.02,0.22, 0.4),
         fontsize = 0.7,
         refLabel = "reference",
         noDigits = 2)
```

```
## Warning in .get_data(model, data = data): The `data` argument is not provided.
## Data will be extracted from model fit.
```


17 17 17 17 16 15 15 15 15 15 15 15 14 13 5 0
Partial Likelihood Deviance
12.6
12.8
13.0
13.2
13.4
-7
-6
-5
-4
Log(λ)

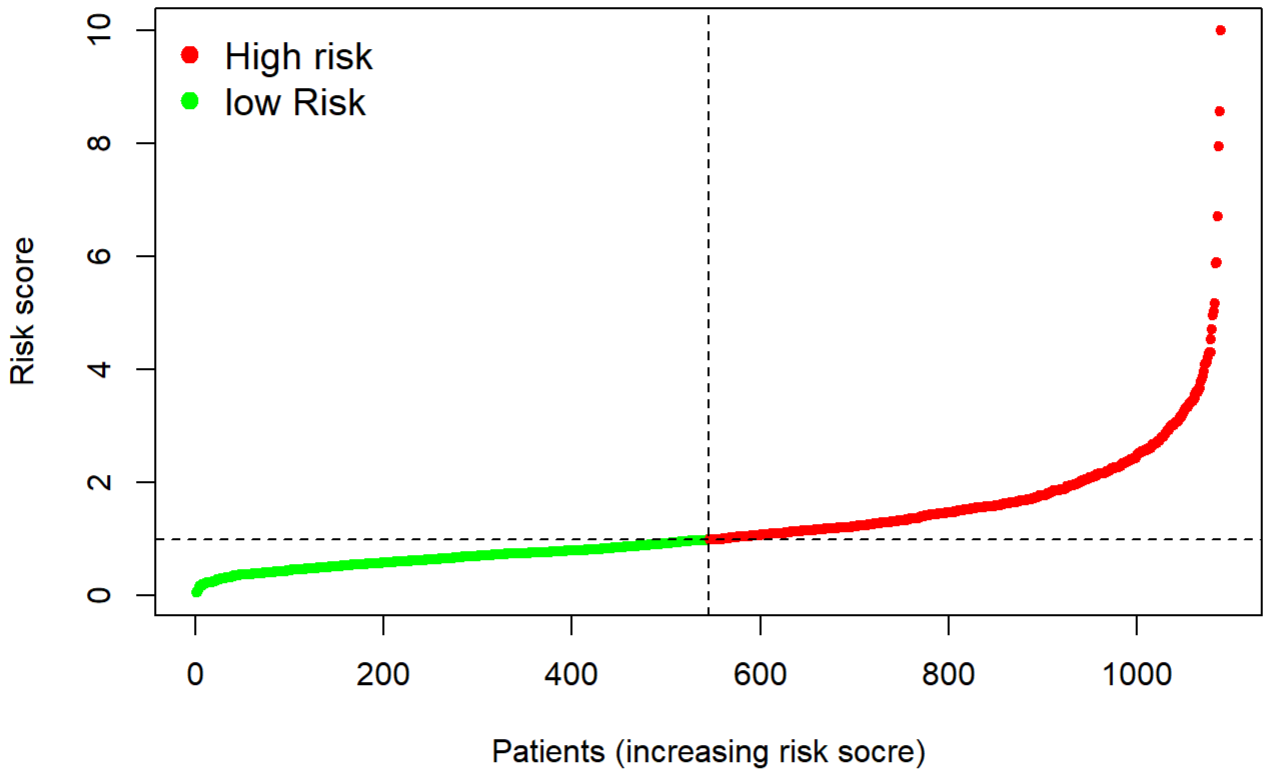

Hazard ratio
B3GALT1 (N=1089) 0.92 (0.83 - 1.01) 0.091
FUT7 (N=1089) 0.91 (0.81 - 1.02) 0.094
COLGALT2 (N=1089) 0.90 (0.79 - 1.03) 0.127
MGAT3 (N=1089) 0.90 (0.79 - 1.03) 0.127
RFNG (N=1089) 0.75 (0.58 - 0.96) 0.024 *
ST3GAL1 (N=1089) 1.18 (1.00 - 1.39) 0.048 *
ST6GALNAC4 (N=1089) 1.22 (1.03 - 1.44) 0.022 *
UGCG (N=1089) 0.76 (0.67 - 0.88) <0.001 ***
UGT2A1 (N=1089) 0.82 (0.67 - 0.99) 0.043 *
UGT3A2 (N=1089) 0.92 (0.86 - 0.99) 0.032 *
# Events: 151; Global p-value (Log-Rank): 1.9564e-08
AIC: 1672.26; Concordance Index: 0.71
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5

```
## survplot
diff=survdiff(Surv(OS.time,OS) ~risk,data = riskOut)
pValue=1-pchisq(diff$chisq,df=1)
pValue=signif(pValue,4)
pValue=format(pValue, scientific = TRUE)

fit <- survfit(Surv(OS.time,OS) ~ risk, data = riskOut)

ggsurvplot(fit,
           data=riskOut,
           conf.int=TRUE,
           pval=paste0("p=",pValue),
           pval.size=4,
           risk.table=TRUE,
           legend.labs=c("High risk", "Low risk"),
           legend.title="Risk",
           xlab="Time(days)",
           risk.table.title="",
           palette=c("red", "blue"),
           risk.table.height=.3)
```



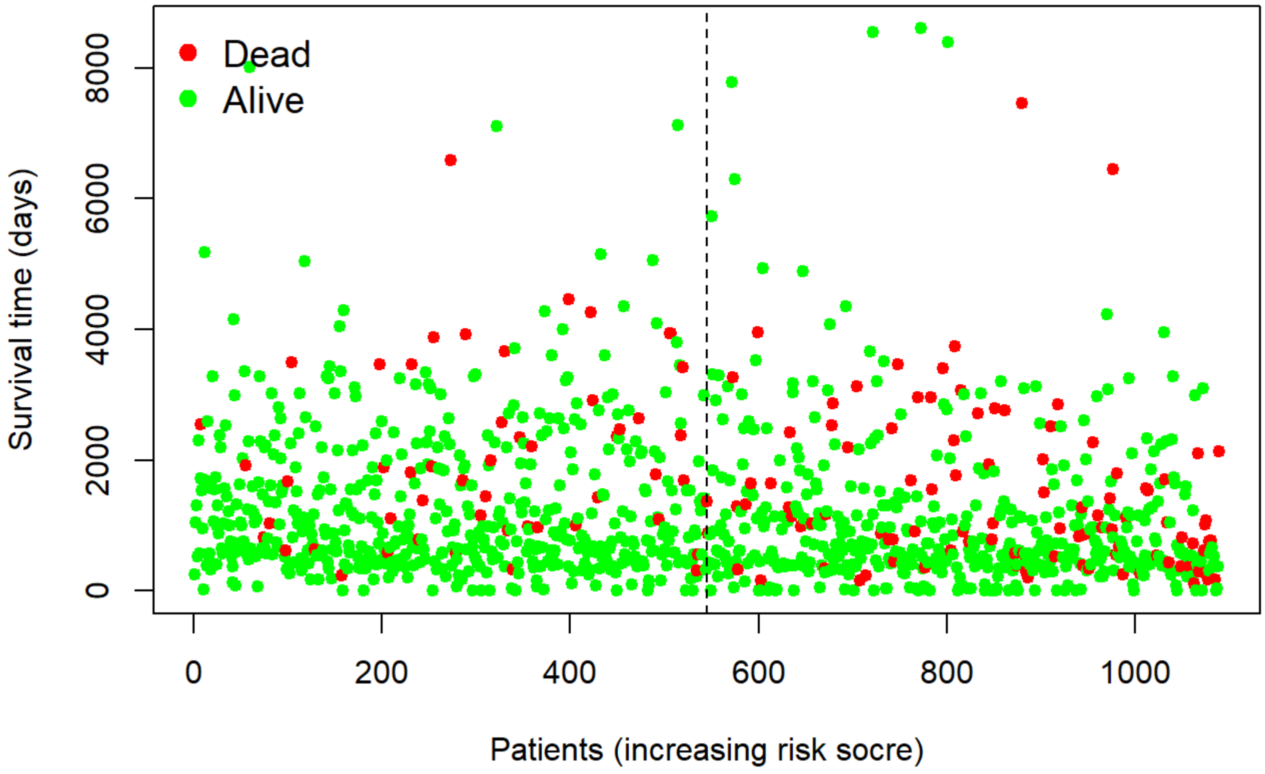
```
riskOut <- riskOut[order(riskOut$riskScore), ]

# risk curve
riskClass=riskOut[,"risk"]
```

```
lowLength=length(riskClass[riskClass=="low"])
highLength=length(riskClass[riskClass=="high"])
line=riskOut[,"riskScore"]
line[line>10]=10

plot(line,
     type="p",
     pch=20,
     xlab="Patients (increasing risk socre)",
     ylab="Risk score",
     col=c(rep("green",lowLength),
     rep("red",highLength)))
abline(h=median(riskOut$riskScore),v=lowLength,lty=2)
legend("topleft", c("High risk", "low Risk"),bty="n",pch=19,col=c("red","green"),cex=1.2)
```

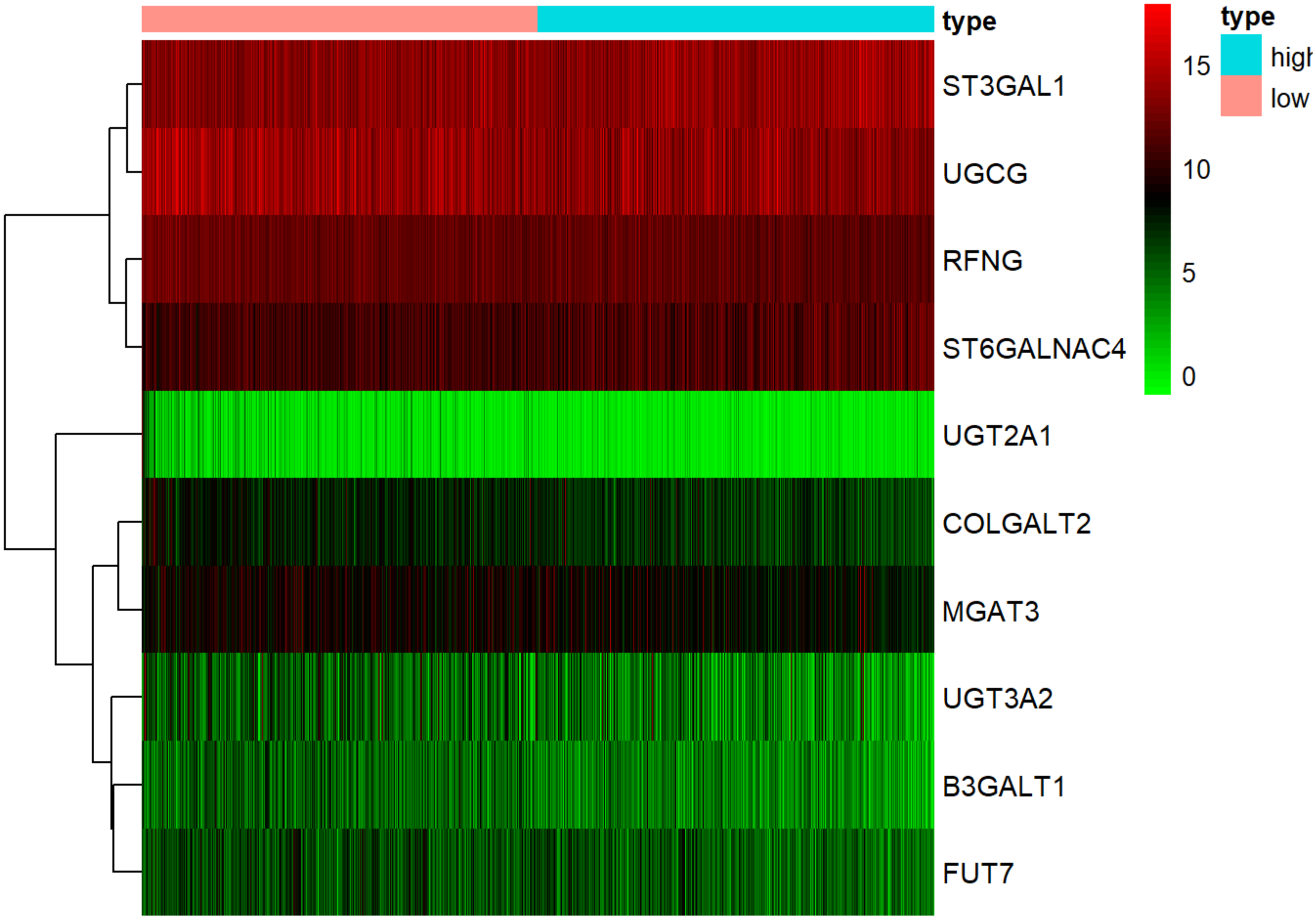


```
# survial status
color=as.vector(riskOut$OS)
color[color==1]="red"
color[color==0]="green"

plot(riskOut$OS.time,
     pch=19,
     xlab="Patients (increasing risk socre)",
     ylab="Survival time (days)",
     col=color,
     cex=.8)
```

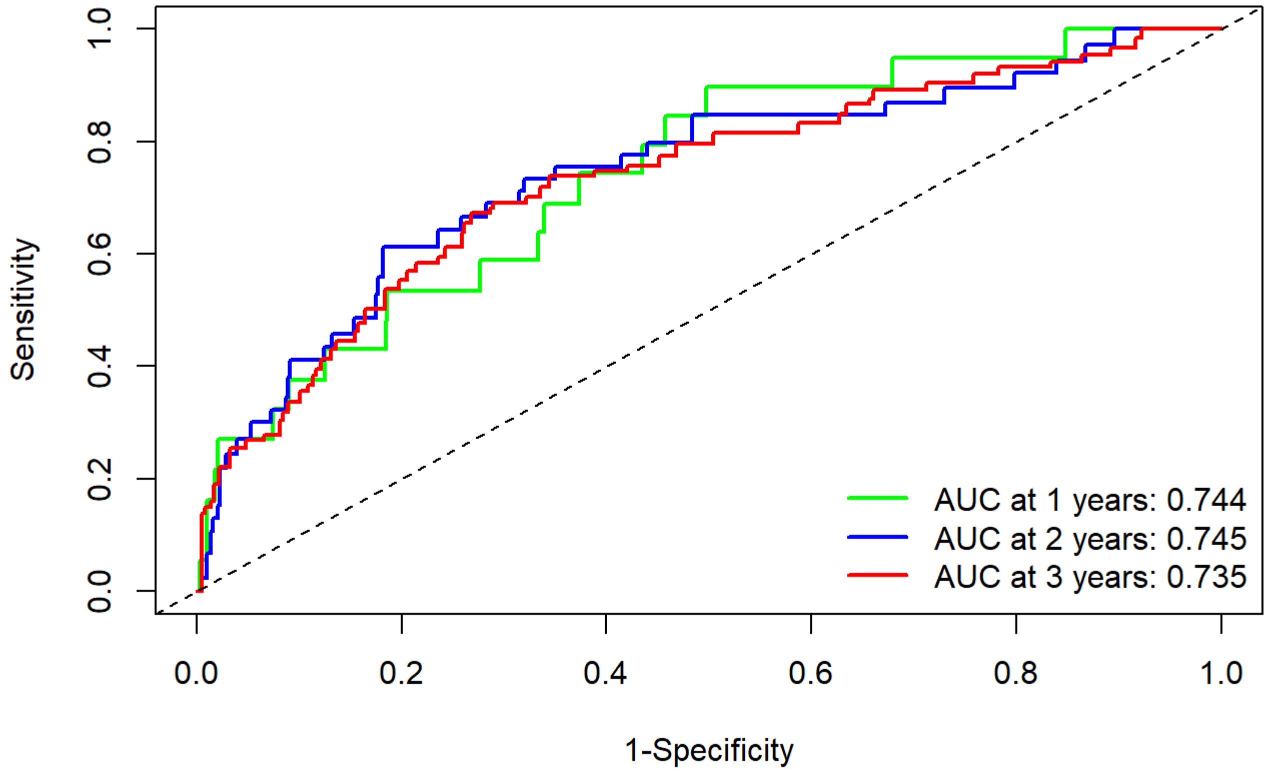
```
legend("topleft", c("Dead", "Alive"),bty="n",pch=19,col=c("red","green"),cex=1.2)
abline(v=lowLength,lty=2)
```



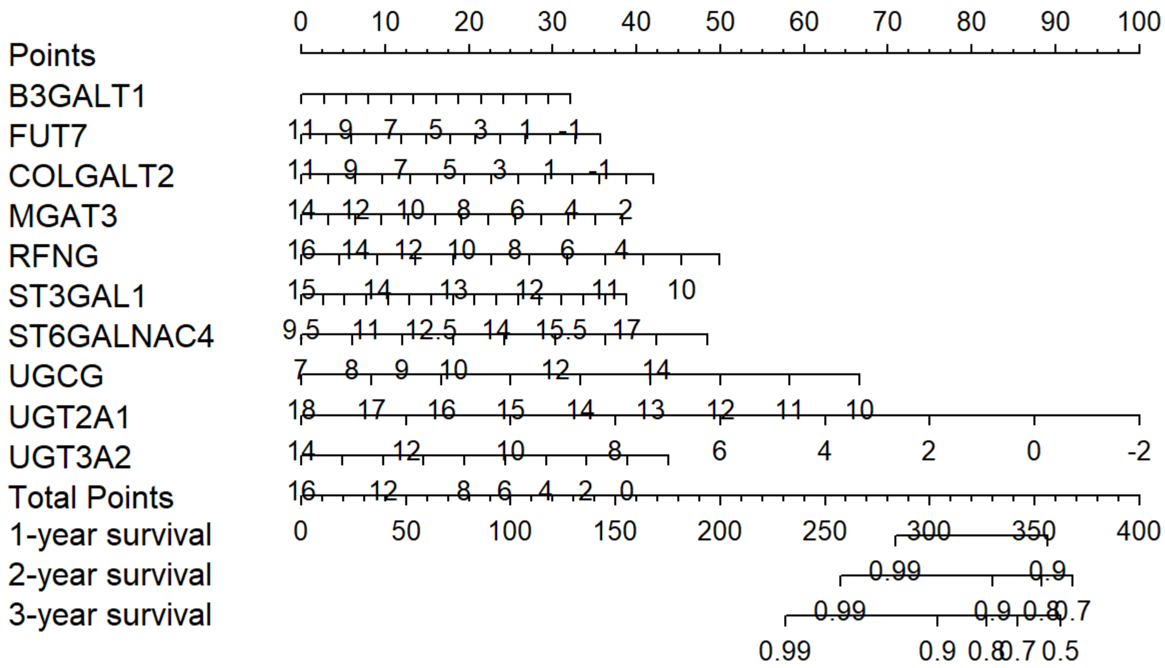
```
# risk_heatmap
riskOut1=riskOut[c(3:(ncol(riskOut)-2))]
riskOut1=t(riskOut1)
annotation=data.frame(type=riskOut[,ncol(riskOut)])
rownames(annotation)=rownames(riskOut)

pheatmap(riskOut1,
         annotation=annotation,
         cluster_cols = FALSE,
         fontsize_row=11,
         show_colnames = F,
         fontsize_col=3,
         color = colorRampPalette(c("green", "black", "red"))(50) )
```



```
## ROC curve
ROC_riskOut=timeROC(T=riskOut$OS.time,delta=riskOut$OS,
               marker=riskOut$riskScore,cause=1,
               weighting='aalen',
               times=c(1:5)*365,ROC=TRUE)

plot(ROC_riskOut,time=1*365,col='green',title=FALSE,lwd=2)
plot(ROC_riskOut,time=2*365,col='blue',add=TRUE,title=FALSE,lwd=2)
plot(ROC_riskOut,time=3*365,col='red',add=TRUE,title=FALSE,lwd=2)
legend('bottomright',
       c(paste0('AUC at 1 years: ',round(ROC_riskOut$AUC[1],3)),
         paste0('AUC at 2 years: ',round(ROC_riskOut$AUC[2],3)),
         paste0('AUC at 3 years: ',round(ROC_riskOut$AUC[3],3))),
       col=c("green",'blue','red'),lwd=2,bty = 'n')
```



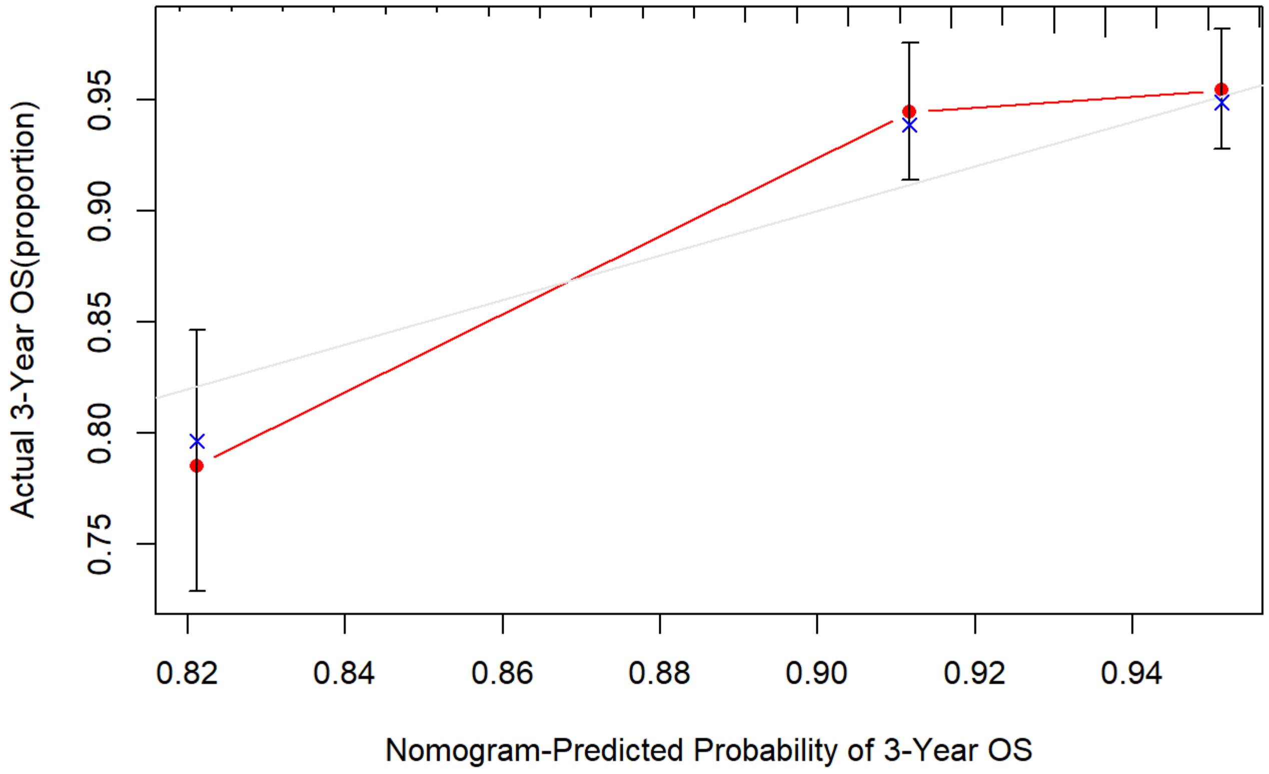
```
## nomogram
rt=riskOut[,1:(ncol(riskOut)-2)]
dd <- datadist(rt[, ])
options(datadist="dd")

f <- cph(Surv(OS.time, OS) ~ B3GALT1 + FUT7 + COLGALT2 + MGAT3 + RFNG + ST3GAL1 + ST6GALNAC4 +
 UGCG + UGT2A1 + UGT3A2, x=T, y=T, surv=T, data=rt, time.inc=1)
surv <- Survival(f)
#建立nomogram
nom <- nomogram(f, fun=list(function(x) surv(1*365, x), function(x) surv(2*365, x), function(x
) surv(3*365, x)),
                lp=F, funlabel=c("1-year survival", "2-year survival", "3-year survival"),
                maxscale=100,
                fun.at=c(0.99, 0.9, 0.8, 0.7, 0.5, 0.3,0.1,0.01))
#nomogram可视化
plot(nom)
```



```
#calibration curve
time=3*365
f <- cph(Surv(OS.time, OS) ~ B3GALT1 + FUT7 + COLGALT2 + MGAT3 + RFNG + ST3GAL1 + ST6GALNAC4 +
 UGCG + UGT2A1 + UGT3A2, x=T, y=T, surv=T, data=rt, time.inc=3*365)
cal <- calibrate(f, cmethod="KM", method="boot", u=time, m=300, B=1000)
```

```
## Using Cox survival estimates at 1095 Days
```

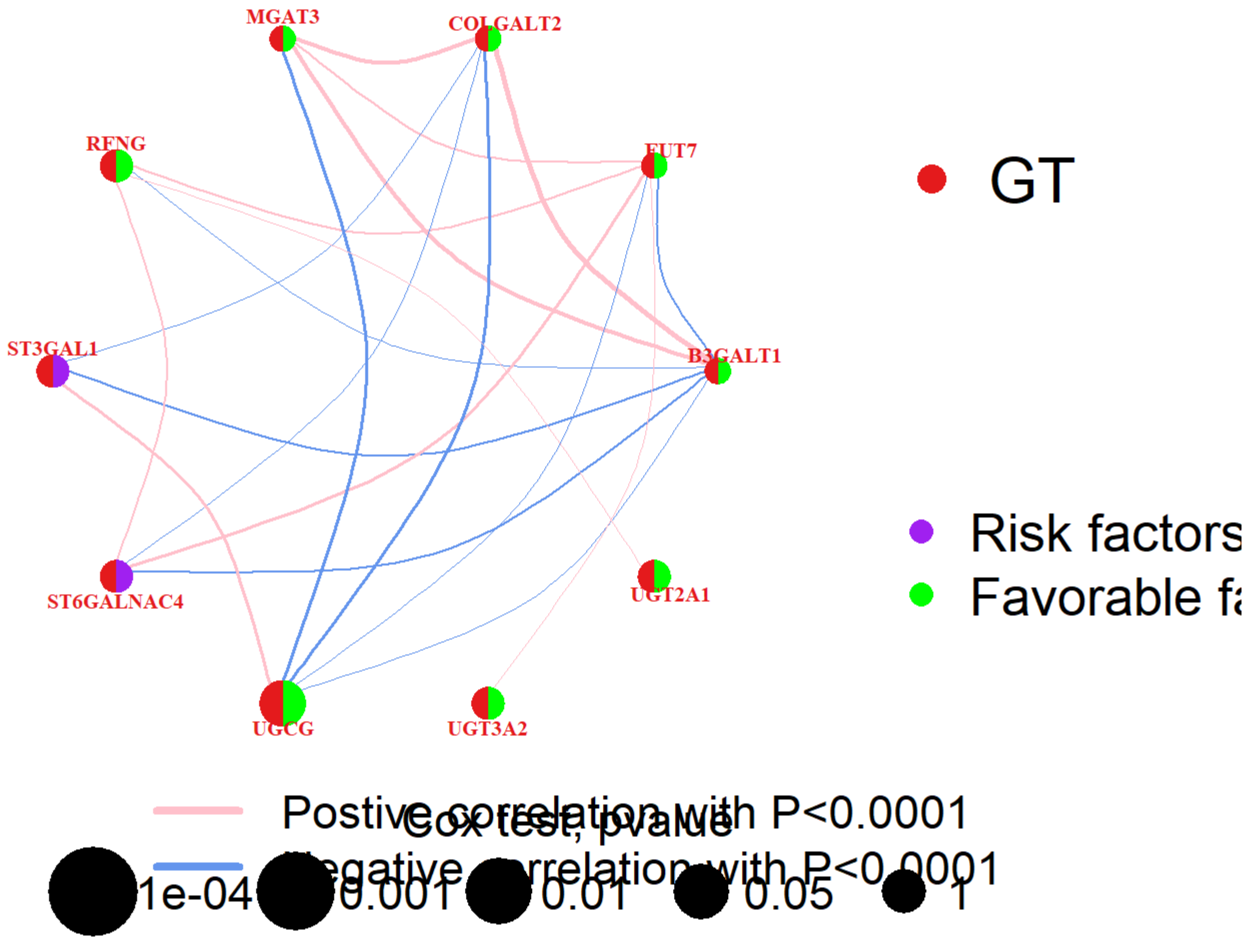
```
plot(cal,xlab="Nomogram-Predicted Probability of 3-Year OS",ylab="Actual 3-Year OS(proportion)
",col="red",sub=F)
```



cli_network

```
outTab <- as.data.frame(outTab)
t <- GT_ensembel[GT_ensembel$SYMBOL %in% rownames(outTab), ]
gene.exp <- as.data.frame(GTrnaExpr[match(t$ENSEMBL, rownames(GTrnaExpr)), ])
rownames(gene.exp) <- t$SYMBOL
gene.cox <- outTab[match(t$SYMBOL, rownames(outTab)), ]

#准备网络文件
gene.cor <- corr.test(t(gene.exp))
gene.cor.cor <- gene.cor$r
gene.cor.pvalue <- gene.cor$p
gene.cor.cor[upper.tri(gene.cor.cor)] = NA
gene.cor.pvalue[upper.tri(gene.cor.pvalue)] = NA
gene.cor.cor.melt <- melt(gene.cor.cor)    #gene1 \t gene2 \t cor
gene.cor.pvalue.melt <- melt(gene.cor.pvalue)
gene.melt <- data.frame(from = gene.cor.cor.melt$Var2,to=gene.cor.cor.melt$Var1,cor=gene.cor.cor.melt$value,pvalue=gene.cor.pvalue.melt$value)
gene.melt <- gene.melt[gene.melt$from!=gene.melt$to&!is.na(gene.melt$pvalue),,drop=F]
gene.edge <- gene.melt[gene.melt$pvalue<0.0001,,drop=F]
gene.edge$color <- ifelse(gene.edge$cor>0,'pink','#6495ED')
gene.edge$weight <- abs(gene.edge$cor)*6

#准备节点属性属性文件
gene.node <- as.data.frame(t$SYMBOL)
gene.node$group <- "GT"
names(gene.node)[1] <- "id"
group.color <- colorRampPalette(brewer.pal(9, "Set1"))(length(unique(gene.node$group)))
gene.node$color <- group.color[as.numeric(as.factor(gene.node$group))]
gene.node$shape <- "circle"
gene.node$frame <- ifelse(gene.cox$HR>1,'purple',"green")
gene.node$pvalue <- gene.cox$pvalue
# pvalue size
pvalue.breaks <- c(0,0.0001,0.001,0.01,0.05,1)
pvalue.size <- c(16,14,12,10,8)
cutpvalue <- cut(gene.node$pvalue,breaks=pvalue.breaks)
gene.node$size <- pvalue.size[as.numeric(cutpvalue)]

g <- graph.data.frame(gene.edge, directed = F)
node <- gene.node[match(names(components(g)$membership), gene.node$id), ]
if(!is.na(match('color',colnames(node)))) V(g)$color = node$color
if(!is.na(match('size',colnames(node)))) V(g)$size = node$size
if(!is.na(match('shape',colnames(node)))) V(g)$shape = node$shape
if(!is.na(match('frame',colnames(node)))) V(g)$frame = node$frame

# plot
# pdf(file="network.pdf", width=10, height=8)
par(mar=c(0,0,0,0))
layout(matrix(c(1,1,4,2,3,4),nc=2),height=c(4,4,2),width=c(8,3))

#节点坐标
coord = layout_in_circle(g)
degree.x = acos(coord[,1])
degree.y = asin(coord[,2])
degree.alpha = c()
for(i in 1:length(degree.x)){
    if(degree.y[i]<0) degree.alpha=c(degree.alpha,2*pi-degree.x[i]) else degree.alpha=c(degree
.alpha,degree.x[i])
}
degree.cut.group = (0:8)/4*pi
degree.cut.group[1] = -0.0001
degree.cut = cut(degree.alpha,degree.cut.group)
degree.degree = c(-pi/4,-pi/4,-pi/2,-pi/2,pi/2,pi/2,pi/2,pi/4)
degree = degree.degree[as.numeric(degree.cut)]

#定义饼图,左半圆颜色代表基因的属性,右半圆代表基因的风险,哪些基因是高风险基因,还是低风险基因
values <- lapply(node$id,function(x)c(1,1))
V(g)$pie.color = lapply(1:nrow(node),function(x)c(node$color[x],node$frame[x]))
V(g)$frame = NA

#绘制图形
plot(g,layout=layout_in_circle,vertex.shape="pie",vertex.pie=values,
    vertex.label.cex=V(g)$lable.cex,edge.width = E(g)$weight,edge.arrow.size=0,
    vertex.label.color=V(g)$color,vertex.frame.color=V(g)$frame,edge.color=E(g)$color,
    vertex.label.cex=2,vertex.label.font=2,vertex.size=V(g)$size,edge.curved=0.4,
    vertex.color=V(g)$color,vertex.label.dist=1,vertex.label.degree=degree)
# label.degree : zero means to the right; and pi means to the left; up is -pi/2 and down is pi/2;  The default value is -pi/4
# label.dist If it is 0 then the label is centered on the vertex; If it is 1 then the label is displayed beside the vertex.
```

```
#绘制节点属性图例(基因的属性)
par(mar=c(0,0,0,0))
plot(1,type="n",xlab="",ylab="",axes=F)
groupinfo = unique(data.frame(group=node$group,color=node$color))
legend("left",legend=groupinfo$group,col=groupinfo$color,pch=16,bty="n",cex=3)
#绘制基因风险的图例(哪些基因是高风险的基因,哪些基因是低风险的基因)
par(mar=c(0,0,0,0))
plot(1,type="n",xlab="",ylab="",axes=F)
legend("left",legend=c('Risk factors','Favorable factors'),col=c('purple','green'),pch=16,bty=
"n",cex=2.5)
#绘制预后pvalue图例
par(mar=c(0,0,0,0))
plot(1,type="n",xlab="",axes=F,ylab="")
legend("top",legend=c('Postive correlation with P<0.0001','Negative correlation with P<0.0001'
),lty=1,lwd=4,col=c('pink','#6495ED'),bty="n",cex=2.2)
legend('bottom',legend=c(0.0001,0.001,0.01,0.05,1),pch=16,pt.cex=c(1.6,1.4,1.2,1,0.8)*6,bty="n
",ncol=5,cex=2.2,col="black",title="Cox test, pvalue")
```



```
# dev.off()
```

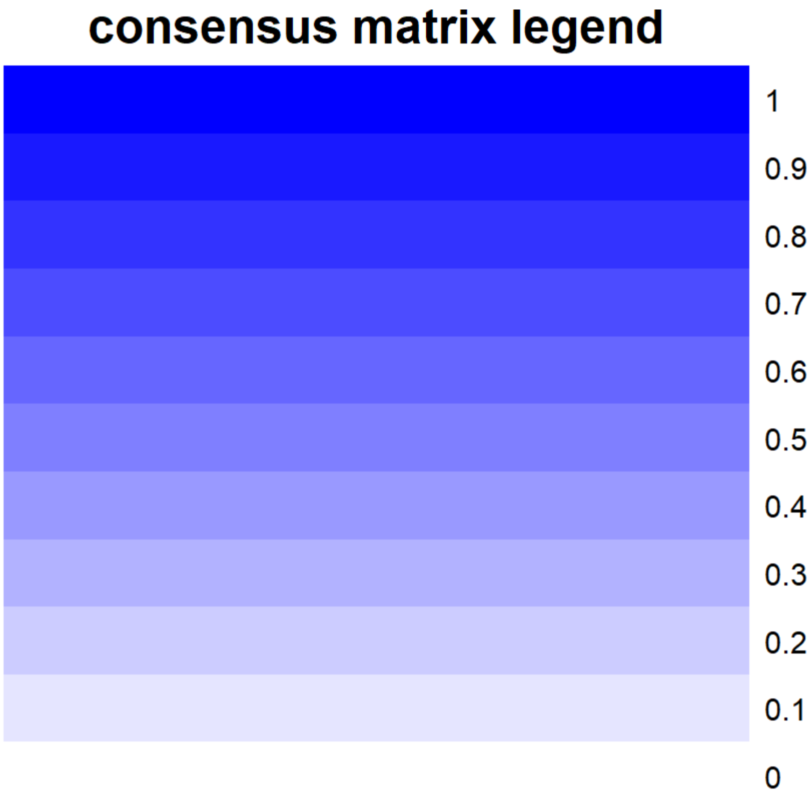
cluster

```
GTrnaExpr_symbol <- GTrnaExpr
rownames(GTrnaExpr_symbol) <- GT_ensembel[match(rownames(GTrnaExpr), GT_ensembel$ENSEMBL), ]$S
```

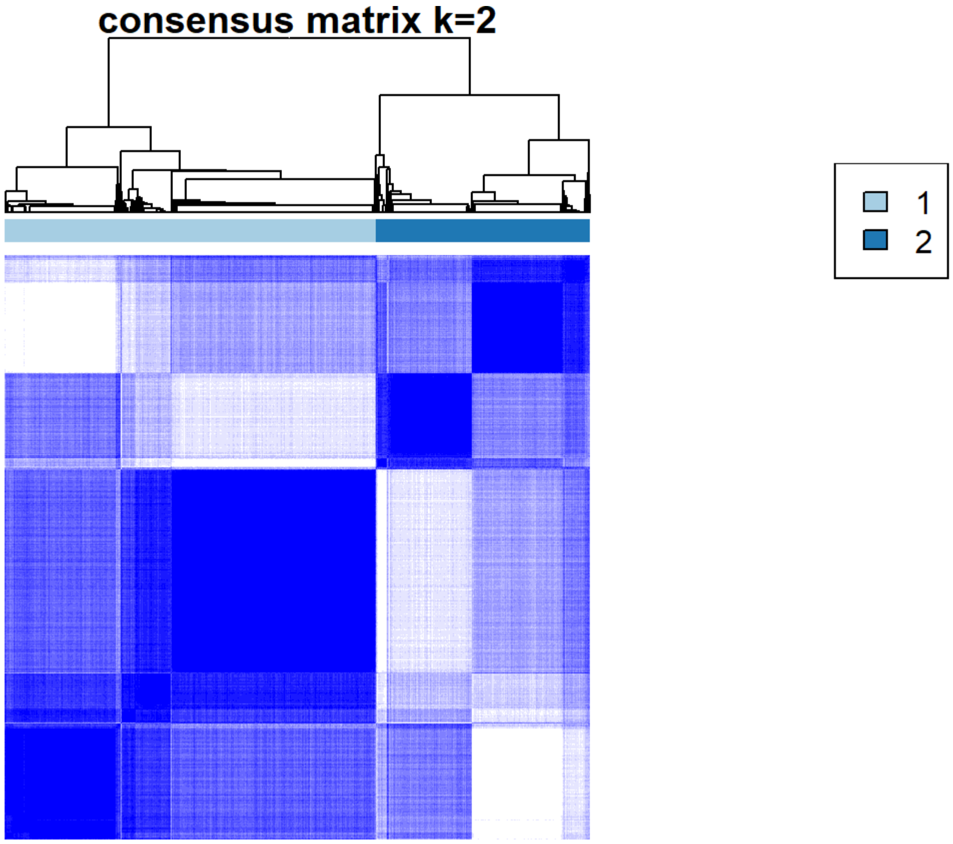
```
YMBOL
maxK <- 9
results <- ConsensusClusterPlus(GTrnaExpr_symbol,
              maxK=maxK,
              reps=50,
              pItem=0.8,
              pFeature=1,
              clusterAlg="km",
              distance="euclidean",
              seed=123456,
              )
```

```
## end fraction
```

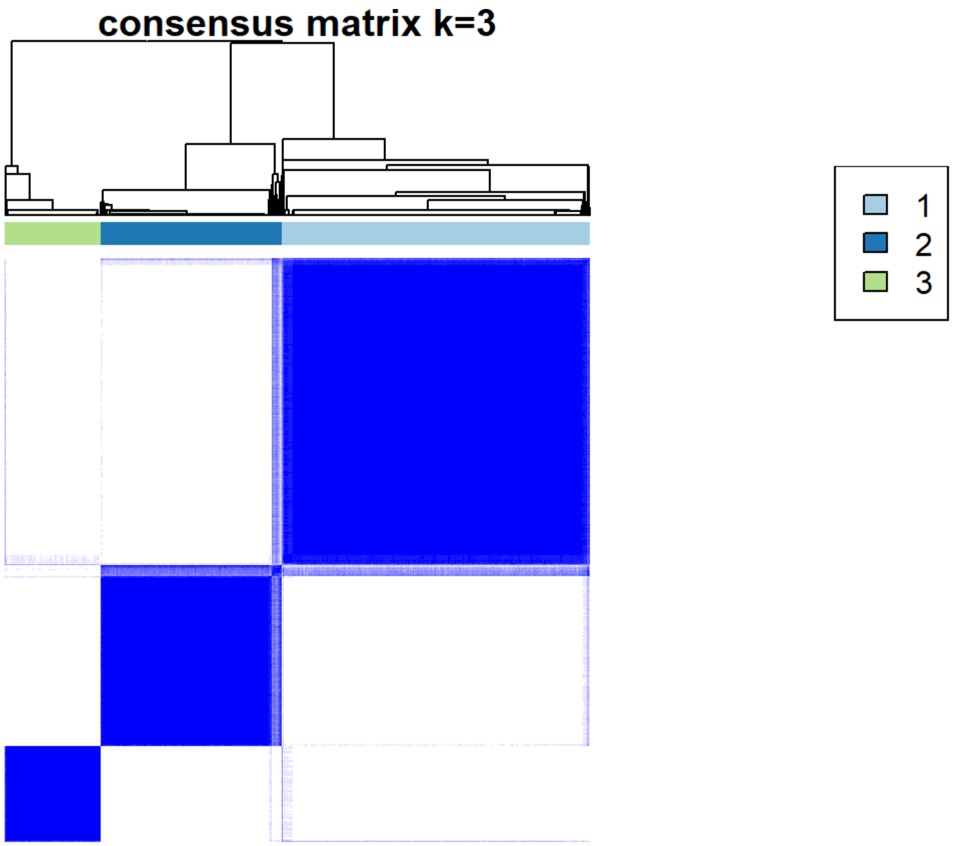
```
## clustered
```



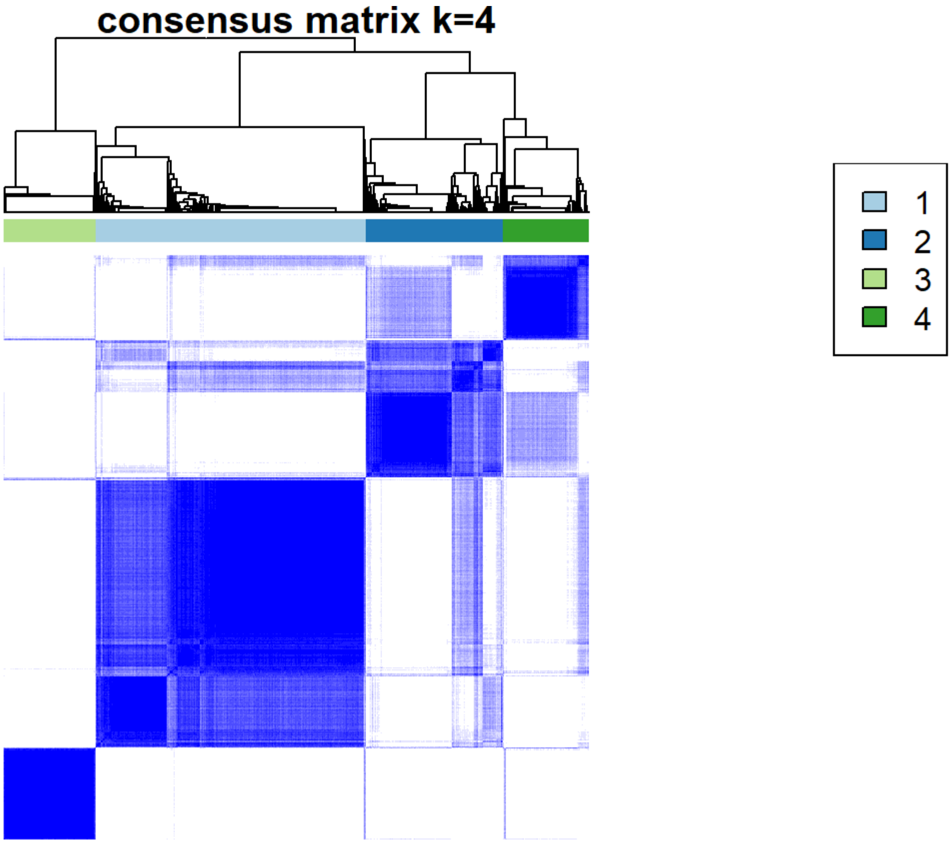
```
## clustered
```



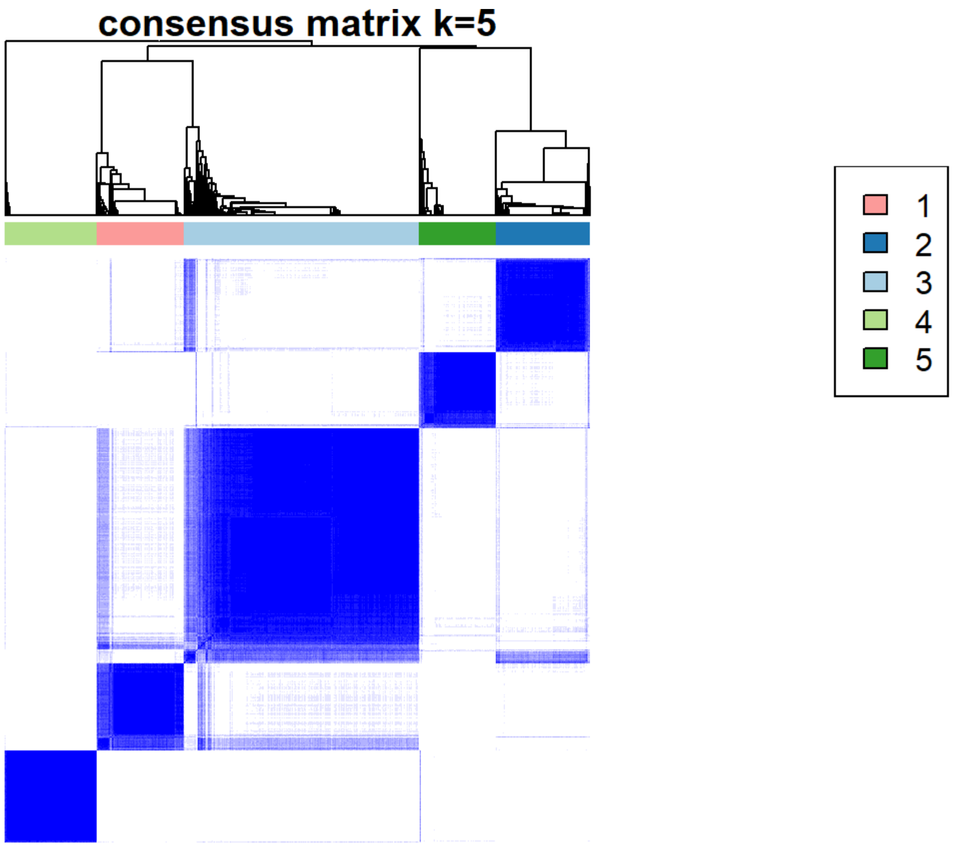
```
## clustered
```



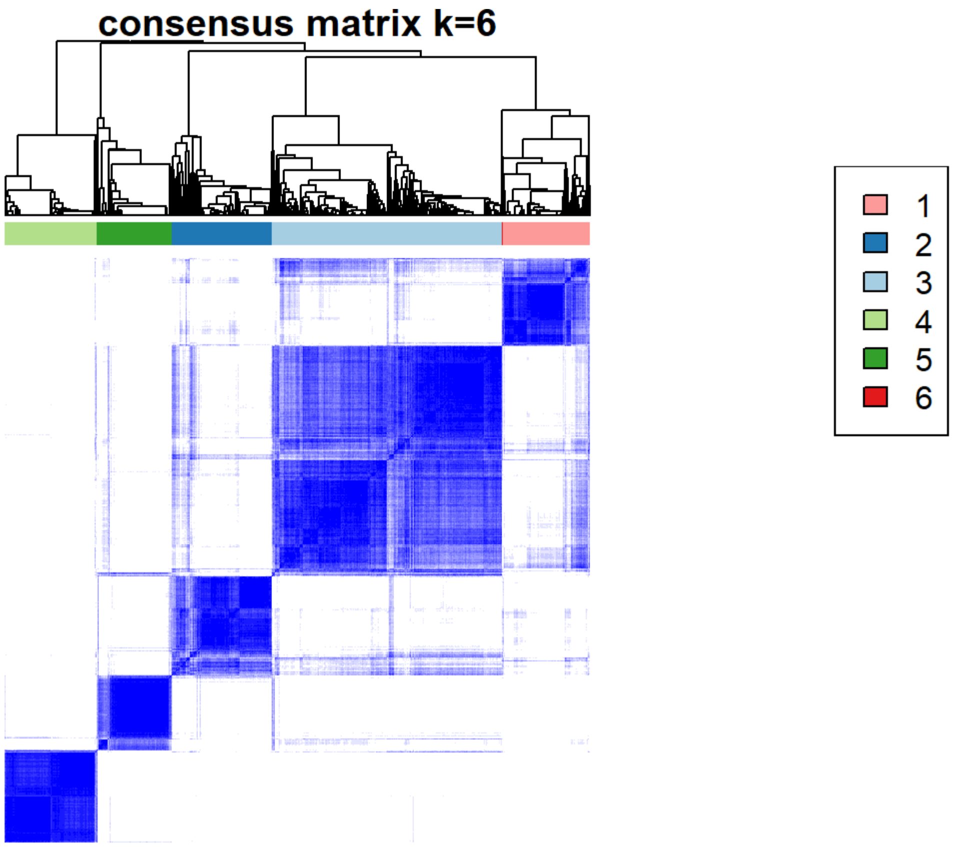
```
## clustered
```



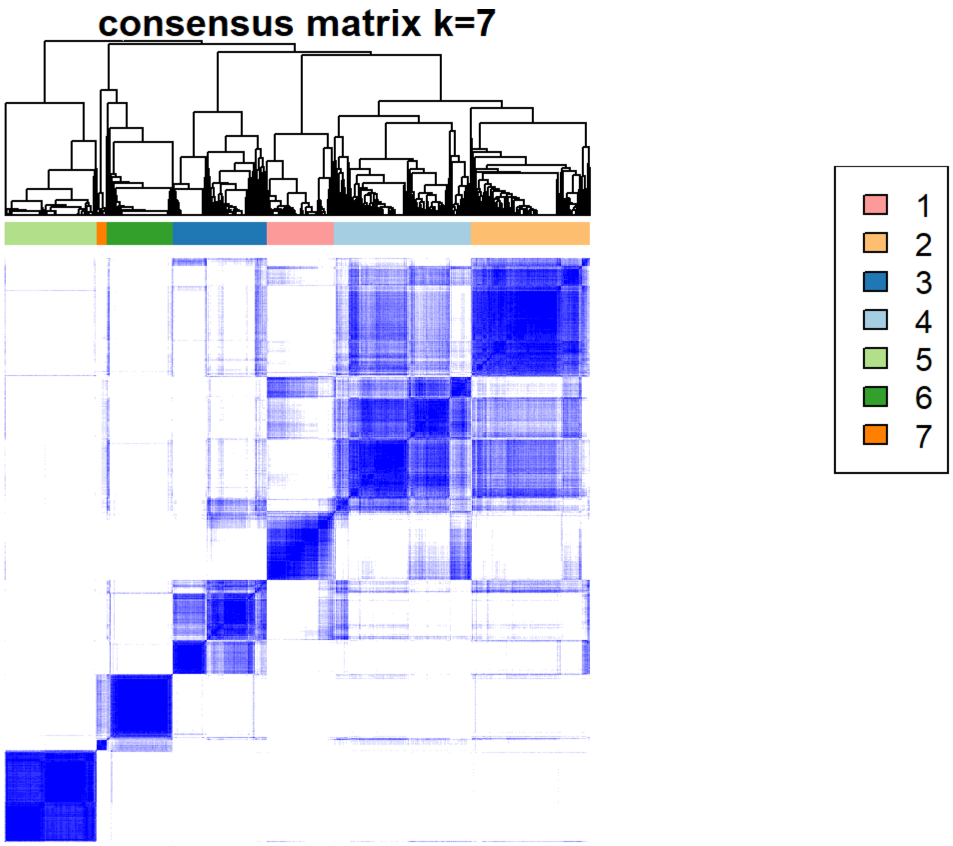
```
## clustered
```



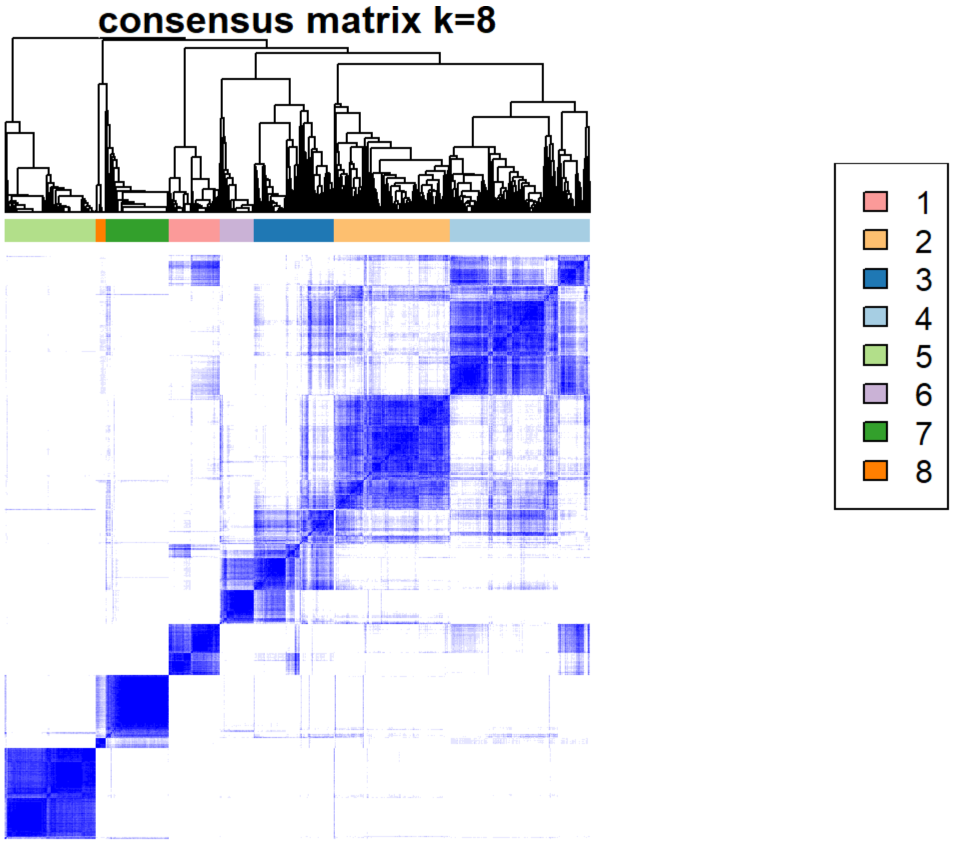
```
## clustered
```

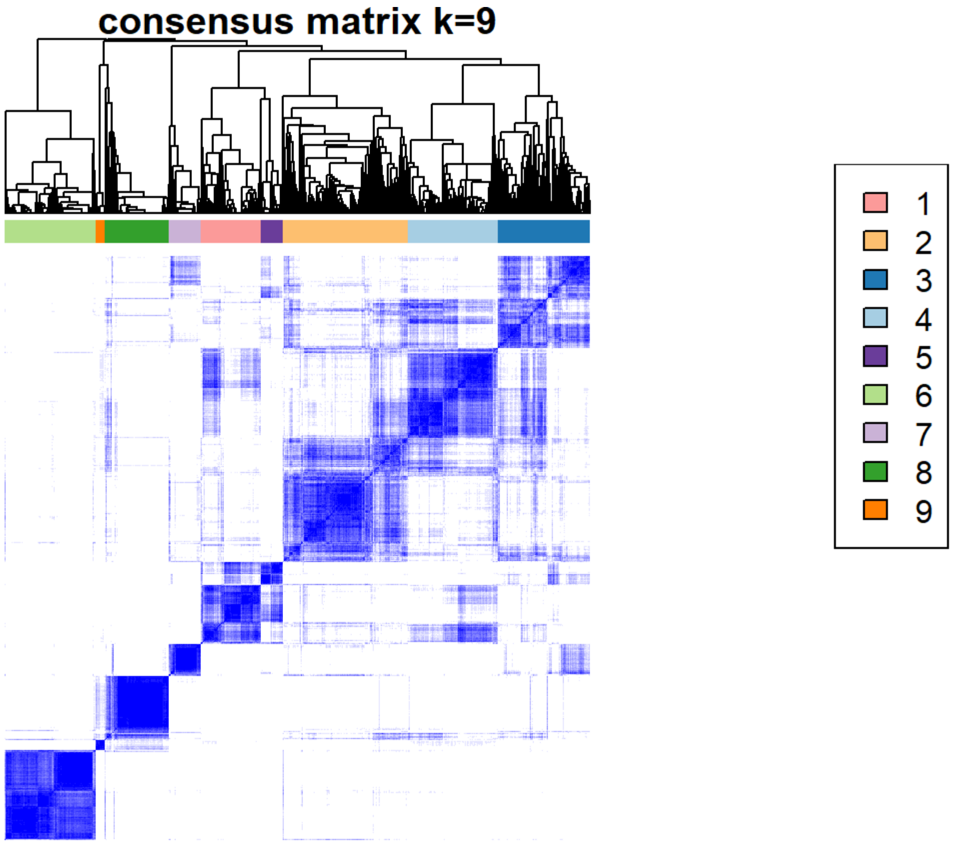


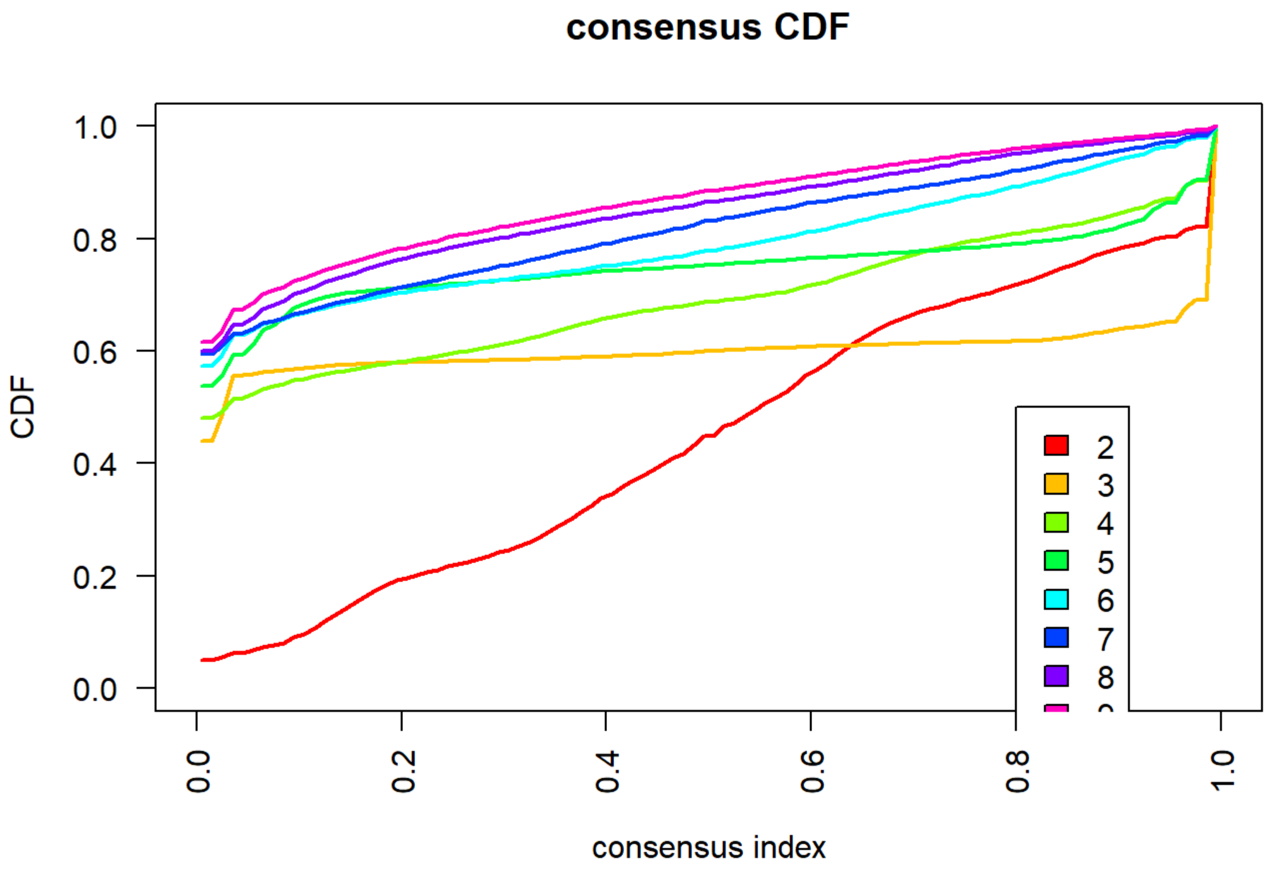
```
## clustered
```

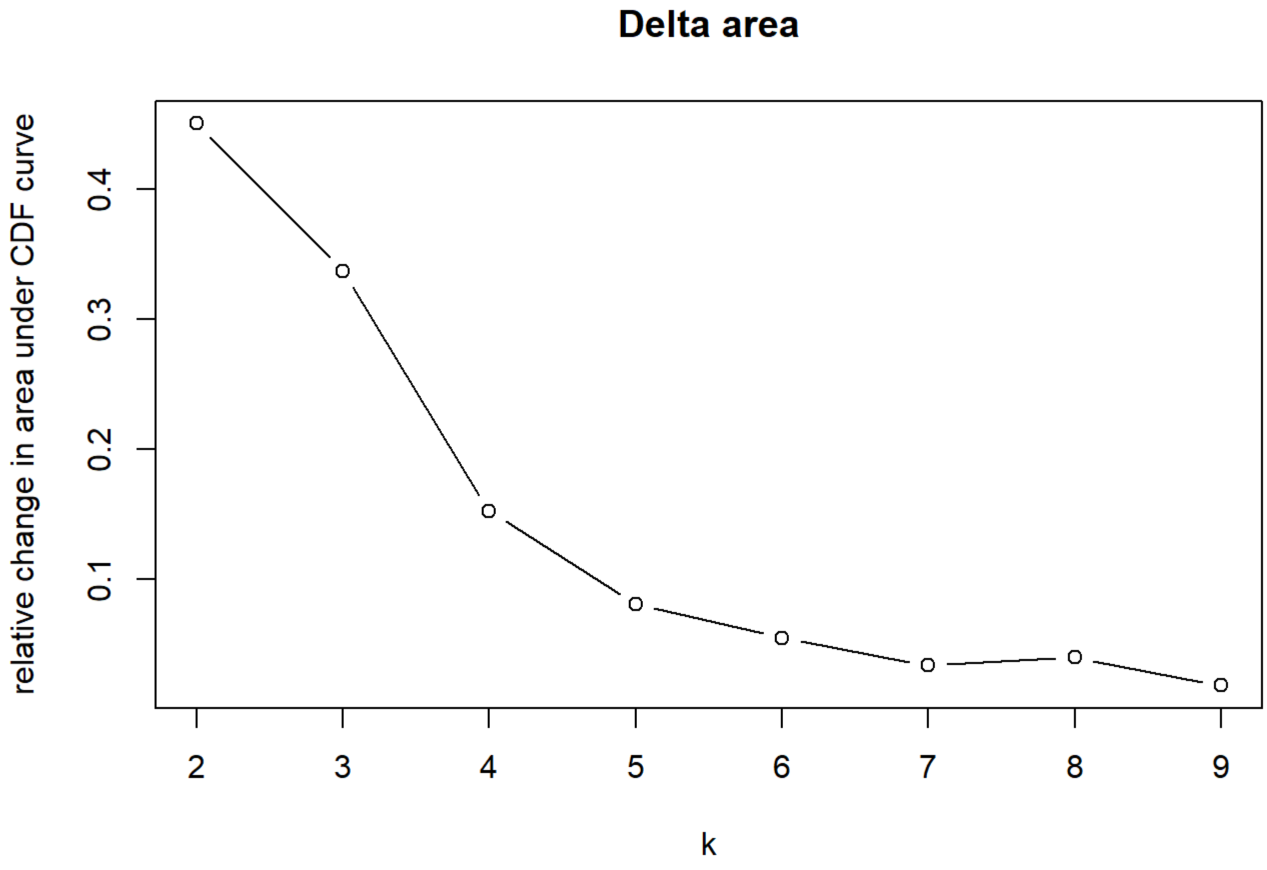


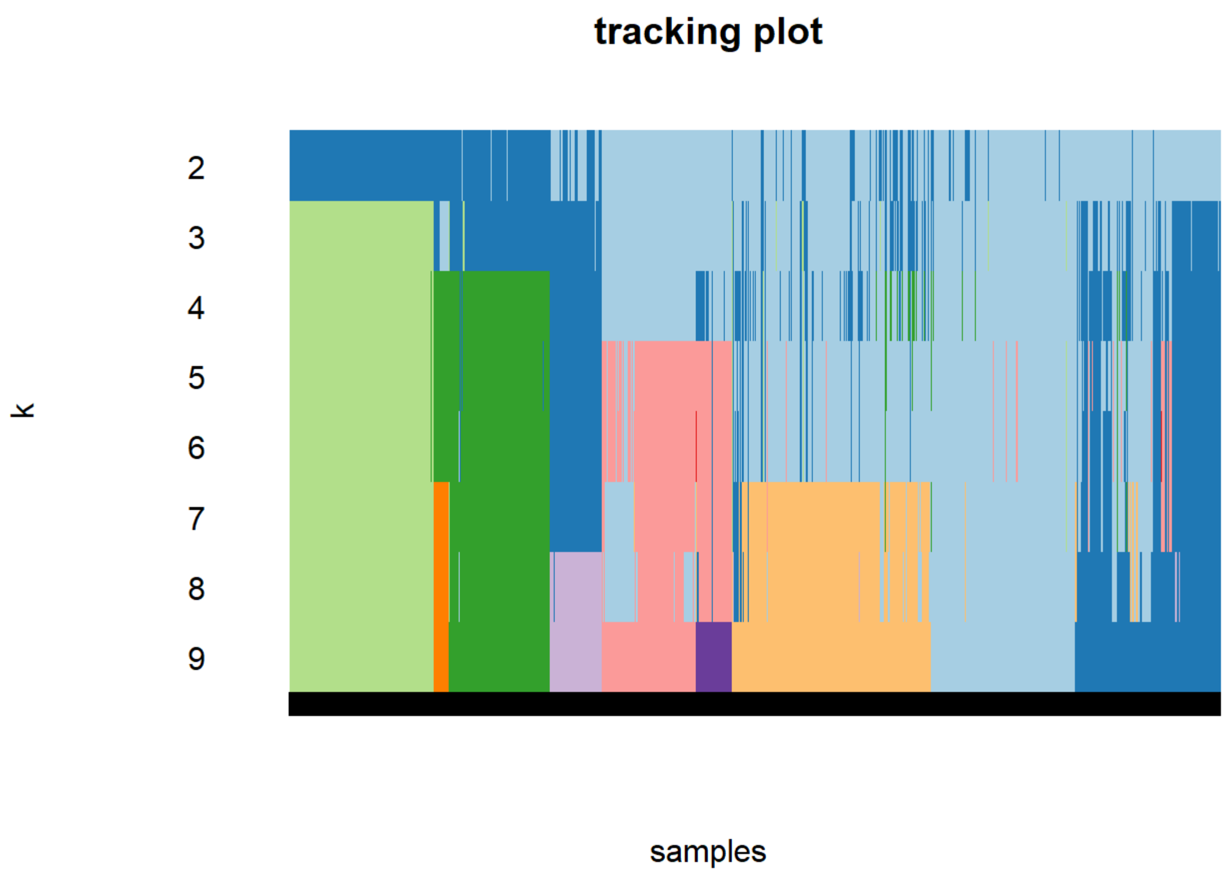
```
## clustered
```


consensus matrix k=8
1
2
3
4
5
6
7
8


consensus matrix k=9
1
2
3
4
5
6
7
8
9


consensus CDF
CDF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
consensus index
2
3
4
5
6
7
8

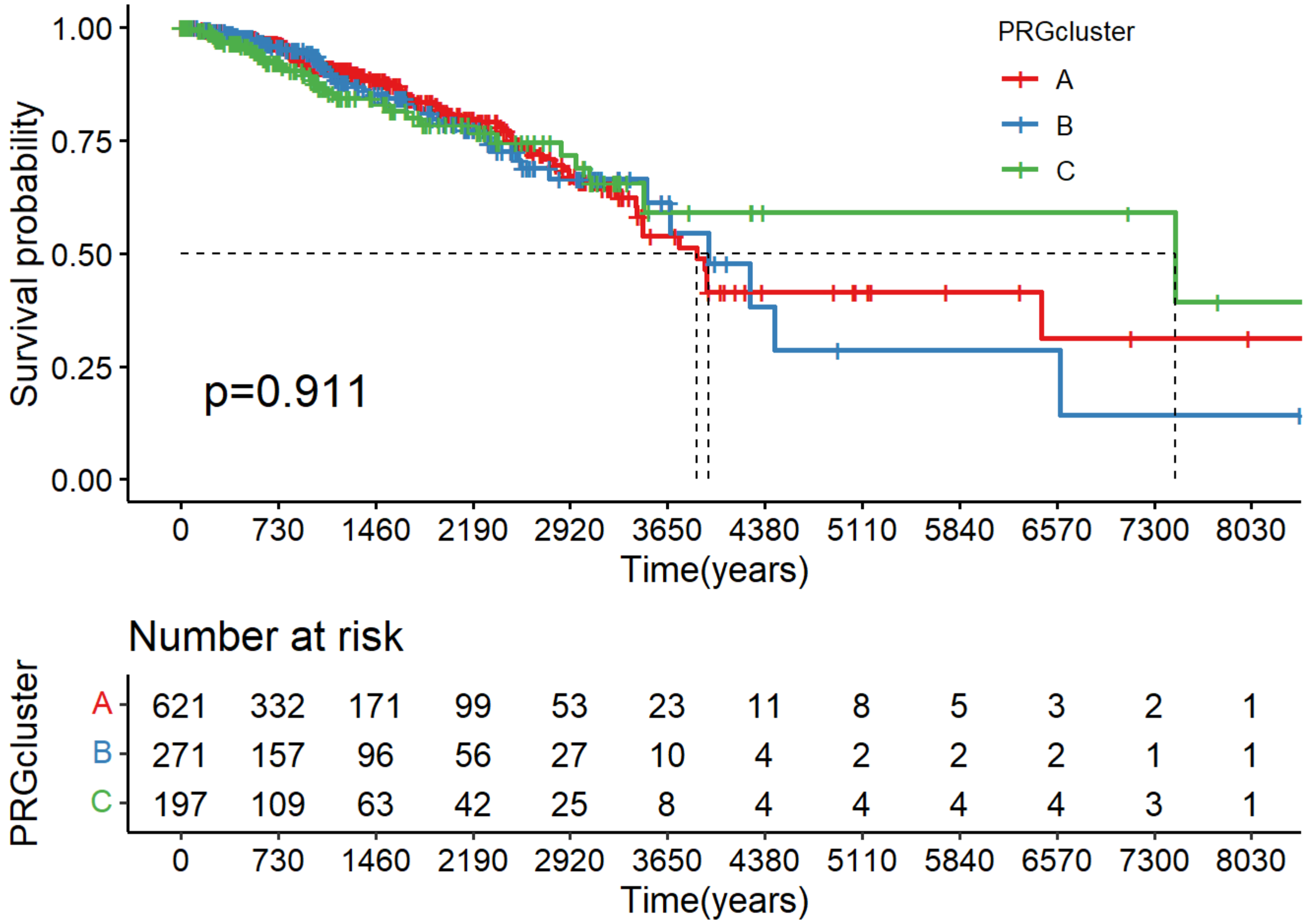

Delta area
relative change in area under CDF curve
0.4
0.3
0.2
0.1
2
3
4
5
6
7
8
9
k



```
#输出分型结果
clusterNum <- 3         #分成几个亚型
cluster <- results[[clusterNum]][["consensusClass"]]
cluster <- as.data.frame(cluster)
colnames(cluster) <- c("PRGcluster")
letter <- c("A","B","C","D","E","F","G")
uniqClu <- levels(factor(cluster$PRGcluster))
cluster$PRGcluster <- letter[match(cluster$PRGcluster, uniqClu)]
```

cluster_cli

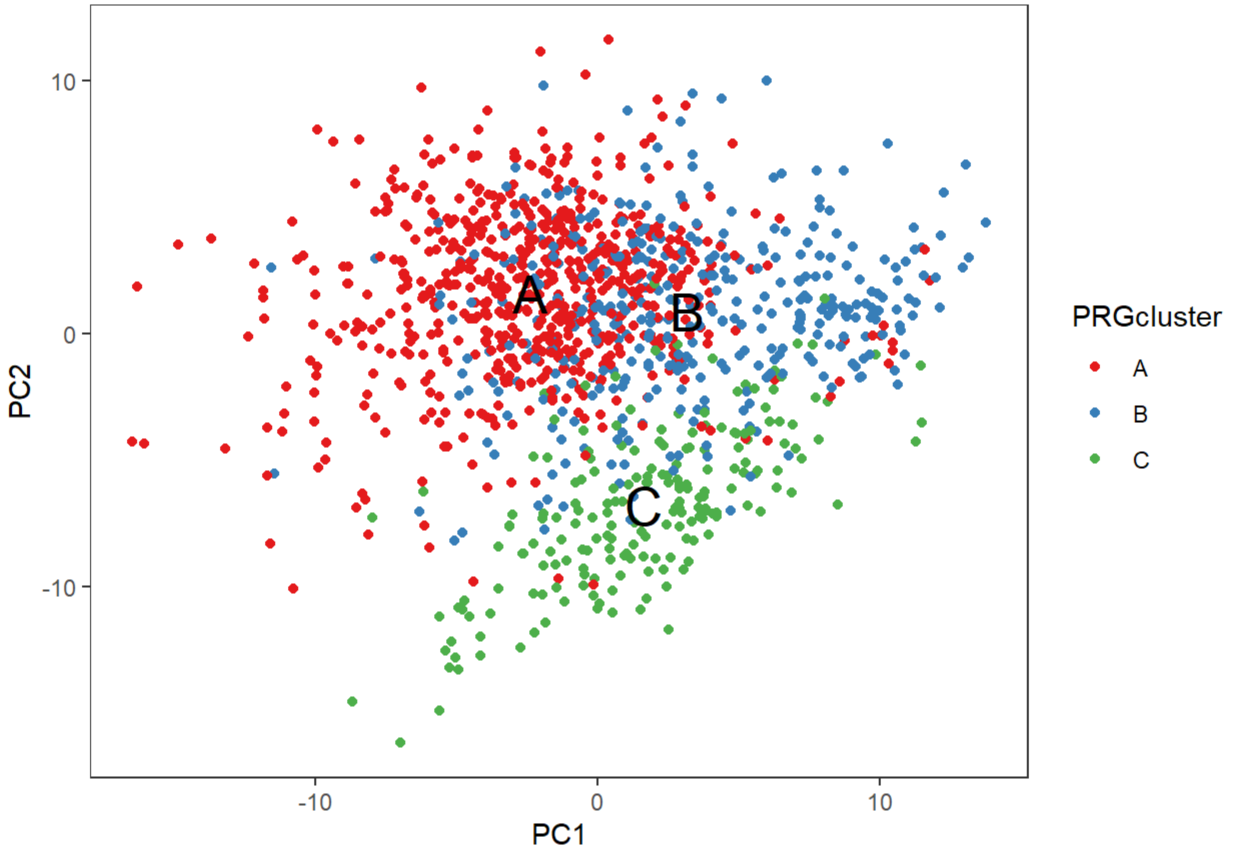
```
t <- intersect(rownames(exp_time), rownames(cluster))
rt <- cbind(exp_time[t, , drop=F], cluster[t, , drop=F])
length <- length(levels(factor(rt$PRGcluster)))
diff <- survdiff(Surv(OS.time, OS) ~ PRGcluster, data = rt)
pValue <- 1-pchisq(diff$chisq, df=length-1)
if(pValue<0.001){
    pValue <- "p < 0.001"
}else{
    pValue <- paste0("p=",sprintf("%.03f",pValue))
}
fit <- survfit(Surv(OS.time, OS) ~ PRGcluster, data = rt)
surPlot=ggsurvplot(fit,
                   data=rt,
                   conf.int=F,
                   pval=pValue,
                   pval.size=6,
                   legend.title="PRGcluster",
                   legend.labs=levels(factor(rt[,"PRGcluster"])),
                   legend = c(0.8, 0.8),
                   font.legend=10,
                   xlab="Time(years)",
                   break.time.by = 730,
                   palette = brewer.pal(length, "Set1"),
                   surv.median.line = "hv",
                   risk.table=T,
                   cumevents=F,
                   risk.table.height=.35)
print(surPlot)
```



	0	730	1460	2190	2920	3650	4380	5110	5840	6570	7300	8030
A	621	332	171	99	53	23	11	8	5	3	2	1
B	271	157	96	56	27	10	4	2	2	2	1	1
C	197	109	63	42	25	8	4	4	4	4	3	1

cluster pca

```
data <- GTrnaExpr_symbol
data <- t(data)
pca <- prcomp(data, scale. = T)
pcaPredict <- predict(pca)
PCA=data.frame(PC1=pcaPredict[,1], PC2=pcaPredict[,2], PRGcluster=as.vector(cluster[,1]))
PCA.mean=aggregate(PCA[,1:2], list(PRGcluster=PCA$PRGcluster), mean)
ggplot(data = PCA, aes(PC1, PC2)) + geom_point(aes(color = PRGcluster)) +
    scale_colour_manual(name="PRGcluster", values = brewer.pal(clusterNum, "Set1"))+
    theme_bw()+
    theme(plot.margin=unit(rep(1.5,4),'lines'))+
    annotate("text",x=PCA.mean$PC1, y=PCA.mean$PC2, label=PCA.mean$PRGcluster, cex=7)+
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
```

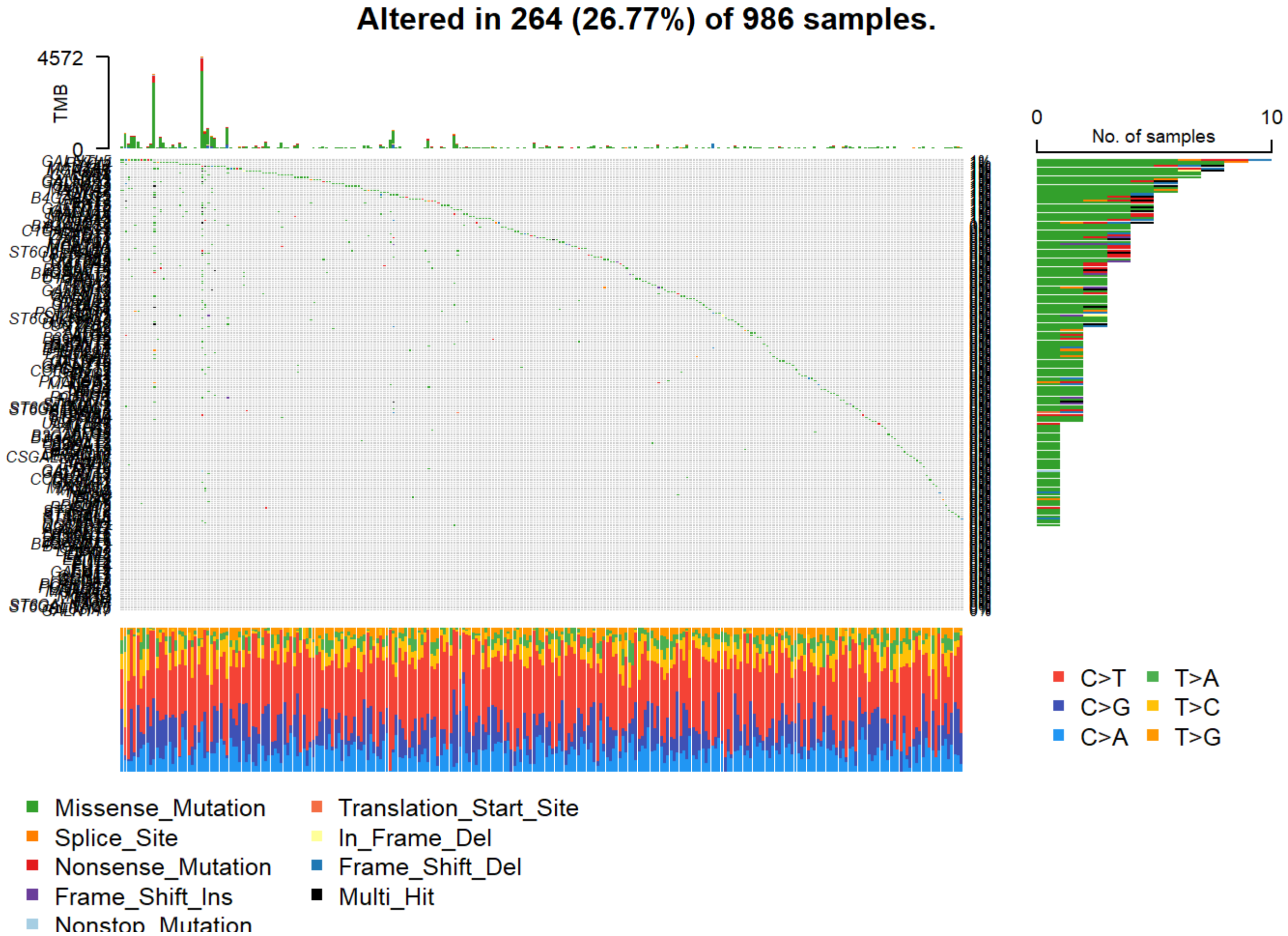


snp

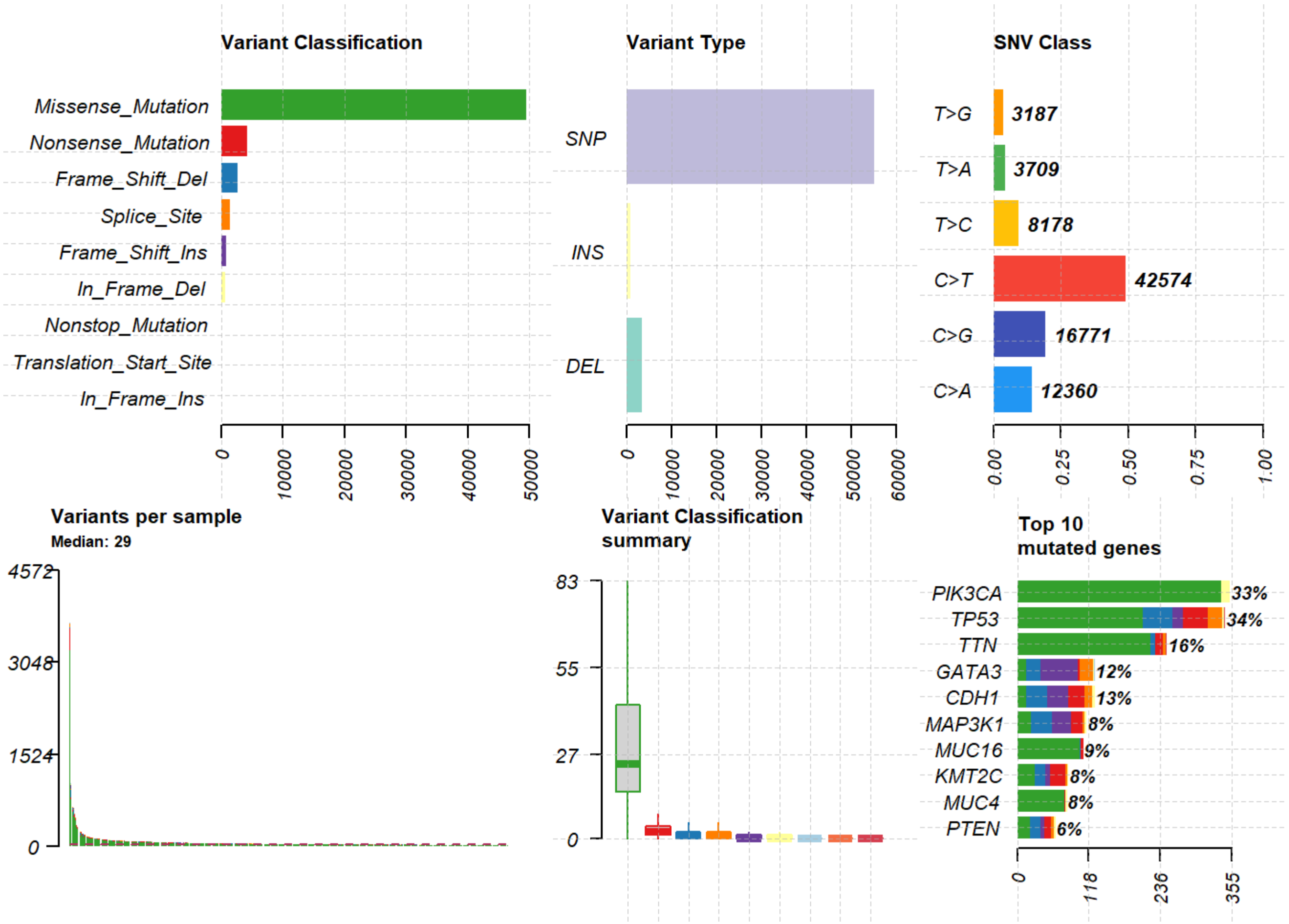
```
snp <- read.maf("raw_data/snp_varscan.maf")
```

```
## -Reading
## -Validating
## -Silent variants: 34035
## -Summarizing
## --Possible FLAGS among top ten genes:
##   TTN
##   MUC16
## -Processing clinical data
## --Missing clinical data
## -Finished in 8.550s elapsed (3.420s cpu)
```

```
GT <- unique(GT_ensembel$SYMBOL)
oncoplot(maf = snp, genes = GT, fontSize = 0.5, draw_titv = T)
```



```
plotmafSummary(maf = snp, rmOutlier = TRUE, addStat = 'median', dashboard = TRUE, titvRaw = FALSE)
```

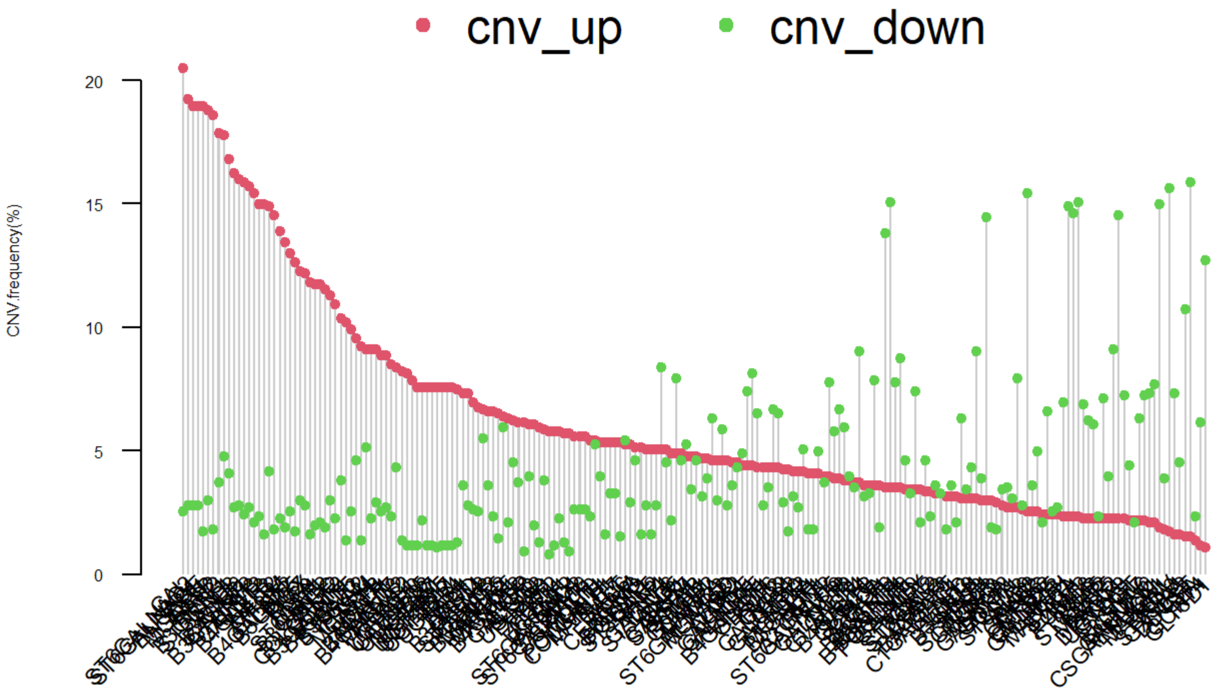


```
tmb <- snp@variants.per.sample
tmb$TMB <- tmb$Variants / 35 # total exons as 35MB
```

cnv

```
# cnv <- read.table("raw_data/TCGA-BRCA.gistic.tsv", sep = "\t", header = T)
# gtf <- rtracklayer::import("raw_data/gencode.v22.annotation.gtf")
# gtf_df <- as.data.frame(gtf)
# t <- gtf_df[gtf_df$gene_id %in% cnv$Gene.Symbol, ]
# names(t)[names(t) == "gene_id"] <- "Gene.Symbol"
# cnv <- merge(t[c("Gene.Symbol", "gene_name")], cnv, by="Gene.Symbol", all=T)
# gtf_df <- gtf_df[c("gene_name", "gene_id", "start", "end", "seqnames")]
# save(list=c("cnv", "gtf_df"), file = "cnv.Rdata")
load("cnv.Rdata")

cnv <- cnv[cnv$gene_name %in% GT, ]
cnv <- cnv[!duplicated(cnv$gene_name), ]
rownames(cnv) <- cnv$gene_name
cnv_up <- rowSums(cnv > 0)
cnv_down <- rowSums(cnv < 0)
cnv_up <- cnv_up / ncol(cnv) * 100
cnv_down <- cnv_down / ncol(cnv) * 100
data <- cbind(cnv_up, cnv_down)
data <- data[order(data[, "cnv_up"], decreasing = T), ]
```

```
data.max <- apply(data, 1, max)

# frequency plot
cex <- 0.5
par(cex.lab=cex, cex.axis=cex, font.axis=1, las=1, xpd=T)
bar=barplot(data.max, col="grey80", border=NA,
            xlab="", ylab="CNV.frequency(%)", space=1.5,
            xaxt="n", ylim=c(0,1.2*max(data.max)))
points(bar,data[,"cnv_up"], pch=20, col=2, cex=1)
points(bar,data[,"cnv_down"], pch=20, col=3, cex=1)
legend("top", legend=c('cnv_up','cnv_down'), col=2:3, pch=20, bty="n", cex=1.5, ncol=2)
par(srt=45)
text(bar, par('usr')[3]-0.2, rownames(data), adj=1, cex=0.7)
```



```
# circle plot
gtf_df <- gtf_df[gtf_df$gene_name %in% rownames(cnv), ]
gtf_df <- gtf_df[!duplicated(gtf_df$gene_name), ]

cytoBandIdeogram=read.table("raw_data/refer.txt", header=T, sep="\t")
chr.exclude <- NULL
cyto.info <- cytoBandIdeogram
tracks.inside <- 5
tracks.outside <- 0

RCircos.Set.Core.Components(cyto.info, chr.exclude, tracks.inside, tracks.outside)
```

```
##
## RCircos.Core.Components initialized.
## Type ?RCircos.Reset.Plot.Parameters to see how to modify the core components.
```

```
rcircos.params <- RCircos.Get.Plot.Parameters()
rcircos.params$text.size=0.8
rcircos.params$point.size=5
RCircos.Reset.Plot.Parameters(rcircos.params)
RCircos.Set.Plot.Area()
RCircos.Chromosome.Ideogram.Plot()

t <- data[,1] - data[,2]
t <- as.data.frame(t)
t[t>0, ] <- 1
t[t<0, ] <- -1
t$gene_name <- rownames(t)
t1 <- gtf_df[c("seqnames", "start", "end", "gene_name")]
t <- merge(t1, t, all=T)

RCircos.Scatter.Plot(scatter.data = t[2:5], track.num = 1, data.col = 4, "in", by.fold=0.1)

RCircos.Gene.Connector.Plot(t[c(2:4, 1)], track.num = 2, "in")
```

```
## Not all labels will be plotted.
```

```
## Type RCircos.Get.Gene.Name.Plot.Parameters()
```

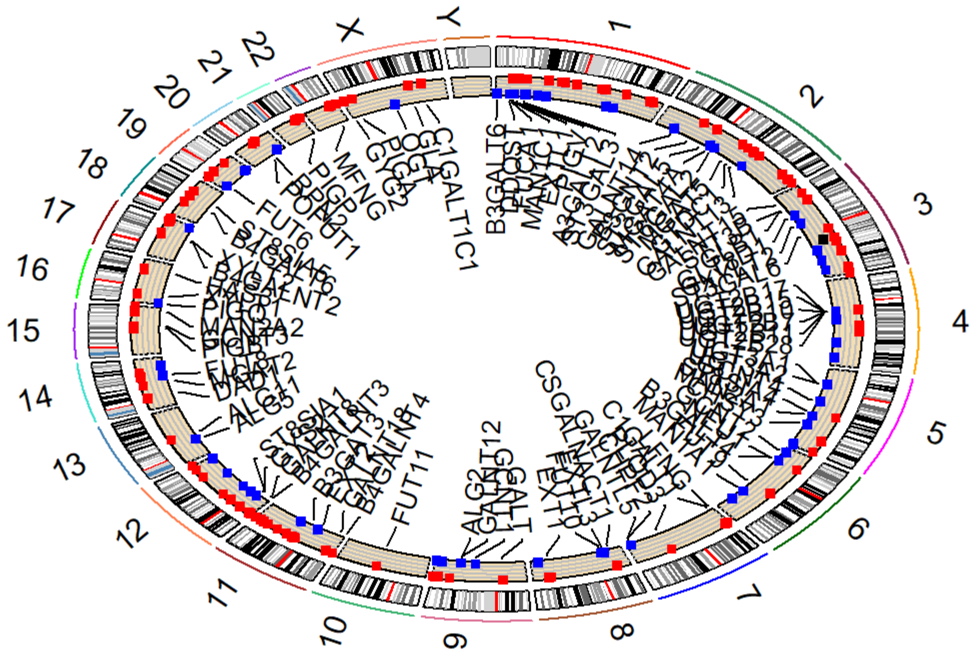
```
## to see the number of labels for each chromosome.
```

```
RCircos.Gene.Name.Plot(t[c(2:4, 1)], name.col = 4, track.num = 3, "in")
```

```
## Not all labels will be plotted.
```

```
## Type RCircos.Get.Gene.Name.Plot.Parameters()
```

```
## to see the number of labels for each chromosome.
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
X
Y
B3GALT6
DDOST
FUCA1
MAN1C1
ST3GAL3
C1GALT1C1
GLA
OGT
PIGA
GYG2
MFNG
PIGP
RPN2
POFUT1
FUT6
ST8SIA5
B4GALT6
MAN2A2
PIGB
DAD1
ALG5
ALG11
ST8SIA1
B4GALNT3
B3GAT3
EXT2
B4GALNT4
FUT11
ALG2
GALNT12
CSGALNACT1
EXT1
FUT10
C1GALT1
B3GAT1
MAN1A1