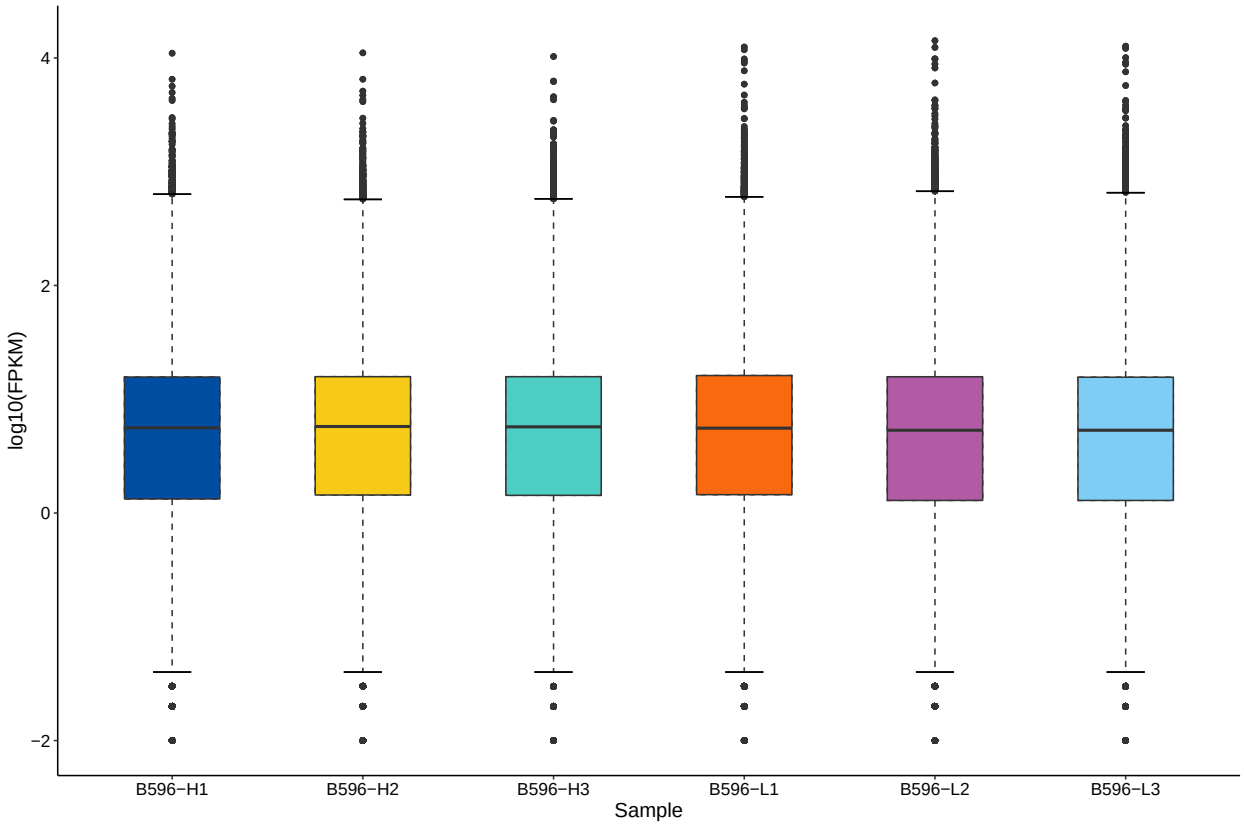
DEG analysis

Changfu Jia

2022-09-10

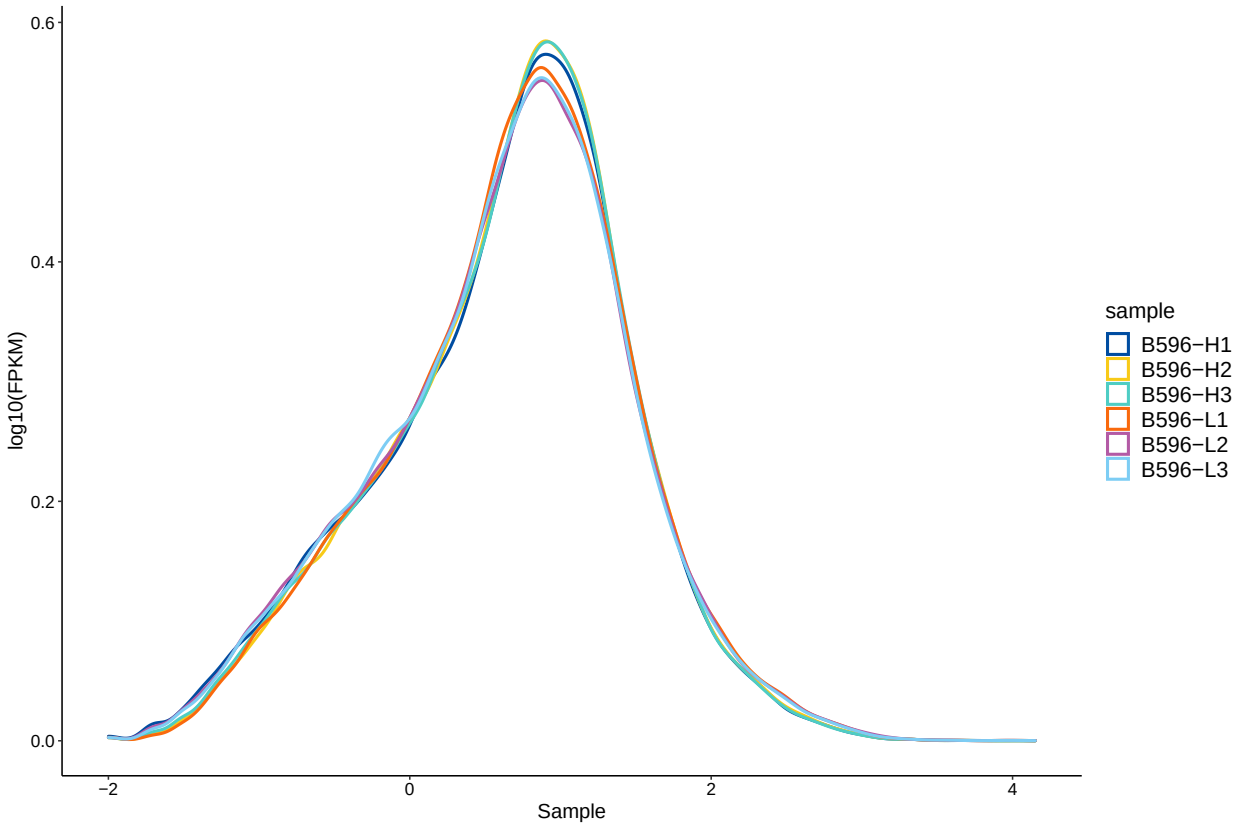
```
setwd("F:/dir/")
options(warn=-1)

library(tidyverse)
library(pheatmap)
#install.packages("FactoMineR")
library(FactoMineR)
#library(devtools)
#install_github("vqv/ggbiplot")
library(ggbiplot)
library(DESeq2)

expression<-list()
for (i in  grep("isoforms.re",dir("F:/dir/"),value = T ) ) {
  j<-read.table(paste("F:/dir/",i, sep="/"), sep = "\t", header = T )
  g<-sub("_fastp.isoforms.results","",i)
  expression[[g]]<-j

}



FPKM<-do.call( cbind,  lapply(names(expression), function(x){ expression[[x]]$FPKM   }   ) )

count<-do.call( cbind,  lapply(names(expression), function(x){ expression[[x]]$expected_count   }   ) )

TPM<-do.call( cbind,  lapply(names(expression), function(x){ expression[[x]]$TPM   }   ) )


rownames(count)<-expression[[1]][,1]
colnames(count)<-gsub( "_L4.*", "", names(expression))

rownames(FPKM)<-expression[[1]][,1]
colnames(FPKM)<-gsub( "_L4.*", "", names(expression))



logFPKM<-log10(FPKM)

as.data.frame(logFPKM) %>%
  gather(key="sample", value = "log2Fpkm") %>%
  ggplot(aes(x=sample, y=log2Fpkm, fill=sample)) +
```

```
  geom_boxplot(linetype="dashed",width=0.5) +
  stat_boxplot(aes(ymin=..lower..,ymax=..upper..),width=0.5) +
  #stat_boxplot(geom = "errorbar",aes(ymin=..ymax) ,width=0.3)
  stat_boxplot(geom = "errorbar",aes(ymin=..ymax..),
               width=0.2)+
  stat_boxplot(geom = "errorbar",aes(ymax=..ymin..),
               width=0.2) +
  theme_classic()  +
  theme(legend.position ="none",
        axis.title.x=element_text(size=14,color="black",hjust=0.5),

        axis.title.y=element_text(size=14,color="black"),
        axis.text.x=element_text(size=12,color="black"),
        axis.text.y=element_text(size=12,color="black"),
        legend.text=element_text(size=15,color="black"),
        legend.title=element_text(size=15,color="black"),
        plot.title=element_text(size=15,color="black",hjust=0.5)) +
  ylab("log10(FPKM)")+
  xlab("Sample") +
  scale_fill_manual(values=c( "#004DA1", "#F8CA18", "#4ECDC4" , "#F9690D", "#B35AA5", "#7DCDF4"  ) )
```

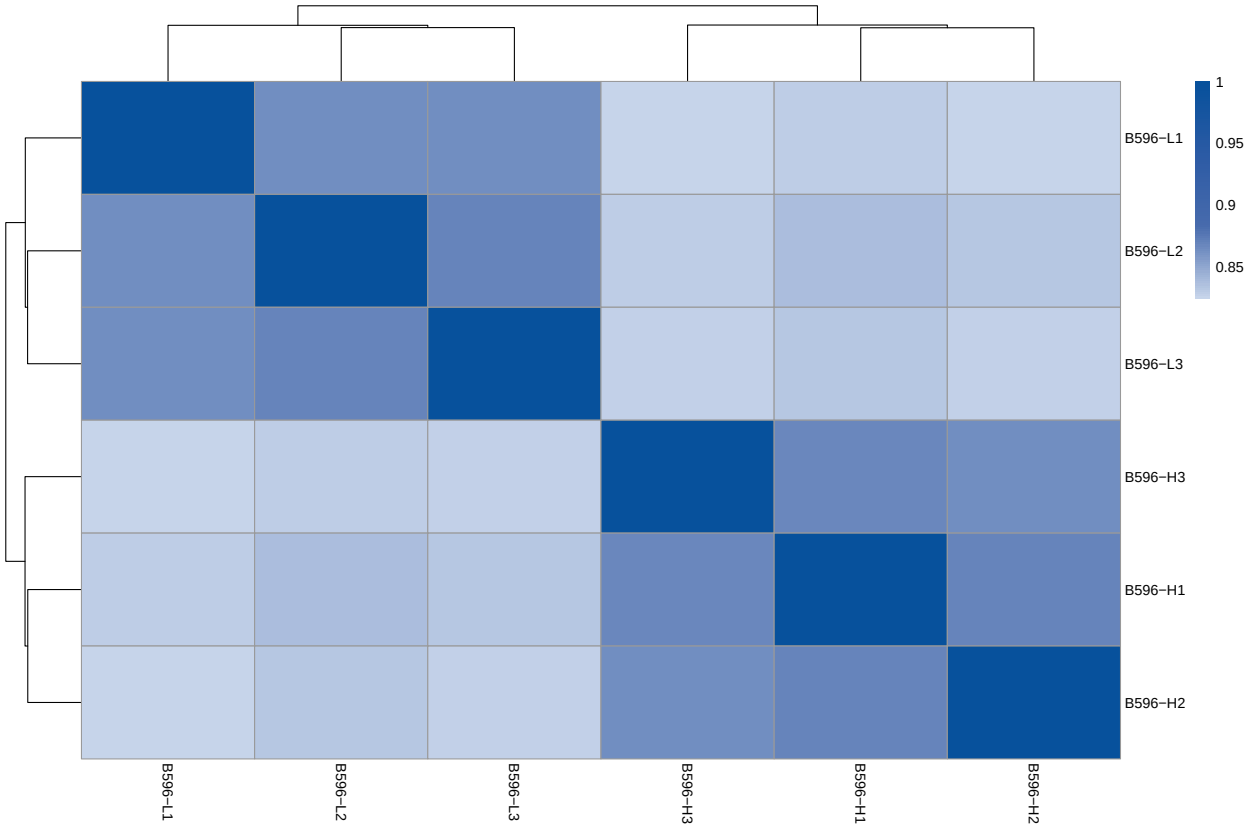


Including Plots

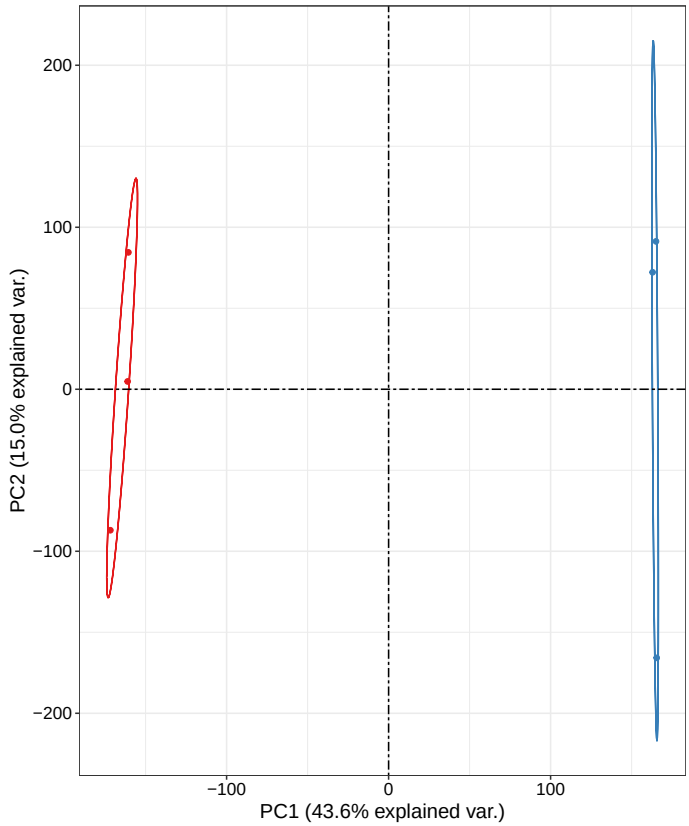
You can also embed plots, for example:




```
logFPKM<-log10(FPKM+0.01)


logFPKM<-
  as.data.frame(logFPKM) %>%
  filter_all(any_vars(.>-2))

as.matrix( cor(logFPKM) ) %>%
  #pheatmap( color  = colorRampPalette(c("#327EB8","#EB5B24","#F7EB18", "#E41F26"))(100) )
  pheatmap( color  = colorRampPalette(c("#C6D4EA","#486BAD","#2C5BA5", "#07519C"))(100) )
```

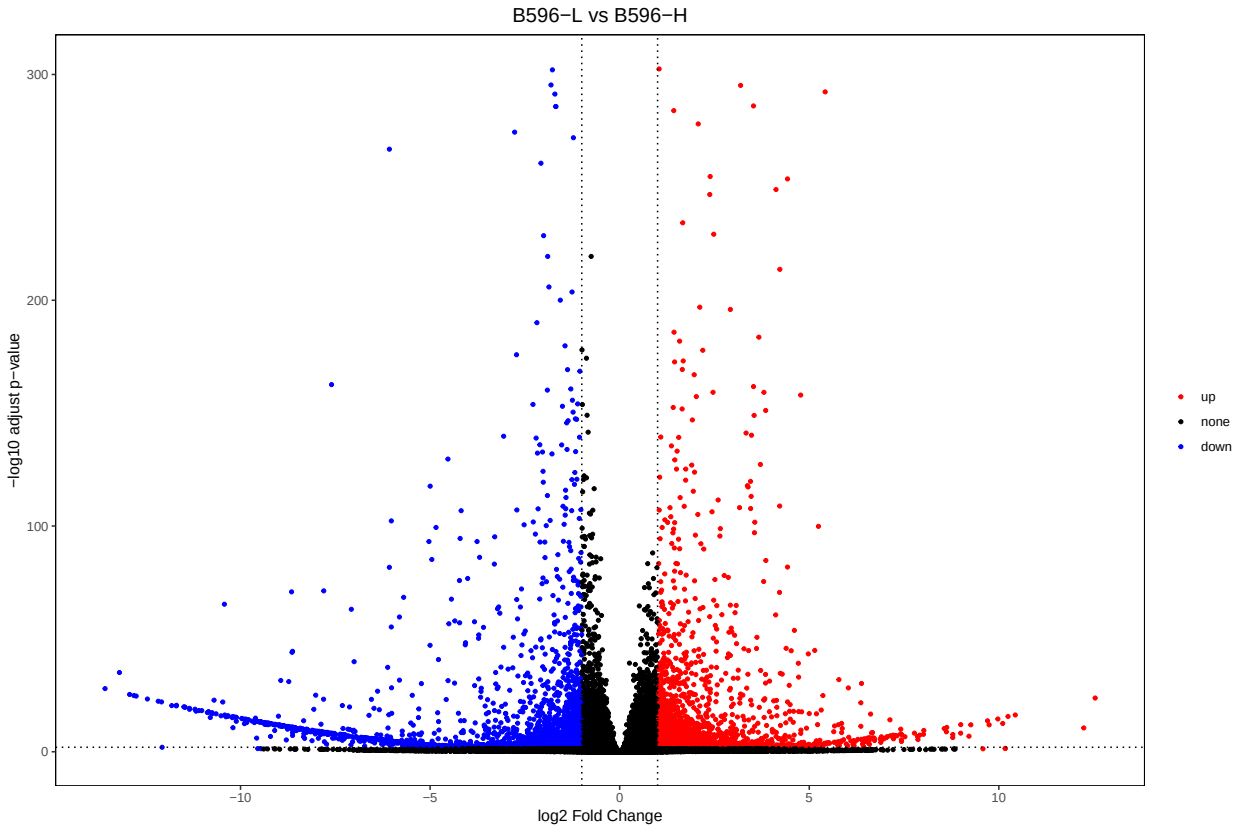



```
data.pca <- prcomp(t(logFPKM), scale. = T)
pca.sum=summary(data.pca)


group=  c( rep("B596-H",3), rep("B596-L", 3)  )
ggbiplot(data.pca,obs.scale = 1,var.scale = 1,
         groups = group,ellipse = T,var.axes = F, choices = c(1,2)) +
  scale_color_brewer(palette = "Set1")  +
  theme(legend.direction = 'horizontal',legend.position = 'top') +
  geom_vline(xintercept = c(0), linetype = 6,color = "black") +
  geom_hline(yintercept = c(0),linetype = 6,color = "black") +
  theme_bw() +
  theme(legend.position ="none",
    axis.title.x=element_text(size=14,color="black",hjust=0.5),

    axis.title.y=element_text(size=14,color="black"),
    axis.text.x=element_text(size=12,color="black"),
    axis.text.y=element_text(size=12,color="black"),
    legend.text=element_text(size=15,color="black"),
    legend.title=element_text(size=15,color="black"),
    plot.title=element_text(size=15,color="black",hjust=0.5))
```



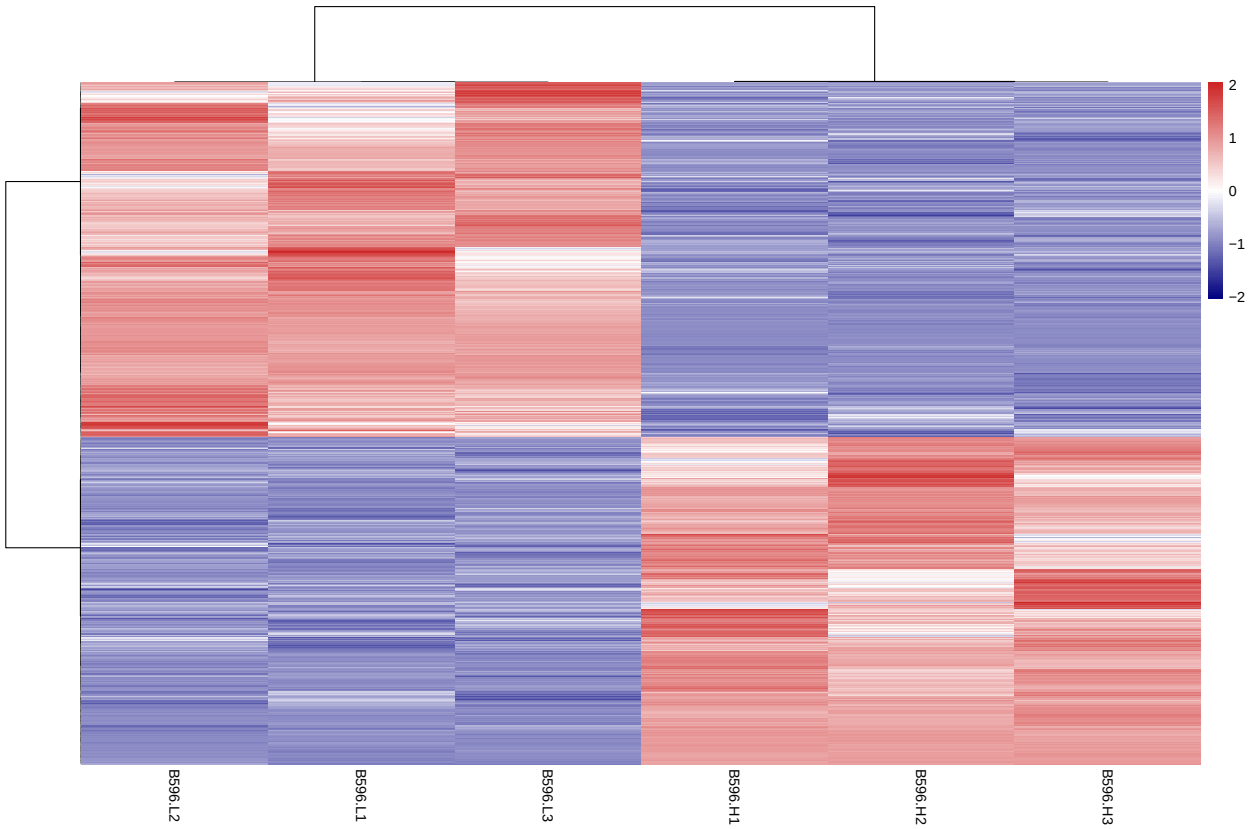

```
options(warn=-1)



#colnames(logFPKM)
coldata <- data.frame(condition = factor(rep(c('B596-H', 'B596-L'), each = 3), levels = c('B596-H', 'B5

count<-read.csv(file = "counts.csv" ,header = T)

rownames(count) <- count$X
count<-count[,-1]

#logFPKM
count<-
as.data.frame(count) %>%
  filter_all(any_vars(.>0)) %>%
  round(.)

dds <- DESeqDataSetFromMatrix(countData =count , colData = coldata, design= ~condition)
dds1 <- DESeq(dds, fitType = 'mean', minReplicatesForReplace = 7, parallel = FALSE)

#results(dds1)

res <- results(dds1, contrast = c('condition', 'B596-H', 'B596-L'))

res1 <- data.frame(res, stringsAsFactors = FALSE, check.names = FALSE) %>% na.omit()
```

```r
#write.table(res1, 'B596-L_B596-H.DESeq2.txt', col.names = NA, sep = '\t', quote = FALSE)

res1 <- res1[order(res1$padj, res1$log2FoldChange, decreasing = c(FALSE, TRUE)), ]

res1[which(res1$log2FoldChange >= 1 & res1$padj < 0.05),'sig'] <- 'up'
res1[which(res1$log2FoldChange <= -1 & res1$padj < 0.05),'sig'] <- 'down'
res1[which(abs(res1$log2FoldChange) <= 1 | res1$padj >= 0.05),'sig'] <- 'none'

res1_select <- subset(res1, sig %in% c('up', 'down'))

#write.csv(res1_select, file = 'B596-L_B596-H.DESEQ2.selected.csv')



res1_up <- subset(res1, sig == 'up')
res1_down <- subset(res1, sig == 'down')


ggplot(data = res1%>%filter(padj!=0) , aes(x = log2FoldChange, y = -log10(padj), color = sig)) +
  geom_point(size = 1) +
  scale_color_manual(values = c("#FF0000", "black", "#0000FF" ), limits = c('up', 'none', 'down')) +
  labs(x = 'log2 Fold Change', y = '-log10 adjust p-value', title = 'B596-L vs B596-H', color = '') +
  theme(plot.title = element_text(hjust = 0.5, size = 14), panel.grid = element_blank(),
        panel.background = element_rect(color = 'black', fill = 'transparent'),
        legend.key = element_rect(fill = 'transparent')) +
  geom_vline(xintercept = c(-1, 1), lty = 3, color = 'black') +
  geom_hline(yintercept = 2, lty = 3, color = 'black')
```




```
FPKM<- read.csv("FPKM.csv")
rownames(FPKM) <- FPKM$X
FPKM<-FPKM[,-1]



FPKM_deg<-FPKM[which(rownames(FPKM)%in%rownames(res1_select)),]


dist.obs<-as.dist(1-cor(t(FPKM_deg)))
dist.obs.tre<- hclust(dist.obs, method = "ward.D")

#dist.obs.tre<- readRDS("tre.RDS")

dist.obs.tis<-as.dist(1-cor(FPKM_deg))
dist.obs.tis.tre<- hclust(dist.obs.tis, method = "ward.D")


pheatmap(mat=as.matrix(FPKM_deg),
         scale = "row",
         show_rownames = FALSE,
         #   color = colorRampPalette(c("blue","white","red"))(10),
         color  = colorRampPalette(c("navy", "white", "firebrick3"))(100),
         #treeheight_row = 200,
         #treeheight_col = 100,
         cluster_rows = dist.obs.tre,
         cluster_cols = dist.obs.tis.tre,
```

```
        #annotation_row=annotation_row,
        #annotation_legend = FALSE,
        #annotation_names_row = F,
        #annotation_colors = ann_colors
)
```



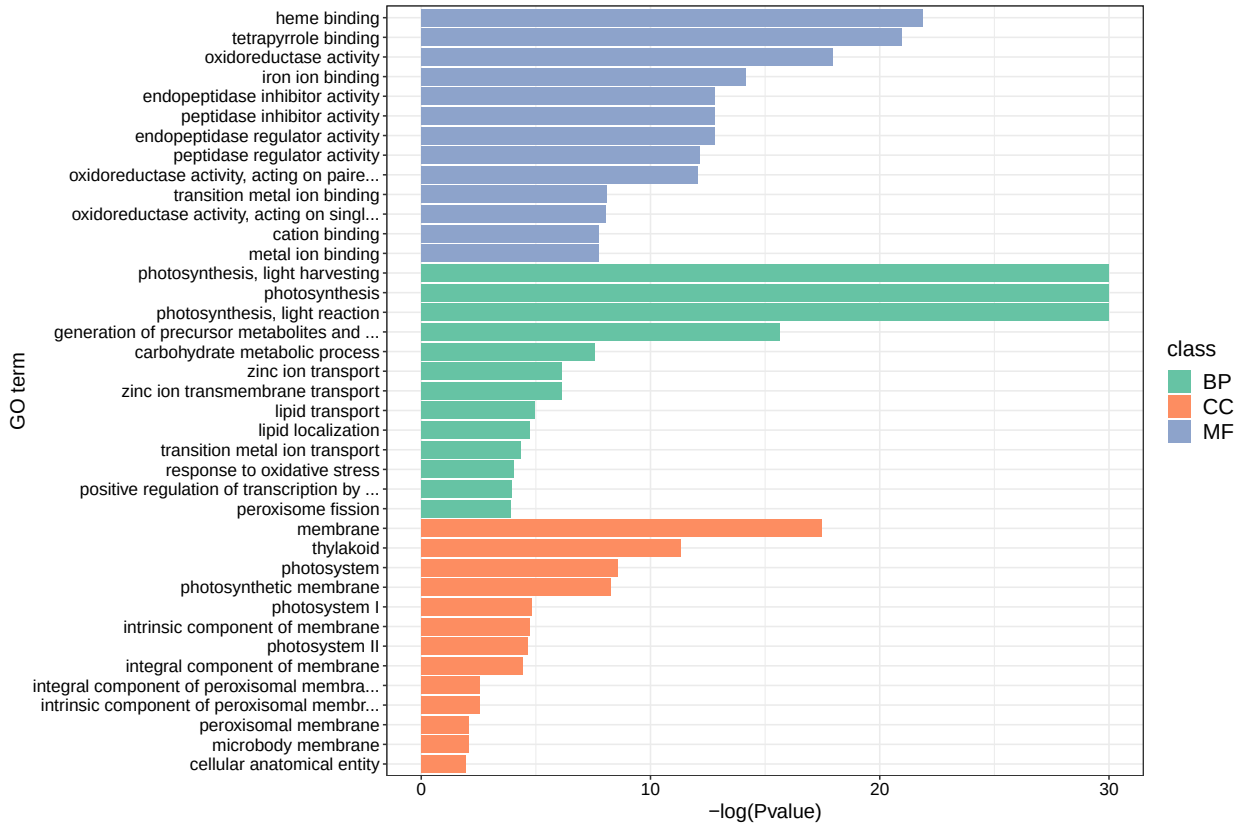
#GO and KEGG enrichment

```
library(clusterProfiler)


L596_go<-read.table("F:/dir/2.Structral_analysis/function_annonation/5.GO_KEGG/m64032_191231_031514.subi
H596_go<-read.table("F:/dir/2.Structral_analysis/function_annonation/5.GO_KEGG/m64032_191231_031514.subi


res1<-read.csv( 'B596-L_B596-H.DESEQ2.result.csv', sep = ',', header = T)




L596<-read.table("../m64032_191231_031514.subreads.6--6_75.ccs.lima.refine.cluster.hq..corrected.fastq.
H596<-read.table("../m64032_191231_031514.subreads.7--7_75.ccs.lima.refine.cluster.hq..corrected.fastq.

L596$V3<-gsub("G", "T", L596$V3)
H596$V3<-gsub("G", "T", H596$V3)
L596$V3<-gsub(";", "", L596$V3)
H596$V3<-gsub(";", "", H596$V3)
```

```
#res1_select <- subset(res1, sig %in% c('up', 'down'))


res1_select<-read.csv('B596-L_B596-H.DESEQ2.selected.csv')
res1_select<- res1_select[,-1]

#colnames(res1_select) <- c("X"  ,  "baseMean" ,"log2FoldChange" ,"lfcSE"  ,"stat" , "pvalue" , "padj"


res1_select<-
  res1_select %>%
  mutate(L596_nov =ifelse(is.na(class), NA, ifelse(class=="L596", Novelid, NA  ) ), H596_nov =ifelse(is


res1<-
  res1%>%
  mutate(geneid=X,gene= X )

res1$geneid <- gsub("\\..*", "",res1$geneid )



res1<-left_join(res1,L596, by=c("geneid"="V3")) %>%
  select(-V2,-V4) %>%
  left_join(H596, by=c("geneid"="V3")) %>%
  select(-V2,-V4, -geneid) %>%
  separate(col = gene, sep = "\\|", into = c("gene","Novelid","class"))



res1<-
  res1%>%
  mutate(L596_nov =ifelse(is.na(class), NA, ifelse(class=="L596", Novelid, NA  ) ), H596_nov =ifelse(is


res1_go<-
  res1 %>%
  select(gene,V1.x,V1.y, L596_nov,H596_nov) %>%
  gather(key="class", value="isoform",2:5) %>%
  na.omit() %>%
  mutate(type= ifelse(class=="V1.x", "L596", ifelse(class=="V1.y", "H596", class))) %>%
  select(-class)




res1_go<-rbind(left_join( res1_go %>% filter(grepl("H596", type)), H596_go, by=c("isoform" = "V1" )  ),

res1_go<-
res1_go %>%
  select(gene, V2 ) %>%
  unique(.)
```

```
down<-res1_select %>%  filter(sig=="down") %>% select(gene)
up<-res1_select %>%  filter(sig=="up") %>% select(gene)


#F:/dir/2.Structral_analysis/function_annonation/5.GO_KEGG
library("clusterProfiler")


term<-go2term(res1_go[,2])
ont<-go2ont(res1_go[,2])
colnames(res1_go)<-c("V1","V2")
#colnames(up)<-c("V1")
colnames(term)<-c("V1","V2")
res1_go<-res1_go[,c("V2","V1")]



library(topGO)
library(dplyr)
library(data.table)
#setwd("F:/GO/")
#d=read.table("ref.go",header=F)
#res1_go<-res1_go[,c("V2","V1")]


topGO_func<-function(gene, golist){
  pd<-list()
  out<-list()
  for (i in c("MF", "BP", "CC" )){
#d<-res1_go[,c("V1","V2")]
d<-golist
colnames(d)=c("gene_id","go_id")

d$gene_id=as.character(d$gene_id)
d$go_id=as.character(d$go_id)
all_go <- lapply(split(d, sub("\\.\\d+$", "", d[, 1])), function(x) unique(x[, 2]))

geneNames=names(all_go)

#gene=fread("ZL2020-SXHZ.dispensable.id",header=F)
gene<-as.data.frame(gene)
colnames(gene)=c("geneid")
outlier_gene = as.vector(gene$geneid)
head(outlier_gene)
geneList=factor(as.integer(geneNames %in% outlier_gene))
names(geneList)=geneNames
head(geneList)
GOdata <- new("topGOdata", ontology = i, allGenes = geneList,annot=annFUN.gene2GO, gene2GO = all_go)
restRes=runTest(GOdata,algorithm="classic",statistic="fisher")



#genetable <- GenTable(GOdata, p.value = restRes, orderBy = "p.value" )
```

```
#gene_table=GenTable(GOdata,Fisher.p=restRes, topNodes=)

gene_table=GenTable(GOdata,Fisher.p=restRes,topNodes=82)
gene_table_total=GenTable(GOdata,Fisher.p=restRes,topNodes=length(nodeData( graph(GOdata) )) )



allGO = usedGO(GOdata)
pvalues=gene_table_total$Fisher.p
gene_table$adjust.p=round(head(p.adjust(pvalues,method="BH"),82),10)


out[[i]]<- gene_table %>% mutate(class=i)
}
#write.
  return(out)
}

out_up<-topGO_func(res1_select[,"gene"] ,res1_go[,c(2,1)]  )
#gene<-res1_select[,"gene"]
#golist<-res1_go[,c(2,1)]

GO_out<-out_up
GO<-do.call("rbind", GO_out)

detach("package:ggbiplot", unload=TRUE)
detach("package:plyr", unload=TRUE)

GO %>%
  group_by(class) %>%
  #filter(Fisher.p=="< 1e-30")
  mutate_at(.vars=vars(Fisher.p), .funs=function(x){ x=ifelse(x=="< 1e-30", "1e-30", x ) }  ) %>%
  mutate( cumProp = seq(0,1,length.out = n())) %>%
  filter(cumProp<0.15) %>%
  ungroup() %>%
  ggplot(aes(y= factor(Term,levels = rev(Term)),x=-log10(as.numeric(Fisher.p)), fill=class))+
  geom_col()+
  theme_bw() +
  theme(legend.position="right",
        axis.title.x=element_text(size=14,color="black",hjust=0.5),

        axis.title.y=element_text(size=14,color="black"),
        axis.text.x=element_text(size=12,color="black"),
        axis.text.y=element_text(size=12,color="black"),
        legend.text=element_text(size=15,color="black"),
        legend.title=element_text(size=15,color="black"),
        plot.title=element_text(size=15,color="black",hjust=0.5))+
  scale_fill_manual(values=c("#66C3A4", "#FD8D61", "#8DA3CB")) +
  xlab("-log(Pvalue)")+
  ylab("GO term")
```




```
#go_up<-go_up[which(!duplicated(go_up$Term)),]
GO<-GO[which(!duplicated(GO$Term)),]


GO %>%
  mutate_at(.vars=vars(Fisher.p), .funs=function(x){ x=ifelse(x=="< 1e-30", "1e-30", x ) }  ) %>%
  filter(adjust.p <=0.05) %>%
  group_by(class) %>%
 # mutate( cumProp = seq(0,1,length.out = n())) %>%
#  filter(cumProp<0.15) %>%
#  ungroup() %>%
  mutate(GeneRatio= Significant/length(res1_select$gene) , BgRatio = Annotated/23575 ,fold_enrichment =
 # ggplot(aes(x=fold_enrichment, y= factor(nTerm,levels = rev(nTerm)), colour=-1*log10(as.numeric(Fishe
  ggplot(aes(x=fold_enrichment, y= factor(Term,levels = rev(Term)), colour= -log10(as.numeric(adjust.p)+
  geom_point() +
  theme_bw() +
  theme(legend.position="right",
        axis.title.x=element_text(size=14,color="black",hjust=0.5),
        axis.title.y=element_text(size=14,color="black"),
        axis.text.x=element_text(size=12,color="black"),
        axis.text.y=element_text(size=12,color="black"),
        legend.text=element_text(size=15,color="black"),
        legend.title=element_text(size=15,color="black"),
        plot.title=element_text(size=15,color="black",hjust=0.5)) +
  #scale_size_continuous(name = "proportion\nof genes\nin pathway", range = c(0.2, 6), breaks = c(0.1,
 # scale_color_continuous(name = "-Log10(FDR)") +
  scale_color_gradient( name = "-Log10(FDR)", low = "#0000FF",high = "#FE0000") +
```

```
  xlab("Fold Enrichment") +
  ylab("GO Term")
```