MAGeCKFlute analysis for Chunyang's CRISPR KO screening data

by Jeon Lee on May 17, 2019

To run MAGeCKFlute pipeline, we need gene summary file generated by running MAGeCK RRA or MAGeCK MLE. MAGeCK MLE can be applied only when an experiment contains more than two conditions, for example, day0, drug treatment and DMSO treatment. So, only MAGeCK RRA results are explored here

Import libraries and define customized functions

```
library(MAGeCKFlute)
library(knitr)
library(kableExtra)
library(ggplot2)
library(ggrepel)
library(caRpools)
library(DESeq2)
library(edgeR)
library(pheatmap)
library(RColorBrewer)
library(gridExtra)
source("my_carpools.read.distribution.R")

setwd("Z:/BICF/BICF_Core/s167719/MichaelBuszczak/CRISPR_screen_ChunyangNi")

Sys.setenv(http_proxy="proxy.swmed.edu:3128")
Sys.setenv(https_proxy="proxy.swmed.edu:3128")


# define my_VolcanoView function
my_VolcanoView <- function(df, x = "logFC", y = "adj.P.Val", Label = NA, top = 5,
                        topnames = NULL, filename = NULL, x_cutoff = log2(1.5), y_cut
off = 0.05,
                        main = NULL, xlab = "Log2 Fold Change", ylab = "-Log10(Adjust
.P)", ...){
    requireNamespace("ggrepel", quietly=TRUE) || stop("need ggrepel package")
    gg = df[, c(x, y)]
    gg$group="no"
    gg$group[gg[,x]>x_cutoff & gg[,y]<y_cutoff] = "up"
    gg$group[gg[,x]< -x_cutoff & gg[,y]<y_cutoff] = "down"

    gg[, y] = -log10(gg[, y])
    if(!(top==0 & is.null(topnames))){
      gg$Label = rownames(gg)
      if(!is.na(Label)) gg$Label = df[, Label]
      gg = gg[order(abs(gg[,x]), gg[,y], decreasing = TRUE), ]
      idx1 = idx2 = c()
      if(top>0){
        idx1 = which(gg$group=="up")[1:min(top, sum(gg$group=="up"))]
        idx2 = which(gg$group=="down")[1:min(top, sum(gg$group=="down"))]
      }
      idx = unique(c(idx1, idx2, which(gg$Label %in% topnames)))
      gg$Label = as.character(gg$Label)
      gg$Label[setdiff(1:nrow(gg), idx)] = ""
      gg$Label = factor(gg$Label, levels = setdiff(unique(gg$Label), ""))
    }
    mycolour=c("no"="gray80",  "up"="#e41a1c","down"="#377eb8")
    #=========
    p = ggplot(gg, aes(x=gg[,x], y=gg[,y], colour=group, fill=group))
    p = p + geom_jitter(position = "jitter", show.legend = FALSE, alpha=0.8, size = 1)
    p = p + theme(text = element_text(colour="black",size = 14, family = "Helvetica"),
                  plot.title = element_text(hjust = 0.5, size=16),
                  axis.text = element_text(colour="gray10"))
    p = p + theme(axis.line = element_line(size=0.5, colour = "black"),
                  panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
                  panel.border = element_blank(), panel.background = element_blank())
    p = p + geom_hline(yintercept = -log10(y_cutoff), linetype = "dotted")
    p = p + geom_vline(xintercept = c(-x_cutoff, x_cutoff), linetype = "dotted")
    p = p + labs(x=xlab, y=ylab, title=main)

    if(!(top==0 & is.null(topnames)))
      p = p + ggrepel::geom_text_repel(aes(x=gg[idx,x],y=gg[idx,y], label = Label), data=gg[idx,],
                                       fontface = 'bold', size = 4,
                                       box.padding = unit(0.4, "lines"), segment.color = 'grey50',
                                       point.padding = unit(0.3, "lines"), segment.size = 0.3)
    p = p + scale_color_manual(values=mycolour)
    p = p + scale_fill_manual(values=mycolour)
    p = p + theme(legend.position = "none")
    return(p)
}


# define my_EnrichedGeneView
my_EnrichedGeneView=function(enrichment, geneList,
                          rank_by = "p.adjust",
                          top = 5, bottom = 5,
                          custom_pid = NULL,
                          keytype = "Symbol",
                          gene_cutoff = c(-log2(1.5), log2(1.5)),
                          custom_gene = NULL,
                          charLength = 40,
                          filename = NULL,
                          width = 7, height = 5, ...){

  if(is(enrichment, "enrichResult")) enrichment = enrichment@result
  if(is(enrichment, "gseaResult")) enrichment = enrichment@result

  ## No enriched pathways ##
  if(is.null(enrichment) || nrow(enrichment)==0){
    p1 = noEnrichPlot("No enriched terms")
    if(!is.null(filename)){
      ggsave(plot=p1,filename=filename, units = "in", width=width, height=height, ...
)
    }
    return(p1)
  }

  ## Rank enriched pathways ##
  enrichment$logP = round(-log10(enrichment$p.adjust), 1)
  enrichment = enrichment[!is.na(enrichment$ID), ]
  if(tolower(rank_by) == "p.adjust"){
    enrichment = enrichment[order(enrichment$p.adjust), ]
  }else if(tolower(rank_by) == "nes"){
    enrichment = enrichment[order(abs(enrichment$NES), decreasing = TRUE), ]
  }

  ## Normalize term description ##
  terms = as.character(enrichment$Description)
  terms = lapply(terms, function(x,k){
    x = as.character(x)
    if(nchar(x)>k){x=substr(x,start=1,stop=k)}
    return(x)}, charLength)
  enrichment$Description = do.call(rbind, terms)
  enrichment = enrichment[!duplicated(enrichment$Description),]
  ## Select pathways to show ##
  pid_neg <- pid_pos <- NULL
  if(bottom>0){
    tmp = enrichment[enrichment$NES<0, ]
    pid_neg = tmp$ID[1:min(nrow(tmp), bottom)]
  }
  if(top>0){
    tmp = enrichment[enrichment$NES>0, ]
    pid_pos = tmp$ID[1:min(nrow(tmp), top)]
  }
  idx = enrichment$ID %in% c(custom_pid, pid_neg, pid_pos)
  if(sum(idx)==0) stop("No input pathway found !!!")

  ## Prepare data for plotting ##
  enrichment = enrichment[idx, ]
  enrichment$Description = factor(enrichment$Description,
                                  levels=enrichment$Description)
  geneNames = strsplit(enrichment$geneName, "\\/")
  geneIds = strsplit(enrichment$geneID, "\\/")
  gg = data.frame(ID = rep(enrichment$ID, enrichment$Count),
                  Term = rep(enrichment$Description, enrichment$Count),
                  Size = rep(enrichment$logP, enrichment$Count),
                  Gene = unlist(geneNames), geneIds = unlist(geneIds),
                  stringsAsFactors = FALSE)

  ## Select genes to show ##
  names(geneList) = toupper(names(geneList))
  geneList = geneList[geneList<gene_cutoff[1] | geneList>gene_cutoff[2] |
                        names(geneList) %in% custom_gene]
  if(keytype == "Symbol") gg$GeneScore = geneList[gg$Gene]
  if(keytype == "Entrez") gg$GeneScore = geneList[gg$geneIds]

  ## Rank pathways and genes ##
  gg = gg[!is.na(gg$GeneScore), ]
  gg$Term = factor(gg$Term, levels = unique(gg$Term))
  gg = gg[order(gg$GeneScore), ]
  gg$Gene = factor(gg$Gene, levels = unique(gg$Gene))
  # Plot the dot heatmap
  p1 = ggplot(data=gg, aes(x=Gene, y=Term, size=Size, color = GeneScore))
  p1 = p1 + geom_point()
  p1 = p1 + scale_color_gradient2(low = "#081087", high = "#c12603")
  p1 = p1 + theme(panel.grid.major=element_line(colour="gray90"),
                  panel.grid.minor=element_blank(),
                  panel.background=element_blank())
  p1 = p1 + labs(x=NULL, y=NULL, color = "Gene score", size = "LogP")
  # p1 = p1 + theme(legend.position="top")
  # p1 = p1 + scale_size_continuous(guide = FALSE)
  p1 = p1 + theme(legend.key = element_rect(fill = "transparent", colour = "transpare
nt"))
  p1 = p1 + theme(text = element_text(colour="black",size = 14, family = "Helvetica")
,
                  plot.title = element_text(hjust = 0.5, size=18),
                  axis.text = element_text(colour="gray10"),
                  axis.text.x = element_text(angle = 60, hjust = 1, vjust = 1))
  p1 = p1 + theme(axis.line = element_line(size=0.5, colour = "black"),
                  panel.grid.major = element_blank(), panel.grid.minor = element_blan
k(),
                  panel.border = element_blank(), panel.background = element_blank(),
                  legend.key = element_rect(fill = "transparent"))

}
```

Section I: Quality control

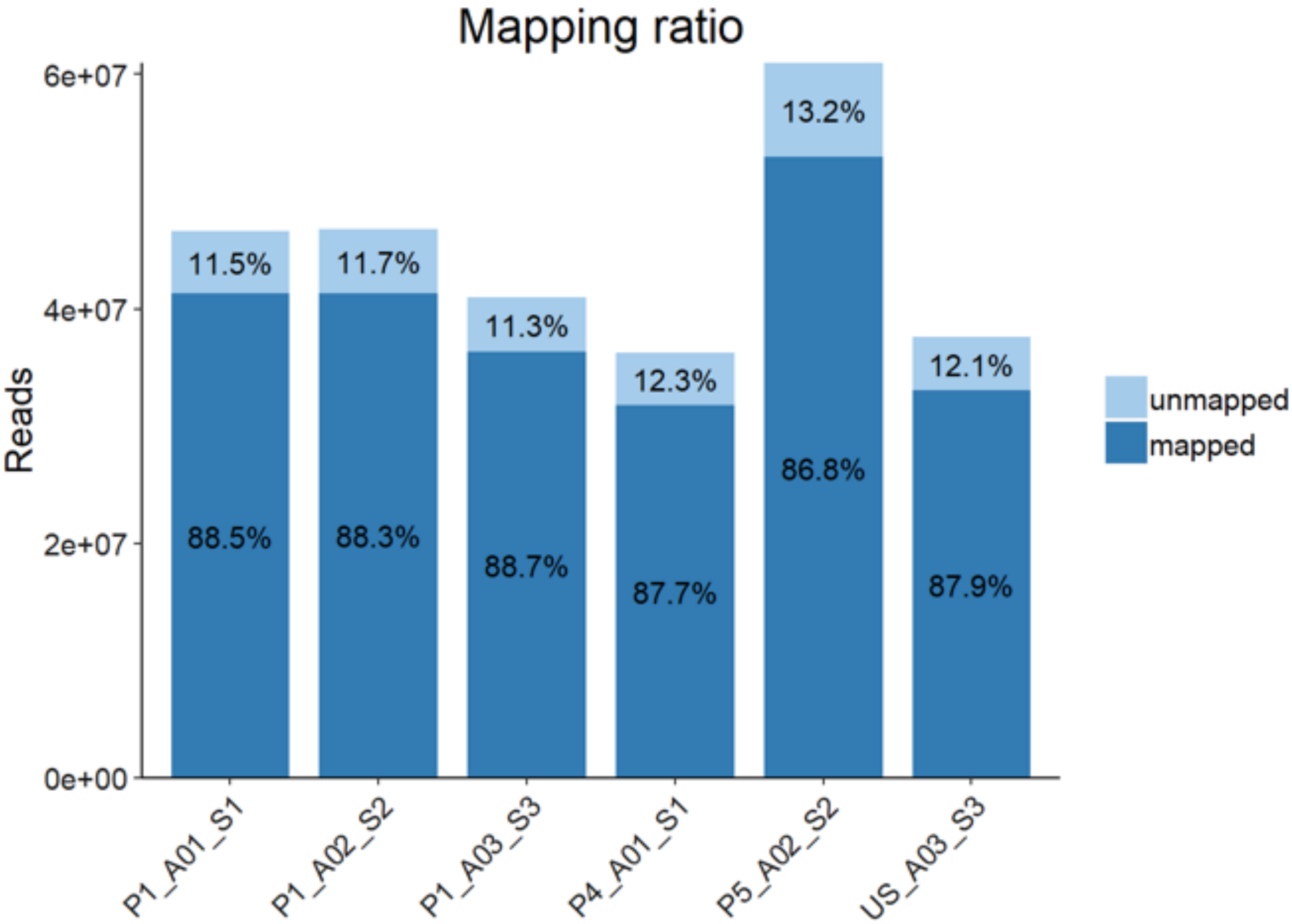
1. Read mapping QC

MAGeCK Count in MAGeCK generates a count summary file, which summarizes some basic QC scores at raw count level, including map ratio, Gini index, and NegSelQC.

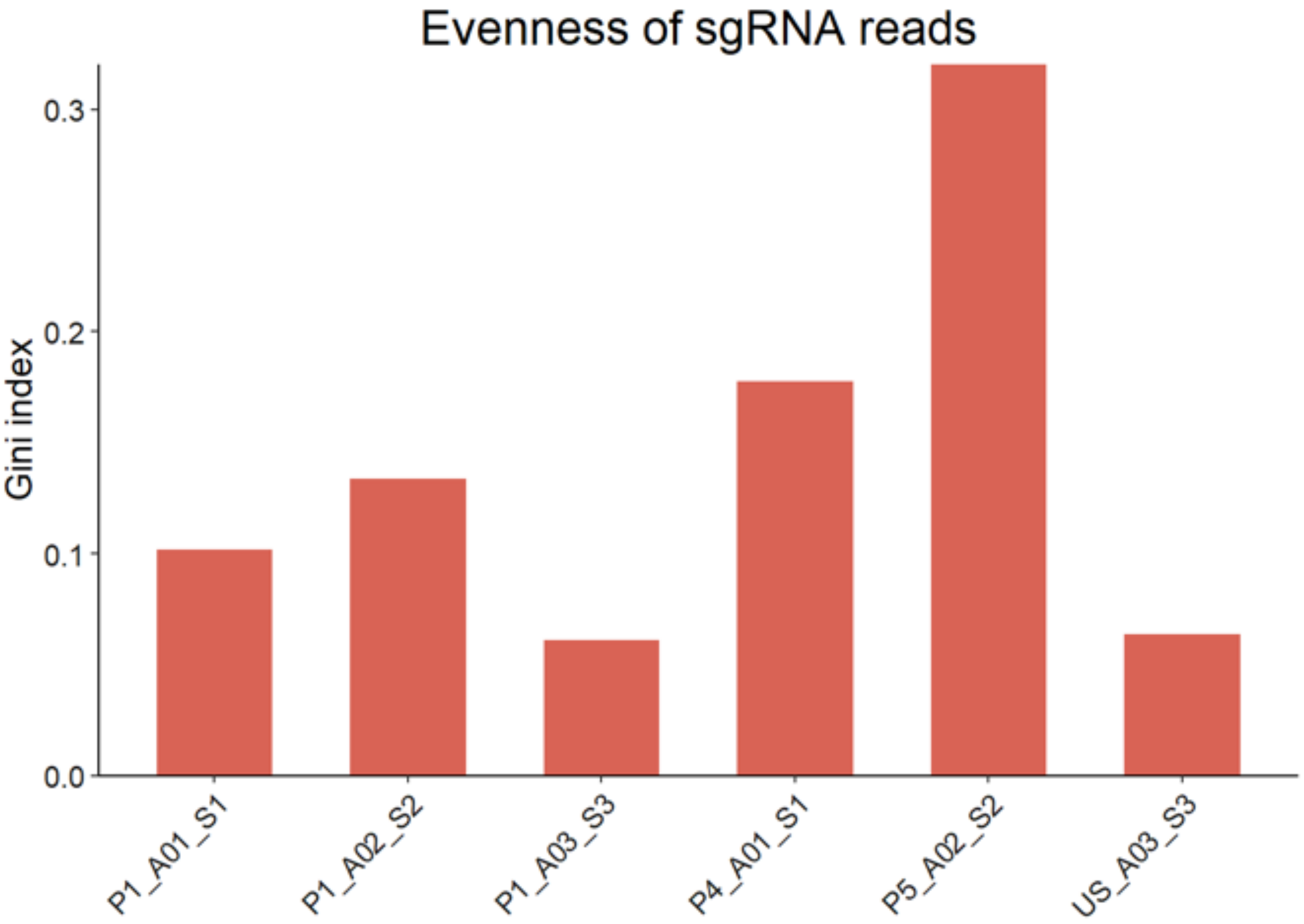
```
countsummary <- read.table("Run056.countsummary.txt", sep='\t', header=T)
kable(countsummary) %>% kable_styling() %>% scroll_box(width = "100%")
```

File	Label	Reads	Mapped	Percentage
DATA/P4_A01_S1_cutadapt.fastq	P4_A01_S1	36210762	31762656	0.8772
DATA/P5_A02_S2_cutadapt.fastq	P5_A02_S2	60985024	52932944	0.8680
DATA/US_A03_S3_cutadapt.fastq	US_A03_S3	37562457	33009520	0.8788
DATA/Brunello_L28snap_P1_A01_S1_cutadapt.fastq	P1_A01_S1	46658350	41274188	0.8846
DATA/Brunello_L28snap_P1_A02_S2_cutadapt.fastq	P1_A02_S2	46827038	41332493	0.8827
DATA/Brunello_L28snap_P1_A03_S3_cutadapt.fastq	P1_A03_S3	40935502	36296265	0.8867

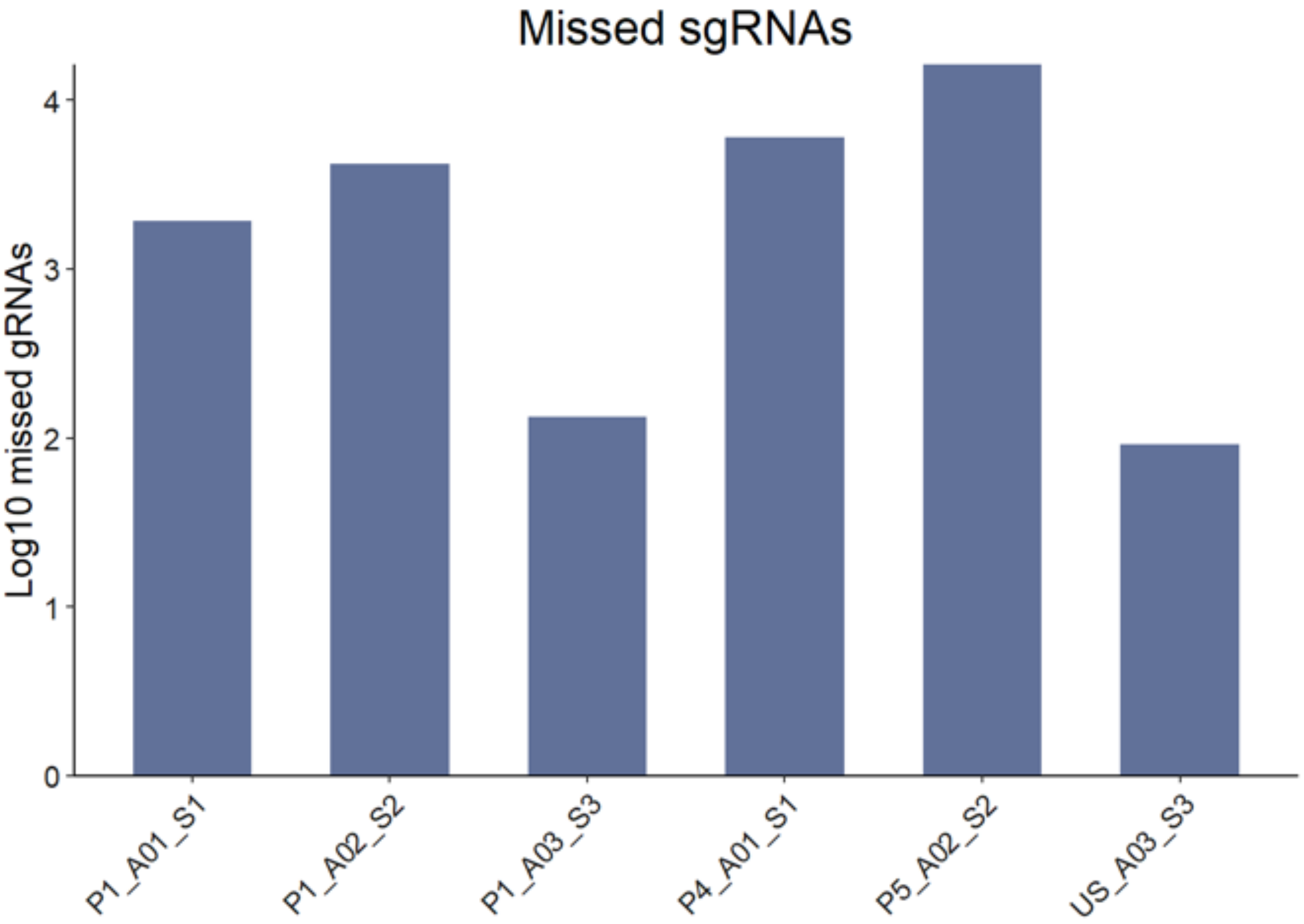
```
MapRatesView(countsummary)
```



```
IdentBarView(countsummary, x = "Label", y = "GiniIndex",
             ylab = "Gini index", main = "Evenness of sgRNA reads")
```



```
countsummary$Missed = log10(countsummary$Zerocounts)
IdentBarView(countsummary, x = "Label", y = "Missed", fill = "#394E80",
             ylab = "Log10 missed gRNAs", main = "Missed sgRNAs")
```



2. SampleQC

1) CRISPR library count summary

```
#== Read the count summary file
Count_summary <- read.table("Run056.countsummary.txt", sep='\t', header=T)
kable(Count_summary) %>% kable_styling() %>% scroll_box(width = "100%")
```

File	Label	Reads	Mapped	Percentage
DATA/P4_A01_S1_cutadapt.fastq	P4_A01_S1	36210762	31762656	0.8772
DATA/P5_A02_S2_cutadapt.fastq	P5_A02_S2	60985024	52932944	0.8680
DATA/US_A03_S3_cutadapt.fastq	US_A03_S3	37562457	33009520	0.8788

DATA/Brunello_L28snap_P1_A01_S1_cutadapt.fastq	P1_A01_S1	46658350	41274188	0.8846
DATA/Brunello_L28snap_P1_A02_S2_cutadapt.fastq	P1_A02_S2	46827038	41332493	0.8827
DATA/Brunello_L28snap_P1_A03_S3_cutadapt.fastq	P1_A03_S3	40935502	36296265	0.8867

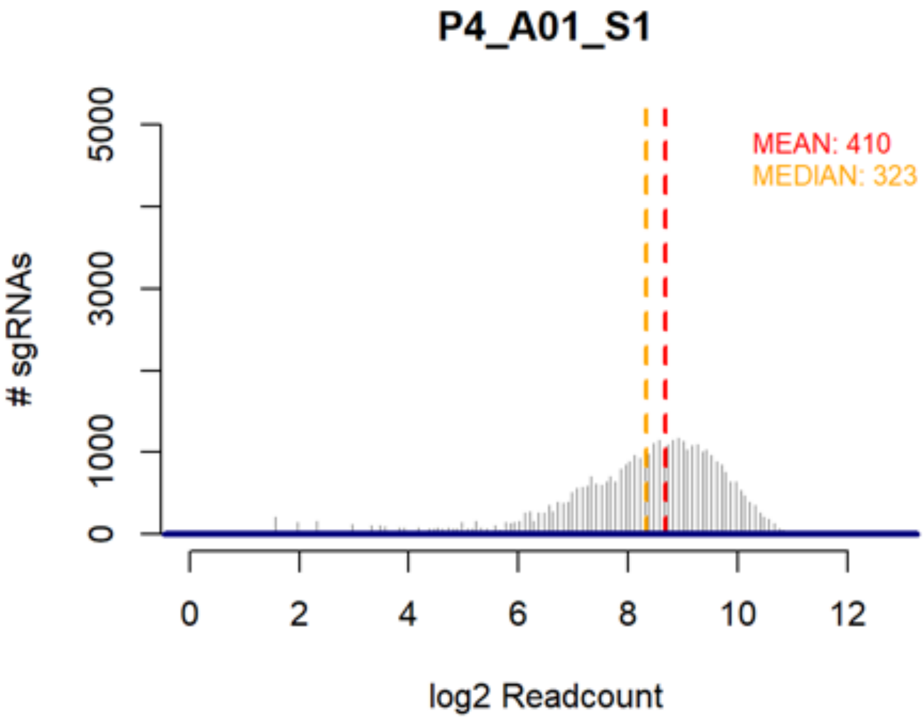
2) sgRNA count statistics

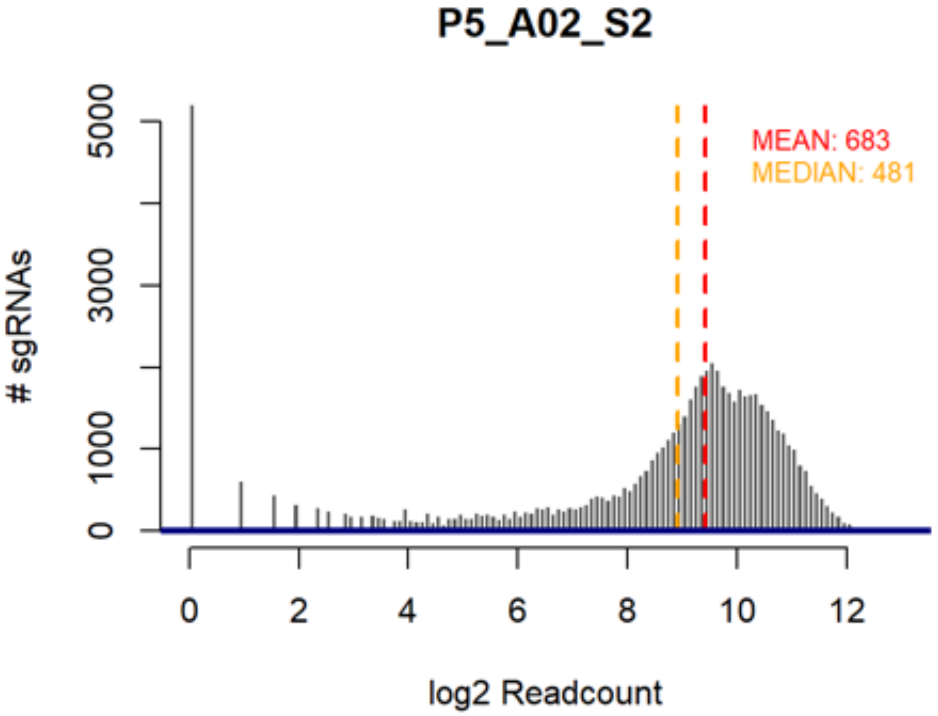
```
#== Read the sgRNA count table
my_data <- read.table("Run056.count.txt", sep='\t', header=T)
sgRNA_table <- my_data[,c(3:8)]
rownames(sgRNA_table) <- my_data$sgRNA

# sgRNA count statistics
Mean <- apply(sgRNA_table, 2, mean)
Median <- apply(sgRNA_table, 2, median)
SD <- apply(sgRNA_table, 2, sd)
Min <- apply(sgRNA_table, 2, min)
Max <- apply(sgRNA_table, 2, max)

sgRNACount_stat <- data.frame(Mean, Median, SD, Min, Max)
kable(sgRNACount_stat) %>% kable_styling()
```

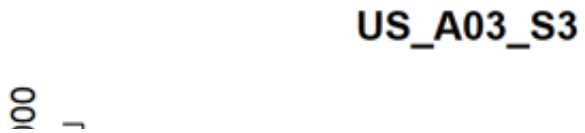
	Mean	Median	SD	Min	Max
P4_A01_S1	410.1530	323	370.0146	0	7217
P5_A02_S2	683.5261	481	795.9970	0	26819
US_A03_S3	426.2538	367	251.4700	0	3863
P1_A01_S1	532.9759	441	416.0654	0	6236
P1_A02_S2	533.7288	433	456.0498	0	20131
P1_A03_S3	468.6957	405	271.1725	0	2361

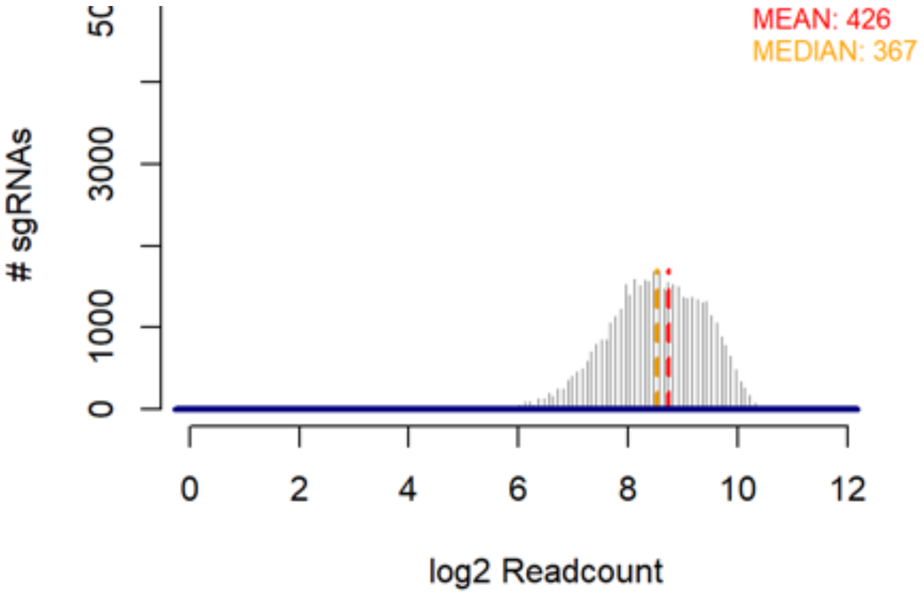
3) log2(sgRNA count) histogram

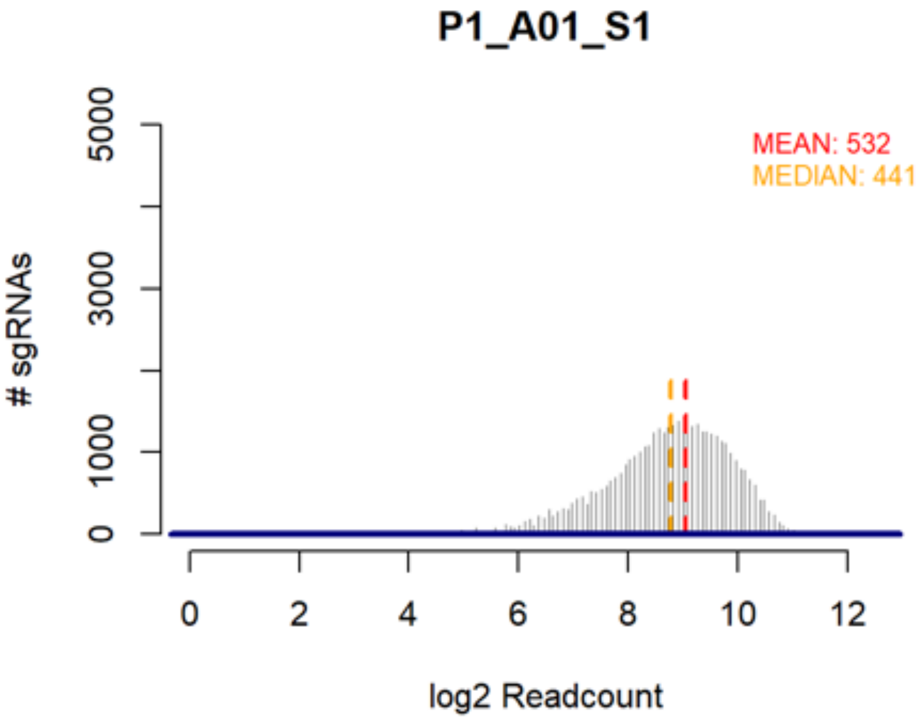
```
# log2 sgRNA count histogram
for (i in c(3:8)) {
  my_carpools.read.distribution(my_data, namecolumn=1,fullmatchcolumn=i,breaks=200,
                              title=names(my_data)[i], xlab="log2 Readcount", ylab="# sgRNAs",statistics=TRUE)
}
```

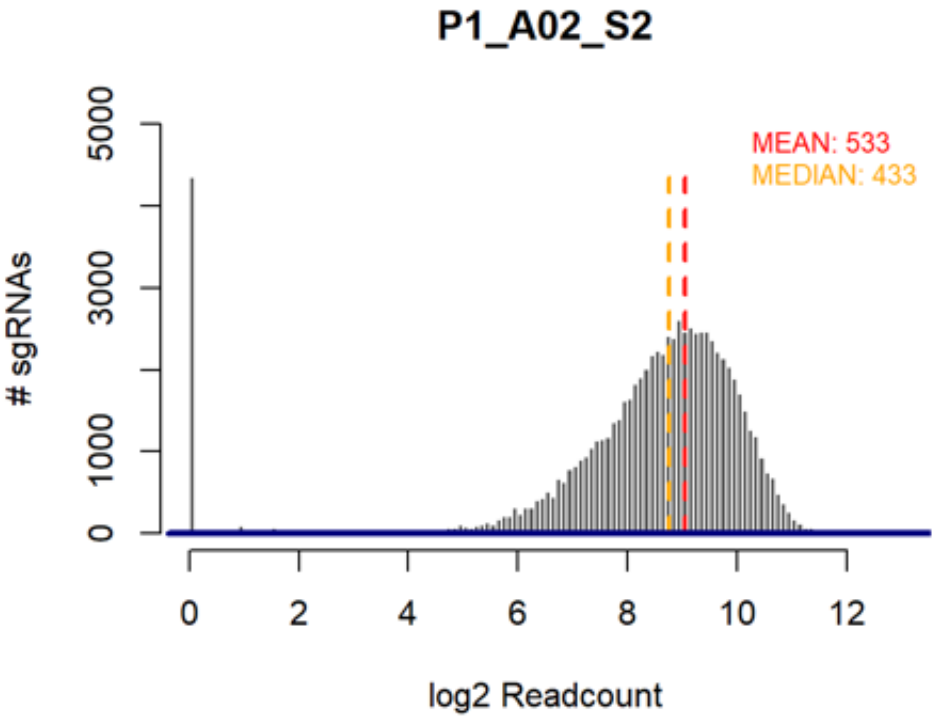

P4_A01_S1
# sgRNAs
0
1000
3000
5000
MEAN: 410
MEDIAN: 323
0
2
4
6
8
10
12
log2 Readcount

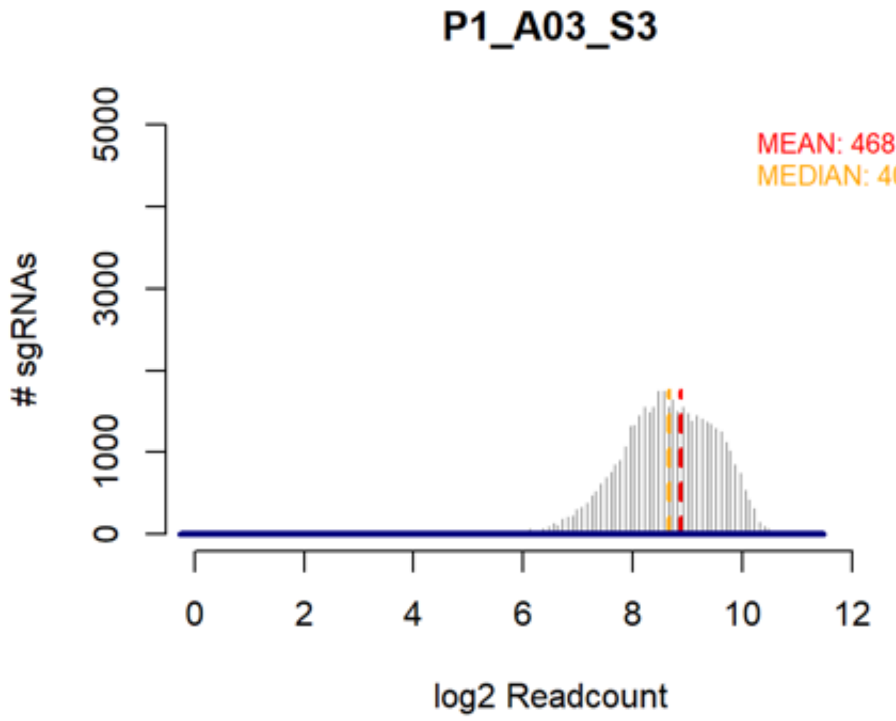

P5_A02_S2
# sgRNAs
0
1000
3000
5000
MEAN: 683
MEDIAN: 481
0
2
4
6
8
10
12
log2 Readcount


US_A03_S3
000


MEAN: 426
MEDIAN: 367
# sgRNAs
0
1000
3000
5C
0
2
4
6
8
10
12
log2 Readcount


P1_A01_S1
MEAN: 532
MEDIAN: 441
# sgRNAs
0
1000
3000
5000
0
2
4
6
8
10
12
log2 Readcount


P1_A02_S2
MEAN: 533
MEDIAN: 433
# sgRNAs
0
1000
3000
5000
0
2
4
6
8
10
12
log2 Readcount



4) Gene count statistics

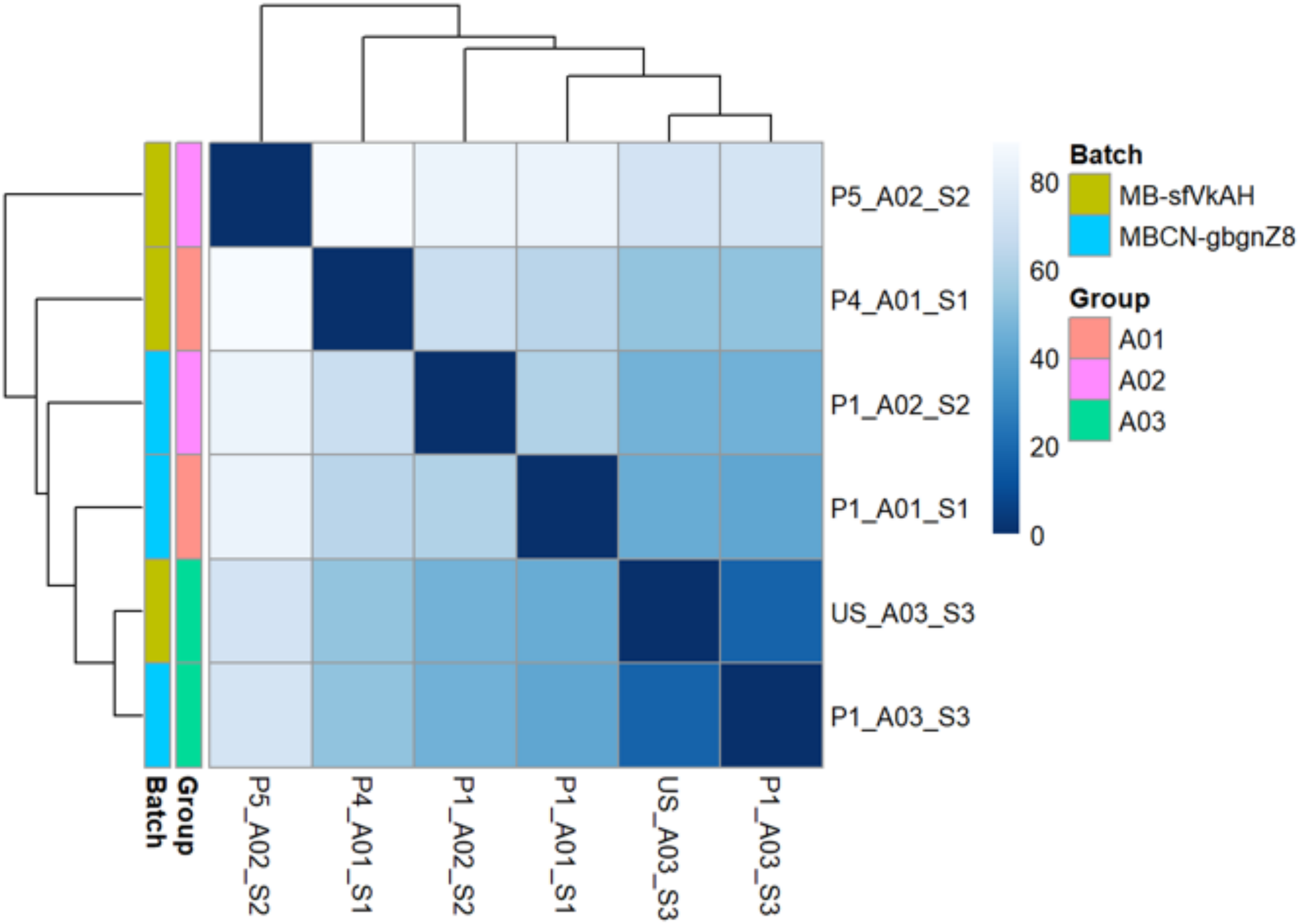
```
#== Generate a gene count table
idx_NonTargeting <- grep("^Non-Targeting", my_data$Gene)
my_data <- my_data[-c(idx_NonTargeting),]
gene_table <- aggregate(. ~Gene, data=my_data, FUN=sum)

# Correct wrong gene names.
# Note that gene names starting with a numeric value were ordered first
full_gene_list <- gene_table$Gene
idx <- grep("^[0-9]",full_gene_list)
wrong_names <- as.character(full_gene_list[idx])
tmp <- strsplit(wrong_names,"-")
correct_names <- vector(mode="character", length=length(idx))

for (i in idx) {
  if (tmp[[i]][2]=="Dec") {
    correct_names[i] <- paste("DEC",tmp[[i]][1],sep="")
  }
  if (tmp[[i]][2]=="Mar") {
    correct_names[i] <- paste("MARCH",tmp[[i]][1],sep="")
  }
  if (tmp[[i]][2]=="Sep") {
    correct_names[i] <- paste("SEPT",tmp[[i]][1],sep="")
  }
}
#
gene_table <- gene_table[,3:8]
rownames(gene_table) <- full_gene_list
rownames(gene_table)[1:length(idx)] <- correct_names


# Gene count statistics
Mean <- apply(gene_table, 2, mean)
Median <- apply(gene_table, 2, median)
SD <- apply(gene_table, 2, sd)
Min <- apply(gene_table, 2, min)
Max <- apply(gene_table, 2, max)

geneCount_stat <- data.frame(Mean, Median, SD, Min, Max)
kable(geneCount_stat) %>% kable_styling()
```

	Mean	Median	SD	Min	Max
P4_A01_S1	1619.397	1463.0	876.6324	0	12842
P5_A02_S2	2695.492	2412.5	1774.1094	0	40650

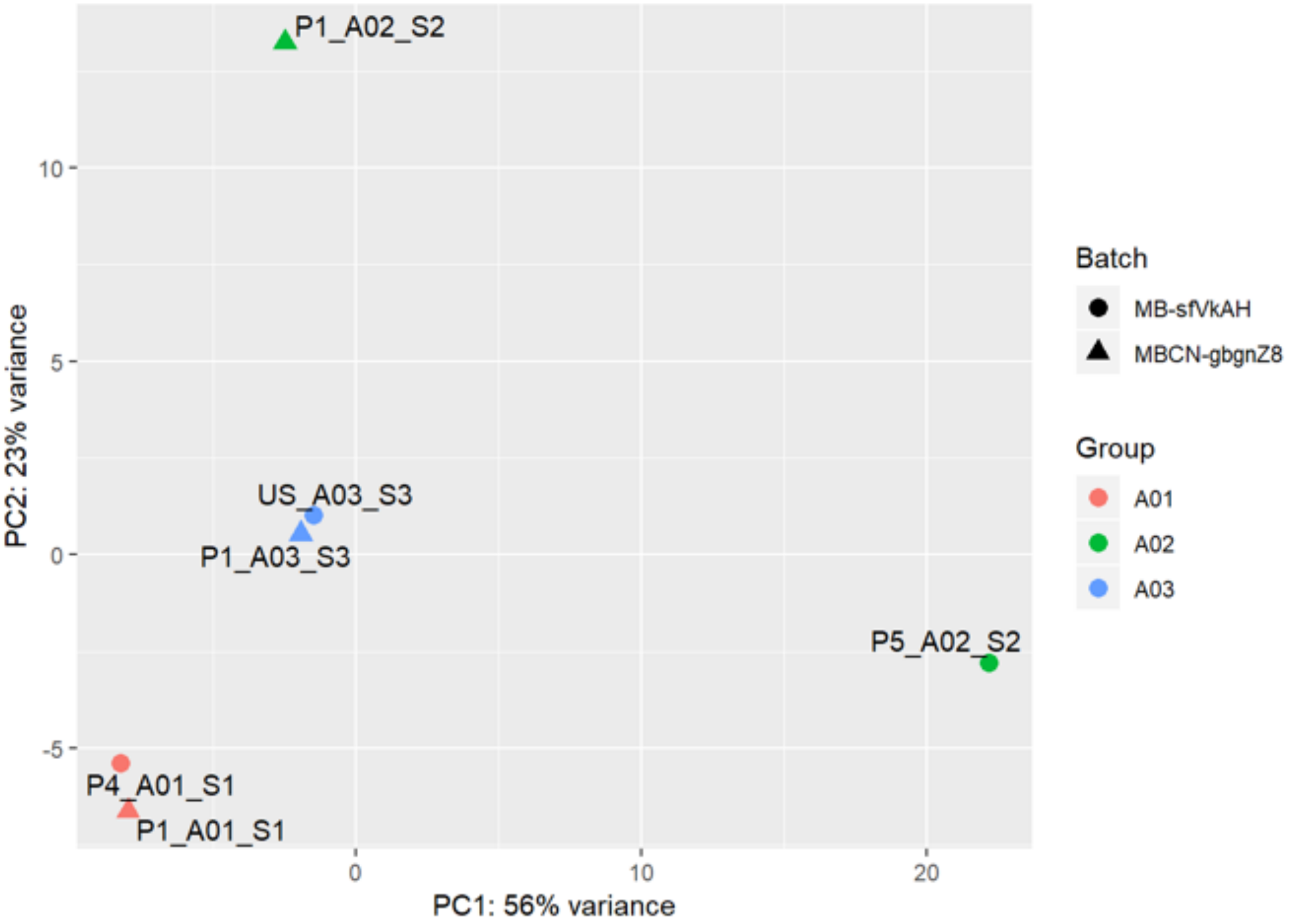
US_A03_S3	1681.005	1544.0	701.3298	107	5156
P1_A01_S1	2107.170	1907.0	1044.1605	26	16522
P1_A02_S2	2107.465	1909.0	1112.6047	1	32253
P1_A03_S3	1851.022	1698.0	751.4913	255	5223

5) Sample heatmap and PCA plot

```
#=== DESeq analysis
countData <- gene_table
SampleID <- gsub("(Brunello_L28snap_)(.*)","\\2", colnames(countData))
colnames(countData) <- SampleID

colData <- data.frame(Group=c("A01","A02","A03","A01","A02","A03"),
                      Batch=c("MB-sfVkAH","MB-sfVkAH","MB-sfVkAH","MBCN-gbgnZ8","MBCN
-gbgnZ8","MBCN-gbgnZ8"))
rownames(colData) <- SampleID

# Data normalization
dds <- DESeqDataSetFromMatrix(countData = countData, colData = colData, design= ~ Gro
up + Batch)
vsd <- vst(dds, blind=FALSE)

##=Plot a heatmap using sample distances
#Euclidean distance
sampleDist_Euc <- dist(t(assay(vsd))) #distance matrix (lower trianglular matrix)
sampleDist_Euc_full <- as.matrix(sampleDist_Euc) #full matrix
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
#pheatmap(sampleDistMatrix, clustering_distance_rows=sampleDists, clustering_distance
_cols=sampleDists, col=colors)
pheatmap(sampleDist_Euc_full, clustering_distance_rows=sampleDist_Euc, clustering_dis
tance_cols=sampleDist_Euc,
         annotation_row=colData[,c(1,2)], col=colors)
```



```
##Correlation distance
#sampleDist_Corr <- as.dist(1-cor(assay(vsd)))
#sampleDist_Corr_full <- as.matrix(sampleDist_Corr)
#
#pheatmap(sampleDist_Corr_full, clustering_distance_rows=sampleDist_Corr, clustering_
distance_cols=sampleDist_Corr,
#         annotation_row=colData[,c(1,2)], col=colors)

##=Plot a PCA plot
pcaData <- plotPCA(vsd, intgroup=c("Group", "Batch"), returnData=TRUE)
#pcaData
percentVar <- round(100 * attr(pcaData, "percentVar"))
ggplot(pcaData, aes(PC1, PC2, color=Group, shape=Batch, label=rownames(colData))) +
geom_point(size=3) +
  geom_text_repel(force=10, color="black") +
  xlab(paste0("PC1: ",percentVar[1],"% variance")) +
  ylab(paste0("PC2: ",percentVar[2],"% variance"))
```



On the PCA plot, P1_A01 and its replicate(P4-A01) samples cluster tightly and it's true to P1_A03 and its replicate(US_A03). So, we can confidently perform a group comparison between A01 vs. A03. However, for P1_A02 and P5_A02, considering the wide variation observed, it's not sure if we can take them as relicates for a group comparison–P5_A02 presumably seems to be an outlier. For this reason, in general, an ideal CRISPR screening experiment requires triplicates for each condition.

Section II: Downstream analysis of MAGeCK RRA

For experiments with two experimental conditions, MAGeCK-RRA is used to identify essential genes from CRISPR/Cas9 knockout screens. It tests the statistical significance of each observed change between two states. Gene summary file in MAGeCK-RRA results summarizes the statistical significance of positive selection and negative selection. In this experiment, we are interested in the positive selection only so that positive selection results are explored accordingly.

1. A01_vs_A03 comparison

1) Load gene and sgrna summary data in MAGeCK RRA results

```
rra.gene_summary = read.table("Run056_A01vsA03.gene_summary.txt", sep='\t', header=TR
UE)
kable(head(rra.gene_summary)) %>% kable_styling() %>% scroll_box(width = "100%")
```

id	num	neg.score	neg.p.value	neg.fdr	neg.rank	neg.goodsgrna	neg.lfc	pos.sco
CLCN5	4	2.80e-06	0.0000106	0.212871	1	4	-3.7046	1.000
C19orf40	4	7.40e-06	0.0000318	0.319307	2	3	-1.7679	0.774
NLGN4Y	4	1.84e-05	0.0000820	0.549505	3	3	-2.7906	0.482
ZNF222	4	2.58e-05	0.0001145	0.575495	4	3	-4.8987	0.718
LIMA1	4	4.40e-05	0.0001799	0.622937	5	4	-3.3333	0.999
FAM19A2	4	4.59e-05	0.0001858	0.622937	6	3	-4.8461	0.164

```
rra.sgrna_summary = read.table("Run056_A01vsA03.sgrna_summary.txt", sep='\t', header
= TRUE)
kable(head(rra.sgrna_summary)) %>% kable_styling() %>% scroll_box(width = "100%")
```

sgrna	Gene	control_count	treatment_count	control_mean	treat_mean	LFC
sgRNA_ID_23292	ZPR1	210.84/329.28	5570.1/5195.1	263.49	5379.3	4.3464
sgRNA_ID_50299	RPP21	95.121/357.84	3088.6/3957.1	184.52	3496.0	4.2365
sgRNA_ID_12207	NARS	96.102/241.83	2305.1/2971	152.46	2616.9	4.0925
sgRNA_ID_02325	CARS	178.47/328.39	1780.7/5162.1	242.10	3031.9	3.6411
sgRNA_ID_23290	ZPR1	759.99/866.49	5744/5180.7	811.49	5455.1	2.7474

sgRNA_ID_50301	RPP21	251.04/614.84	3208.6/3194.9	392.88	3201.8	3.0235

2) Top-10 genes and sgRNAs for positive selection

```
dd.rra = ReadRRA(rra.gene_summary, organism = "hsa")
dd.rra <- dd.rra[order(-dd.rra$LFC),]
kable(head(dd.rra, 10)) %>% kable_styling(full_width = F)
```

	Official	EntrezID	LFC	FDR
8882	ZPR1	8882	2.7150	0.001650
1938	EEF2	1938	2.4784	0.004950
833	CARS	833	2.4593	0.001650
7407	VARS	7407	2.3906	0.033130
51728	POLR3K	51728	1.9280	0.015752
54955	C1orf109	54955	1.8334	0.014851
5687	PSMA6	5687	1.7632	0.886904
1937	EEF1G	1937	1.7062	0.009194
4677	NARS	4677	1.6988	0.049196
1915	EEF1A1	1915	1.6909	0.095002

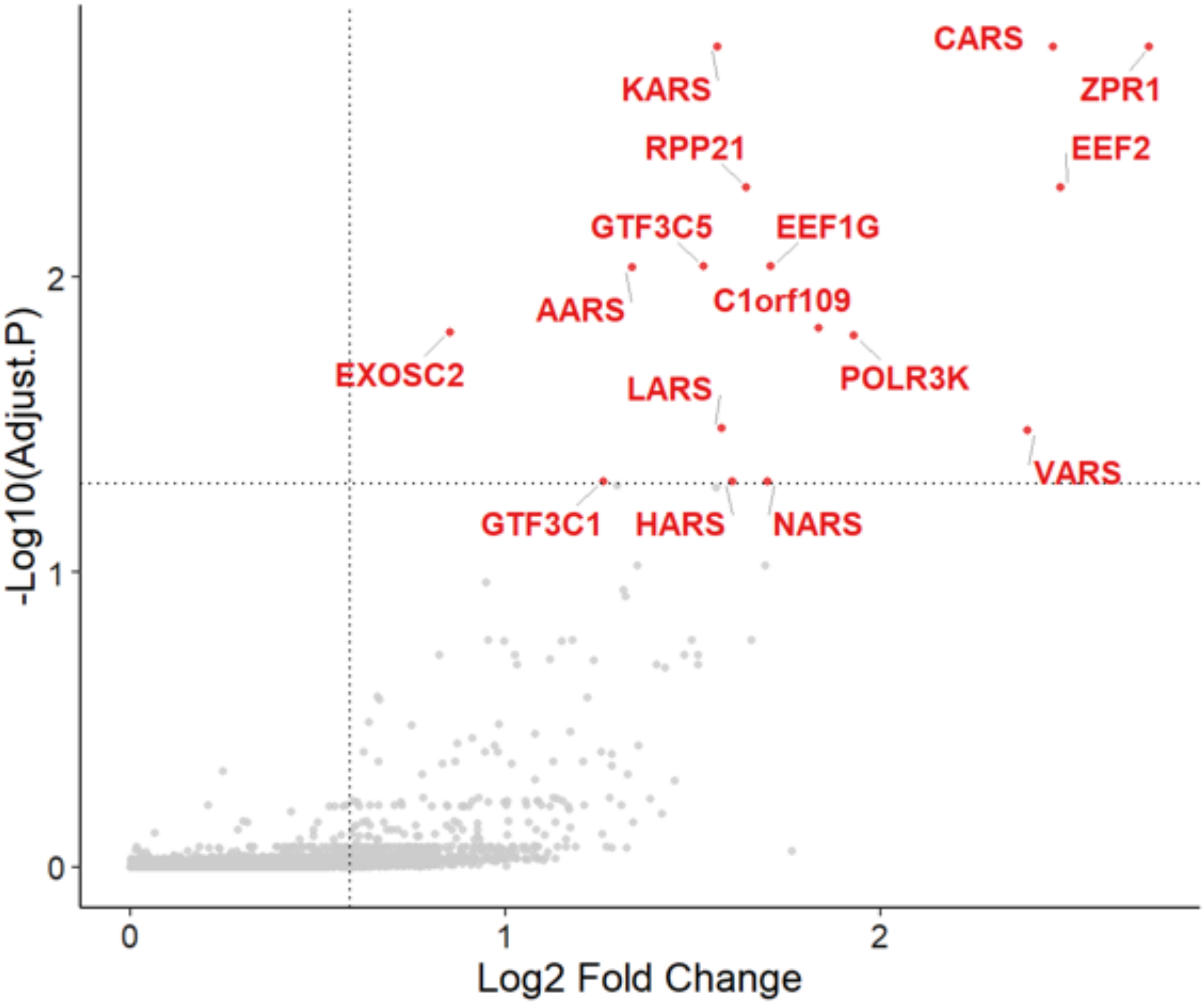
```
dd.sgrna = ReadsgRRA(rra.sgrna_summary)
dd.sgrna <- dd.sgrna[order(-dd.sgrna$LFC),]
kable(head(dd.sgrna,10)) %>% kable_styling(full_width = F)
```

	sgrna	Gene	LFC	FDR
1	sgRNA_ID_23292	ZPR1	4.3464	0
2	sgRNA_ID_50299	RPP21	4.2365	0

3	sgRNA_ID_12207	NARS	4.0925	0
51	sgRNA_ID_38611	POP5	4.0450	0
25	sgRNA_ID_05327	EEF2	4.0125	0
8	sgRNA_ID_37780	TRNT1	3.8698	0
334	sgRNA_ID_59548	NKAIN4	3.7323	0
11	sgRNA_ID_20083	VARS	3.6951	0
239	sgRNA_ID_20408	YWHAZ	3.6523	0
4	sgRNA_ID_02325	CARS	3.6411	0

3) VolcanoView for positive selected genes

```
p1 = my_VolcanoView(dd.rra[dd.rra$LFC >=0, ], x = "LFC", y = "FDR", Label = "Official
", top=20)
p1 <- p1 + xlim(0,NA)
print(p1)
```



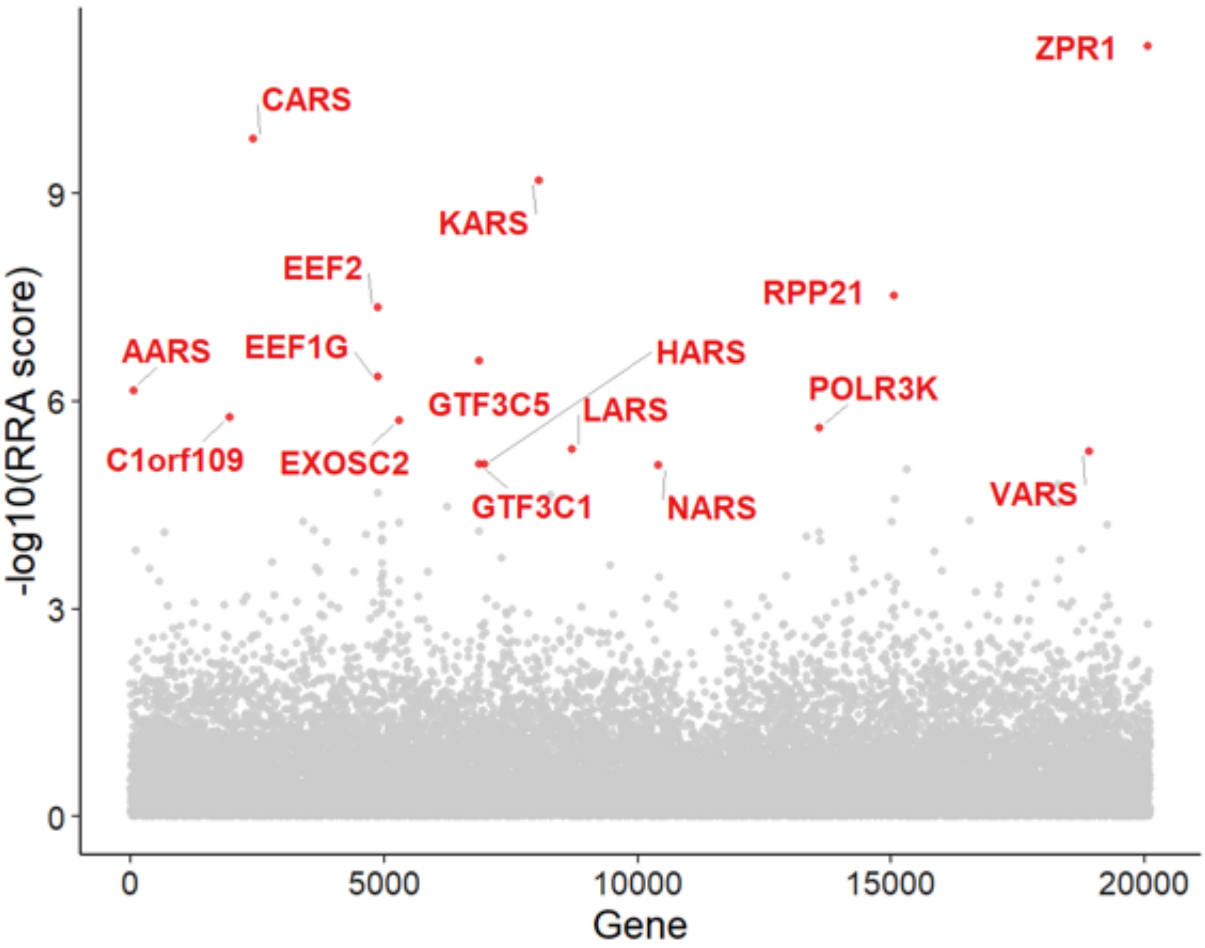
4) -log10(RRAscore) and -log10(pos.fdr) plots

```
df <- rra.gene_summary[,c('id','neg.score', 'neg.fdr', 'neg.lfc', 'pos.score', 'pos.f
dr', 'pos.lfc')]

x_cutoff = log2(1.5); y_cutoff = 0.05
df$group="NoSig"
df$group[df[,'pos.lfc']>x_cutoff & df[,'pos.fdr']<y_cutoff] = "Up"
df$group[df[,'neg.lfc']< -x_cutoff & df[,'neg.fdr']<y_cutoff] = "Down"
df$group <- as.factor(df$group)
levels(df$group) <- c("NoSig", "Up", "Down")
kable(table(df$group)) %>% kable_styling(full_width = F)
```

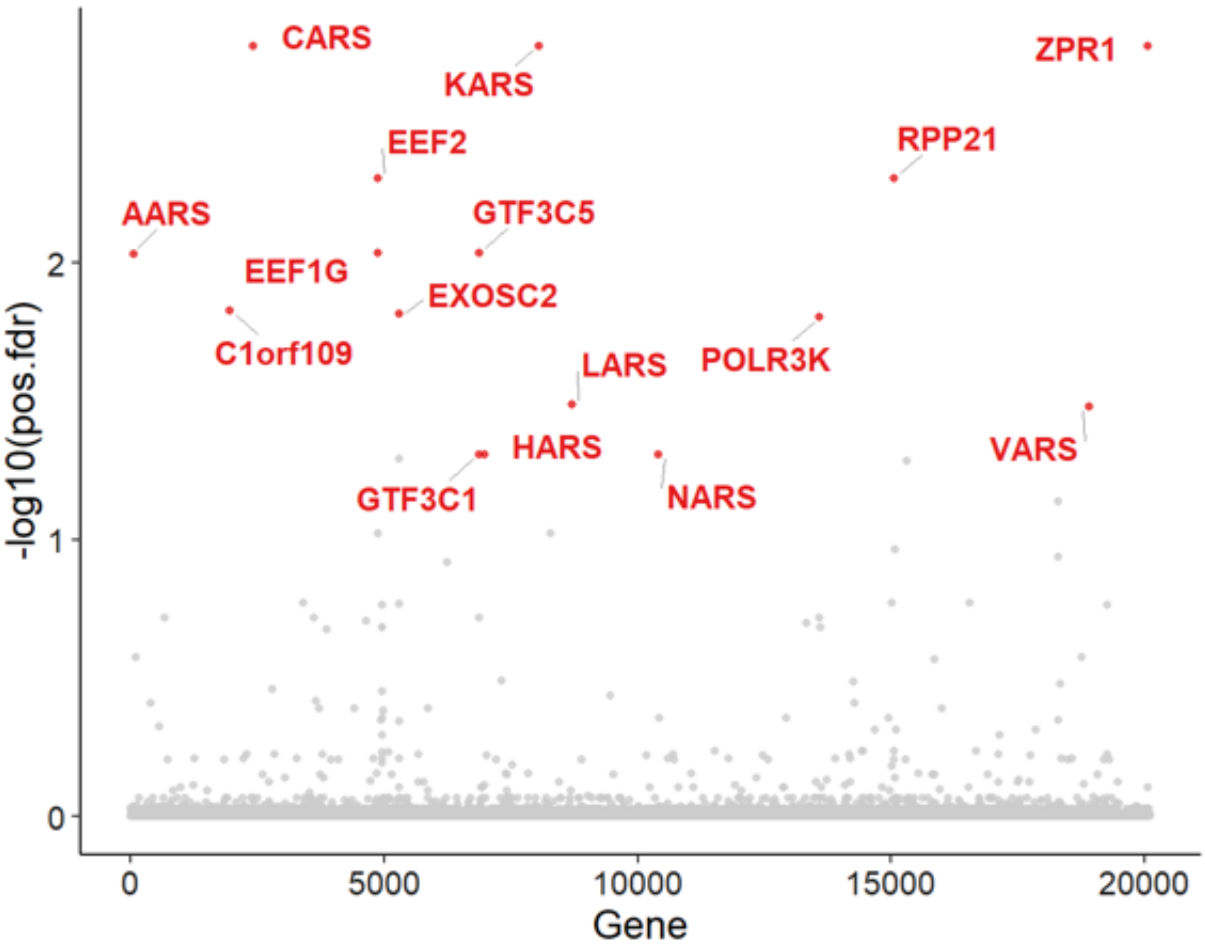
Var1	Freq
NoSig	20096
Up	16

Down	0

```
df[, c('neg.fdr', 'pos.fdr')] = -log10(df[, c('neg.fdr', 'pos.fdr')])
df[, c('neg.score', 'pos.score')] = -log10(df[, c('neg.score', 'pos.score')])

mycolour=c("NoSig"="gray80",  "Up"="#e41a1c", "Down"="#377eb8")

# Sort gene symbol in an alphabetical order
df = df[order(df$id),]
df$GeneOrder = c(1:nrow(df))

df$Label = as.character(df$id)
idx_up = which(df$group=="Up")
idx_down = which(df$group=="Down")
idx = unique(idx_up, idx_down)
df$Label[setdiff(1:nrow(df), idx)] = ""
df$Label = factor(df$Label, levels = setdiff(unique(df$Label), ""))

# -log10 RRA scrore plot
p_sc = ggplot(df, aes(x=GeneOrder, y=pos.score, colour=group, fill=group))
p_sc = p_sc + geom_jitter(position = "jitter", show.legend = FALSE, alpha=0.8, size = 1)
p_sc = p_sc + theme(text = element_text(colour="black",size = 14, family = "Helvetica"),
              plot.title = element_text(hjust = 0.5, size=16),
              axis.text = element_text(colour="gray10"))
p_sc = p_sc + theme(axis.line = element_line(size=0.5, colour = "black"),
              panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
              panel.border = element_blank(), panel.background = element_blank())
p_sc = p_sc + labs(x='Gene', y='-log10(RRA score)', title='')
p_sc = p_sc + geom_text_repel(aes(x=df[idx,'GeneOrder'], y=df[idx,'pos.score'], label = Label), data=df[idx,],
                                         fontface = 'bold', size = 4,
                                         box.padding = unit(0.4, "lines"), segment.color = 'grey50',
                                         point.padding = unit(0.3, "lines"), segment.size = 0.3)
p_sc = p_sc + scale_color_manual(values=mycolour)
p_sc = p_sc + scale_fill_manual(values=mycolour)
p_sc = p_sc + theme(legend.position = "none")
p_sc
```

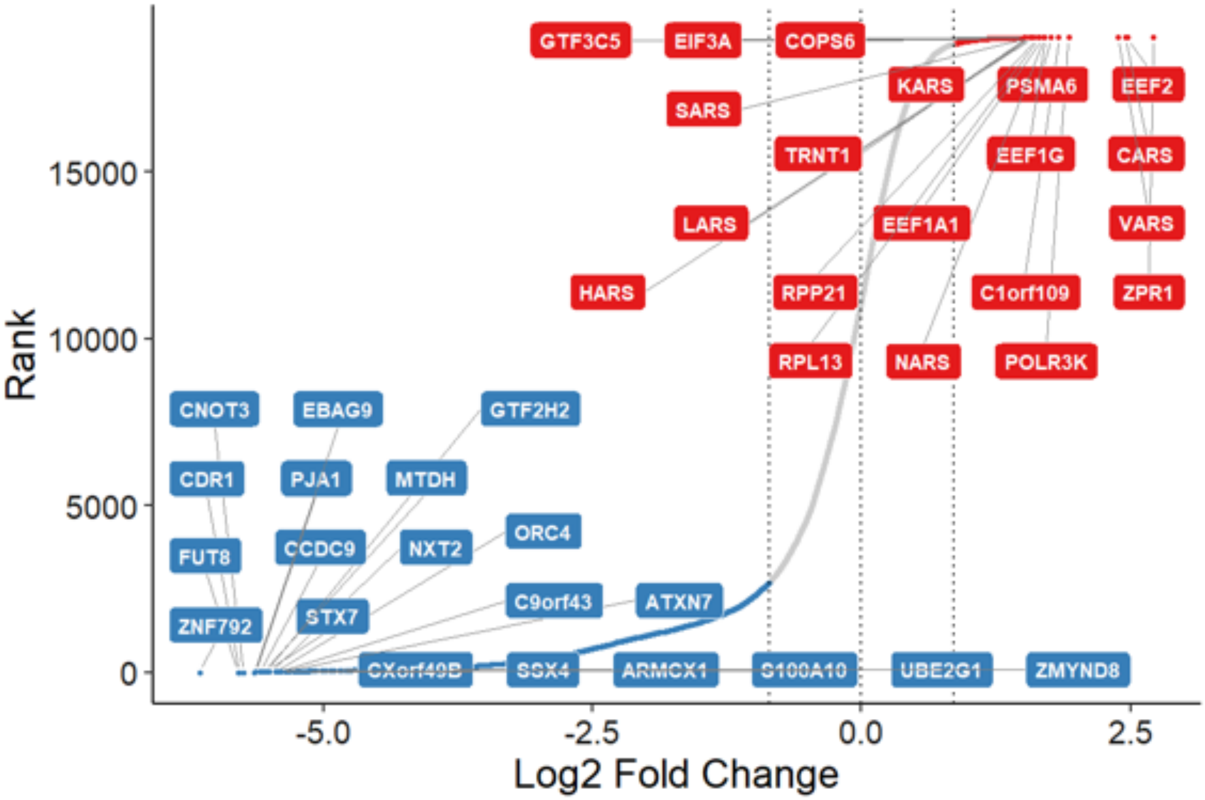

ZPR1
CARS
KARS
EEF2
RPP21
AARS
EEF1G
HARS
GTF3C5
LARS
POLR3K
C1orf109
EXOSC2
GTF3C1
NARS
VARS
-log10(RRA score)
9
6
3
0
0
5000
10000
15000
20000
Gene

```
# -log10 pos.fdr plot
p_fdr = ggplot(df, aes(x=GeneOrder, y=pos.fdr, colour=group, fill=group))
p_fdr = p_fdr + geom_jitter(position = "jitter", show.legend = FALSE, alpha=0.8, size
= 1)
p_fdr = p_fdr + theme(text = element_text(colour="black",size = 14, family = "Helveti
ca"),
              plot.title = element_text(hjust = 0.5, size=16),
              axis.text = element_text(colour="gray10"))
p_fdr = p_fdr + theme(axis.line = element_line(size=0.5, colour = "black"),
              panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
              panel.border = element_blank(), panel.background = element_blank())
p_fdr = p_fdr + labs(x='Gene', y='-log10(pos.fdr)', title='')
p_fdr = p_fdr + geom_text_repel(aes(x=df[idx,'GeneOrder'], y=df[idx,'pos.fdr'], label
= Label), data=df[idx,],
                                       fontface = 'bold', size = 4,
                                       box.padding = unit(0.4, "lines"), segment.colo
r = 'grey50',
                                       point.padding = unit(0.3, "lines"), segment.si
ze = 0.3)
p_fdr = p_fdr + scale_color_manual(values=mycolour)
p_fdr = p_fdr + scale_fill_manual(values=mycolour)
p_fdr = p_fdr + theme(legend.position = "none")
p_fdr
```

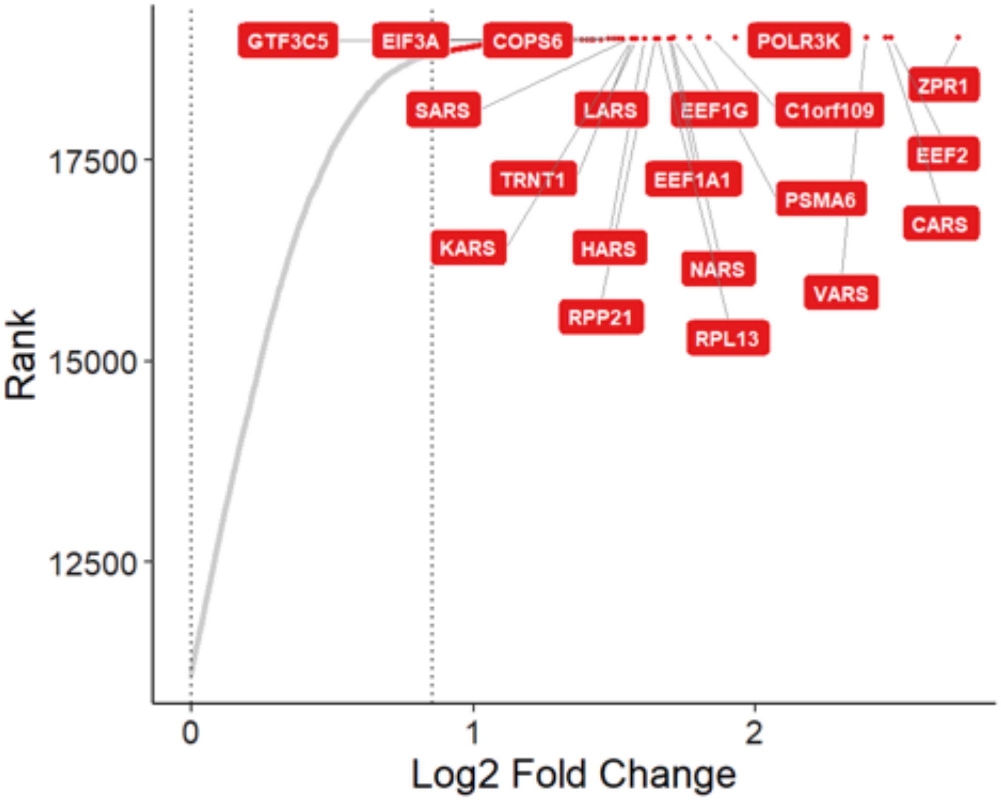


5) RankView to visualize top positive and negative selected genes

```
geneList= dd.rra$LFC
names(geneList) = dd.rra$Official
p4 = RankView(geneList)
p4 = p4 + labs(x = "Log2 Fold Change")
print(p4)
```



```
#positive selection only
no_neg <- sum(dd.rra$LFC<0)
p4 <- p4 + xlim(0,NA) + ylim(no_neg, NA)
print(p4)
```

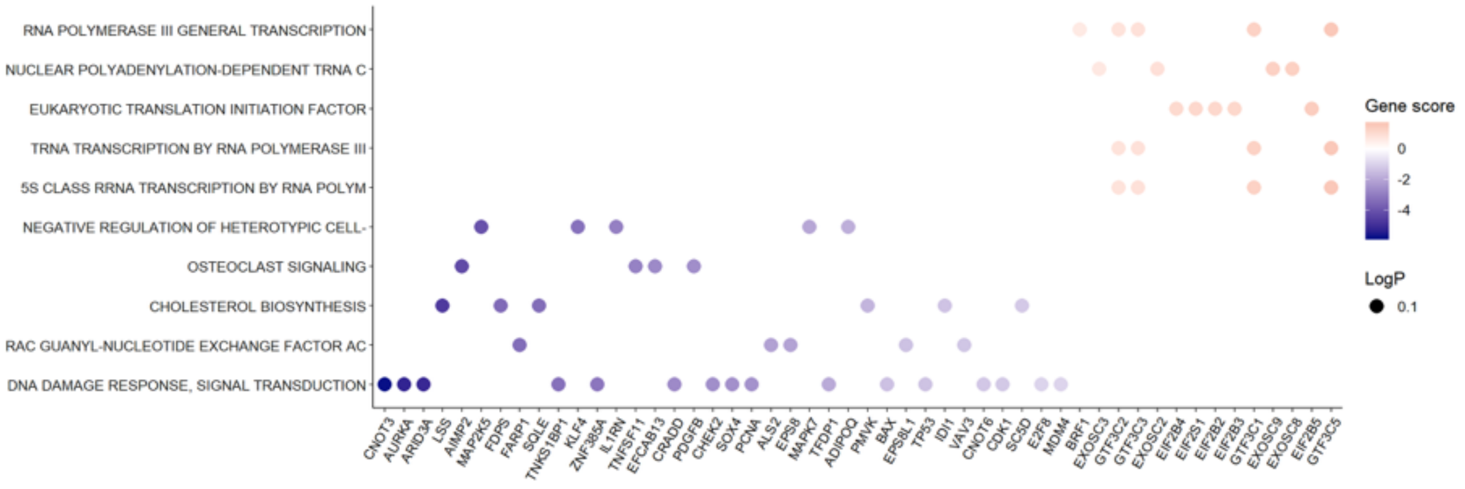


6) Enrichment analysis(GSEA, Gene Set Erichment Analysis)

```
universe = dd.rra$EntrezID
geneList= dd.rra$LFC
names(geneList) = universe

enrich = enrich.GSE(geneList=geneList, keytype = "Entrez", type = "All", organism = "
hsa", pvalueCutoff = 1, pAdjustMethod = "BH",limit = c(3, 100), gmtpath = NA)
```

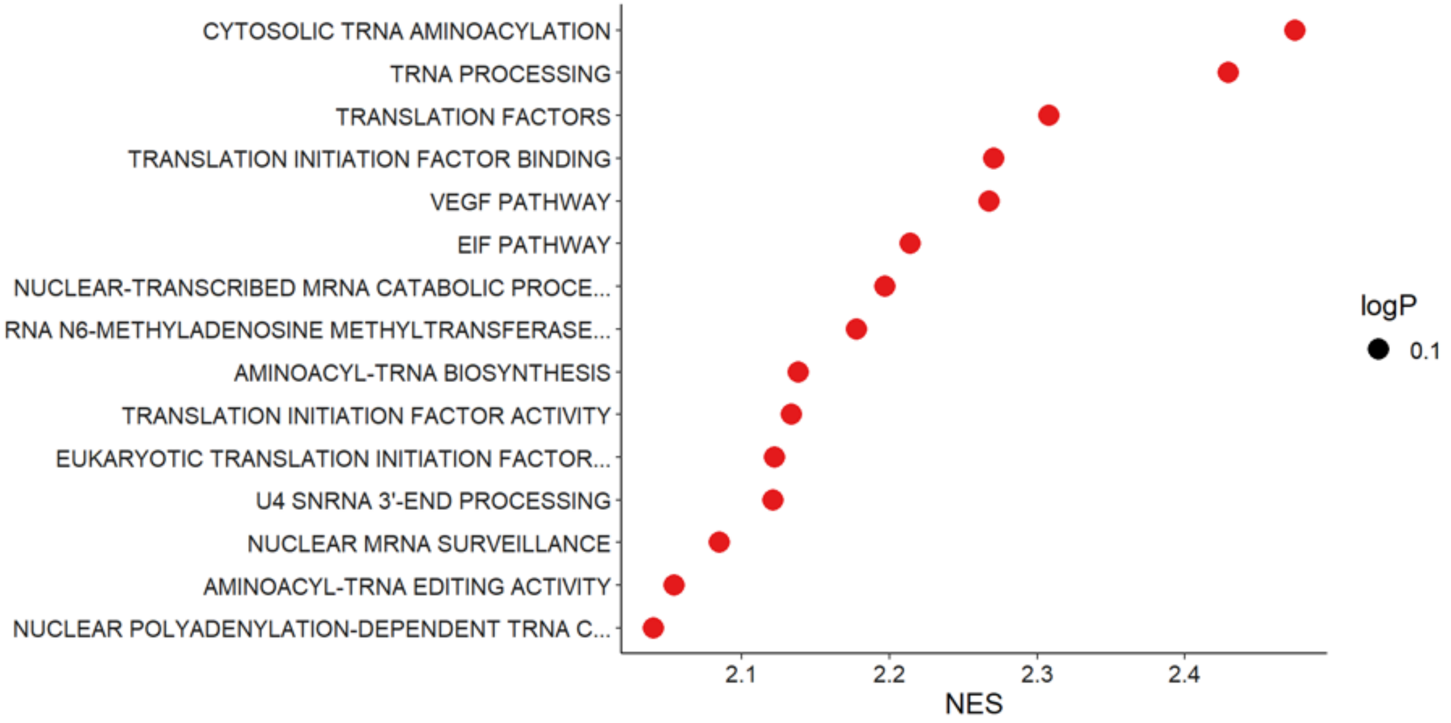
6-a) Visualize selected genes in enriched genesets

```
my_EnrichedGeneView(as.data.frame(enrich), geneList, keytype = "Entrez", gene_cutoff
= c(-log2(1.5), log2(1.5)), top = 5, bottom = 5, charLength = 40) + theme(text = elem
ent_text(colour="black",size = 13, family = "Helvetica"), axis.text.x = element_text(
color = "black", size = 10)) + labs(x=NULL, y=NULL, color = "Gene score", size = "Log
P")
```



6-b) Grid plot for enriched terms in GSEA

```
#EnrichedGSEView(as.data.frame(enrich), decreasing = FALSE, plotTitle = NULL,type = "
All", termNum = 15, charLength = 40)
EnrichedGSEView(as.data.frame(enrich), decreasing = TRUE, plotTitle = NULL, type = "A
ll", termNum = 15, charLength = 40)
```



7) Functional analysis of selected genes (ORT, Over-Representing Test)

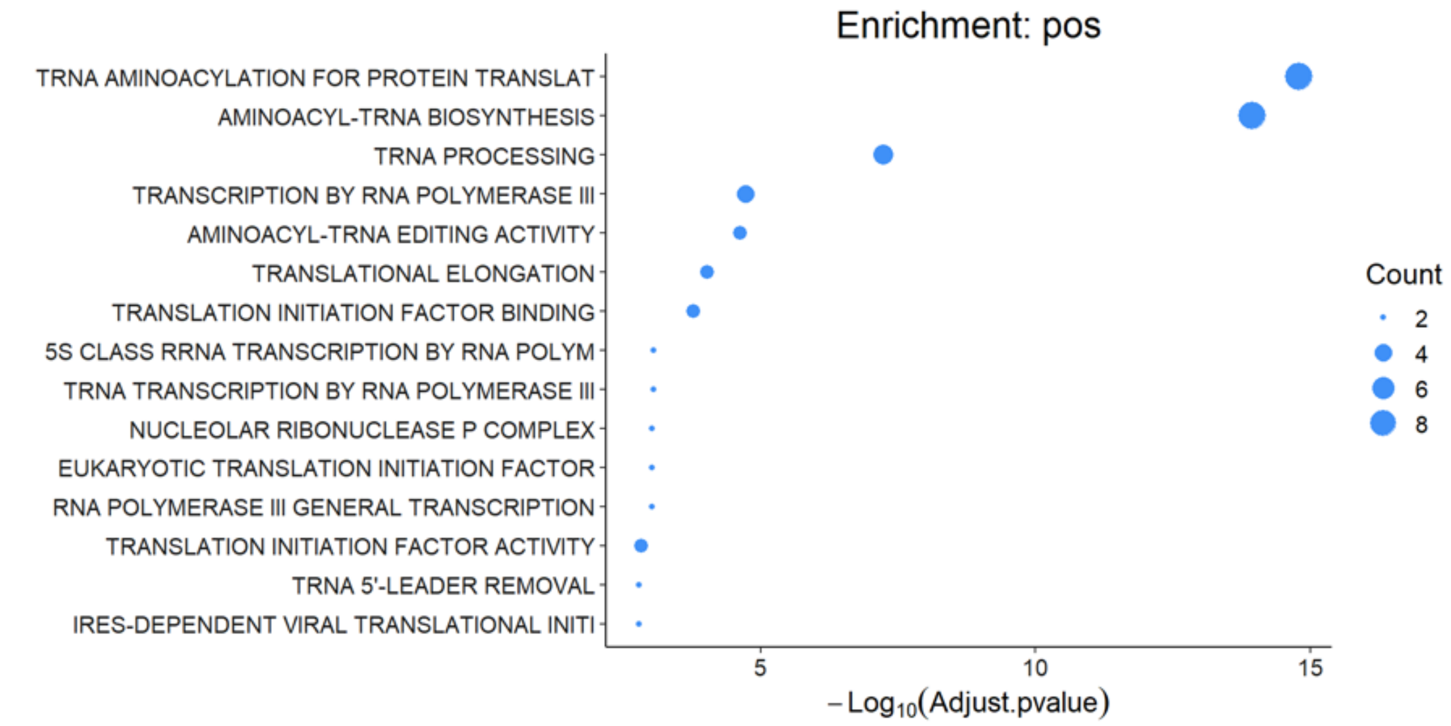
```
#universe = dd$EntrezID
#geneList = dd$LFC; names(geneList) = dd$EntrezID
lfcCutoff <- c(-1,1)
idx1 = (dd.rra$LFC<lfcCutoff[1] & dd.rra$FDR<0.30) ; idx2 = (dd.rra$LFC>lfcCutoff[2]
& dd.rra$FDR<0.30)

#positive selected genes
dd.rra$Official[idx2]
```

```
##  [1] ZPR1     EEF2     CARS     VARS     POLR3K   C1orf109 EEF1G
##  [8] NARS     EEF1A1   RPL13    RPP21    HARS     LARS     TRNT1
## [15] KARS     SARS     GTF3C5   EIF3A    COPS6    CLP1     POLR3B
## [22] CSTF3    POP7     KIAA1429 AARS     GARS     TRMT112  EXOSC8
## [29] GTF3C1   PKMYT1   ABCF1    SPATA5   EIF3F    DPH3     EIF2B4
## [36] ANAPC11
## 20112 Levels: 1-Dec 1-Mar 1-Sep 10-Mar 10-Sep 11-Mar 11-Sep 12-Sep ... ZZZ3
```

```
kegg.pos = enrich.ORT(geneList=geneList[idx2], universe=universe, keytype = "Entrez",
type = "CORUM+GOBP+GOMF+GOCC+KEGG", organism="hsa", pvalueCutoff=1, pAdjustMethod = "
BH", limit = c(3, 100), gmtpath = NA)
```

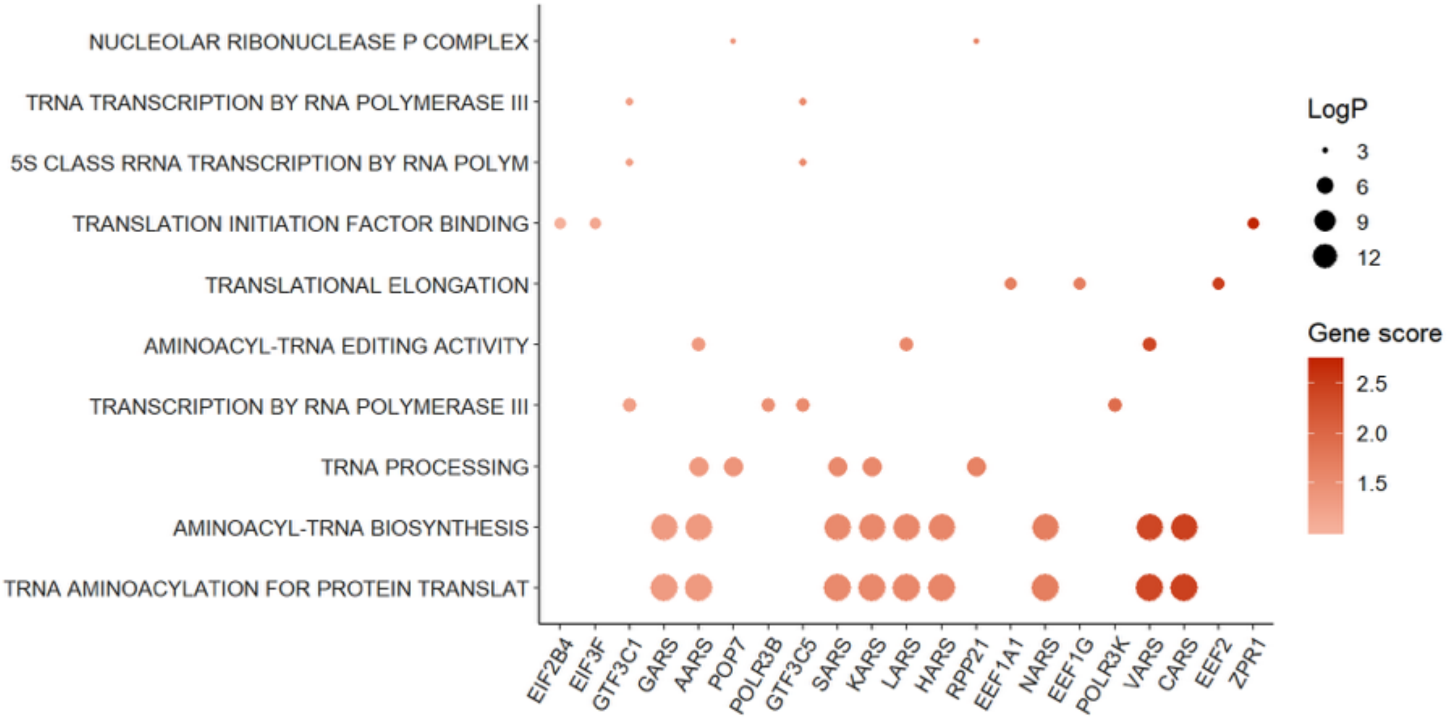
7-a) Grid plot for positively enriched terms

```
EnrichedView(kegg.pos@result, top = 5) + labs(title = "Enrichment: pos")
```



7-b) Visualize selected genes in Top-10 enriched genesets

```
my_EnrichedGeneView(kegg.pos@result, geneList, keytype = "Entrez",gene_cutoff = lfcCu
toff, top = 10, bottom = 0) + theme(text = element_text(colour="black",size = 13, fam
ily = "Helvetica"))
```



For GSE anlysis, no significantly enriched pathway is detectd. For functional analysis(Over-Representing Test) of genes selected by criteria of (LFC>1 & FDR<0.30), 36 genes are found significantly enriched in the A01 samples, compared to the A03 samples. The functions enriched by the 36 genes are mainly related to tRNA and translation, and some others.

2. A02_vs_A03 comparison

1) Load gene and sgrna summary data in MAGeCK RRA results

```
rra.gene_summary = read.table("Run056_A02vsA03.gene_summary.txt", sep='\t', header=TR
UE)
kable(head(rra.gene_summary)) %>% kable_styling() %>% scroll_box(width = "100%")
```

id	num	neg.score	neg.p.value	neg.fdr	neg.rank	neg.goodsgrna	neg.lfc	pos.sco
MAP3K10	4	1.10e-05	0.0000480	0.29648	1	4	-5.9462	0.9635
CTDSP1	4	1.73e-05	0.0000756	0.29648	2	3	-2.1675	0.5017
CAMKK1	4	1.81e-05	0.0000795	0.29648	3	2	-3.4280	0.889
GNS	4	2.13e-05	0.0000953	0.29648	4	4	-4.8297	0.968
ANO5	4	2.56e-05	0.0001140	0.29648	5	3	-2.6450	0.848
TAOK1	4	2.58e-05	0.0001145	0.29648	6	2	-2.5611	0.139

```
rra.sgrna_summary = read.table("Run056_A02vsA03.sgrna_summary.txt", sep='\t', header
= TRUE)
kable(head(rra.sgrna_summary)) %>% kable_styling() %>% scroll_box(width = "100%")
```

sgrna	Gene	control_count	treatment_count	control_mean	treat_mean	LFC
sgRNA_ID_34403	GLCE	570.63/571	25495/17753	570.8200	21275.00	5.2175

sgRNA_ID_34404	GLCE	564.63/613.8	10125/7827.4	588.7000	8902.50	3.9163
sgRNA_ID_47270	TSPYL2	0/17.303	876.5/66.14	1.2192	240.87	6.7680
sgRNA_ID_16406	RPL28	233.26/298.7	2804.4/3441.9	263.9600	3106.80	3.5521
sgRNA_ID_06971	GCG	362.4/377.93	18647/666.69	370.0900	3526.00	3.2486
sgRNA_ID_29878	RAB35	329.37/369.74	2524.9/4083.9	348.9700	3211.20	3.1982

2) Top-10 genes and sgRNAs for positive selection

```
dd.rra = ReadRRA(rra.gene_summary, organism = "hsa")
dd.rra <- dd.rra[order(-dd.rra$LFC),]
kable(head(dd.rra, 10)) %>% kable_styling(full_width = F)
```

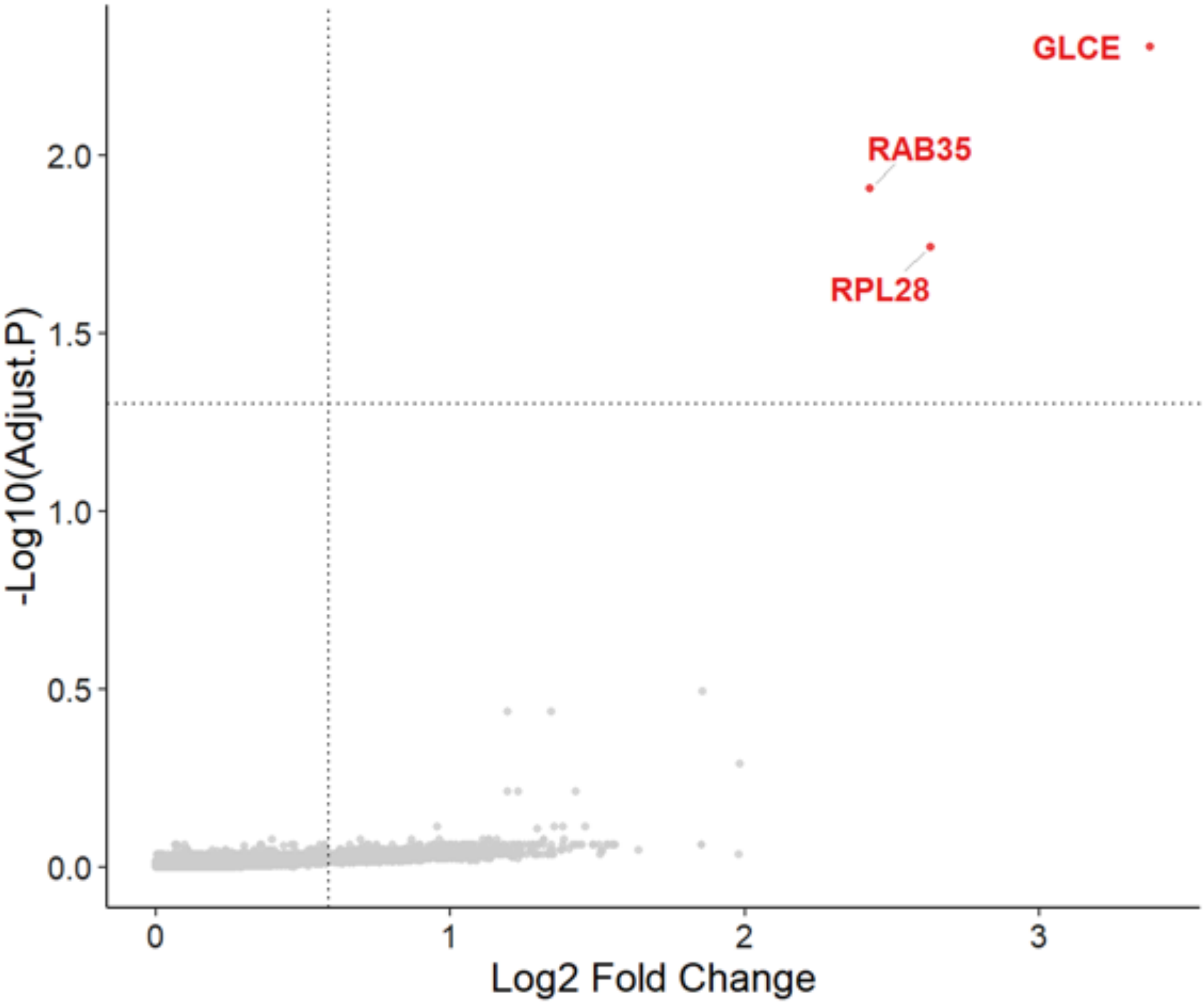
	Official	EntrezID	LFC	FDR
26035	GLCE	26035	3.3772	0.004950
6158	RPL28	6158	2.6289	0.018152
11021	RAB35	11021	2.4239	0.012376
382	ARF6	382	1.9828	0.514144
5438	POLR2I	5438	1.9796	0.921571
5901	RAN	5901	1.8573	0.320545
3107	HLA-C	3107	1.8540	0.864242
431707	LHX8	431707	1.8510	0.864242
51477	ISYNA1	51477	1.6381	0.899025
5148	PDE6G	5148	1.5576	0.864242

```
dd.sgrna = ReadsgRRA(rra.sgrna_summary)
dd.sgrna <- dd.sgrna[order(-dd.sgrna$LFC),]
kable(head(dd.sgrna,10)) %>% kable_styling(full_width = F)
```

	sgrna	Gene	LFC	FDR
3	sgRNA_ID_47270	TSPYL2	6.7680	0
8	sgRNA_ID_14348	POLE	5.3416	0
1	sgRNA_ID_34403	GLCE	5.2175	0
2	sgRNA_ID_34404	GLCE	3.9163	0
11	sgRNA_ID_10444	AFF3	3.5928	0
4	sgRNA_ID_16406	RPL28	3.5521	0
16	sgRNA_ID_16407	RPL28	3.4793	0
9	sgRNA_ID_18230	SRPR	3.3077	0
5	sgRNA_ID_06971	GCG	3.2486	0
6	sgRNA_ID_29878	RAB35	3.1982	0

3) VolcanoView for positive selected genes

```
p1 = my_VolcanoView(dd.rra[dd.rra$LFC >=0, ], x = "LFC", y = "FDR", Label = "Official
", top=20)
p1 <- p1 + xlim(0,NA)
print(p1)
```



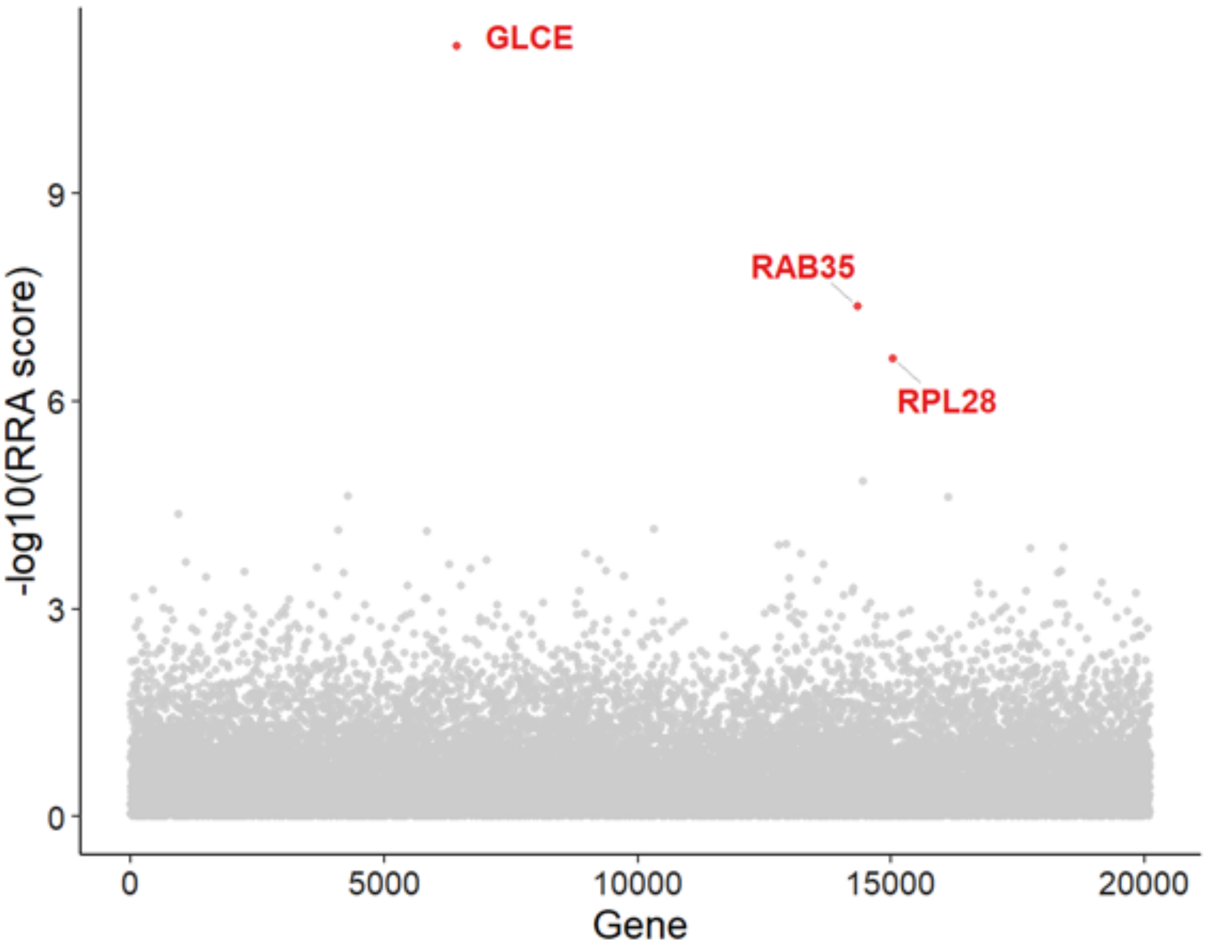
4) -log10(RRAscore) and -log10(pos.fdr) plots

```
df <- rra.gene_summary[,c('id','neg.score', 'neg.fdr', 'neg.lfc', 'pos.score', 'pos.f
dr', 'pos.lfc')]

x_cutoff = log2(1.5); y_cutoff = 0.05
df$group="NoSig"
df$group[df[,'pos.lfc']>x_cutoff & df[,'pos.fdr']<y_cutoff] = "Up"
df$group[df[,'neg.lfc']< -x_cutoff & df[,'neg.fdr']<y_cutoff] = "Down"
df$group <- as.factor(df$group)
levels(df$group) <- c("NoSig", "Up", "Down")
kable(table(df$group)) %>% kable_styling(full_width = F)
```

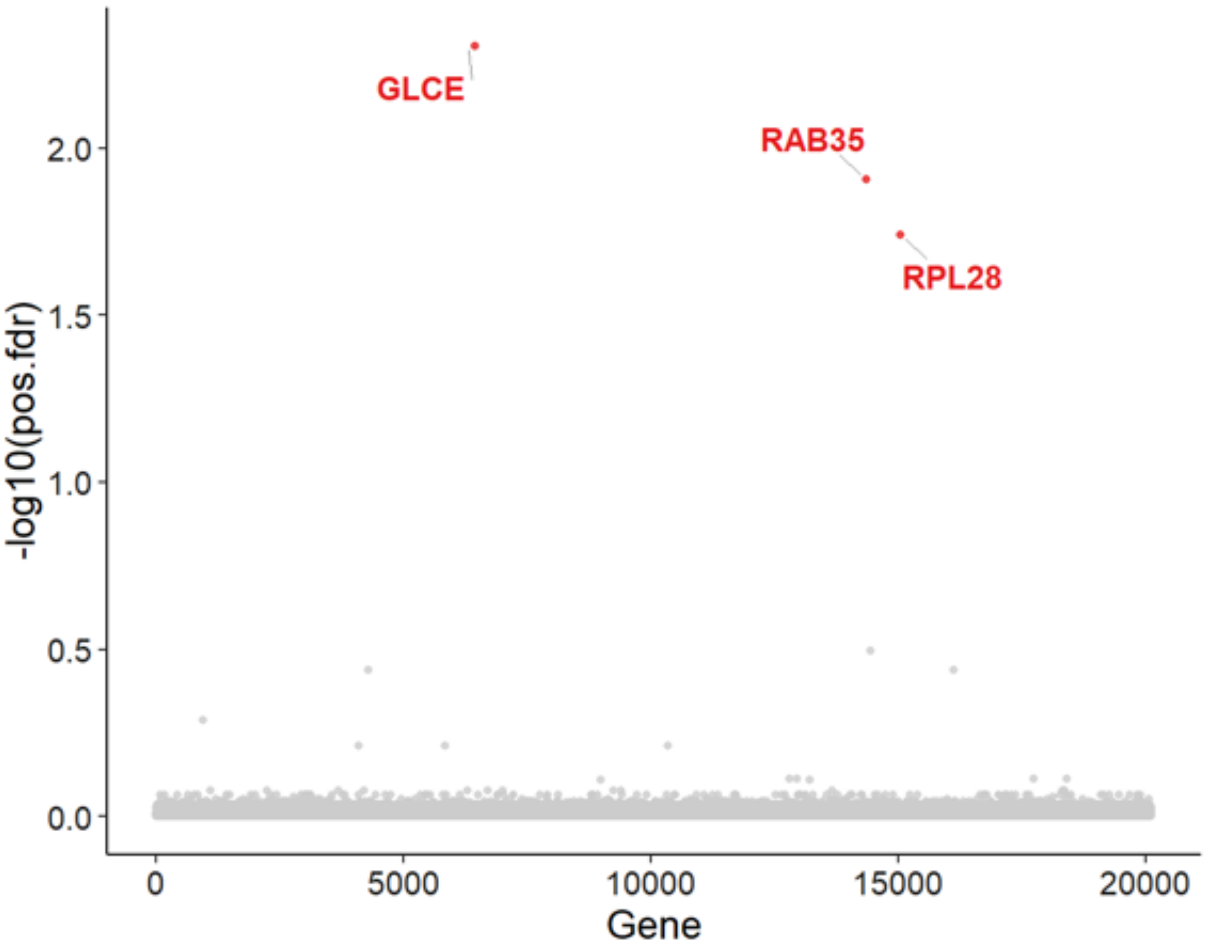
Var1	Freq
NoSig	20109
Up	3

Down	0

```
df[, c('neg.fdr', 'pos.fdr')] = -log10(df[, c('neg.fdr', 'pos.fdr')])
df[, c('neg.score', 'pos.score')] = -log10(df[, c('neg.score', 'pos.score')])

mycolour=c("NoSig"="gray80",  "Up"="#e41a1c", "Down"="#377eb8")

# Sort gene symbol in an alphabetical order
df = df[order(df$id),]
df$GeneOrder = c(1:nrow(df))

df$Label = as.character(df$id)
idx_up = which(df$group=="Up")
idx_down = which(df$group=="Down")
idx = unique(idx_up, idx_down)
df$Label[setdiff(1:nrow(df), idx)] = ""
df$Label = factor(df$Label, levels = setdiff(unique(df$Label), ""))

# -log10 RRA scrore plot
p_sc = ggplot(df, aes(x=GeneOrder, y=pos.score, colour=group, fill=group))
p_sc = p_sc + geom_jitter(position = "jitter", show.legend = FALSE, alpha=0.8, size = 1)
p_sc = p_sc + theme(text = element_text(colour="black",size = 14, family = "Helvetica"),
              plot.title = element_text(hjust = 0.5, size=16),
              axis.text = element_text(colour="gray10"))
p_sc = p_sc + theme(axis.line = element_line(size=0.5, colour = "black"),
              panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
              panel.border = element_blank(), panel.background = element_blank())
p_sc = p_sc + labs(x='Gene', y='-log10(RRA score)', title='')
p_sc = p_sc + geom_text_repel(aes(x=df[idx,'GeneOrder'], y=df[idx,'pos.score'], label = Label), data=df[idx,],
                                       fontface = 'bold', size = 4,
                                       box.padding = unit(0.4, "lines"), segment.color = 'grey50',
                                       point.padding = unit(0.3, "lines"), segment.size = 0.3)
p_sc = p_sc + scale_color_manual(values=mycolour)
p_sc = p_sc + scale_fill_manual(values=mycolour)
p_sc = p_sc + theme(legend.position = "none")
p_sc
```

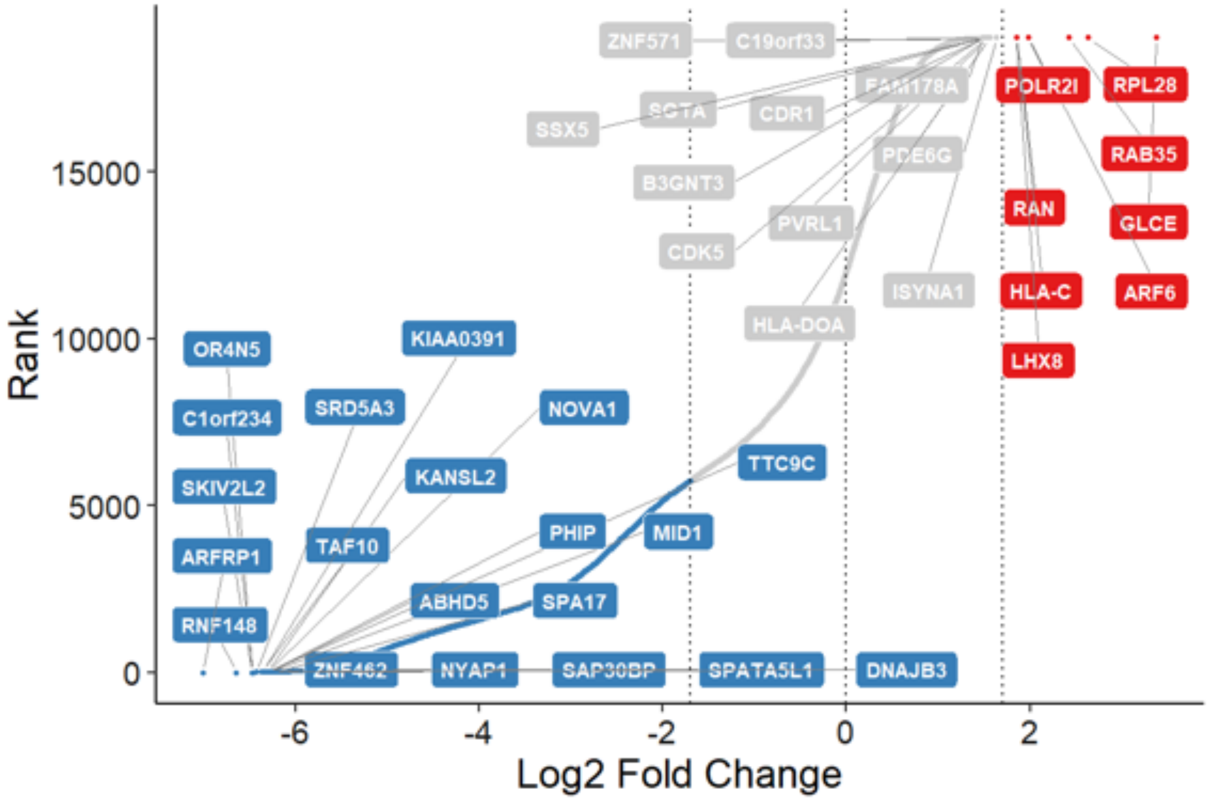

GLCE
RAB35
RPL28
-log10(RRA score)
9
6
3
0
0
5000
10000
15000
20000
Gene

```
# -log10 pos.fdr plot
p_fdr = ggplot(df, aes(x=GeneOrder, y=pos.fdr, colour=group, fill=group))
p_fdr = p_fdr + geom_jitter(position = "jitter", show.legend = FALSE, alpha=0.8, size = 1)
p_fdr = p_fdr + theme(text = element_text(colour="black",size = 14, family = "Helvetica"),
              plot.title = element_text(hjust = 0.5, size=16),
              axis.text = element_text(colour="gray10"))
p_fdr = p_fdr + theme(axis.line = element_line(size=0.5, colour = "black"),
              panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
              panel.border = element_blank(), panel.background = element_blank())
p_fdr = p_fdr + labs(x='Gene', y='-log10(pos.fdr)', title='')
p_fdr = p_fdr + geom_text_repel(aes(x=df[idx,'GeneOrder'], y=df[idx,'pos.fdr'], label = Label), data=df[idx,],
                                       fontface = 'bold', size = 4,
                                       box.padding = unit(0.4, "lines"), segment.color = 'grey50',
                                       point.padding = unit(0.3, "lines"), segment.size = 0.3)
p_fdr = p_fdr + scale_color_manual(values=mycolour)
p_fdr = p_fdr + scale_fill_manual(values=mycolour)
p_fdr = p_fdr + theme(legend.position = "none")
p_fdr
```

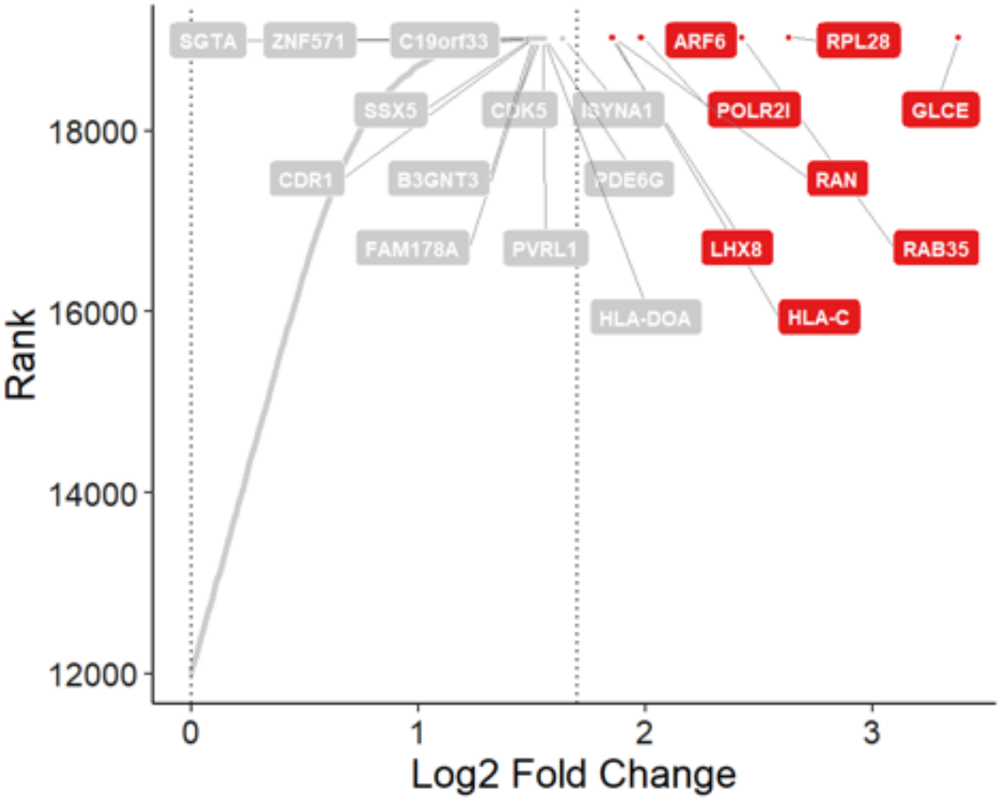


5) RankView to visualize top positive and negative selected genes

```
geneList= dd.rra$LFC
names(geneList) = dd.rra$Official
p4 = RankView(geneList)
p4 = p4 + labs(x = "Log2 Fold Change")
print(p4)
```



```
#positive selection only
no_neg <- sum(dd.rra$LFC<0)
p4 <- p4 + xlim(0,NA) + ylim(no_neg, NA)
print(p4)
```

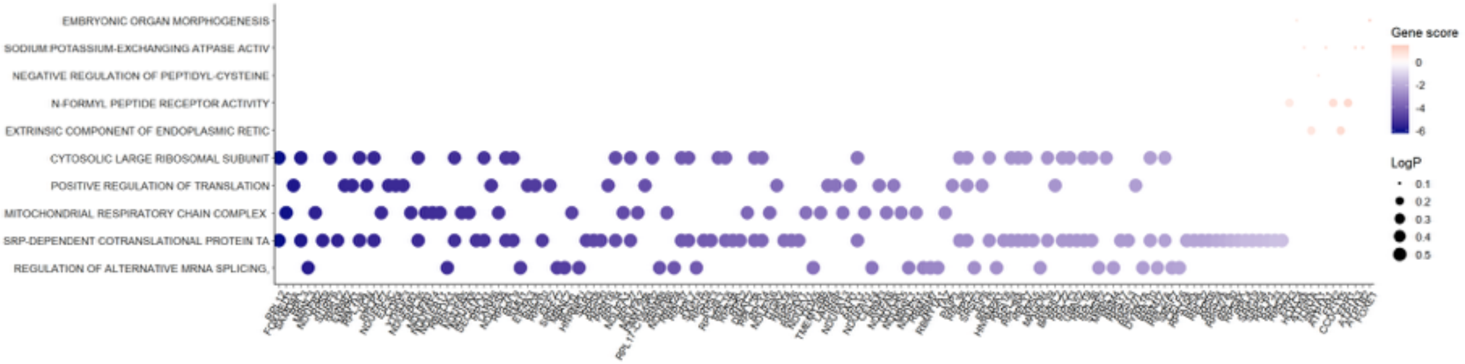


6) Enrichment analysis(GSEA, Gene Set Erichment Analysis)

```
universe = dd.rra$EntrezID
geneList= dd.rra$LFC
names(geneList) = universe

enrich = enrich.GSE(geneList=geneList, keytype = "Entrez", type = "All", organism = "
hsa", pvalueCutoff = 1, pAdjustMethod = "BH",limit = c(3, 100), gmtpath = NA)
```

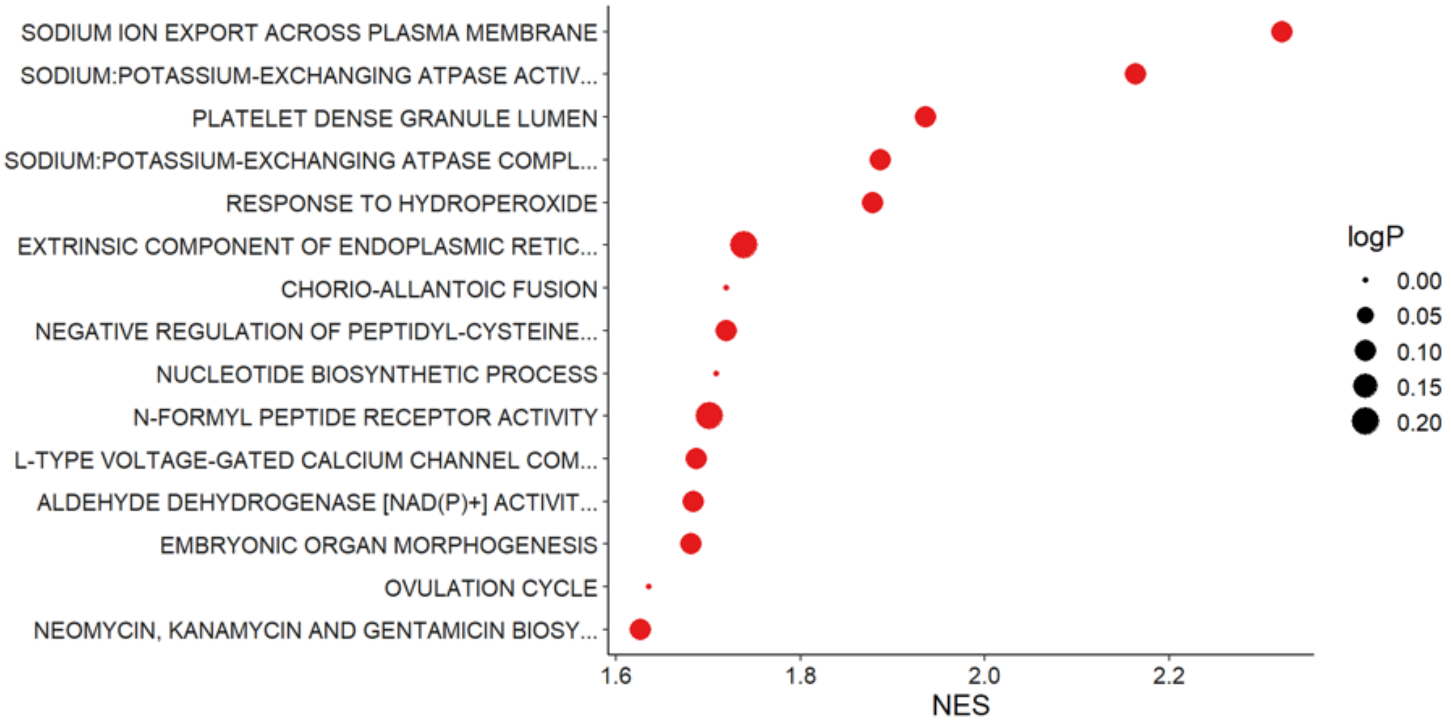
6-a) Visualize selected genes in enriched genesets

```
my_EnrichedGeneView(as.data.frame(enrich), geneList, keytype = "Entrez", gene_cutoff
= c(-log2(1.5), log2(1.5)), top = 5, bottom = 5, charLength = 40) + theme(text = elem
ent_text(colour="black",size = 13, family = "Helvetica"), axis.text.x = element_text(
color = "black", size = 10)) + labs(x=NULL, y=NULL, color = "Gene score", size = "Log
P")
```



6-b) Grid plot for enriched terms in GSEA

```
#EnrichedGSEView(as.data.frame(enrich), decreasing = FALSE, plotTitle = NULL,type = "
All", termNum = 15, charLength = 40)
EnrichedGSEView(as.data.frame(enrich), decreasing = TRUE, plotTitle = NULL, type = "A
ll", termNum = 15, charLength = 40)
```



7) Functional analysis of selected genes (ORT, Over-Representing Test)

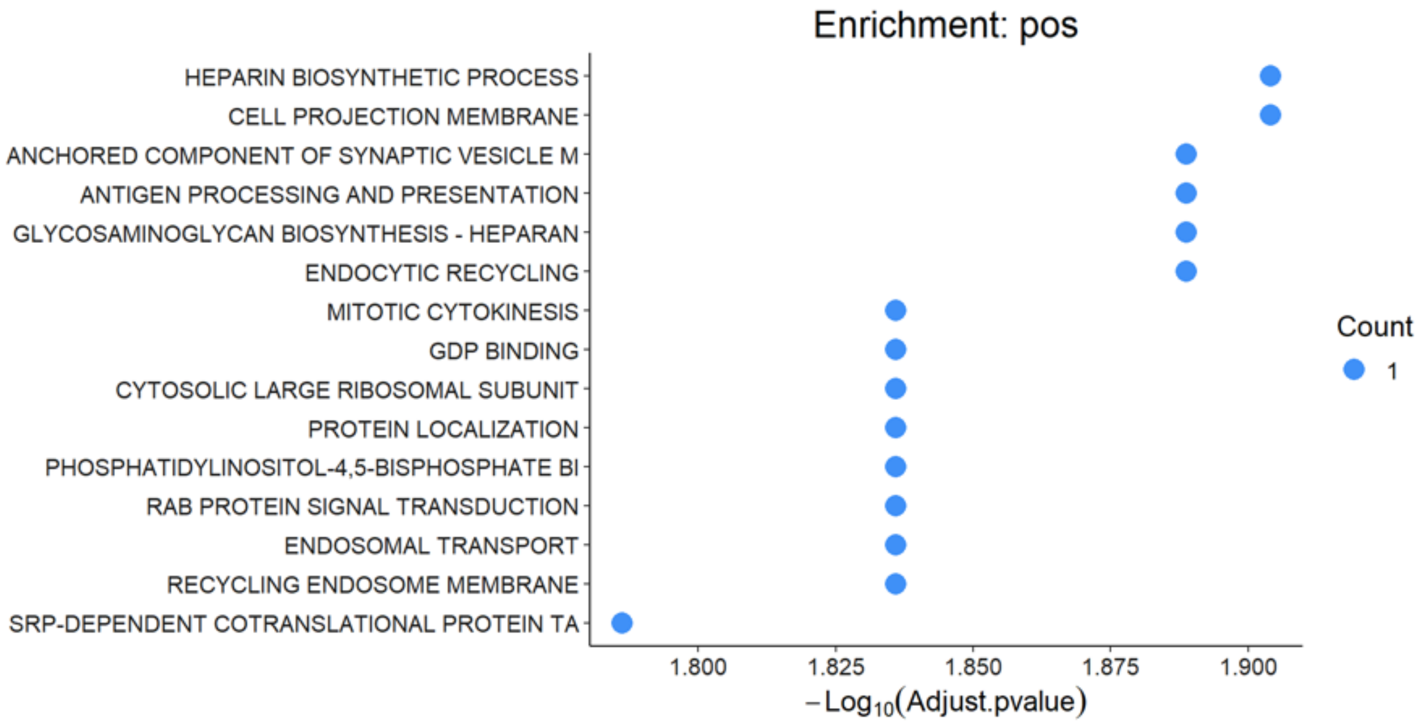
```
#universe = dd$EntrezID
#geneList = dd$LFC; names(geneList) = dd$EntrezID
lfcCutoff <- c(-1,1)
idx1 = (dd.rra$LFC<lfcCutoff[1] & dd.rra$FDR<0.30) ; idx2 = (dd.rra$LFC>lfcCutoff[2]
& dd.rra$FDR<0.30)

#positive selected genes
dd.rra$Official[idx2]
```

```
## [1] GLCE  RPL28 RAB35
## 20112 Levels: 1-Dec 1-Mar 1-Sep 10-Mar 10-Sep 11-Mar 11-Sep 12-Sep ... ZZZ3
```

```
kegg.pos = enrich.ORT(geneList=geneList[idx2], universe=universe, keytype = "Entrez",
type = "CORUM+GOBP+GOMF+GOCC+KEGG", organism="hsa", pvalueCutoff=1, pAdjustMethod = "
BH", limit = c(3, 100), gmtpath = NA)
```

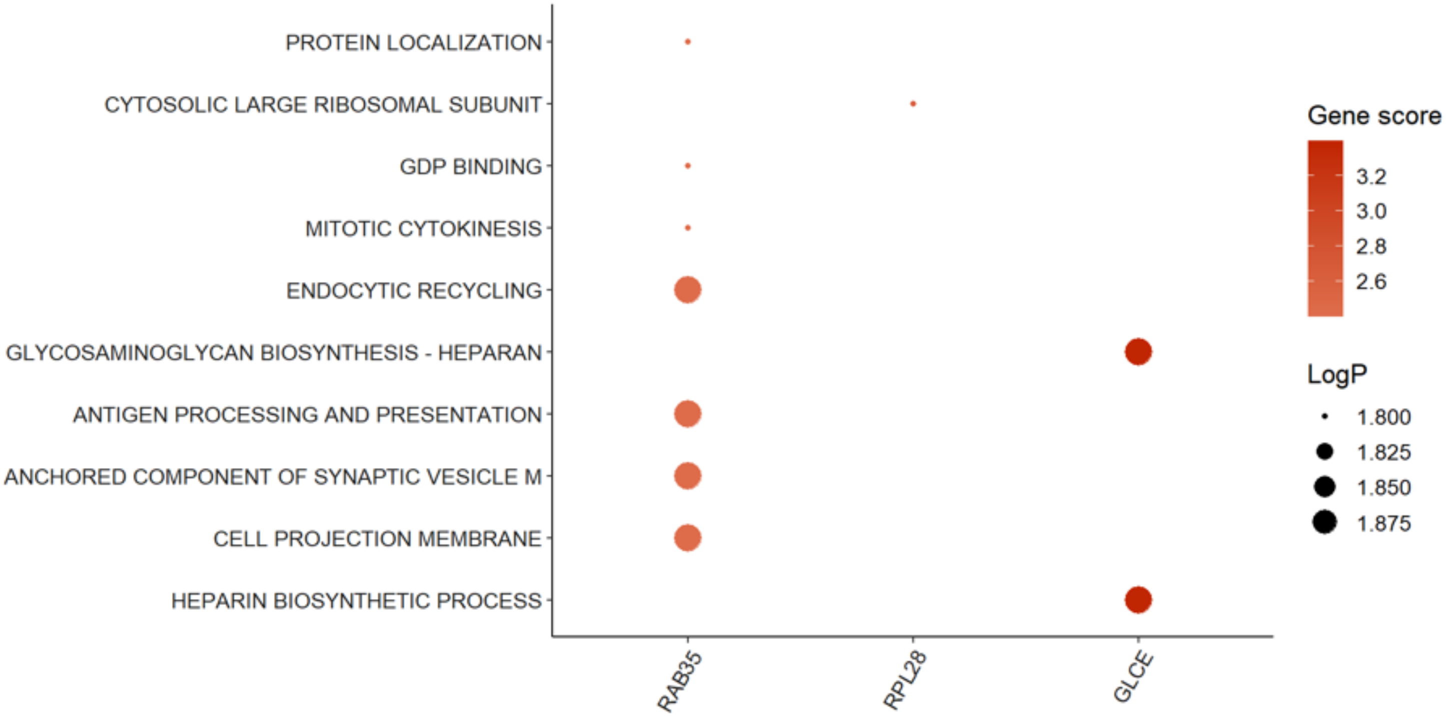
7-a) Grid plot for positively enriched terms

```
EnrichedView(kegg.pos@result, top = 5) + labs(title = "Enrichment: pos")
```



7-b) Visualize selected genes in Top-10 enriched genesets

```
my_EnrichedGeneView(kegg.pos@result, geneList, keytype = "Entrez",gene_cutoff = lfcCutoff, top = 10, bottom = 0) + theme(text = element_text(colour="black",size = 13, family = "Helvetica"))
```



For GSE anlysis, no significantly enriched pathway is detectd. For functional analysis(Over-Representing Test) of genes selected by criteria of (LFC>1 & FDR<0.30), 3 genes are found significantly enriched in the A02 samples, compared to the A03 samples. The three genes are GLCE, RPL28, and RAB35.

=== End of Report ===