```
###load libraries
library(affy)              #Version 1.64.0
library(AnnotationDbi)      #Version 1.48.0
library(beeswarm)           #Version 0.2.3
library(biomaRt)           #Version 2.42.0
library(cgdsr)             #Version 1.3.0
library(clusterProfiler)    #Version 3.14.3
library(data.table)         #Version 1.12.8
library(export)            #Version 0.2.2.9000
library(fgsea)             #Version 1.12.0
library(gcrma)             #Version 2.58.0
library(GenomicFeatures)     #Version 1.38.0
library(GenomicRanges)      #Version 1.38.0
library(ggbeeswarm)         #Version 0.6.0
library(ggcorrplot)         #Version 0.1.3
library(ggplot2)           #Version 3.3.1
library(ggpubr)            #Version 0.3.0
library(glmnet)            #Version 3.0-2
library(gplots)            #Version 3.0.3
library(gtools)            #Version 3.8.1
library(hgu133plus2.db )     #Version 3.2.3
library(Hmisc)             #Version 4.4-0
library(httr)              #Version 1.4.1
library(limma)             #Version 3.42.0
library(magrittr)           #Version 1.5
library(openxlsx)           #Version 4.1.4
library(org.Hs.eg.db)       #Version 3.10.0
library(PerformanceAnalytics) #Version 2.0.4
library(psych)              #Version 1.9.12.31
library(R.utils)            #Version 2.9.2
library(randomForestSRC)     #Version 2.9.3
library(ReactomePA)          #Version 1.30.0
library(RColorBrewer)        #Version 1.1-2
library(readxl)             #Version 1.3.1
library(reshape2)           #Version 1.4.3
library(survminer)           #Version 0.4.6
library(tidyverse)          #Version 1.3.0
library(UCSCXenaTools)       #Version 1.3.1
library(vctrs)              #Version 0.3.1
library(VennDiagram)         #Version 1.6.2
library(xtable)             #Version 1.8-4

###Read the KM survival analysis data
Dataset<-read.xlsx("Sup Table 5_DDX kmplot data_surival.xlsx")

#Heatmap plotting of the -LOG (P VALUE) (Figure 1A)
windows(width=5, height=7)
heatmap.2(as.matrix(Dataset[,c(12:14)]), scale="column",
        col = bluered(100), margins=c(2,7), cexRow=0.6, cexCol=1, keysize=1.5,
        trace="none", symkey=T, srtCol=0,Colv=F,
        dendrogram="row", lhei = c(2,9.5), main="-log (p value)",
        labRow=Dataset$Gene, density.info="none")
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

### Heatmap plotting of the HR (Figure 1B)
windows(width=5, height=7)
heatmap.2(as.matrix(Dataset[,c(9:11)]), scale="column",
        col = bluered(100), margins=c(2,7), cexRow=0.6, cexCol=1, keysize=1.5,
        trace="none", symkey=T, srtCol=0,Colv=F,
        dendrogram="row", lhei = c(2,9.5), main="Hazard ratio",
        labRow=Dataset$Gene, density.info="none")
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

###Prepare the LUAD RNAseq dataset
#Clinical information (LUAD and LUSC, respectively)
to_browse<-XenaGenerate(subset = XenaHostNames=="tcgaHub") %>%
  XenaFilter(filterDatasets = "clinical") %>%
  XenaFilter(filterDatasets = "LUAD")
XenaBrowse(to_browse)
XenaBrowse(to_browse, type = "cohort")
dfclinic<-
fread("https://tcga.xenahubs.net/download/TCGA.LUAD.sampleMap/LUAD_clinicalMatrix.gz")
write.csv(dfclinic, file = "LUAD_clinic.csv",row.names=FALSE)

#RNASeq data (LUAD and LUSC, respectively)
BROWSE("http://github.com/mskcc/RNAseqDB/tree/master/data/normalized")
df1 <-fread("https://github.com/mskcc/RNAseqDB/raw/master/data/normalized/luad-rsem-fpkm-
tcga-t.txt.gz")
write.csv(df1, file = "luad-rsem-fpkm-tcga-t.csv",row.names=FALSE)
df2 <-fread("https://github.com/mskcc/RNAseqDB/raw/master/data/normalized/luad-rsem-fpkm-
tcga.txt.gz")
write.csv(df2, file = "luad-rsem-fpkm-tcga.csv",row.names=FALSE)
df3 <-fread("https://github.com/mskcc/RNAseqDB/raw/master/data/normalized/lung-rsem-fpkm-
gtex.txt.gz")
write.csv(df3, file = "lung-rsem-fpkm-gtex.csv",row.names=FALSE)

df1<-read.csv(file = "luad-rsem-fpkm-tcga-t.csv")
df1 = df1[,-2]
df2<-read.csv(file = "luad-rsem-fpkm-tcga.csv")
df2 = df2[,-2]
df3<-read.csv(file = "\\lung-rsem-fpkm-gtex.csv")
df3 = df3[,-2]

dfmerg<-list(df1,df2, df3) %>% reduce(full_join, by = "Hugo_Symbol")
write.csv(dfmerg, file = "LUAD_TCGA&GTEX.csv",row.names=FALSE)
dfmerg<-read.csv(file = "LUAD_TCGA&GTEX.csv",header=TRUE, sep=",", row.names=NULL)
dfmerg1<-names(dfmerg) %<>%substr(0, 15) # Correc the sample names
dfmerg1 <- dfmerg %>% setNames(gsub("\\.","-",names(.))) #Change "." to "_"

dfclinic<-read.csv(file ="LUAD_clinic.csv")
dfclinic1<-t(dfclinic)
dfclinic1<-data.frame(dfclinic1, stringsAsFactors = FALSE)
```

```
names(dfclinic1) <- as.matrix(dfclinic1[1,])
dfclinic1 = dfclinic1[c(-1),]
dfclinic1<-rownames_to_column(dfclinic1, var="Hugo_Symbol")

all_data <- rbind.fill(dfclinic1, dfmerg1) #combine clinic and rnaseq data
saveRDS(all_data, file="LUAD_TCGA&GTEX.rds")
dfsurv<-readRDS(file="LUAD_TCGA&GTEX.rds")
dfsurv<-t(dfsurv)
dfsurv<-data.frame(dfsurv, stringsAsFactors = FALSE)

names(dfsurv) <- as.matrix(dfsurv[1,])
dfsurv = dfsurv[c(-1),]
dfsurv<-rownames_to_column(dfsurv, var="sampleID")
dfsurv$sample_type[is.na(dfsurv$sample_type)] <- "GTEX Normal"
dfsurv$sample_type <- gsub('Primary Tumor', 'LUAD', dfsurv$sample_type)
dfsurv[is.na(dfsurv)] <- ""
saveRDS(dfsurv, file="LUAD_TCGA&GTEX_all.rds")

#Prepare the database for the top20 genes
dfsurv<-readRDS(file="LUAD_TCGA&GTEX_all.rds")
#list10<- Dataset %>% slice(1:20) %>% as.list(as.data.frame(t(Gene)))
list10$Gene
match(c("DDX11","DDX17","DDX18","DDX20","DDX25","DDX26B","DDX31",
    "DDX3Y","DDX46","DDX47","DDX49","DDX5","DDX50","DDX55","DDX56","DHX15",
    "DHX29","DHX32","DHX35","DHX40") ,colnames(dfsurv))
top20<-dfsurv[, c(1:143, 7195, 12888, 15083,  6007, 13269, 12756, 16620,
          14960, 14325, 19501, 7882, 13365, 13903, 7947, 15666, 12899,
          14406, 9765, 7415, 6268)]
DT1<-melt(top20, id.vars=c(colnames(top20)[1:143]))
DT1<-data.frame(DT1)
DT1<-DT1 %>% mutate(type=ifelse(grepl("Normal", sample_type), "Normal",
                  "Tumor"))
DT1<-DT1 %>%mutate(stage = case_when(grepl("T1", pathologic_T) ~ "T1",
                     grepl("T2", pathologic_T) ~ "T2",
                     grepl("T3", pathologic_T) ~ "T3",
                     grepl("T4", pathologic_T) ~ "T4",
                     TRUE ~ "TX"))
DT1<-DT1 %>%mutate(met = case_when(grepl("M0", pathologic_M) ~ "M0",
                    grepl("M1", pathologic_M) ~ "M1",
                    TRUE ~ "MX"))
DT1<-DT1 %>%mutate(smoking = case_when(grepl("<", tobacco_smoking_history_indicator) ~
"Pre<=15yrs",
                      grepl(">", tobacco_smoking_history_indicator) ~ "Pre>15yrs",
                      grepl("Current", tobacco_smoking_history_indicator) ~ "Current",
                      TRUE ~ "Non"))
DT1<-DT1 %>%mutate(age = case_when(
  age_at_initial_pathologic_diagnosis %in% 20:40 ~ "21-40",
  age_at_initial_pathologic_diagnosis %in% 40:60 ~ "41-60",
  age_at_initial_pathologic_diagnosis %in% 60:80 ~ "61-80",
  age_at_initial_pathologic_diagnosis %in% 80:100 ~ "81-100",
  T ~ ""))
```

```
write.xlsx(DT1, file="Top20 DDX in LUAD.xlsx", row.names=FALSE)

###Correlation matrix (Figure 4A&4B)
ddxdhxlist<-dfsurv[ , c(7195, 12888, 15083,  6007, 13269, 12756,
                      16620, 14960, 14325, 19501, 7882, 13365, 13903, 7947, 15666, 12899,
                      14406, 9765, 7415, 6268)]
windows(width=9, height=9)
list<-na.omit(ddxdhxlist)
corr <- round(cor(list), 1)
p.mat <- cor_pmat(list)
ggcorrplot(corr, hc.order = T, type = "lower", outline.col = "white",
          lab = T, insig = "blank", p.mat = p.mat, lab_size = 3, tl.cex=5)+
  font("xy.text", size = 8, color = "black", face = "bold")
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

###Differential expression plots (Figure 2, Figure 3, Sup. Fig. 1 and Sup. Fig. 2)
####################################
# Loop differential expression: tumor vs normal
#####################################
Dataset <- read.xlsx("Top20 DDX in LUAD.xlsx", sheet = 1)
#Dataset[,147]<-factor(Dataset[,147], levels = c("T1", "T2", "T3", "T4","TX"))
graphics.off()
plots <- list()  # new empty list
for(i in unique(Dataset$variable)) {
  windows(width=5.5, height=5.5)
  p1<-ggboxplot(subset(Dataset, variable==i), x = "type",
               y = "value", color="type", xlab="",
               ylab = "RFKM", add = "jitter",
               add.params = list(size = 0.1, jitter = 0.05),
               x.text.angle=90, outlier.size=0, palette = "lancet",
               title =paste0(i, ", LUAD"))
  p2<-p1+stat_compare_means(aes(label = paste0("p = ", ..p.format..)), size=2.5)+
    rremove("legend")+
    #theme(axis.text=element_text(size=rel(0.5), vjust=0.5, hjust=0))+
    #stat_summary(fun.data = n_fun, geom = "text", color="black")+
    #geom_quasirandom(method="tukey",cex=0.4)+
    #font("title", size = 6, color = "black", face = "bold")+
    theme(text = element_text(size = 4, face = "bold"))+
    theme(title = element_text(size = 6, face = "bold"))+
    theme(axis.text.x = element_text(size = 7, face = "bold"))
  #+scale_color_brewer(palette="Set1")
  #stat_compare_means(aes(label = paste0("p = ", ..p.format..)))
  #print(p2)
  #graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
  plots[[i]] <- p2
}
figure <- ggarrange(plotlist = plots, ncol = 5, nrow = 4)
print(figure)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

####################################
```

```
# Loop differential expression: stage
######################################
#Dataset <- read.xlsx("Top20 DDX in LUAD.xlsx", sheet = 1)
Dataset$stage_new<-factor(Dataset$stage_new, levels= c("I", "II", "III", "IV"))
graphics.off()
plots <- list()  # new empty list
for(i in unique(Dataset$variable)) {
  windows(width=5.5, height=5.5)
  p1<-ggboxplot(subset(Dataset, variable==i&stage_new!="[Discrpncy]"&stage_new!=""), x =
"stage_new",
              y = "value", color="stage_new", xlab="",
              ylab = "RFKM", add = "jitter", palette = "lancet",
              add.params = list(size = 0.1, jitter = 0.05),
              x.text.angle=90, outlier.size=0,
              title =paste0(i, ", LUAD"))
  p2<-p1+stat_compare_means(aes(label = paste0("p = ", ..p.format..)), size=2.5)+
    rremove("legend")+
    #theme(axis.text=element_text(size=rel(0.5), vjust=0.5, hjust=0))+
    #stat_summary(fun.data = n_fun, geom = "text", color="black")+
    #geom_quasirandom(method="tukey",cex=0.4)+
    #font("title", size = 6, color = "black", face = "bold")+
    theme(text = element_text(size = 4, face = "bold"))+
    theme(title = element_text(size = 6, face = "bold"))+
    theme(axis.text.x = element_text(size = 7, face = "bold"))
  #+scale_color_brewer(palette="Set1")
  #stat_compare_means(aes(label = paste0("p = ", ..p.format..)))
  #print(p2)
  #graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
  plots[[i]] <- p2
}
figure <- ggarrange(plotlist = plots, ncol = 5, nrow = 4)
print(figure)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

#####################################
# Loop differential expression: pathologic_N
######################################
#Dataset <- read.xlsx("Top20 DDX in LUAD.xlsx", sheet = 1)
Dataset$pathologic_N<-factor(Dataset$pathologic_N, levels = c("N0", "N1", "N2", "N3","NX"))
summary(Dataset$pathologic_N)
Dataset$pathologic_N<-na.omit(Dataset$pathologic_N)
gc()
graphics.off()
par(font=2)
plots <- list()  # new empty list
for(i in unique(Dataset$variable)) {
  windows(width=5.5, height=5.5)
  p1<-ggboxplot(subset(Dataset, variable==i&pathologic_N!="NX"), x = "pathologic_N",
              y = "value", color="pathologic_N", xlab="",
              ylab = "RFKM", add = "jitter",
              add.params = list(size = 0.1, jitter = 0.1),
              x.text.angle=90, outlier.size=0, palette = "lancet",
              title =paste0(i, ", LUAD"))
  p2<-p1+stat_compare_means(aes(label = paste0("p = ", ..p.format..)), size=2.5)+
    rremove("legend")+
    #theme(axis.text=element_text(size=rel(0.5), vjust=0.5, hjust=0))+
    #stat_summary(fun.data = n_fun, geom = "text", color="black")+
    #geom_quasirandom(method="tukey",cex=0.4)+
    #font("title", size = 6, color = "black", face = "bold")+
    theme(text = element_text(size = 4, face = "bold"))+
    theme(title = element_text(size = 6, face = "bold"))+
    theme(axis.text.x = element_text(size = 7, face = "bold"))
  #+scale_color_brewer(palette="Set1")
  #stat_compare_means(aes(label = paste0("p = ", ..p.format..)))
  #print(p2)
  #graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
  plots[[i]] <- p2
}
figure <- ggarrange(plotlist = plots, ncol = 5, nrow = 4)
print(figure)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

#####################################
# Loop differential expression: met
######################################
Dataset <- read.xlsx("Top20 DDX in LUAD.xlsx", sheet = 1)
Dataset$met<-factor(Dataset$met, levels = c("M0", "M1", "MX"))
Dataset$met<-as.factor(Dataset$met)
summary(Dataset$met)
Dataset$met<-na.omit(Dataset$met)
gc()
graphics.off()
par(font=2)
plots <- list()  # new empty list
for(i in unique(Dataset$variable)) {
  windows(width=5.5, height=5.5)
  p1<-ggboxplot(subset(Dataset, variable==i&met!="MX"), x = "met",
              y = "value", color="met", xlab="",
              ylab = "RFKM", add = "jitter",
              add.params = list(size = 0.1, jitter = 0.1),
              x.text.angle=90, outlier.size=0, palette = "lancet",
              title =paste0(i, ", LUAD"))
  p2<-p1+stat_compare_means(aes(label = paste0("p = ", ..p.format..)), size=2.5)+
    rremove("legend")+
    #theme(axis.text=element_text(size=rel(0.5), vjust=0.5, hjust=0))+
    #stat_summary(fun.data = n_fun, geom = "text", color="black")+
    #geom_quasirandom(method="tukey",cex=0.4)+
    #font("title", size = 6, color = "black", face = "bold")+
    theme(text = element_text(size = 4, face = "bold"))+
    theme(title = element_text(size = 6, face = "bold"))+
    theme(axis.text.x = element_text(size = 7, face = "bold"))
  #+scale_color_brewer(palette="Set1")
  #stat_compare_means(aes(label = paste0("p = ", ..p.format..)))
  #print(p2)
  #graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
  plots[[i]] <- p2
}
figure <- ggarrange(plotlist = plots, ncol = 5, nrow = 4)
print(figure)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

###################################
# Loop differential expression: smoking
####################################
# Dataset <- read.xlsx("Top20 DDX in LUAD.xlsx", sheet = 1)
Dataset$smoking<-factor(Dataset$smoking, levels = c("Non", "Current", "Pre<=15yrs", "Pre>15yrs"))
#Dataset$smoking<-as.factor(Dataset$smoking)
summary(Dataset$smoking)
Dataset$smoking<-na.omit(Dataset$smoking)
gc()
graphics.off()
par(font=2)
plots <- list()  # new empty list
for(i in unique(Dataset$variable)) {
  windows(width=5.5, height=5.5)
  p1<-ggboxplot(subset(Dataset, variable==i&smoking!=""), x = "smoking",
              y = "value", color="smoking", shape="smoking",xlab="",
              ylab = "RFKM", add = "jitter",
              add.params = list(size = 0.1, jitter = 0.1),
              x.text.angle=90, outlier.size=0, palette = "lancet",
              title =paste0(i, ", LUAD"))
  p2<-p1+stat_compare_means(aes(label = paste0("p = ", ..p.format..)), size=2.5)+
    rremove("legend")+
    #theme(axis.text=element_text(size=rel(0.5), vjust=0.5, hjust=0))+
    #stat_summary(fun.data = n_fun, geom = "text", color="black")+
    #geom_quasirandom(method="tukey",cex=0.4)+
    #font("title", size = 6, color = "black", face = "bold")+
    theme(text = element_text(size = 4, face = "bold"))+
    theme(title = element_text(size = 6, face = "bold"))+
    theme(axis.text.x = element_text(size = 7, face = "bold"))
  #print(p2)
  #graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
  plots[[i]] <- p2
}
figure <- ggarrange(plotlist = plots, ncol = 5, nrow = 4)
print(figure)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

###################################
# Loop differential expression: age
####################################
#Dataset <- read.xlsx("Top20 DDX in LUAD.xlsx", sheet = 1)
Dataset$age<-as.factor(Dataset$age)
```

```
summary(Dataset$age)
Dataset$age<-na.omit(Dataset$age)
gc()
graphics.off()
par(font=2)
plots <- list()  # new empty list
for(i in unique(Dataset$variable)) {
  windows(width=5.5, height=5.5)
  p1<-ggboxplot(subset(Dataset, variable==i&age!=""), x = "age",
              y = "value", color="age", xlab="",
              ylab = "RFKM", add = "jitter",
              add.params = list(size = 0.1, jitter = 0.1),
              x.text.angle=90, outlier.size=0, palette = "lancet",
              title =paste0(i, ", LUAD"))
  p2<-p1+stat_compare_means(aes(label = paste0("p = ", ..p.format..)), size=2.5)+
    rremove("legend")+
    #geom_quasirandom(method="tukey",cex=0.4)+
    #font("title", size = 6, color = "black", face = "bold")+
    theme(text = element_text(size = 4, face = "bold"))+
    theme(title = element_text(size = 6, face = "bold"))+
    theme(axis.text.x = element_text(size = 7, face = "bold"))
  plots[[i]] <- p2
}
figure <- ggarrange(plotlist = plots, ncol = 5, nrow = 4)
print(figure)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

####################################
# Loop differential expression: gender
#####################################
#Dataset <- read.xlsx("Top20 DDX in LUAD.xlsx", sheet = 1)
Dataset$gender<-as.factor(Dataset$gender)
summary(Dataset$gender)
Dataset$gender<-na.omit(Dataset$gender)
gc()
graphics.off()
par(font=2)
plots <- list()  # new empty list
for(i in unique(Dataset$variable)) {
  windows(width=5.5, height=5.5)
  p1<-ggboxplot(subset(Dataset, variable==i&gender!=""), x = "gender",
              y = "value", color="gender", xlab="",
              ylab = "RFKM", add = "jitter",
              add.params = list(size = 0.1, jitter = 0.1),
              x.text.angle=90, outlier.size=0, palette = "lancet",
              title =paste0(i, ", LUAD"))
  p2<-p1+stat_compare_means(aes(label = paste0("p = ", ..p.format..)), size=2.5)+
    rremove("legend")+
    theme(text = element_text(size = 4, face = "bold"))+
    theme(title = element_text(size = 6, face = "bold"))+
    theme(axis.text.x = element_text(size = 7, face = "bold"))
```

```
   plots[[i]] <- p2
}
figure <- ggarrange(plotlist = plots, ncol = 5, nrow = 4)
print(figure)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

####################################
# Loop differential expression: tobacco_smoking_history
#####################################
# Dataset <- read.xlsx("Top20 DDX in LUAD.xlsx", sheet = 1)
#Dataset$tobacco_smoking_history<-factor(Dataset$tobacco_smoking_history, levels = c("Non",
"Current", "Pre<=15yrs", "Pre>15yrs"))
Dataset$tobacco_smoking_history<-as.factor(Dataset$tobacco_smoking_history)
summary(Dataset$tobacco_smoking_history)
Dataset$tobacco_smoking_history<-na.omit(Dataset$tobacco_smoking_history)
gc()
graphics.off()
par(font=2)
plots <- list()  # new empty list
for(i in unique(Dataset$variable)) {
  windows(width=5.5, height=5.5)
  p1<-ggboxplot(subset(Dataset,
variable==i&tobacco_smoking_history!="[Discrepancy]"&tobacco_smoking_history!=""), x =
"tobacco_smoking_history",
                y = "value", color="tobacco_smoking_history", shape="tobacco_smoking_history",xlab="",
                ylab = "RFKM", add = "jitter",
                add.params = list(size = 0.1, jitter = 0.1),
                x.text.angle=90, outlier.size=0, palette = "lancet",
                title =paste0(i, ", LUAD"))
  p2<-p1+stat_compare_means(aes(label = paste0("p = ", ..p.format..)), size=2.5)+
    rremove("legend")+
    theme(text = element_text(size = 4, face = "bold"))+
    theme(title = element_text(size = 6, face = "bold"))+
    theme(axis.text.x = element_text(size = 7, face = "bold"))
  print(p2)
  plots[[i]] <- p2
}
figure <- ggarrange(plotlist = plots, ncol = 5, nrow = 4)
print(figure)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

####################################
# Loop differential expression: stage_c
#Dataset <- read.xlsx("Top20 DDX in LUAD.xlsx", sheet = 1)
Dataset[,147]<-factor(Dataset[,147], levels = c("T1", "T2", "T3", "T4","TX"))
graphics.off()
plots <- list()  # new empty list
for(i in unique(Dataset$variable)) {
  windows(width=5.5, height=5.5)
  p1<-ggboxplot(subset(Dataset, variable==i&stage_c!="TX"), x = "stage_c",
                y = "value", color="stage_c", xlab="",
            ylab = "RFKM", add = "jitter", palette = "lancet",
            add.params = list(size = 0.1, jitter = 0.05),
            x.text.angle=90, outlier.size=0,
            title =paste0(i, ", LUAD"))
  p2<-p1+stat_compare_means(aes(label = paste0("p = ", ..p.format..)), size=2.5)+
    rremove("legend")+
    theme(text = element_text(size = 4, face = "bold"))+
    theme(title = element_text(size = 6, face = "bold"))+
    theme(axis.text.x = element_text(size = 7, face = "bold"))
  plots[[i]] <- p2
}
figure <- ggarrange(plotlist = plots, ncol = 5, nrow = 4)
print(figure)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

##################
#P value table
##################
#Dataset <- read.xlsx("Top20 DDX in LUAD.xlsx", sheet = 1)
Pvalue1<-compare_means(value ~ type,  data = subset(Dataset), group.by=c("variable"),
method="wilcox.test")
Pvalue2<-compare_means(value ~ stage,  data = subset(Dataset, stage!="TX"),
group.by=c("variable"), method="wilcox.test")
Pvalue3<-compare_means(value ~ met,  data = subset(Dataset, met!="MX"), group.by=c("variable"),
method="wilcox.test")
Pvalue4<-compare_means(value ~ age,  data = subset(Dataset, age!=""), group.by=c("variable"),
method="wilcox.test")
Pvalue5<-compare_means(value ~ gender,  data = subset(Dataset, gender!=""),
group.by=c("variable"), method="wilcox.test")
Pvalue6<-compare_means(value ~ pathologic_N,  data = subset(Dataset,
                                      pathologic_N!="NX"&pathologic_N!=""), group.by=c("variable"),
method="wilcox.test")
Pvalue7<-compare_means(value ~ stage_c,  data = subset(Dataset, stage_c!="TX"),
group.by=c("variable"), method="wilcox.test")
Pvalue<-as.data.frame(mapply(c, Pvalue1, Pvalue2, Pvalue3, Pvalue4,Pvalue5, Pvalue6, Pvalue7))
write.xlsx(Pvalue, file="Top20 DDX in LUAD p value.xlsx", row.names=FALSE)

###Heatmap plotting of the top 20 DDX genes (Figure 4C)
Dataset <- read.xlsx("Top20 DDX in LUAD & LUSC p value.xlsx", sheet = 1)
Dataset<-type.convert (Dataset)
Dataset <- Dataset %>% mutate(nlogluad = -log10(LUAD), nloglusc = -log10(LUSC))

windows(width=5, height=5)
par(font=2)
heatmap.2(as.matrix(Dataset[,c(4:5)]), cellnote=round(Dataset[,c(4:5)],2),
       notecex=1.0, notecol="black", scale="column", col = bluered(100), margins=c(2,7),
       cexRow=0.8, cexCol=1.5, keysize=1.5, trace="none",
       symkey=T, density.info="none", srtCol=0,Colv=F,
       dendrogram="row", lhei = c(2,7), main="-log (p value)",
       labRow=as.expression(lapply(Dataset[,1], function(a) bquote(bold(.(a))))),
       distfun = function(x) dist(x, method="euclidean"),
        hclustfun = function(x) hclust(x, method="ward.D2"))
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

#Radar chart (Figure 4D)
data <- read.xlsx("Top20 DDX in LUAD & LUSC p value.xlsx", sheet = 2)
colnames(data)[1]<-"group"
data[] <- data.frame(lapply(data, function(x)
  type.convert(as.character(x), as.is = T)))
#str(data)
#summary(data)
cols<-c("LUAD-TUMOR", "LUAD-T", "LUAD-N", "LUAD-M", "LUSC-TUMOR", "LUSC-T",
      "LUSC-N", "LUSC-M")
data<-setDT(data)[, paste0("nlog_", cols) := lapply(.SD, function(x) -log10(x)), .SDcols = cols]
#write.xlsx(data2, "Data for radarchart.xlsx", rowNames = F)
data2<-data[, c(1, 10:17)] %>% mutate_each(funs(rescale), -group)
ggradar(data2, group.point.size = 4, axis.label.size = 3, group.line.width = 1,
      grid.line.width = 0.3,background.circle.colour = "white",
      legend.text.size = 10, legend.position = "bottom")
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

###GSEA analysis (Figure 5)
rank1 <- read.csv("LUAD volcano DDX5.csv",stringsAsFactors=FALSE)
rank2<-rank1[,c(1,8)]
ranks<-deframe(rank2)
#barplot
graphics.off()
windows(width=2.5, height=2.5)
par(mar=c(3,2,2,2))
barplot(sort(ranks, decreasing = T), cex.names=0.4, cex.axis=0.4,
      cex.main=0.6, font = 2, main="LUAD DDX5")
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
#fileNames=c("h.all.v7.0.symbols.gmt",
#         "c5.all.v7.0.symbols.gmt",
#         "c5.mf.v7.0.symbols.gmt",
#         " c5.bp.v7.0.symbols.gmt",
#         " c5.cc.v7.0.symbols.gmt")
fileNames <- Sys.glob("*.gmt")
for (i in fileNames){
  pathways <- gmtPathways(i)
  str(head(pathways))
  #Peform fgsea
  fgseaRes <- fgsea(pathways, ranks, minSize=15, maxSize=100, nperm=1000)
  fgseaRes<-subset(fgseaRes, pathway!="HALLMARK_PANCREAS_BETA_CELLS")
  head(fgseaRes[order(padj, -abs(NES)), ], n=10)
  pathsig<-(subset(fgseaRes, pval < 0.05&pathway!="HALLMARK_PANCREAS_BETA_CELLS"))
  pathsig<-pathsig[order(pval),]
  #Plot gsea
  windows(width=4, height=4)
  #barplot(sort(ranks, decreasing = T))
  p1<-plotEnrichment(pathways[[head(fgseaRes[order(pval), ], 1)$pathway]],
                 ranks) +
    labs(title=head(fgseaRes[order(pval), ], 1)$pathway, adjust = TRUE)+
    geom_text(data=head(fgseaRes[order(pval), ], 1), aes(x=15000, y=0.2,
                                              label=paste0("LUAD DDX5, p=", round(pval,4), ", NES=",
round(NES,4))), size=3)
  print(p1)
  graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
  #plotEnrichment(pathways[["HALLMARK_MYC_TARGETS_V2"]], rank2)
  #Table top10 bottom 10
  topPathwaysUp <- fgseaRes[ES > 0&pval<0.05][head(order(pval), n=5), pathway]
  topPathwaysDown <- fgseaRes[ES < 0&pval<0.05][head(order(pval), n=5), pathway]
  topPathways <- c(topPathwaysUp, rev(topPathwaysDown))
  windows(width=4, height=4)
  p2<-plotGseaTable(pathways[topPathways], ranks, fgseaRes,
                gseaParam = 0.5)
  print(p2)
  graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
}

### Gene function pathway profiling analysis (Figure 6)
#genelist preparation
dfid<-fread("luad-rsem-fpkm-tcga.txt.gz")
dfid<-dfid[,c(1:2)]
colnames(dfid) <- c("Gene","ID")
#Pathway analysis for individual DDXs in LUAD and LUSA, respectively
df<-read_excel("Sup Table 2_pathview volcano pooled data.xlsx",  sheet=1) # Repeat with 8 sheets
#df1<- df %>% rownames_to_column("gene")
df1<-df %>% mutate(fc=high/low) #%>%    arrange(fc)
df2<-df1[c(1:100,19534:19633), c(1,9) ]
df3 <- merge(df2,dfid, by="Gene")
geneList = df3[,2]
names(geneList) = as.character(df3[,3])
geneList = sort(geneList, decreasing = TRUE)
de <- names(geneList)
edo <- enrichDGN(de)
x <- enrichPathway(gene=de,pvalueCutoff=0.05, readable=T)
head(as.data.frame(x))
X11(w=10, h=7)
cnetplot(x, categorySize="pvalue", foldChange=geneList, circular = TRUE, colorEdge = TRUE)+  #Most
important!
  ggplot2::labs(title = "DDX in LU")
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
#Multiple comparison
dfgc<-read.csv(file = "Pooled DDX rank 200 dif gene list.csv", row.names=1) #Pooled from 8 groups.
res <- compareCluster(dfgc, fun="enrichPathway")
X11()
dotplot(res)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

###MICRORNA analysis (Figure 7)
#View(showTCGA())
#cli<-fread("https://tcga.xenahubs.net/download/TCGA.LUAD.sampleMap/LUAD_clinicalMatrix")
```

```
#exp_gene<-downloadTCGA(project = "LUAD", data_type = "Gene Expression RNASeq", file_type =
"IlluminaHiSeq RNASeqV2", destdir = "J:\\TCGA")
#exp_gene<-fread("J:\\TCGA\\TCGA.LUAD.sampleMap\\HiSeqV2.gz")
#exp_target<-exp_gene %>% dplyr::filter(sample %in% c("DDX11","DDX5","DDX55","DDX56",
"DHX36"))
#t_exp = setNames(data.frame(t(exp_target[,-1])),exp_target[,1])
#t_exp <- t_exp %>%rownames_to_column(var = "sample")
#df<-read.csv(file="t_exp.csv", row.names=1)
#df1 <- df %>% mutate(DDX5_level = if_else(DDX5 >= median(df$DDX5), 'High', 'Low'))
#df1 <- df1 %>% mutate(DDX11_level = if_else(DDX11 >= median(df$DDX11), 'High', 'Low'))
#df1 <- df1 %>% mutate(DDX55_level = if_else(DDX55 >= median(df$DDX55), 'High', 'Low'))
#df1 <- df1 %>% mutate(DDX56_level = if_else(DDX56 >= median(df$DDX56), 'High', 'Low'))
#df1 <- df1 %>% mutate(DHX36_level = if_else(DHX36 >= median(df$DHX36), 'High', 'Low'))
#df2<-subset(df1[, c(1, 7:11)])
#df2<-type.convert(df2)
#write.csv(df2, file="TCGA DDX exp levels.csv")

#Load DDX level file
dflevel<-read.csv(file="TCGA DDX exp levels.csv", row.names=1)

#Load MIRNA data from Xena
mir_gene<-
fread("https://tcga.xenahubs.net/download/TCGA.LUAD.sampleMap/miRNA_HiSeq_gene.gz")
head(mir_gene[1:5, 1:5])
mir_gene<-type.convert(mir_gene)
t_mir <- t(mir_gene)
colnames(t_mir) <- as.character(t_mir[1,])
t_mir <- t_mir[-1,]
t_mir <- t_mir %>%data.frame() %>% rownames_to_column(var = "sample")

#Merge mir aarray with the ddx level file
df<-merge(t_mir, dflevel, by="sample")
write.csv(df, file="TCGA exp MIRNA merged data.csv")

#Build the table of gouped mean
df<-read.csv(file="TCGA exp MIRNA merged data.csv", row.names=1, stringsAsFactors=FALSE)
df1 <- df %>% select_if(.predicate=funs(sum(is.na(.))<=442)) #select n>5%

#create individual data frame for lgfc and pval
dfmean<- aggregate(df1[, -1], by=list(df$DDX56_level), FUN = mean, na.rm=TRUE)
dfsd<-aggregate(df1[, -1], by=list(df$DDX56_level), FUN = sd, na.rm=TRUE)
dfmerge<-rbind(dfmean, dfsd)
dfvol = setNames(data.frame(t(dfmerge[,-1])),dfmerge[,1])
dfvol <- dfvol %>%rownames_to_column(var = "sample") %>% na.omit()
names(dfvol)[2:5] <- c("meanhigh","meanlow","sdhigh","sdlow")
dfvol <- dfvol %>% mutate(log2fc = log2(meanhigh)-log2(meanlow)) %>%
  arrange(desc(log2fc))
dftop<-dfvol %>% top_n(20)
dfbot<-dfvol %>% top_n(-20)
dftb20<-rbind(dftop, dfbot)
#Setect the df for t.test
```

```
dflist<-dplyr::pull(dftb20, sample)
dflist <- paste(dflist, collapse = "|")
dftest<- df1%>% select(c(sample, matches(dflist), DDX56_level))
dfpv<-dftest %>% summarise_each(funs(t.test(.[DDX56_level == "High"],.[DDX56_level ==
"Low"],na.rm=TRUE)$p.value), vars=c(2:41))
tdfpv<-as.data.frame(t(dfpv)) %>% rename(pval = V1)
dfmir<-data.frame(dftb20, tdfpv, check.rows = FALSE) %>%
  mutate(padj = p.adjust(dfpv,method="BH")) %>%
  mutate(nlogpadj = -log10(padj)) %>%
  mutate(rank = nlogpadj/log2fc) %>%
  arrange(pval) %>%
  rename(DDX56 = sample)
write.csv(dfmir, file="DDX56 MIR rank40 lgfc pval_new2.csv",row.names=FALSE)
#ggbarplot
x11(w=3, h=4)
ggbarplot(subset(dfmir,pval<0.05), x = "DDX56", y = "log2fc", x.text.angle = 90,
        font.tickslab=c(8), fill="pval", orientation = "horiz", sort.val = c("desc"),
        legend=c("right"), main="DDX56")
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

#Venn diagram plot
dataset <- read_excel("DDX all MIR rank20 lgfc pval.xlsx")
dataset<-as.data.frame(dataset)
dataset[is.na(dataset)]<-""
dflist <- list(DDX5=dataset$DDX5, DDX56=dataset$DDX56,
             DDX11=dataset$DDX11, DDX55=dataset$DDX55)
x11()
venn.plot <- venn.diagram(dflist, NULL, fill = c("cornflowerblue", "green", "yellow", "darkorchid1"),
                     alpha = 0.50, main = "miRNA Venn Diagram")
grid.draw(venn.plot)
graph2ppt(file=paste0(Sys.Date(), "_miRNA plot",".pptx"), append =TRUE)
# To get the list of gene present in each Venn compartment
dfshare <- venn(dflist, show.plot=FALSE)
str(dfshare)
dfshare

#heatmap rank 40 each
mirddx5<-fread("DDX5 MIR rank40 lgfc pval_new2.csv")
mirddx5<-mirddx5 %>% rename(sample=DDX5, DDX5_lg2fc = log2fc)
mirddx11<-fread("DDX11 MIR rank40 lgfc pval_new2.csv")
mirddx11<-mirddx11 %>% rename(sample=DDX11, DDX11_lg2fc = log2fc)
mirddx55<-fread("DDX55 MIR rank40 lgfc pval_new2.csv")
mirddx55<-mirddx55 %>% rename(sample=DDX55, DDX55_lg2fc = log2fc)
mirddx56<-fread("DDX56 MIR rank40 lgfc pval_new2.csv")
mirddx56<-mirddx56 %>% rename(sample=DDX56, DDX56_lg2fc = log2fc)
mirmap<-list(mirddx5, mirddx11, mirddx55, mirddx56) %>% reduce(full_join, by = "sample")
fwrite(mirmap, file="DDX miR rank40 merged heatmap data.csv")
#Heatmap plot
mirmap<-fread("DDX miR rank40 merged heatmap data.csv")
mirmap2<-mirmap %>% filter(dense_rank(P.Value.x) <= 50|dense_rank(P.Value.y) <= 50 |
                       dense_rank(P.Value.x.x) <= 50|dense_rank(P.Value.y.y) <= 50)
```

```
mirmap[is.na(mirmap)]<-0
mirmap2<-mirmap[,c(1,6,15,24,33)]
x11(width=5, height=7)
heatmap.2(as.matrix(mirmap2[,c(2:5)]), scale="column", col = bluered(100), margins=c(3.5,5),
        cexRow=0.5, cexCol=0.6, keysize=3, trace="none", symkey=T,
        density.info="none", Colv=T, dendrogram="both", lhei = c(2,7),
        main="lg2fc", distfun = function(x) dist(x, method="euclidean"),
        hclustfun = function(x) hclust(x, method="ward.D2"), labRow=mirmap$sample,
        srtCol=45)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)

###Mutation (Figure 8A and 8B)
#Load mutation data from Xena
mut_gene<-
fread("https://tcga.xenahubs.net/download/mc3_gene_level/LUAD_mc3_gene_level.txt.gz")
#Ref: Chen 2013.GMB
mut_target <- mut_gene %>% dplyr::filter(sample %in%
                                   c("EGFR","KRAS","TP53", "ALK", "CDKN2A", "STK11", "BRAF","PIK3CA",
"RB1","ERBB2",
                                    "PTEN", "NFE2L2", "MET","CTNNB1","NRAS", "ROS1","RET","NF1","ERBB4",
"MAP2",
                                    "AKT1", "MAP2K1","PTPN11","CBL", "FGFR4", "MUC16", "SMARCA4",
"ATM", "NTRK3", "HRAS"))
t_mut = setNames(data.frame(t(mut_target[,-1])),mut_target[,1])
t_mut <- t_mut %>%rownames_to_column(var = "sample")

#Merge selected expression and mutation data
df<-merge(t_exp, t_mut, by="sample")
write.csv(df, file="TCGA exp mut merged data.csv")

#Set the levels of DDX
df<-fread(file="TCGA exp mut merged data.csv")
df1 <- df %>% mutate(DDX5_level = if_else(DDX5 >= median(df$DDX5), 'High', 'Low'))
df1 <- df1 %>% mutate(DDX11_level = if_else(DDX11 >= median(df$DDX11), 'High', 'Low'))
df1 <- df1 %>% mutate(DDX55_level = if_else(DDX55 >= median(df$DDX55), 'High', 'Low'))
df1 <- df1 %>% mutate(DDX56_level = if_else(DDX56 >= median(df$DDX56), 'High', 'Low'))
df1 <- df1 %>% mutate(DHX36_level = if_else(DHX36 >= median(df$DHX36), 'High', 'Low'))
df1<-type.convert(df1)
lapply(df1, class)
write.csv(df1, file="TCGA exp mut merged data_cat.csv")

#mirror bar plot wtih frequency
df1<-fread(file="TCGA exp mut merged data_cat.csv")
df1<-df1[,c(-1,-2, -4:-8)]
match(c("DDX5_level"), colnames(df1))
df1<-type.convert(df1)
sapply(df1, class)
df2<-melt(df1, id=c(1, 32:36))  #restructure
df2$value<-factor(df2$value)    #numeric to factor
df3<-df2 %>% rename( gene=variable, mutation=value)  #rename
```

```
#Select the first 20 most mutated genes
dfrank<-df3 %>% group_by(gene, mutation, add = TRUE) %>%
  summarise(counts = n()) %>% arrange(mutation, desc(counts))
dfrank<-dfrank[c(31:50, 11:30),]
t20<-vapply(dfrank$gene, paste, collapse = ", ", character(1L))
df4<-filter(df3, gene %in% t20)
write.csv(df4, file="TCGA exp mut merged data_top20.csv")

#Add a coumn of percentage
df4<-read.csv(file="TCGA exp mut merged data_top20.csv")
dfDDX5 <- df4 %>%
  group_by(DDX5_level, gene, mutation, add = TRUE) %>%
  summarise(counts = n()) %>%
  filter(mutation=="1") %>%
  mutate(freq=100*counts/509) %>%
  mutate(freq_s = if_else(DDX5_level == "Low", -freq, freq)) %>%
  mutate(group="DDX5") %>%
  rename(level = 1)

dfDDX11 <- df4 %>%
  group_by(DDX11_level, gene, mutation, add = TRUE) %>%
  summarise(counts = n()) %>%
  filter(mutation=="1") %>%
  mutate(freq=100*counts/509) %>%
  mutate(freq_s = if_else(DDX11_level == "Low", -freq, freq)) %>%
  mutate(group="DDX11") %>%
  rename(level = 1)

dfDDX55 <- df4 %>%
  group_by(DDX55_level, gene, mutation, add = TRUE) %>%
  summarise(counts = n()) %>%
  filter(mutation=="1") %>%
  mutate(freq=100*counts/509) %>%
  mutate(freq_s = if_else(DDX55_level == "Low", -freq, freq)) %>%
  mutate(group="DDX55") %>%
  rename(level = 1)

dfDDX56 <- df4 %>%
  group_by(DDX56_level, gene, mutation, add = TRUE) %>%
  summarise(counts = n()) %>%
  filter(mutation=="1") %>%
  mutate(freq=100*counts/509) %>%
  mutate(freq_s = if_else(DDX56_level == "Low", -freq, freq)) %>%
  mutate(group="DDX56") %>%
  rename(level = 1)

dfplot<-rbind(dfDDX5, dfDDX11, dfDDX55, dfDDX56)
write.csv(dfplot, file="TCGA exp mut merged data_frequency.csv")

#Back-to-back bar plot
dfplot<-read.csv(file="TCGA exp mut merged data_frequency.csv", row.names = 1)
```

```
dev.new()
for (i in unique(dfplot$group)){
  dev.new()
  p <-
    dfplot %>%
    filter(group==i) %>%
    ggplot(aes(x = gene, y = freq_s, group = level, fill = level)) +
    geom_bar(stat = "identity", width = 0.75) +
    coord_flip() +
    #scale_x_discrete(limits = the_order) +
    # another trick!
    scale_y_continuous(breaks = seq(-50, 50, 10),
                       labels = abs(seq(-50, 50, 10))) +
    labs(x = "Gene", y = "%", title = paste0(i)) +
    theme(legend.position = "bottom",
          legend.title = element_blank(),
          plot.title = element_text(hjust = 0.5),
          panel.background = element_rect(fill =  "grey90")) +
    scale_fill_manual(values=c("red", "blue"),
                      name="",
                      breaks=c("High", "Low"),
                      labels=c("High", "Low"))
  print(p)
  graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
}

###Facet bar plot
#Creat a list of order
dfplot<-read.csv(file="TCGA exp mut merged data_frequency.csv", row.names = 1)
dp1<-dfplot %>% arrange(gene, group, desc(level)) %>%
  mutate(dif=c(0, diff(counts))) %>%
  mutate(dif_s= ifelse((row_number() %% 2) == 1, NA, dif))
dfgene<-subset(dp1, group!="DHX36") %>% arrange(level, desc(freq)) %>%
  distinct(gene)
t20<- vapply(dfgene[,1], paste, collapse = ", ", character(1L))
#barplot with facet_grid and order
dp1$group <- factor(dp1$group, levels = c("DDX5", "DDX11", "DDX55", "DDX56", "DDX36"))
dp1$level <- factor(dp1$level, levels = c("Low", "High"))
dev.new()
p1 <- subset(dp1, group!="DHX36") %>%
  arrange(level, desc(freq)) %>%
  ggbarplot(x="gene", y="freq", color="level", fill="level", x.text.angle=90,
            position = position_dodge(0.9), font.tickslab=c(8),
            palette = "aaas", order=t20)
p2<-p1+ facet_grid(group ~ .)
print(p2)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
write.csv(dp1, file="Sup Table_mutation vs DDX.csv")
X11(w=4,h=4)
p3<-ggbarplot(na.omit(subset(dp1, group!="DHX36")), x="gene", y="dif_s", color="group",
order=t20,
          x.text.angle=90,palette = "npg", shape="group", fill="group", position = position_dodge(0.8))
print(p3)
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)


###Pan-cancer cell line gene expression figures (Figure 9A to 9D)
# Heatmap plots of pan-cancer cell line gene expression
#browseURL("http://cellexpress.cgm.ntu.edu.tw") #data source
data<-read_excel("Cell line Microarray gene expression_figure.xlsx")
data$Gene_symbol<-factor(data$Gene_symbol, levels =unique(data$Gene_symbol), ordered=T)
graphics.off()
for (i in unique(data$Gene_symbol)){
  windows(w=3.8, h=3)
  p1<-ggboxplot(subset(data, Gene_symbol==i), x = "Cell_Line_name", y = "value",
            fill="Primary_site",       # change fill color by ...
            color = "black",           # Set bar border colors
            palette = "jco",           # jco journal color palett. see ?ggpar
            sort.val = "desc",         # Sort the value in dscending order
            sort.by.groups = T,        # Don't sort inside each group
            x.text.angle = 90,
            xlab="", ylab="Expresssion value",
            font.tickslab = c("bold","4", "black"),
            title =paste0(i),
            font.legend =c("bold","3"), legend.title=""
  )
  p2<-p1+geom_hline(aes(yintercept = mean(value)), colour = 'red')+
    theme(axis.text.x=element_text(vjust=0.2, hjust=1))
  print(p2)
  graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
}

###Heatamap of lung cancer cell line expression (Figure 9E)
#browseURL("https://www.ebi.ac.uk/gxa/experiments") #data source
df1<-read.table("DDX cell line expression E-MTAB-2770-query-results.tsv", sep = "\t",
na.strings=c("", "NA"))
df2<-as.data.frame(t(df1))
names(df2) <- as.matrix(df2[2,])
df2 = df2[c(-1,-2),-6]
df2<-type.convert(df2)
sapply(df2, class)
windows(width=5, height=5.5)
heatmap.2(as.matrix(df2[,c(2:5)]), scale="column", col = bluered(100), margins=c(2,11),
        cexRow=0.5, cexCol=0.6, keysize=2.5, trace="none", symkey=T,
        density.info="none", srtCol=0, Colv=T, dendrogram="both", lhei = c(2,7),
        main="KFPM", labRow=df1[,1],
        distfun = function(x) dist(x, method="euclidean"),
        hclustfun = function(x) hclust(x, method="ward.D2"))
graph2ppt(file=paste0(Sys.Date(), "_plot",".pptx"), append =TRUE)
```

```
> sessionInfo()
R version 3.6.3 (2020-02-29)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 17763)

Matrix products: default

locale:
  [1] LC_COLLATE=English_United Kingdom.1252  LC_CTYPE=English_United 
Kingdom.1252   
[3] LC_MONETARY=English_United Kingdom.1252 LC_NUMERIC=C                           
[5] LC_TIME=English_United Kingdom.1252    

attached base packages:
  [1] grid      stats4    parallel  stats     graphics  grDevices utils     
datasets  methods  
[10] base     

other attached packages:
  [1] xtable_1.8-4               VennDiagram_1.6.20         
futile.logger_1.4.3       
[4] vctrs_0.3.1                UCSCXenaTools_1.3.1        forcats_0.4.0             
[7] stringr_1.4.0              dplyr_0.8.5                purrr_0.3.3               
[10] readr_1.3.1                tidyr_1.0.2                tibble_3.0.1              
[13] tidyverse_1.3.0            survminer_0.4.6            reshape2_1.4.3            
[16] readxl_1.3.1               RColorBrewer_1.1-2         
ReactomePA_1.30.0         
[19] randomForestSRC_2.9.3      R.utils_2.9.2              R.oo_1.23.0               
[22] R.methodsS3_1.8.0          psych_1.9.12.31            
PerformanceAnalytics_2.0.4
[25] xts_0.12-0                 zoo_1.8-8                  openxlsx_4.1.4            
[28] magrittr_1.5               limma_3.42.0               httr_1.4.1                
[31] Hmisc_4.4-0                Formula_1.2-3              survival_3.1-8            
[34] lattice_0.20-38            hgu133plus2.db_3.2.3       
org.Hs.eg.db_3.10.0       
[37] gtools_3.8.1               gplots_3.0.3               glmnet_3.0-2              
[40] Matrix_1.2-18              ggpubr_0.3.0               ggcorrplot_0.1.3          
[43] ggbeeswarm_0.6.0           ggplot2_3.3.1              
GenomicFeatures_1.38.0    
[46] GenomicRanges_1.38.0       GenomeInfoDb_1.22.0        gcrma_2.58.0              
[49] fgsea_1.12.0               Rcpp_1.0.4                 
export_0.2.2.9000         
[52] data.table_1.12.8          clusterProfiler_3.14.3     cgdsr_1.3.0               
[55] biomaRt_2.42.0             beeswarm_0.2.3             
AnnotationDbi_1.48.0      
[58] IRanges_2.20.2             S4Vectors_0.24.3           affy_1.64.0               
[61] Biobase_2.46.0             BiocGenerics_0.32.0       

loaded via a namespace (and not attached):
  [1] rappdirs_0.3.1              rtracklayer_1.46.0          acepack_1.4.1              
[4] bit64_0.9-7                 knitr_1.28                  
DelayedArray_0.12.2        
[7] rpart_4.1-15                RCurl_1.98-1.1              generics_0.0.2             
[10] preprocessCore_1.48.0       lambda.r_1.2.4              cowplot_1.0.0              
[13] RSQLite_2.2.0               europepmc_0.3               bit_1.1-15.2               
[16] enrichplot_1.6.1            lubridate_1.7.4             webshot_0.5.2              
[19] xml2_1.2.2                  httpuv_1.5.2                
SummarizedExperiment_1.16.1
[22] assertthat_0.2.1            viridis_0.5.1               xfun_0.12                  
[25] hms_0.5.3                   evaluate_0.14               promises_1.1.0             
```

```
[28] fansi_0.4.1                 progress_1.2.2              caTools_1.18.0
[31] dbplyr_1.4.4                km.ci_0.5-2                 igraph_1.2.4.2
[34] DBI_1.1.0                   htmlwidgets_1.5.1           ellipsis_0.3.0
[37] crosstalk_1.1.0.1           backports_1.1.5             abind_1.4-5
[40] withr_2.1.2                 ggforce_0.3.1
triebeard_0.3.0
[43] checkmate_2.0.0             GenomicAlignments_1.22.1
prettyunits_1.1.1
[46] mnormt_1.5-6                cluster_2.1.0               DOSE_3.12.0
[49] crayon_1.3.4                pkgconfig_2.0.3             tweenr_1.0.1
[52] nlme_3.1-144                vipor_0.4.5                 nnet_7.3-12
[55] rlang_0.4.6                 lifecycle_0.2.0             miniUI_0.1.1.1
[58] affyio_1.56.0               BiocFileCache_1.10.2        modelr_0.1.6
[61] cellranger_1.1.0            polyclip_1.10-0
matrixStats_0.56.0
[64] flextable_0.5.9             graph_1.64.0                urltools_1.7.3
[67] KMsurv_0.1-5                carData_3.0-3               reprex_0.3.0
[70] base64enc_0.1-3             ggridges_0.5.2              png_0.1-7
[73] viridisLite_0.3.0           bitops_1.0-6
KernSmooth_2.23-16
[76] Biostrings_2.54.0           blob_1.2.1                  rgl_0.100.50
[79] shape_1.4.4                 qvalue_2.18.0
manipulateWidget_0.10.1
[82] jpeg_0.1-8.1                rstatix_0.6.0
gridGraphics_0.5-0
[85] ggsignif_0.6.0              reactome.db_1.70.0          scales_1.1.0
[88] graphite_1.32.0             memoise_1.1.0               plyr_1.8.5
[91] gdata_2.18.0                zlibbioc_1.32.0             compiler_3.6.3
[94] tinytex_0.20                cli_2.0.2
Rsamtools_2.2.1
[97] XVector_0.26.0              formatR_1.7
htmlTable_1.13.3
[100] MASS_7.3-51.5               tidyselect_0.2.5            stringi_1.4.6
[103] GOSemSim_2.12.0             askpass_1.1
latticeExtra_0.6-29
[106] ggrepel_0.8.2               survMisc_0.5.5
fastmatch_1.1-0
[109] tools_3.6.3                 rio_0.5.16                  rvg_0.2.1
[112] rstudioapi_0.11             uuid_0.1-2                  foreach_1.4.8
[115] foreign_0.8-75              gridExtra_2.3
stargazer_5.2.2
[118] farver_2.0.3                ggraph_2.0.2                digest_0.6.24
[121] rvcheck_0.1.8               BiocManager_1.30.10         shiny_1.4.0.2
[124] quadprog_1.5-8              car_3.0-8                   broom_0.5.6
[127] later_1.0.0                 gdtools_0.2.1
colorspace_1.4-1
[130] rvest_0.3.5                 fs_1.3.1                    XML_3.99-0.3
[133] splines_3.6.3               graphlayouts_0.6.0
ggplotify_0.0.5
[136] systemfonts_0.1.1           futile.options_1.0.1
jsonlite_1.6.1
[139] tidygraph_1.1.2             R6_2.4.1                    pillar_1.4.3
[142] htmltools_0.4.0             mime_0.9                    glue_1.3.1
[145] fastmap_1.0.1               BiocParallel_1.20.1
codetools_0.2-16
[148] curl_4.3                    officer_0.3.6               zip_2.0.4
[151] GO.db_3.10.0                openssl_1.4.1               rmarkdown_2.1
[154] munsell_0.5.0               DO.db_2.9
GenomeInfoDbData_1.2.2
[157] iterators_1.0.12            haven_2.2.0                 gtable_0.3.0
```