

Functional landscape between WGD gene pairs

Changfu Jia

2022-08-14

```
#knitr::opts_chunk$set(echo = TRUE)
#library(knitr)
knitr::opts_chunk$set(fig.width=12, fig.height=8,
                      echo=TRUE, warning=FALSE, message=FALSE)

#kable(data)
options(warn=-1)

library(tidyverse)
library(ggpubr)
setwd("F:/dir")

orth <- read.table( "input-data/wgdi/dup_gene.tsv" ,header = T,
                  sep = "\t", na.strings = "-")
orth <- orth %>% filter(class=="wgd") %>% select(-OG , -N1 , -N11)

six_tis<-read.table("input-data/expression_data/Atha_Ov_sixtis.tsv",
                  header = T, sep = " ")
six_tis<-six_tis %>% select(-TPC,-TPCA)
#c( "flower" , "leaf" , "root" , "stem" , "silique", "seed" )

six_tis<-
six_tis %>%
  mutate_at( colnames(six_tis), function(x){ ifelse(is.na(x),0,x ) }) %>%
  mutate(A1_status=ifelse( eA1<1&eA2<1&eA3<1&eA4<1&eA5<1&eA6<1 , "none", "yes"),
        P1_status=ifelse( eP1<1&eP2<1&eP3<1&eP4<1&eP5<1&eP6<1 , "none", "yes"),
        P2_status=ifelse( eC1<1&eC2<1&eC3<1&eC4<1&eC5<1&eC6<1 , "none", "yes") )
  #filter( A1_status == "none" )

orth_tbl<-cbind(orth,six_tis)

#orth_tbl %>%
# filter(A1_status=="yes")

Tau<-function(x){
  aa<- apply(x ,1,function(x){x/max(x)})
  tau<-as.data.frame(colSums(1-aa)/(ncol(x)-1))
```

```

return(tau)

}

for (i in 1:nrow(orth_tbl)){
  dup1<-orth_tbl[i,7:12]
  dup2<-orth_tbl[i,13:18]
  anc<-orth_tbl[i,19:24]
  rownames(dup1)<-orth_tbl[i,2]
  rownames(dup2)<-orth_tbl[i,3]
  rownames(anc)<-orth_tbl[i,1]
  #mat<-log(rbind(as.matrix(anc),as.matrix(dup1),as.matrix(dup2))+ 0.01 )
  #EI calculate
  #orth_tbl[i,"EI"]<-dist( mat)[1]
  #cor(t(mat))[1,2]
  #Pearson
  pears<-cor.test(as.matrix(log2(dup1+0.01)), as.matrix(log2(dup2+0.01)))
  A_dup1<-cor.test(as.matrix(log2(anc+0.01)), as.matrix(log2(dup1+0.01)))
  A_dup2<-cor.test(as.matrix(log2(anc+0.01)), as.matrix(log2(dup2+0.01)))
  A_sum<-cor.test( as.matrix(log2(anc+0.01)), log2(as.matrix(dup1)+as.matrix(dup2)+0.01))
  orth_tbl[i,"Pearson"]<-pears$estimate
  orth_tbl[i,"pvalue"]<-pears$p.value
  orth_tbl[i,"A_dup1_pearson"]<-A_dup1$estimate
  orth_tbl[i,"A_dup1_pvalue"]<-A_dup1$p.value
  orth_tbl[i,"A_dup2_pearson"]<-A_dup2$estimate
  orth_tbl[i,"A_dup2_pvalue"]<-A_dup2$p.value
  orth_tbl[i,"Sum_pearson"]<-A_sum$estimate
  orth_tbl[i,"Sum_pvalue"]<-A_sum$p.value
  orth_tbl[i,"Tau_dup1"]<-Tau(dup1)[1,1]
  orth_tbl[i,"Tau_dup2"]<-Tau(dup2)[1,1]
  orth_tbl[i,"Tau_dup1"]<-Tau(dup1)[1,1]
  orth_tbl[i,"Tau_anc"]<-Tau(anc)[1,1]
}

intra_class<-read.table("0.cluster.txt")

intra_class<-
  intra_class%>%
  mutate(geneid=rownames(.))

orth_tbl_func<-
orth_tbl %>%
  filter(A1_status=="yes") %>%
  #na.omit() %>%
  mutate(high_cor=ifelse(A_dup1_pearson>=A_dup2_pearson,"dup1", "dup2") ) %>%
  left_join(intra_class, by=c("dup1"="geneid")) %>%
  left_join(intra_class, by=c("dup2"="geneid")) %>%
  mutate(S2_Tau = ifelse(high_cor=="dup1", Tau_dup1, Tau_dup2 )
    , S1_Tau =ifelse(high_cor=="dup2", Tau_dup1, Tau_dup2 ) ) %>%
  #na.omit() %>%
  mutate(S2= ifelse(high_cor=="dup1", A_dup1_pearson, A_dup2_pearson ),

```

```

      S1 =ifelse(high_cor=="dup2", A_dup1_pearson, A_dup2_pearson ),
      S1_p=ifelse(high_cor=="dup2", A_dup1_pvalue, A_dup2_pvalue ),
      S2_p = ifelse(high_cor=="dup1", A_dup1_pvalue, A_dup2_pvalue ) ) %>%
mutate( neo = ifelse( ((S2 > 0.6 & S2_p<0.05 )&(class.y!=class)) , "neo", "none" ),
      sub=ifelse( (S1<0.6&S2<0.6&class.y!=class), "sub", "none" ) ) %>%
mutate(strict_Sub=ifelse( sub=="sub"&Sum_pearson>0.6, "Strict","none" ) ) %>%
#mutate(non_func=ifelse(P1_status=="yes"&P2_status=="yes",
# "none", ifelse(P1_status=="none"&P2_status=="yes", "P1", "next")))
mutate(non_func=ifelse(P1_status=="yes"&P2_status=="yes", "none",
      ifelse(P1_status=="none"&P2_status=="yes", "P1",
      ifelse(P1_status=="yes"& P2_status=="none", "P2","Both" ) ) )

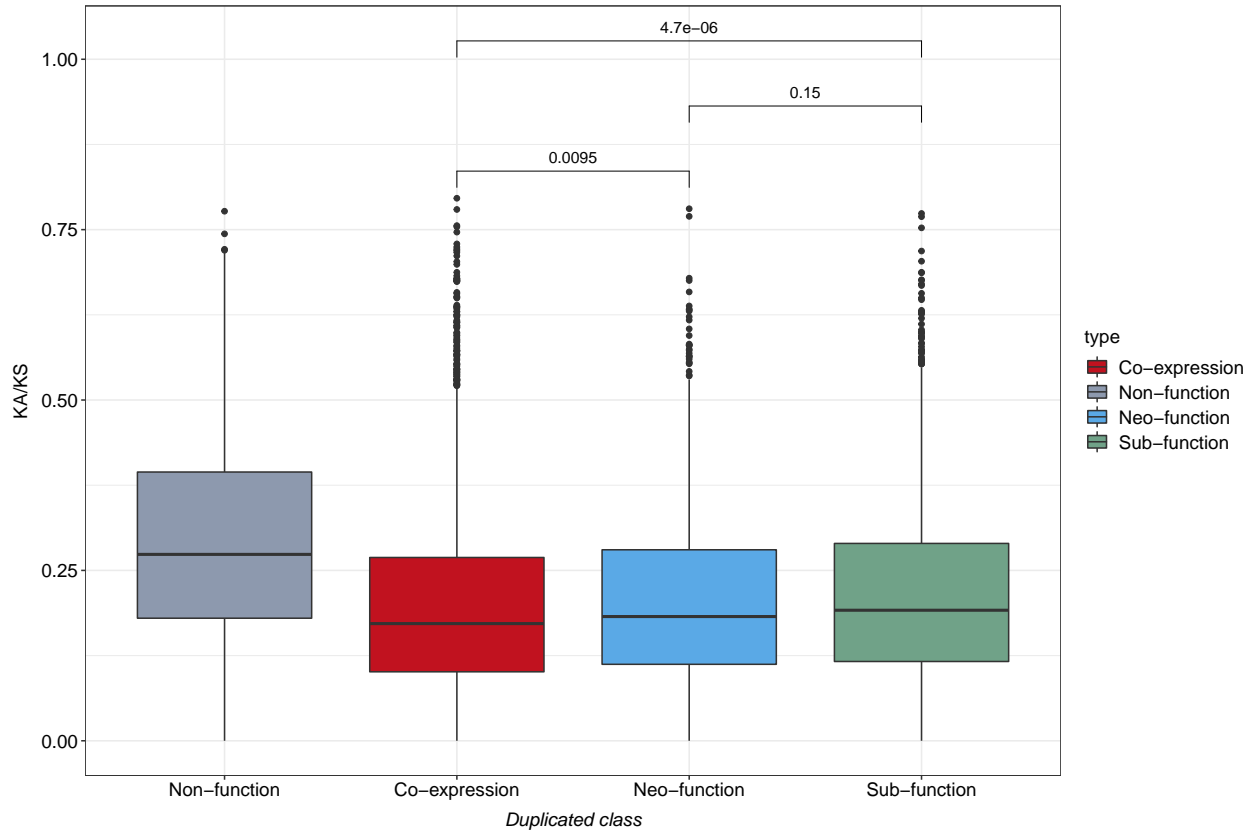
setwd("F:/dir")

orth_intra<-read.table(file= "orth_tbl_intra.txt",header=T)

#orth_intra %>% select(ref,kaks)

left_join(orth_tbl_func , (orth_intra %>% select(ref,kaks)), by=c("ref")) %>%
mutate(type=ifelse(non_func!="none", "Non-function",
      ifelse(neo=="neo" , "Neo-function",
      ifelse(sub=="sub" , "Sub-function", "Co-expression" ) ))) %>%
filter(!is.na(type), kaks>=0, kaks<=0.8) %>%
ggplot(aes(x=factor(type,levels=c("Non-function","Co-expression",
      "Neo-function","Sub-function")) , y=kaks, fill=type))+
geom_boxplot() +
  theme_bw() +
  theme(
axis.title.x=element_text(size=13,color="black",hjust=0.5, face="italic", vjust = -1),
axis.title.y=element_text(size=13,color="black", vjust=2.5),
axis.text.x=element_text(size=13,color="black"),
axis.text.y=element_text(size=13,color="black"),
legend.text=element_text(size=13,color="black"),
legend.title=element_text(size=13,color="black"),
plot.title=element_text(size=13,color="black",face="italic",hjust=0.5)) +
ylab("KA/KS")+
xlab("Duplicated class") +
stat_compare_means(method = 'wilcox.test',
      comparisons = list(c("Co-expression","Neo-function"),
      c("Neo-function", "Sub-function"),
      c("Co-expression", "Sub-function" ) ) +
scale_fill_manual(values=c("Co-expression"= "#c1121f", "Non-function"="#8d99ae",
      "Neo-function"="#5aa9e6", "Sub-function"="#70a288" ) )

```



```
setwd("F:/dir")

blast<-read.table("input-data/ov.blast",header = F,sep = "\t")
blast$V1<- gsub("\\.t.*", "", blast$V1)
blast$V2<- gsub("\\.t.*", "", blast$V2)

orth_tbl_func_identity<-orth_tbl_func %>%mutate(cDNA=0)

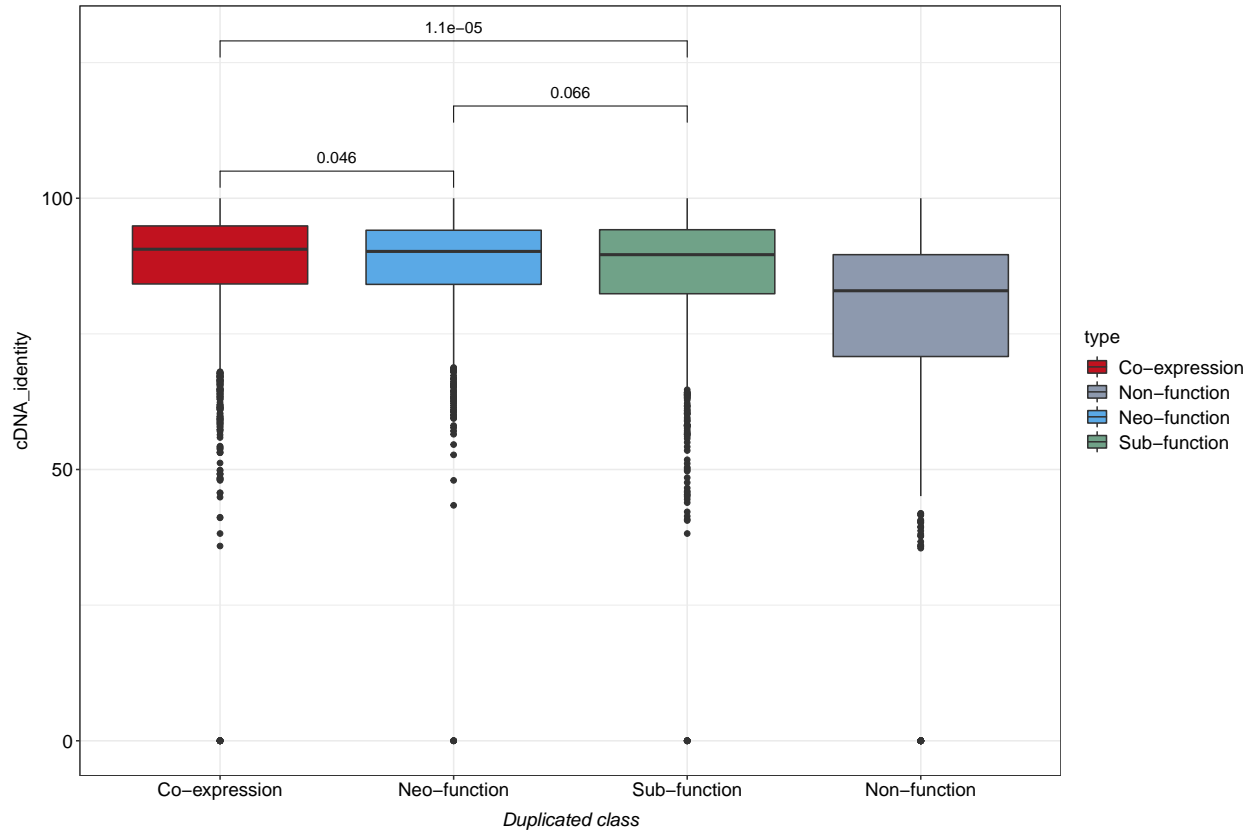
for (i in 1:nrow(orth_tbl_func_identity)){
  #dup1 <- orth_tbl[i,"dup1"]
  #dup2 <- orth_tbl[i,"dup2"]
  L1 <- which(blast$V1 %in% orth_tbl_func_identity[i,"dup2"])
  L2<- which(blast$V2 %in% orth_tbl_func_identity[i,"dup1"])
  #L1[L1 %in% L2]
  if ( ( TRUE %in% (L1 %in% L2 ) ) ){
    orth_tbl_func_identity[i,"cDNA"]<-blast[intersect(L1,L2 ),3]
  }else{
    L1 <- which(blast$V1 %in% orth_tbl_func_identity[i,"dup1"])
    L2<- which(blast$V2 %in% orth_tbl_func_identity[i,"dup2"])
    if ( ( TRUE %in% (L1 %in% L2 ) ) ){
      orth_tbl_func_identity[i,"cDNA"]<-blast[intersect(L1,L2 ),3]
    }else{
      orth_tbl_func_identity[i,"cDNA"]<-0
    }
  }
}
```

```

    }
  }
  #cc<-blast[intersect(L1,L2 ),3]
  #orth_tbl[i,"cDNA"]<-cc
}

orth_tbl_func_identity %>%
  mutate(a=paste(dup1,dup2,sep="_")) %>%
  left_join( (blast %>% mutate(a=paste(V1,V2,sep="_"))) ) %>%
  select(a,neo,sub,strict_Sub,non_func,V3) %>%
  left_join( (blast %>% mutate(b=paste(V2,V1,sep="_"))), by=c("a"="b") ) %>%
  select(a,neo,sub,strict_Sub,non_func,V3.x,V3.y) %>%
  mutate(cDNA_identity= ifelse(!is.na(V3.x), V3.x, ifelse(!is.na(V3.y), V3.y, 0)))%>%
  mutate(type=ifelse(non_func!="none", "Non-function",
                    ifelse(neo=="neo", "Neo-function",
                          ifelse(sub=="sub", "Sub-function", "Co-expression") ))) %>%
  filter(!is.na(type)) %>%
  ggplot(aes(x=factor(type,
                    levels=c("Co-expression","Neo-function","Sub-function","Non-function")) ,
            y=cDNA_identity, fill=type))+
  geom_boxplot() +
  theme_bw() +
  theme(
    axis.title.x=element_text(size=13,color="black",hjust=0.5, face="italic", vjust = -1),
    axis.title.y=element_text(size=13,color="black", vjust=2.5),
    axis.text.x=element_text(size=13,color="black"),
    axis.text.y=element_text(size=13,color="black"),
    legend.text=element_text(size=13,color="black"),
    legend.title=element_text(size=13,color="black"),
    plot.title=element_text(size=13,color="black",face="italic",hjust=0.5)) +
  ylab("cDNA_identity")+
  xlab("Duplicated class") +
  stat_compare_means(method = 'wilcox.test',
                    comparisons = list(c("Co-expression","Neo-function"),
                                       c("Neo-function", "Sub-function"), c("Co-expression", "Sub-function") )) +
  scale_fill_manual(values=c("Co-expression"= "#c1121f", "Non-function"="#8d99ae",
                             "Neo-function"="#5aa9e6", "Sub-function"="#70a288" ) )

```



```
P1<-orth_tbl_func %>%
  filter(non_func=="P2" | non_func=="P1")%>%
  select(grep("Tau$", colnames(.)),A_dup1_pearson,A_dup2_pearson) %>%
  # na.omit() %>%
  gather(key=type, value=Tau, 1:2) %>%
  ggplot(aes(x=type,y=Tau, fill=Tau)) +
  geom_boxplot()+
  stat_compare_means() +
  theme_bw() +
  theme(
    axis.title.x=element_text(size=13,color="black",hjust=0.5, face="italic", vjust = -1),
    axis.text.x=element_text(size=13,color="black"),
    axis.text.y=element_text(size=13,color="black"),
    legend.text=element_text(size=13,color="black"),
    legend.title=element_text(size=13,color="black"),
    plot.title=element_text(size=13,color="black",face="italic",hjust=0.5)) +
  ylab("Tissue specificity")+
  xlab("Non-function")
```

```
P2<-
orth_tbl_func %>%
  filter(neo=="neo") %>%
  filter(S1<0.6) %>%
  filter(S1_p>0.03) %>%
  select(grep("Tau$", colnames(.)),A_dup1_pearson,A_dup2_pearson) %>%
  gather(key=type, value=Tau, 1:2) %>%
```

```

ggplot(aes(x=type, y=Tau, fill=Tau))+
geom_boxplot()+
stat_compare_means() +
  theme_bw() +
  theme(
axis.title.x=element_text(size=13,color="black",hjust=0.5, face="italic", vjust = -1),
axis.text.x=element_text(size=13,color="black"),
axis.text.y=element_text(size=13,color="black"),
legend.text=element_text(size=13,color="black"),
legend.title=element_text(size=13,color="black"),
plot.title=element_text(size=13,color="black",face="italic",hjust=0.5)) +
ylab("Tissue specificity")+
xlab("Neo-function")

```

P3<-

```

orth_tbl_func %>%
  filter(sub=="sub") %>%
  select(grep("Tau$", colnames(.)),A_dup1_pearson,A_dup2_pearson) %>%
  gather(key=type, value=Tau, 1:2) %>%
  ggplot(aes(x=type, y=Tau, fill=Tau))+
  geom_boxplot()+
  stat_compare_means() +
    theme_bw() +
    theme(
axis.title.x=element_text(size=13,color="black",hjust=0.5, face="italic", vjust = -1),
axis.text.x=element_text(size=13,color="black"),
axis.text.y=element_text(size=13,color="black"),
legend.text=element_text(size=13,color="black"),
legend.title=element_text(size=13,color="black"),
plot.title=element_text(size=13,color="black",face="italic",hjust=0.5)) +
ylab("Tissue specificity")+
xlab("Sub-function")

```

```

#orth_tbl_func %>%
# filter(strict_Sub=="Strict")%>%
# select(grep("Tau$", colnames(.)),A_dup1_pearson,A_dup2_pearson) %>%
# gather(key=type, value=Tau, 1:2) %>%
# ggplot(aes(x=type, y=Tau, fill=Tau))+
# geom_boxplot()+
# stat_compare_means() +
#   theme_bw() +
#   theme(
# axis.title.x=element_text(size=13,color="black",hjust=0.5, face="italic", vjust = -1),
# axis.text.x=element_text(size=13,color="black"),
# axis.text.y=element_text(size=13,color="black"),
# legend.text=element_text(size=13,color="black"),
# legend.title=element_text(size=13,color="black"),
# plot.title=element_text(size=13,color="black",face="italic",hjust=0.5)) +
# ylab("Tissue specificity")+
# xlab("Sub-function")

```

P4<-

```

orth_tbl_func %>%

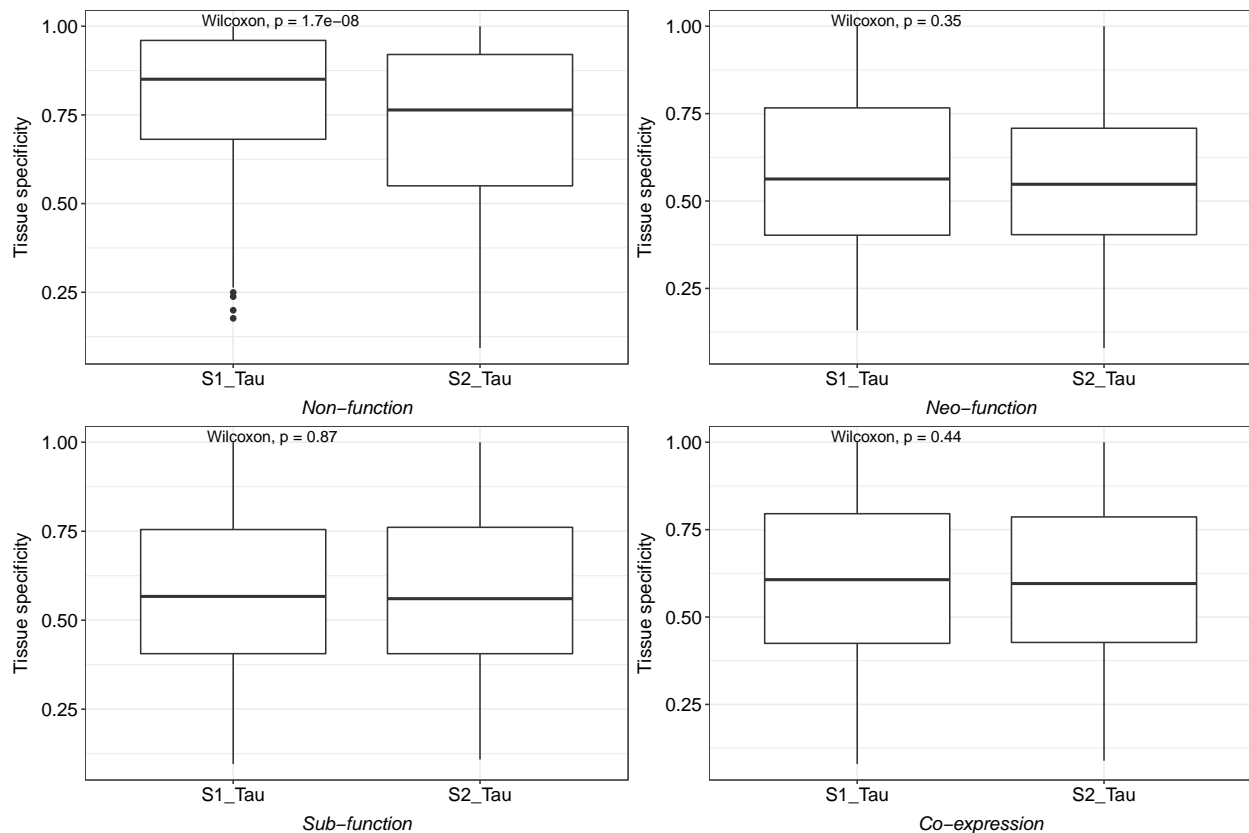
```

```

filter(neo=="none",sub=="none",non_func=="none") %>%
select(grep("Tau$", colnames(.)),A_dup1_pearson,A_dup2_pearson) %>%
gather(key=type, value=Tau, 1:2) %>%
ggplot(aes(x=type, y=Tau, fill=Tau))+
geom_boxplot()+
stat_compare_means() +
  theme_bw() +
  theme(
axis.title.x=element_text(size=13,color="black",hjust=0.5, face="italic", vjust = -1),
axis.title.y=element_text(size=13,color="black", vjust=2.5),
axis.text.x=element_text(size=13,color="black"),
axis.text.y=element_text(size=13,color="black"),
legend.text=element_text(size=13,color="black"),
legend.title=element_text(size=13,color="black"),
plot.title=element_text(size=13,color="black",face="italic",hjust=0.5)) +
ylab("Tissue specificity")+
xlab("Co-expression")

```

```
ggarrange(P1,P2,P3,P4)
```



```

#install.packages('ggpointdensity')
#library(ggpointdensity)

```

```

#orth_tbl_func %>%
# filter(neo=="neo") %>%

```



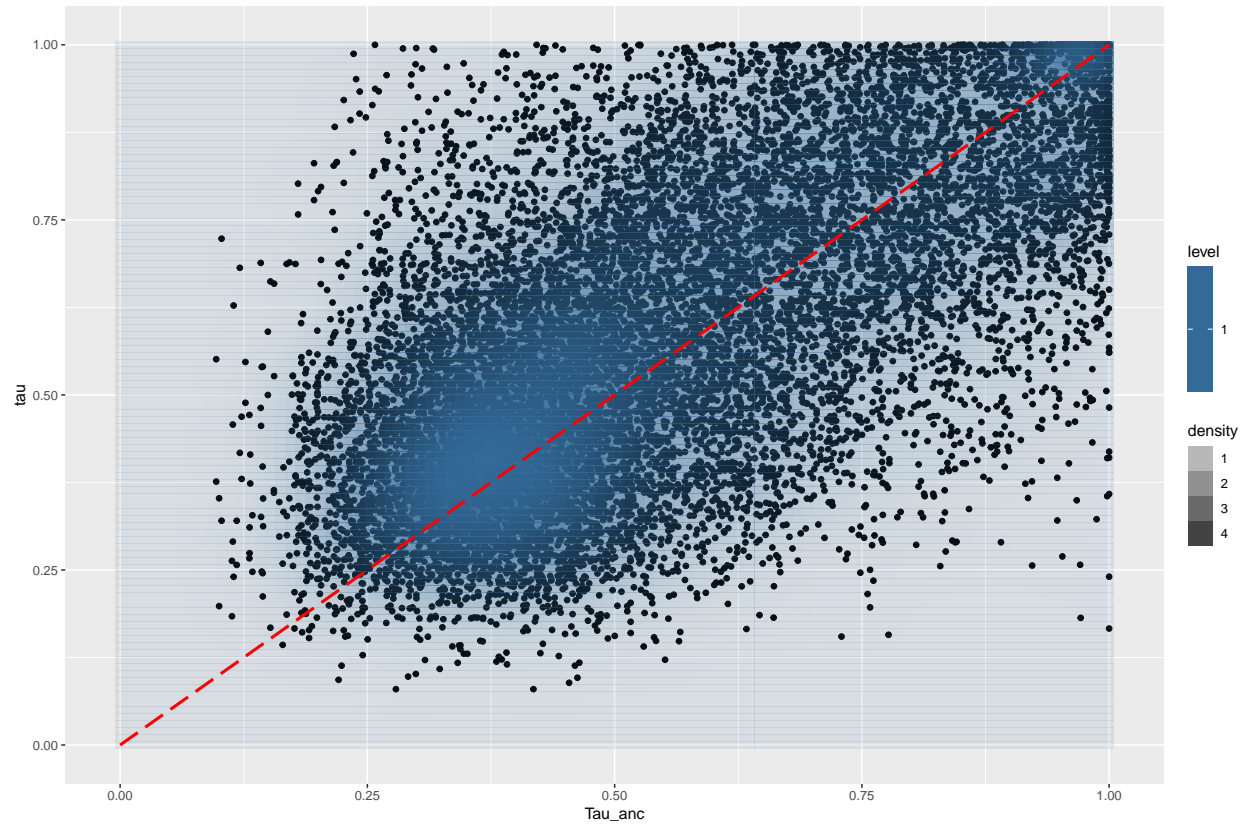
```

# select(grep("Tau", colnames(.)),A_dup1_pearson,A_dup2_pearson)%>%
# ggplot(aes(x=S2_Tau,y=S1_Tau)) +
# geom_point()+
# geom_pointdensity() +
# stat_density_2d_filled(aes(alpha=..density..), geom="tile", contour = F )+
# geom_smooth(color="steelblue2",method="lm",se=FALSE)
# annotate("segment", x=0, xend=1, y=0, yend=1, linetype = "longdash",size=1, color="red" )

orth_tbl_func %>%
# filter(non_func=="P2" | non_func=="P1")%>%
# select(grep("Tau", colnames(.)),A_dup1_pearson,A_dup2_pearson)%>%
# na.omit() %>%
# ggplot(aes(x=S2_Tau,y=S1_Tau)) +
# geom_point()

orth_tbl_func %>%
na.omit()%>%
# filter(neo=="neo") %>%
select(grep("^Tau", colnames(.))) %>%
gather(key = dup, value = tau, 1:2) %>%
ggplot(aes(x=Tau_anc,y=tau))+
geom_point()+
stat_density2d_filled(aes(alpha=..density..), geom="tile", contour = F )+
#geom_smooth(color="steelblue2",method="lm",se=FALSE)
annotate("segment", x=0, xend=1, y=0, yend=1, linetype = "longdash",size=1, color="red" )

```



Boxplot for Tau

```

#orth_tbl_func %>%
# filter(neo=="neo") %>%
# filter(S1<0.6) %>%
# filter(S1_p>0.03) %>%
# select(grep("Tau$", colnames(.)),A_dup1_pearson,A_dup2_pearson) %>%
# gather(key=type, value=Tau, 1:2) %>%
# ggplot(aes(x=type, y=Tau, fill=Tau))+
#   geom_boxplot()+
#   stat_compare_means() +
#   theme_bw() +
#   theme(
#     axis.title.x=element_text(size=13,color="black",hjust=0.5, face="italic", vjust = -1),
#     axis.text.x=element_text(size=13,color="black"),
#     axis.text.y=element_text(size=13,color="black"),
#     legend.text=element_text(size=13,color="black"),
#     legend.title=element_text(size=13,color="black"),
#     plot.title=element_text(size=13,color="black",face="italic",hjust=0.5)) +
#   ylab("Tissue specificity")+
#   xlab("Neo-function")

```

```
ATAC<-list()
```

```

for (i in dir("input-data/ATAC")){
  file<-paste("input-data/ATAC", i,sep="/")
  ID<- gsub(".geneDistance.bed", "", i)

```

```

ATAC[[ID]]<-read.table(file, header = F, sep="\t")
}

#rm(ATAC_plot)
ATAC_plot<-list()

#names(ATAC)
for (i in names(ATAC)) {

  out<-i
  ATAC_plot[[out]]<-
  orth_tbl_func %>%
    #select( -media, -media2, -V3.x, -V3.y ) %>%
    mutate(type=ifelse(non_func!="none", "Non-function", ifelse(neo=="neo", "Neo-function",
      ifelse(sub=="sub", "Sub-function", "Co-expression") ))) %>%
    mutate( S2=ifelse( high_cor=="dup1",dup1 , dup2 ),
      S1= ifelse( high_cor=="dup1",dup2 , dup1 ) ) %>%
    select(S1,S2,type) %>%
    na.omit() %>%

    #mutate(EIclass=ifelse(EI<=2,"0-2", ifelse(EI<=4,"2-4" ,
    #ifelse(EI<=6, "4-6", ifelse(EI<=8, "6-8", "8") ) ) ) ) %>%
    left_join( ATAC[[i]], by=c("S1"="V9") ) %>%

    select(S1,S2,type,V11) %>%
    left_join( ATAC[[i]], by=c("S2"="V9") ) %>%
    select(S1,S2,type,V11.x,V11.y) %>%
    gather(key=Copy,value=distance,4:5) %>%
    na.omit() %>%
    mutate(distance_kb=distance/1000) %>%
    ggplot(aes(x=log10(distance_kb+0.0001),color=type )) +
    geom_density(size=2) +
    geom_vline(xintercept = log10(2+0.0001), linetype = "longdash", color= "red")+
    theme_bw() +
    theme(#legend.position = c(.4, 0.8),
      legend.position = "none",
      axis.ticks = element_line(color = "black", linetype = "solid", size = 1),
      axis.ticks.length.x=unit(0.2, "cm"),
      panel.grid=element_blank(),
      axis.title.x=element_text(size=20,color="black",hjust=0.5, vjust = -1),
      axis.text.x=element_text(size=15,color="black"),
      axis.text.y=element_text(size=15,color="black"),
      #legend.text=element_text(size=20,color="black"),
      #legend.title=element_blank(),
      #legend.key.size = unit(1,"cm"),
      plot.title=element_text(size=13,color="black",face="italic",hjust=0.5)) +
    #ylab("Relative Density")+
    ylab("")+
    #xlab("Distance between ACR and nearest gene (kb)") +
    xlab("")+

```

```

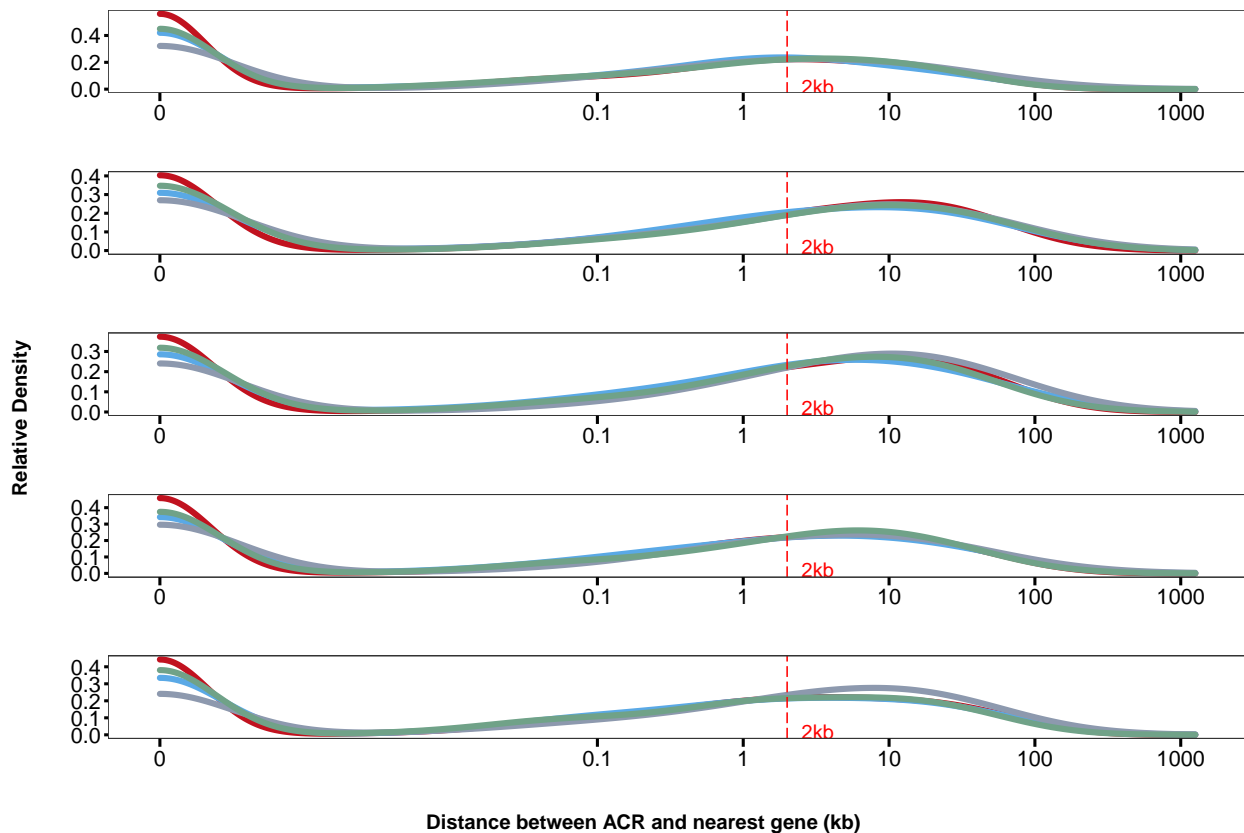
scale_color_manual(values=c("Co-expression"= "#c1121f",
                             "Non-function"="#8d99ae",
                             "Neo-function"="#5aa9e6",
                             "Sub-function"="#70a288") ) +
scale_x_continuous( breaks = c(-4,-1,0,1,2,3), limits = c(-4,3.1)
                    ,labels = c("0","0.1","1","10","100","1000")) +
annotate("text", x=0.4,y= 0.02, hjust=0, color= "red", label="2kb",size=5 )

#xlim(0,1000)
}

p1<- ggarrange(plotlist =ATAC_plot, nrow = 5)

annotate_figure(p1, bottom = text_grob("Distance between ACR and nearest gene (kb)",
                                       color = "black", face = "bold", size = 14),
               left= text_grob("Relative Density", color = "black", face = "bold",
                              size = 14, rot = 90) )

```



```

#ggsave("ATAC_func_distance.pdf",width = 5, height = 15)

setwd("F:/dir")

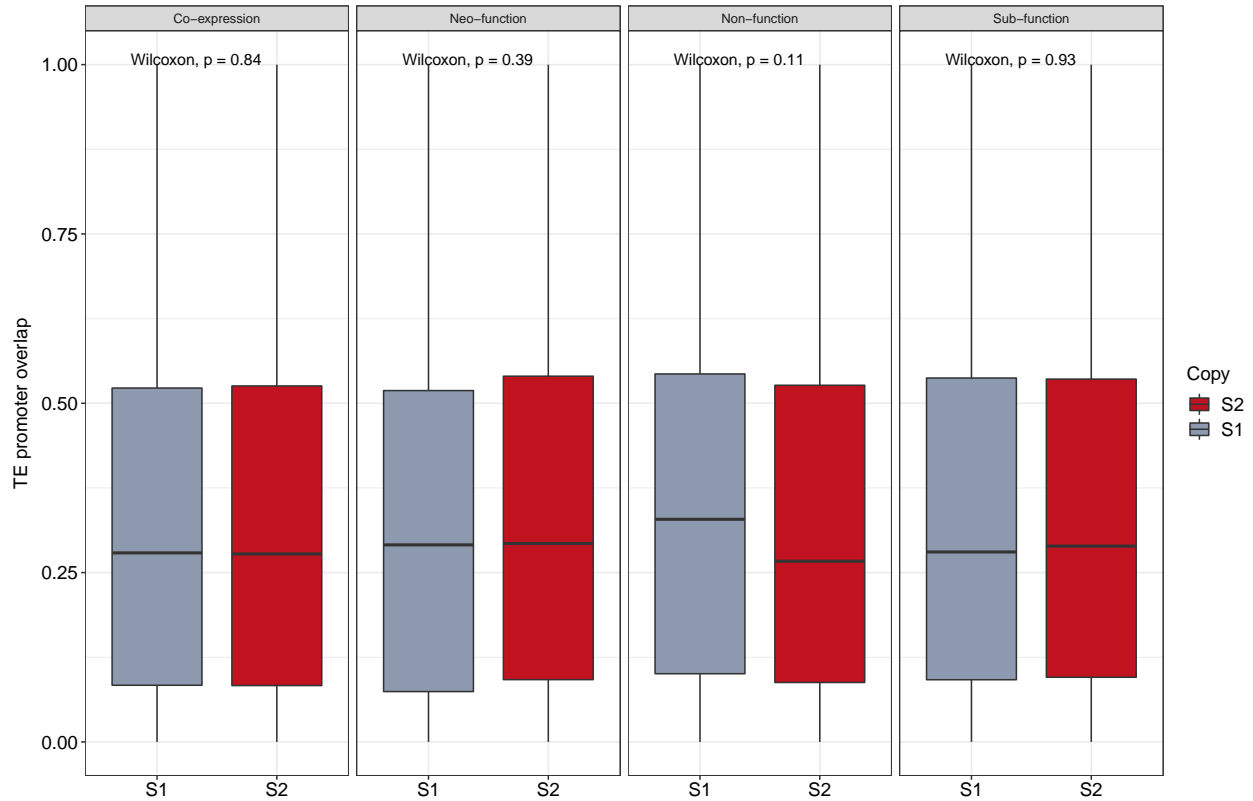
te_all_promoter<-read.table(file="input-data/LTR/ov_promoter_te.overlap"
                             ,header = F,sep = "\t")

```

```

orth_tbl_func %>%
  mutate(type=ifelse(non_func!="none", "Non-function",
                     ifelse(neo=="neo", "Neo-function",
                             ifelse(sub=="sub", "Sub-function", "Co-expression") ))) %>%
  left_join(te_all_promoter, by=c("dup1"="V4")) %>%
  left_join(te_all_promoter, by=c("dup2"="V4")) %>%
  select(type, high_cor, V8.x, V8.y) %>%
  na.omit() %>%
  mutate( S2=ifelse( high_cor=="dup1", V8.x , V8.y ),
          S1= ifelse( high_cor=="dup1", V8.y , V8.x ) ) %>%
  select(-V8.x, -V8.y, -high_cor) %>%
  gather(key=Copy, value=TE, 2:3) %>%
  ggplot(aes(x=Copy, y=TE, fill=Copy))+
  geom_boxplot()+
  facet_grid(~type)+
  stat_compare_means() +
  theme_bw() +
  theme(
axis.title.x=element_text(size=13, color="black", hjust=0.5, face="italic", vjust = -1),
axis.text.x=element_text(size=13, color="black"),
axis.text.y=element_text(size=13, color="black"),
legend.text=element_text(size=13, color="black"),
legend.title=element_text(size=13, color="black"),
plot.title=element_text(size=13, color="black", face="italic", hjust=0.5)) +
ylab("TE promoter overlap")+
xlab("") +
scale_fill_manual(values=c("S2"= "#c1121f", "S1"= "#8d99ae") )

```



```

methy_id<-
orth_tbl_func %>%
  mutate(type=ifelse(non_func!="none", "Non-function",
    ifelse(neo=="neo", "Neo-function",
    ifelse(sub=="sub", "Sub-function", "Co-expression")  )))%>%
  mutate( S2=ifelse( high_cor=="dup1",dup1 , dup2 ),
    S1= ifelse( high_cor=="dup1",dup2 , dup1 ) ) %>%
  select(S1,S2,type)

#write.table(methy_id,file = "function_methy.txt",sep="\t", col.names = F, quote = F)
setwd("F:/dir")

methy_func<-read.table("function.plot",header = F)

methy_plot<-list()

methy_func<-
methy_func %>%
  mutate( type= gsub("_.*", "", V3), copy = gsub(".*_", "", V3),
    tissue = gsub(".flank.bin.meth", "", gsub("_context", "", V4) ) )

for (i in unique(methy_func$tissue)){

methy_plot[[i]]<-methy_func %>%
  filter( grepl(pattern = i , tissue) ) %>%
  ggplot(aes(x=V1,y=V2,color=copy))+

```

```

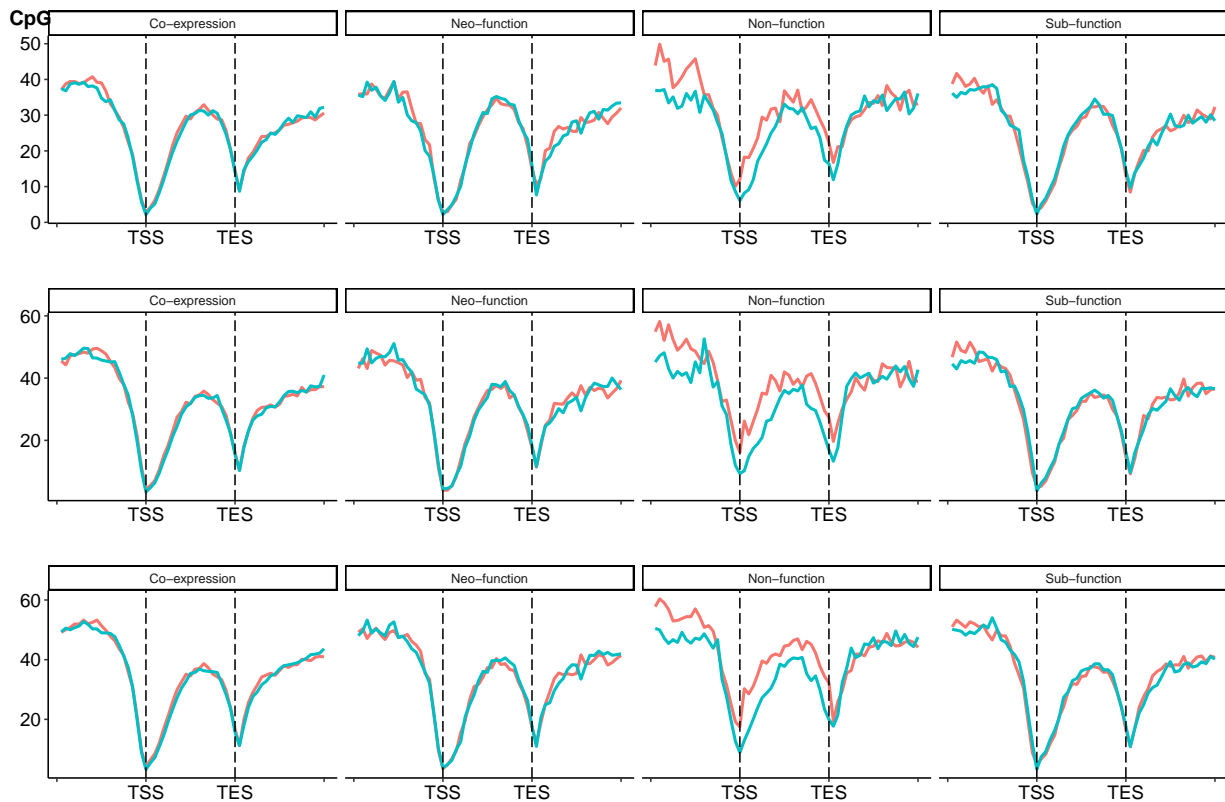
geom_line(size=1) +
facet_wrap(~type, nrow = 1) +
  #scale_fill_manual(values=c("0-2"= "#264653", "2-4"="#e76f51", "4-6"="#f4a261", "6-8"="#e9c46a", "8-10"="#f4a261", "10-12"="#e76f51", "12-14"="#264653")) +
  #scale_color_manual(values=c("0-2"= "#264653", "2-4"="#e76f51", "4-6"="#f4a261", "6-8"="#e9c46a", "8-10"="#f4a261", "10-12"="#e76f51", "12-14"="#264653")) +
theme_classic() +
theme(legend.position = "none",
axis.title.x=element_text(size=15,color="black",hjust=0.5, vjust = -1),
axis.text.x=element_text(size=13,color="black"),
axis.text.y=element_text(size=13,color="black"),
legend.text=element_text(size=13,color="black"),
legend.title=element_text(size=13,color="black"),
plot.title=element_text(size=13,color="black",hjust=0.5)) +
geom_vline(xintercept = 20, linetype = "longdash" ) +
geom_vline(xintercept = 40, linetype = "longdash" ) +
ylab("")+
#ylab("Average Methylation Density") +
xlab("")+
scale_x_continuous(labels = c("", "TSS", "TES", ""))+
guides(color=guide_legend(title = "Distance"))
}

```

```

ggarrange(plotlist = methy_plot[grep("CpG",names(methy_plot),value = T )] , nrow = 3, labels = "CpG")

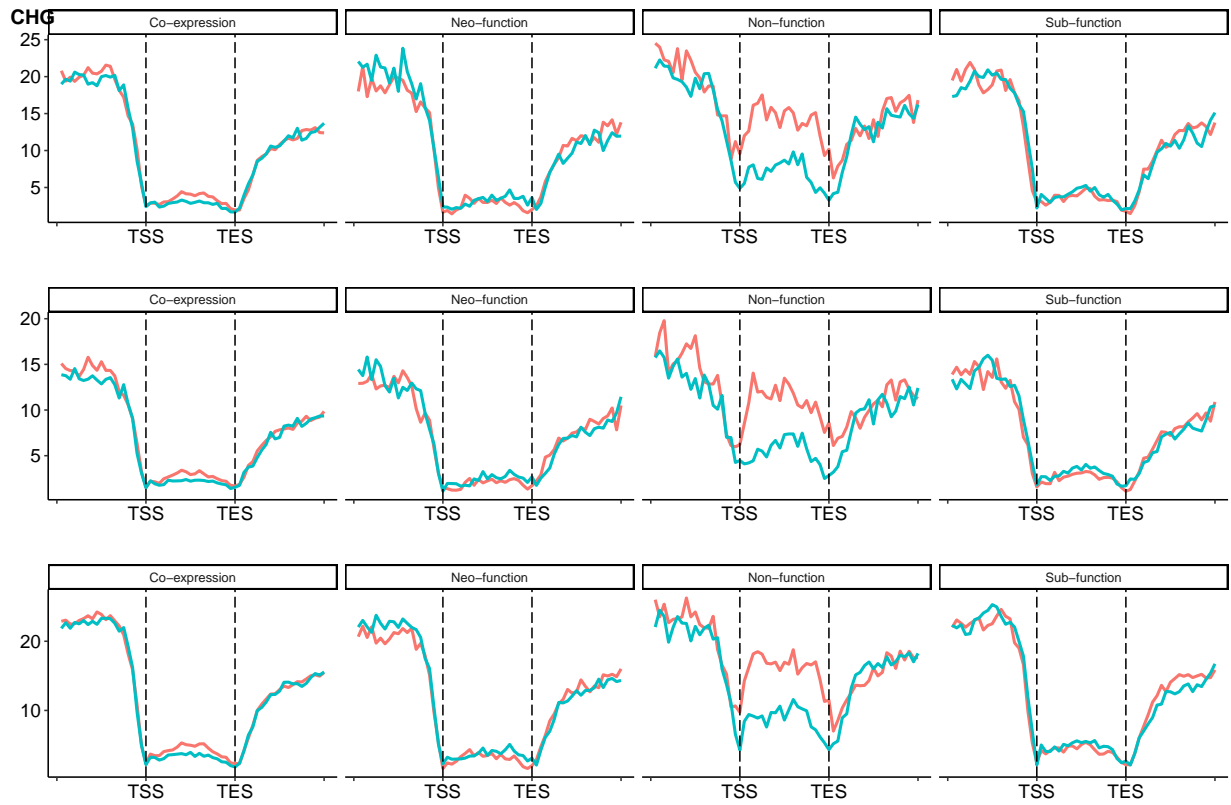
```



```

ggarrange(plotlist = methy_plot[grep("CHG",names(methy_plot),value = T )] , nrow = 3, labels = "CHG")

```



```
ggarrange(plotlist = methy_plot[grep("CHH",names(methy_plot),value = T )] , nrow = 3, labels = "CHH")
```