suppressPackageStartupMessages(library("optparse"))

suppressPackageStartupMessages(library("stats"))

usage = "The prog is used to display Violin of data,the Violin display density distribution of data,and a line to show the median data.

It's used to compare several distributions while they're placed side by side.A violin plot is a kernel density estimate,mirrored so that it forms a symmetrical shape.

Rscript %prog -f -t -n

example: Rscript %prog

-f CLIP\_Peak\_len

-t CLIP\_Peak\_len

-n CLIP\_Peak\_len

"

option\_list <- list(

make\_option(c("-f", "--file"),action = "store",type = "character",

help = "The Input file"),

make\_option(c("-t","--title"),action = "store",type = "character",

help = "The title of the plot"),

make\_option(c("-x","--xaxis"),action = "store",type = "character",default = "Location",

help = "The name of x-axis "),

make\_option(c("-y","--yaxis"),action = "store",type = "character",default = "Frequency of each base(%)",

help = "The name of y-axis"),

make\_option(c("-n", "--filename"),action = "store",type = "character",

help = "The name of outimage"),

make\_option(c("-o", "--outdir"),action = "store",type = "character",default = "./",

help = "The outdir")

)

opt <- parse\_args(OptionParser(option\_list = option\_list))

setwd(opt$outdir) # Set the Outpath

library(ggplot2)

library(reshape2)

library(plotrix)

library(methods)

library(gtable)

library(grid)

library(RColorBrewer)

###################################################################################

colour <- c('#85A2EF','#D285EF','#A2EF85','#4682B4','#A0522D','#87CEEB','#6B8E23','#6A5ACD','#E59B95','#EFD285','#B4B643','#2E9AFE','#A1DDBB','#FF8C00')

# colour1 <- brewer.pal(8,"Dark2")

#######data format sample###########################################################

# Type Peaks

# CLIP\_overlap 351

# CLIP\_overlap 111

# CLIP\_overlap 226

# CLIP\_overlap 316

# CLIP\_overlap 316

# CLIP\_overlap 486

# CLIP\_sp 346

# CLIP\_sp 321

# CLIP\_sp 91

# CLIP\_sp 291

# CLIP\_sp 236

# CLIP\_sp 216

# CLIP\_sp 191

#####################################################################################

######Instruction for data#######

#first column is the X axis---discrete data

#second column is the Y axis to plot -- continuous data

#the data is to display the summary about X axis data in density.

#################################

###################################################################################

Deal with Data）

data <- read.table(file= opt$file,header = T)##### must with header

title <- gsub('\_',' ',opt$title)

colname <- colnames(data)

dim\_data <- dim(data)

Item <- factor(data[,1],levels = data[,1])

x\_axis = gsub('\_',' ',opt$xaxis)

y\_axis = gsub('\_',' ',opt$yaxis)

##################################################################################

###Plot Theme for ABLife

###theme(),Tha last term without comma

##################################################################################

ablife\_theme\_line <- function(base\_size = 12){

library(grid) ####for using unit function

theme(

plot.title = element\_text(size = 12,lineheight = 100,colour = "black",hjust = 0.5),

# axis.text.x=element\_text(angle=75,hjust=1,size = 5,colour = "black"),

axis.title.x = element\_text(size = 12,colour = "black"),

axis.text.y = element\_text(size = 12 ,colour = "black"),

axis.title.y = element\_text(size = 12,colour = "black"),

panel.background = element\_rect(colour = "black"),

legend.title = element\_text(size = 12),

legend.text = element\_text(size = 12),

strip.text.x = element\_text(colour = "black",size = "8"),

strip.background = element\_rect(colour = "black")

)

}

##################################################################################

###Plot by ggplot2

ggplot(data)+

geom\_violin(aes(x = Item,y=data[,2],stat = "identity",fill=Type))+ labs(title = title,y = y\_axis , x = x\_axis)+

theme(axis.text.x=element\_text(angle=60,hjust=1,size = 12,colour = "black"),

axis.ticks.x = element\_blank(),

legend.position="none"

# legend.position = c(0.3, 0.92),

# legend.direction = "horizontal"

) +

ablife\_theme\_line()+

# # scale\_y\_continuous(limits=c(0,100))+

scale\_colour\_hue(name=opt$legend\_name)

# scale\_colour\_manual("Sample Name",values = colour[1:length(sample)])

# scale\_fill\_discrete(name="Sample\_name") ###################################################################################

###Save Plot File

ggsave(file = paste(opt$filename,"violin.pdf",sep='\_'), width = 105,height =150 ,dpi = 450,units = "mm")

ggsave(file = paste(opt$filename,"violin.png",sep='\_'), width = 105,height =150 ,dpi = 450,units = "mm")