

Transcriptomics Final Report

Zunqiu Wang and Lindsay Reisman

8/23/2021

Introduction

The data set we used is called 'RNA-seq of granulocytes derived from control and AK2 depleted human HSPCs', and was obtained from the NCBI GEO database (GSE179320). The data contains 18 total samples which contains two factors cell and condition. There are two levels of the condition, (AAVS1 CRISPR and AK2 CRISPR) and three levels of cell (Promyelocyte, Myelocyte, Neutrophil) and three replicates of all cells.

The study focuses on Reticular Dysgenesis (RD) which is a very rare but most severe form of severe combined immunodeficiency (SCID). Reticular Dysgenesis is caused by mutations in a protein coding gene known as Adenylate Kinase 2 (AK2). AK2 is found in the mitochondria and responsible for catalyzing phosphate groups between ATP, ADP and AMP. Overall, this gene is known to be important to maintaining energy within a cell and participant in ATP metabolic pathways. RD is generally characterized by lower levels of neutrophils and lymphocytes. In this study, AK2 CRISPR knock out and AAVS1 CRISPR, a safe harbor control, conditions are treated to Hematopoietic stem and progenitor cells at different maturation stages. RNA was extracted from HSPCs using a Qiagen RNeasy Plus Micro kit and the libraries were sequenced on Illumina NovaSeq.

The goal of our project is to utilize various transcriptome analysis tools and create visuals to make concise interpretations on a dataset. By using these tools we hope to get a better understanding of why AK2 is depleted and how that affects myelopoiesis and its metabolic pathway.

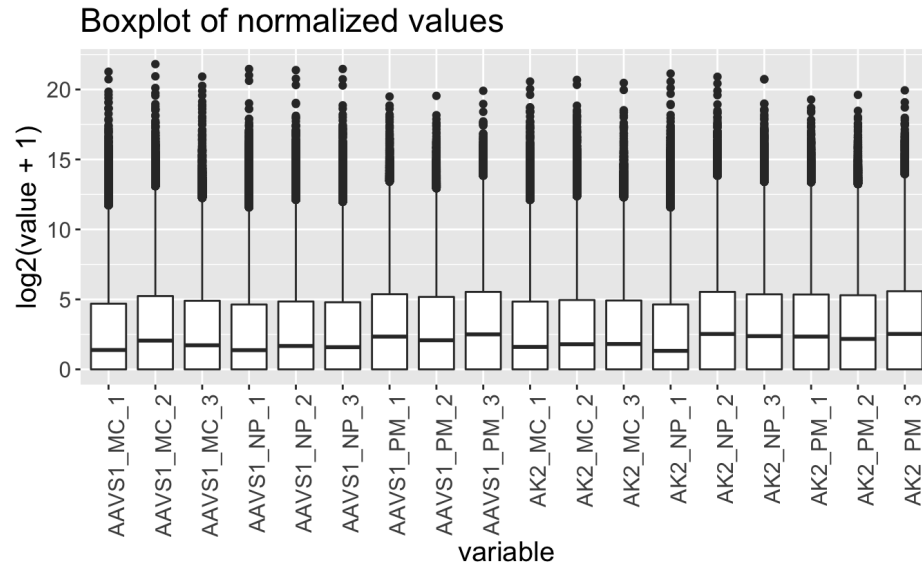
Methods

The workflow that we generated is as follows. We first loaded the dataset into R for manipulation using the `getGEO()` command. After counting the raw counts, we normalized the dataset and created a boxplot to confirm the dataset was properly normalized across all cell types and conditions. Using the `DESeqDataSetFromMatrix()` command storing our count matrix alongside our defined expdesign. We used a `rowSum >= 10` to prefilter out the counts below 10 to help not skew the analysis. We ran our count matrix through `DESeq()` command and identified the coef vectors for each factor then using the `results()` we find the interaction term. Then we filter the genes based on `padj < 0.05` and `log2foldchange` for `>1` or `< -1` for significance. Using the `commdand` order to sort the values in descending order in the `padj` column. From this we were able to identify the 3 samples that were mosly highly signifigant gene. Next, we plotted the first 3 samples and created a boxplot of each sample. We created a volcano plot to visually see what is differentially expressed up regulated vs. downregualted. Using `ggplot2` we created a differential PCA plot using `diffexpvalues` df. Determine the best k value for our dataset calculating for the average sil value and then plotting those values. Using `cluster()` library we were able to cluster with `cuttree` with a `k=9`. Created a silhouette plot based off our differential expression cluster silhouette. We then created a heat map using `k=9` rows. Then we created a Hierarchical clustering plots comparing our expdesign against cell, condition and interaction to see how they all cluster. Using libraries `clusterProfiler` and `org.Hs.eg.db`. we created our GO term analysis and created go terms per cluster group to get a better idea

Results

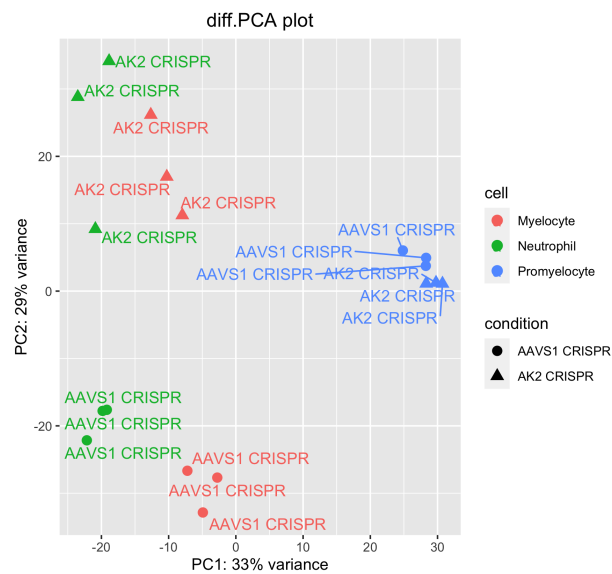
We looked at our normalized boxplot values to confirm that our dataset was normalized correctly.

```
knitr::include_graphics("/Users/lindsayreisman/Desktop/normalizedboxplot.png")
```



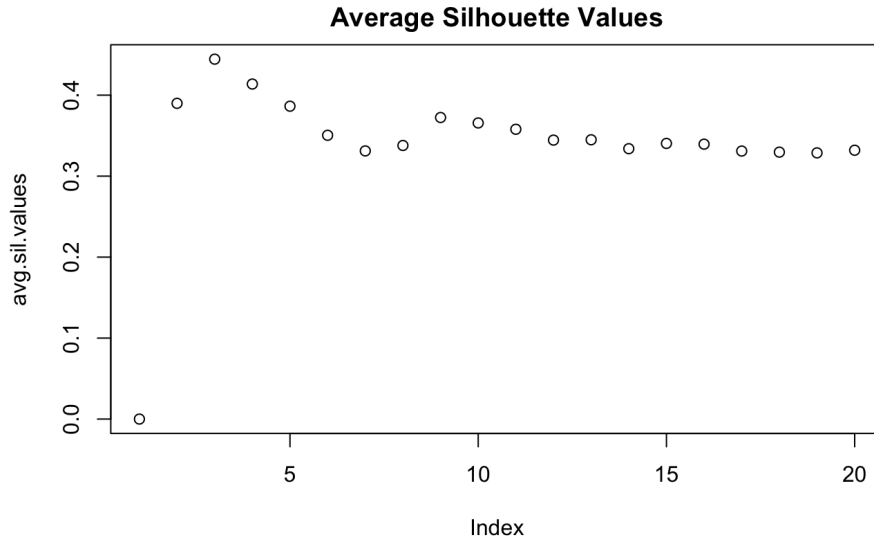
From the differential PCA plot, it is important to note the disparity between the AK2 CRISPR and the AAVS1 CRISPR in both Myelocytes and Neutrophil. For the Promyelocytes, it is hard to differentiate the difference between the AAVS1 and AK2 CRISPR.

```
knitr::include_graphics("/Users/lindsayreisman/diffexpvalues.PCA.png")
```



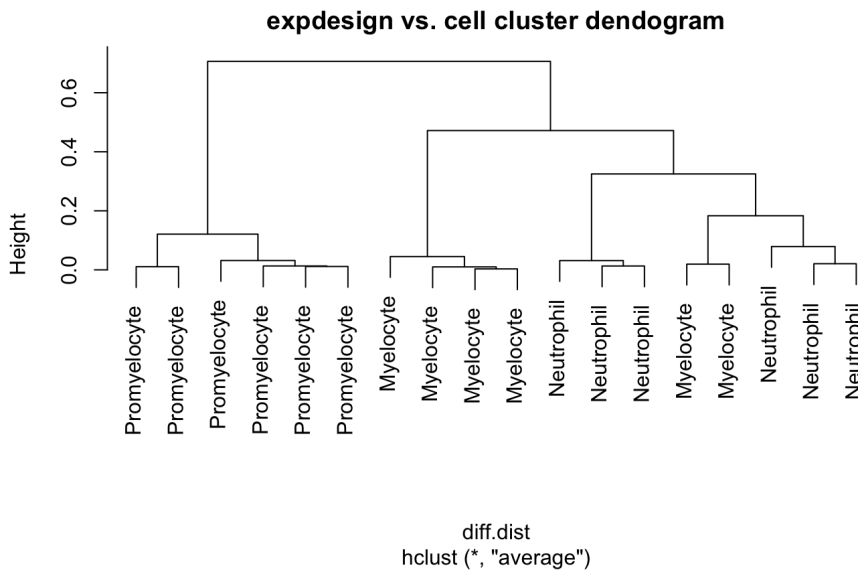
From the average Silhouette model, we can make an assumption as to what our k should be. From our presentation, and conversation with Professor, we decided to target our k = 9.

```
knitr::include_graphics("/Users/lindsayreisman/Downloads/Average Silhouette.png")
```

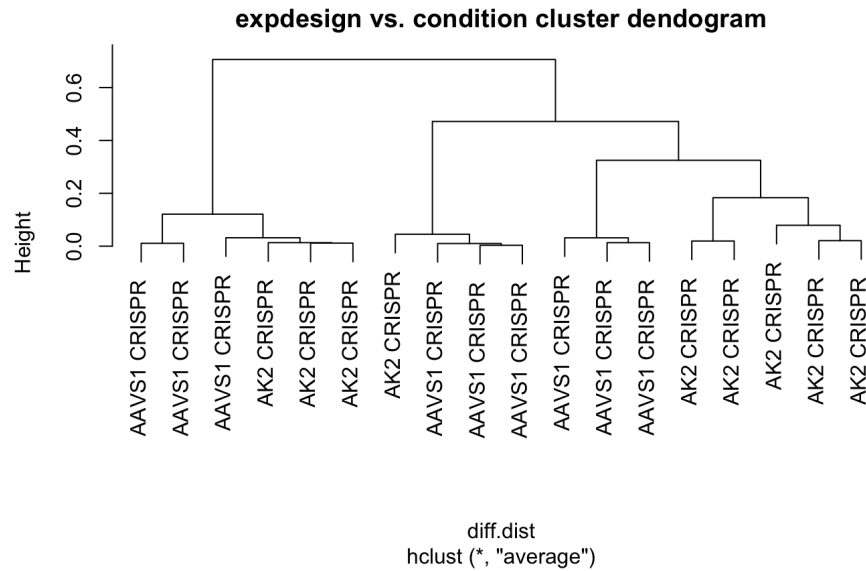


From the cluster dendrogram, we can tell that interesting part is to see how closely the related the myelocytes and the neutrophil are. Which is interesting because we can see the differences from the above PCA. The promyeloblast is the lowest maturation stage, and this leads us to think that maybe the Promyelocyte is not as affected by the AK2 knock-out than the Neutrophil or the Myelocyte. Some samples of the same samples DO NOT cluster together

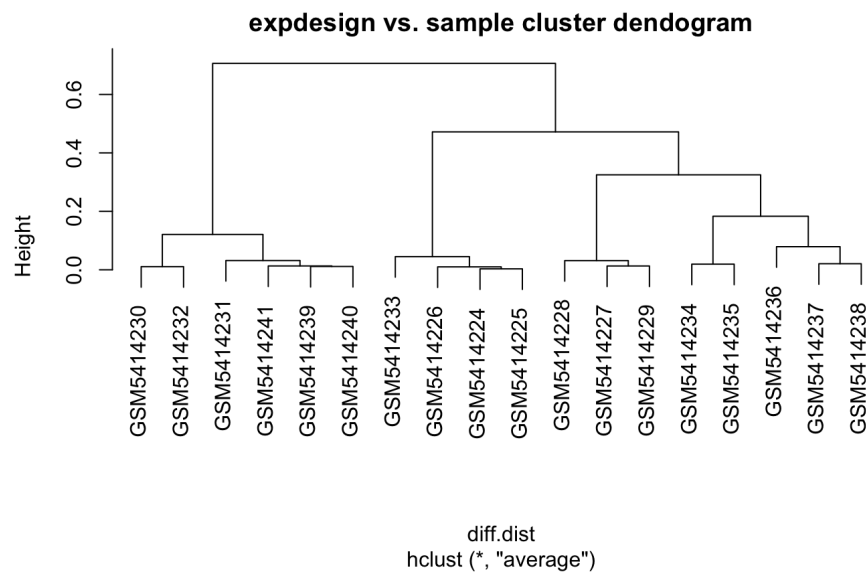
```
knitr::include_graphics("/Users/lindsayreisman/Downloads/Archive/final.proj.int copy/expdesignvs.cell.png")
```



```
knitr::include_graphics("/Users/lindsayreisman/Downloads/Archive/final.proj.int copy/expdesignvscondition.png")
```



```
knitr::include_graphics("/Users/lindsayreisman/Downloads/Archive/final.proj.int copy/expdesignvssample.png")
```



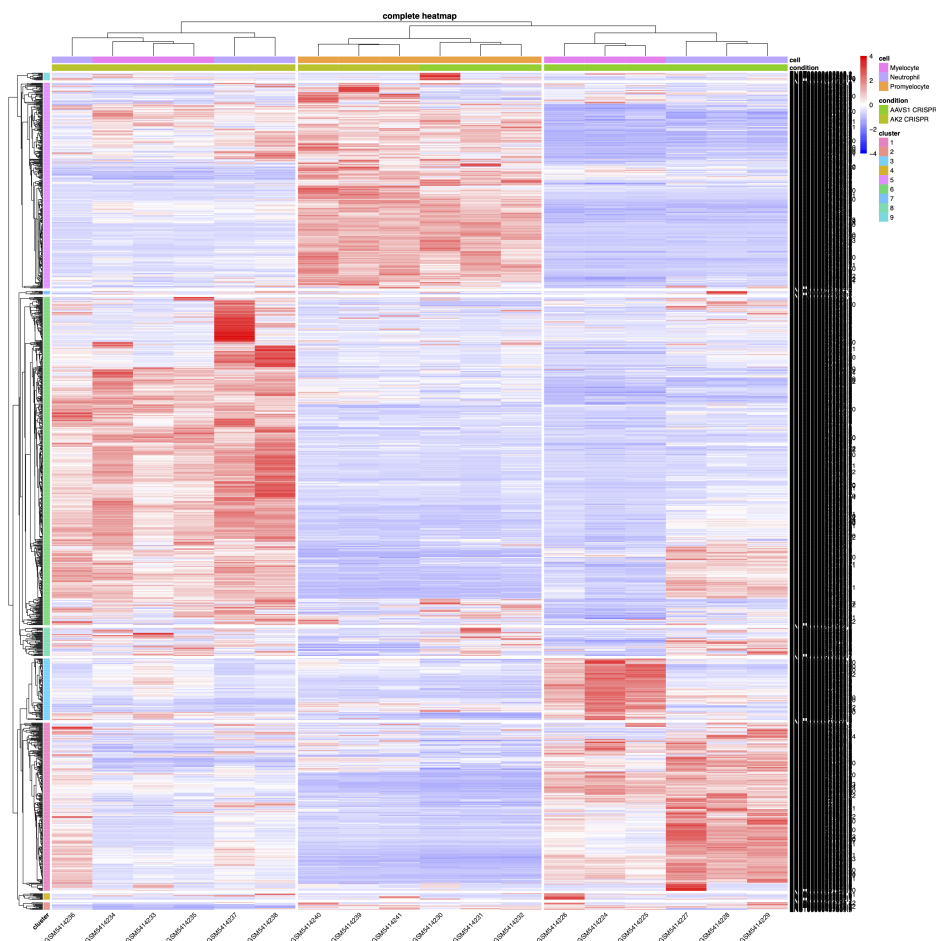
So here we see which genes are differentially expressed using a volcano plot.

```
knitr::include_graphics("/Users/lindsayreisman/Downloads/Archive/final.proj.int copy/volcano.png")
```



Here we see the complete heatmap of our 9 clusters. You can see how large the cluster groups 1,6,5. Again, most of the higher expression is seen under the AK2 condition of Neutrophils and Myelocytes for cluster 5.

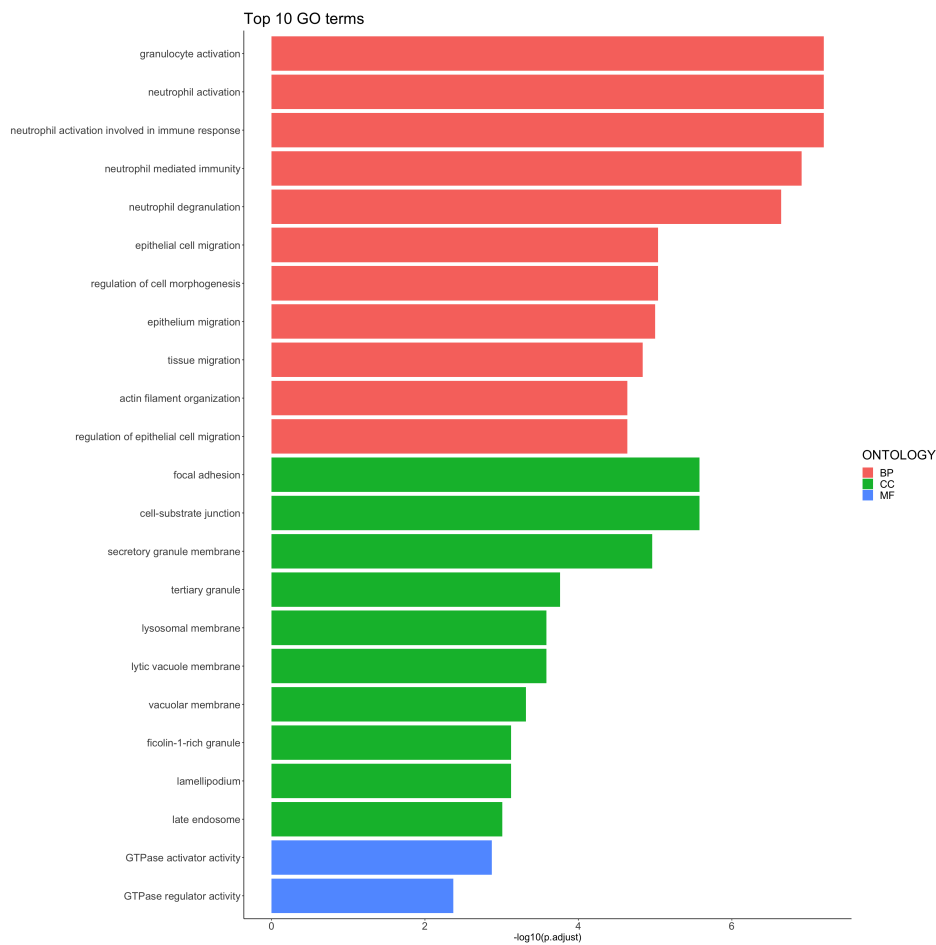
```
knitr::include_graphics("/Users/lindsayreisman/Desktop/complete_heatmap.int.png")
```



From the GO terms, we see that the biological process is more prevalent than the cellular component and

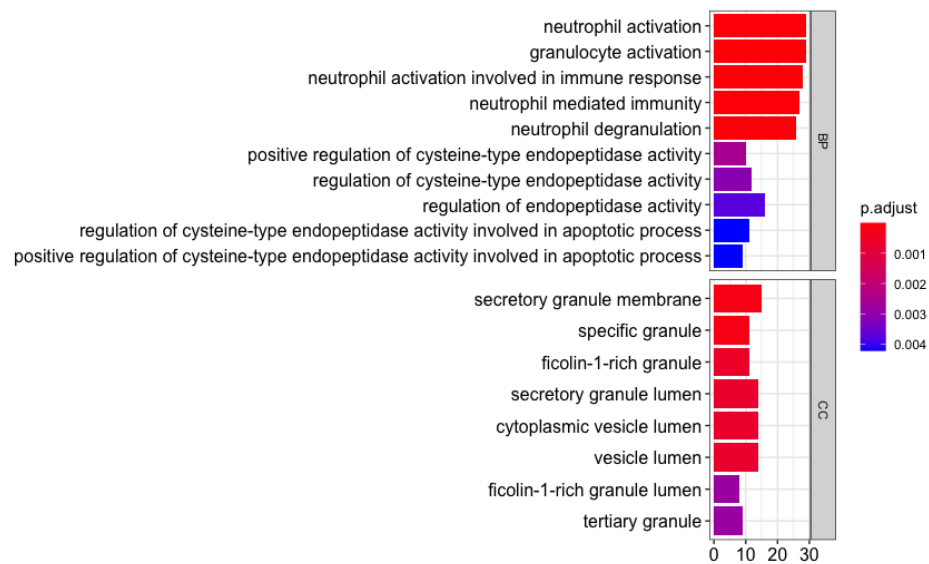
molecular function. Within the biological process, there are a lot of enrichment in neutrophil activation and immune response activation.

```
knitr::include_graphics("/Users/lindsayreisman/Desktop/top10go.png")
```



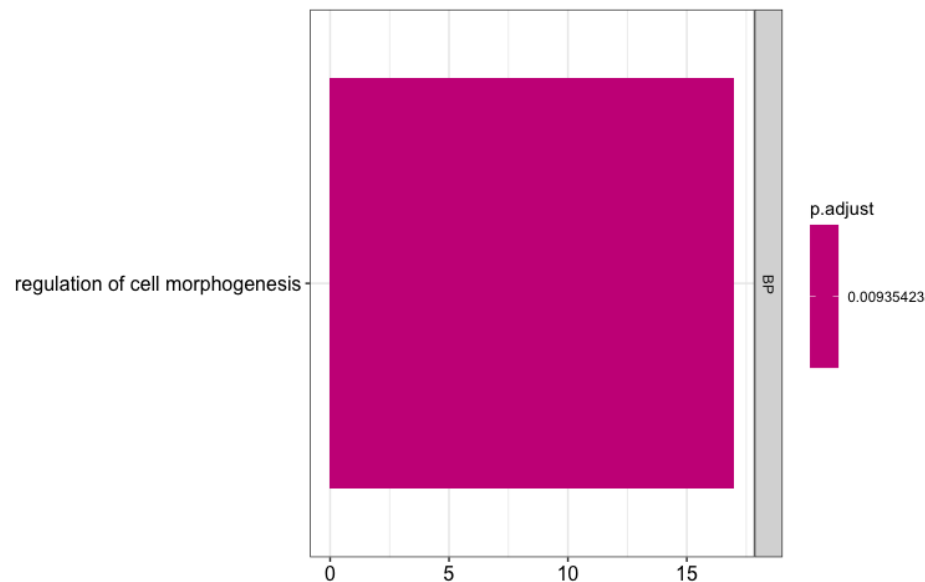
Group1 bar GO enrichment confirm activity/activation in the immune response.

```
knitr::include_graphics("/Users/lindsayreisman/Downloads/final.proj.int only/grp1.GO.bar.png")
```



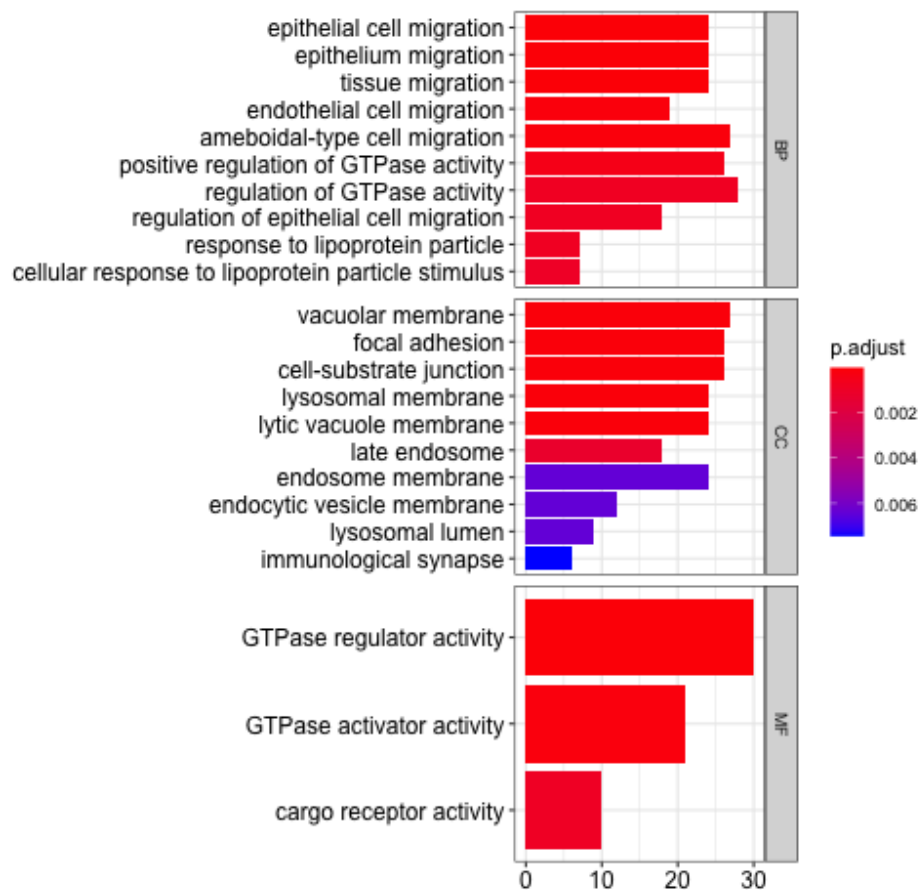
group5 Go enrichment barplot which shows enrichment in the regulation of morphogenesis which contributes to the developmental process of a cell. Due to the cellular energy being depleted, it causes the developmental process to go haywire.

```
knitr::include_graphics("/Users/lindsayreisman/Downloads/final.proj.int only/grp5.G0.bar.png")
```



group6 Go enrichment barplot shows that there is a lot of cell migration activity to the epithelium and other tissues as a result of the lack of oxygen and energy derived from the cells.

```
knitr::include_graphics("/Users/lindsayreisman/Downloads/final.proj.int only/grp6.G0.bar.png")
```



Conclusion

From our interpretations, we have concluded that higher expression of transcripts under AK2 condition is responsible for over activation the immune activation and response. The two cells that are most affected by this knockout gene is Neutrophils and Myelocytes. The immune response is seen to show more expression which we believe is due to the body trying to overcompensate due to the lower levels of lymphocytes and Neutrophils. Unfortunately, this overcompensation can also lead to overproduction of certain phospholipid pathways and metabolic processes which we saw enriched in the earlier graphs which can cause the cell to undergo oxidative stress and eventually lead to apoptosis. Also with AK2 knockout, it suppresses the transfer of phosphate groups, which causes not enough energy for the cell to perform its natural duties.

Appendix

```
# Load dataset into R
#
# library(GEOquery)
#
# gse <- getGEO("GSE179320")
# data <- gse$GSE179320_series_matrix.txt.gz
# data@experimentData
# data$data_processing
# data$geo_accession
# data$`cell type:ch1`
```



```

# data$`treatment:ch1`
#
# expdesign = data.frame(sample=data$geo_accession,
#                         condition=factor(data$`treatment:ch1`),
#                         cell=factor(data$`cell type:ch1`))
# rownames(expdesign) <- data$geo_accession
#
#
# library(reshape2)
# library(ggplot2)
# library(org.Hs.eg.db)
#
# list_of_files <- list.files(path = ".", recursive = TRUE,
#                             pattern = "\\tsv$",
#                             full.names = TRUE)
# tablist <- lapply(list_of_files, function(x) read.table(x, header=T))
# sub.tablist <- lapply(tablist, function(x) x[,c(1,4)])
# raw.count = do.call("cbind", sub.tablist)
# rownames(raw.count) <- raw.count$target_id
# raw.count <- raw.count[!duplicated(as.list(raw.count))]
# raw.count$target_id <- NULL
# names(raw.count) <- c("AAVS1_MC_1", "AAVS1_MC_2", "AAVS1_MC_3", "AAVS1_NP_1",
#                       "AAVS1_NP_2", "AAVS1_NP_3", "AAVS1_PM_1", "AAVS1_PM_2",
#                       "AAVS1_PM_3", "AK2_MC_1", "AK2_MC_2", "AK2_MC_3",
#                       "AK2_NP_1", "AK2_NP_2", "AK2_NP_3", "AK2_PM_1",
#                       "AK2_PM_2", "AK2_PM_3")
#
# Create a boxplot to confirm we normalized the dataset
#
# raw.count.drop.zero <- raw.count[rowSums(raw.count[])>0,]
# raw.count.drop.zero$transcript_id <- rownames(raw.count.drop.zero)
# raw.count.drop.zero.melt = melt(raw.count.drop.zero)
# ggplot(raw.count.drop.zero.melt, aes(x=variable, y=log2(value + 1))) +
# geom_boxplot() +
#   theme(text=element_text(size=15), axis.text.x=element_text(angle=90, hjust=1)) +
#   ggtitle("Boxplot of normalized values")
#
# tab <- read.table("mart_export.txt", sep='\t', header=T)
# idx <- match(rownames(raw.count), tab$Transcript.stable.ID.version)
# meta.table <- tab[idx,]
# meta.table$Gene.name <- NULL
#
#
# xxd <- toTable(org.Hs.egSYMBOL)
# yyd <- toTable(org.Hs.egENSEMBL2EG)
# ann <- merge(xxd, yyd, by="gene_id")
# meta.table$ensembl_id = meta.table$Gene.stable.ID
# meta.table$Gene.stable.ID <- NULL
#
# meta.table.merge <- merge(ann, meta.table, by="ensembl_id")
#
# raw.count.meta.table.drop.dup <-
# drop duplicated gene_id

```

```

#meta.table.merge[!duplicated(meta.table.merge$gene_id),]
#
#
#
# library(DESeq2)
# library(ggplot2)
# library(ggrepel)
#
# set base level in each factor for interaction
# expdesign$condition <- relevel(expdesign$condition, "AAVS1 CRISPR")
# expdesign$cell <- relevel(expdesign$cell, "Myelocyte")
#
#
# run DESeq2 using DESeqDataSetFromMatrix() command.
# cds <- DESeqDataSetFromMatrix(countData =
#                                     round(raw.count[rowSums(raw.count[])>0,]),
#                                     colData = expdesign,
#                                     design = ~ cell + condition + cell:condition)
#
# prefilter counts below 10
# keep <- rowSums(counts(cds)) >= 10
# cds <- cds[keep, ]
# dim(cds)
#
#
# cds <- DESeq(cds)
# resultsNames(cds)
#
# check design matrix
# mod_mat <- model.matrix(design(cds), colData(cds))
#
# coef vectors for each factor
# pro_AAVS1 <- colMeans(mod_mat[cds$cell ==
# "Promyelocyte" & cds$condition == "AAVS1 CRISPR", ]) -->
# pro_AK2 <- colMeans(mod_mat[cds$cell ==
# "Promyelocyte" & cds$condition == "AK2 CRISPR", ]) -->
# neu_AAVS1 <- colMeans(mod_mat[cds$cell == "Neutrophil" & cds$condition ==
# "AAVS1 CRISPR", ]) -->
# neu_AK2 <- colMeans(mod_mat[cds$cell == "Neutrophil" & cds$condition ==
# "AK2 CRISPR", ]) -->
#
# results of cds with interaction term
# res.int <- results(cds,name = "cellPromyelocyte.conditionAK2.CRISPR")
#
# summary(res.int)
#
#
#
# sum(res.int$padj < 0.05 & abs(res.int$log2FoldChange) > 1, na.rm=T)
#
# resSigind = res.int[ which(res.int$padj < 0.05 & res.int$log2FoldChange > 1), ]
# resSigrep = res.int[ which(res.int$padj < 0.05 & res.int$log2FoldChange < -1), ]

```

```

# resSig = rbind(resSigind, resSigrep)
#
#
# diffexpgenes = rownames(resSig)
#
# normvalues <- counts(cds, normalized=T)
#
# find the diffexpvalues
# idx2 <- match(diffexpgenes, rownames(normvalues))
# diffexpvalues <- normvalues[idx2,]
#
# head(resSig)
# dim(diffexpvalues)
#
# resSig$Transcript.stable.ID.version = rownames(resSig)
# resSig.merged.ann <- merge(as.data.frame(resSig),
# diffexpvalues.matched.meta.table.drop.na, by='Transcript.stable.ID.version')

# resSig.merged.ann$diffexpressed <- "NO"
# resSig.merged.ann$diffexpressed[resSig.merged.ann$log2FoldChange > 1 &

# resSig.merged.ann$pvalue < 0.05] <- "UP"
# resSig.merged.ann$diffexpressed[resSig.merged.ann$log2FoldChange < -1 &

# resSig.merged.ann$pvalue < 0.05] <- "DOWN"
#
# mycolors <- c("red", "green", "black")
# names(mycolors) <- c("DOWN", "UP", "NO")
# resSig.merged.ann$delabel <- NA
# resSig.merged.ann$delabel[resSig$diffexpressed != "NO"] <-
# resSig.merged.ann$symbol.x[resSig$diffexpressed != "NO"]
# ordered.resSig.merged.ann <- resSig.merged.ann[order(resSig.merged.ann$padj),]
#
# ordered.filter.resSig.merged.ann <- ordered.resSig.merged.ann %>%
# filter(abs(log2FoldChange) > 1) %>% mutate(top20_symbol = "")
#
# ordered.filter.resSig.merged.ann$top20_symbol[1:20] <- #as.character
# (ordered.filter.resSig.merged.ann$delabel[1:20])
#
# library(ggrepel)
# options(ggrepel.max.overlaps = Inf)
#
# ggplot(data=ordered.filter.resSig.merged.ann, aes(x=log2FoldChange,
# y=-log10(padj), col=diffexpressed)) +
#   geom_point() +
#   theme_minimal() +
#   geom_vline(xintercept=c(-1, 1), col="orange") +
#   geom_hline(yintercept=-log10(0.05), col="orange") +
#   scale_color_manual(values = mycolors) +
#   labs(col = "DE genes") +
#   ggtitle("Differentially expressed")
#   geom_text_repel(aes(label=top20_symbol))
#

```

```

#
# Find the genes with the most significance
# resSig[order(resSig$padj), ]
#
# library(reshape2)
# norm_melt = melt(normvalues)
# gene2draw.ENST00000620254.5 = subset(norm_melt, Var1=="ENST00000620254.5")
#
# library(ggplot2)
#
# Look at the 3 most significant
# gene2draw.ENST00000380554.5 = subset(norm_melt, Var1=="ENST00000380554.5")
# ggplot(gene2draw.ENST00000380554.5) +
#   geom_bar(mapping = aes(x=Var2, y=value), stat="identity")+
#   theme(text=element_text(size=15),axis.text.x=element_text(angle=90,hjust=1)) +
#   ggtitle("ENST00000380554.5")
#
#
# gene2draw.ENST00000505239.1 = subset(norm_melt, Var1=="ENST00000505239.1")
# ggplot(gene2draw.ENST00000505239.1) +
#   geom_bar(mapping = aes(x=Var2, y=value), stat="identity") +
#   theme(text=element_text(size=15),axis.text.x=element_text(angle=90,hjust=1)) +
#   ggtitle("ENST00000505239.1")
#
#
# gene2draw.ENST00000605244.5 = subset(norm_melt, Var1=="ENST00000605244.5")
# ggplot(gene2draw.ENST00000605244.5) +
#   geom_bar(mapping = aes(x=Var2, y=value), stat="identity") +
#   theme(text=element_text(size=15),axis.text.x=element_text(angle=90,hjust=1)) +
#   ggtitle("ENST00000605244.5")
#
#
# match transcript id
# idx3 <- match(rownames(difffexpvalues),
# meta.table.merge$Transcript.stable.ID.version)
#
# subset those matched in difffexpvalues
# difffexpvalues.matched.meta.table <- meta.table.merge[idx3,]
#
# #drop row with na
# difffexpvalues.matched.meta.table.drop.na <- difffexpvalues.matched.
# meta.table[!is.na(difffexpvalues.matched.meta.table$gene_id),]
#
#
# difffexpvalues PCA
# difffexpvalues.prcomp <- prcomp(t(difffexpvalues), scale=TRUE, center=TRUE)
#
# check variances of PC
# plot(difffexpvalues.prcomp)
#
# difffexpvalues.coords2draw <- cbind(as.data.frame(difffexpvalues.prcomp$x),
# expdesign)
# vars_transformed <- apply(difffexpvalues.prcomp$x, 2, var)

```

```

#
# or diffexpvalues.prcomp$sdev^2
# percent.var <- vars_transformed/sum(vars_transformed)
#
# library(ggrepel)
# ggplot(diffexpvalues.coords2draw, aes(x = PC1, y= PC2,
#                                       col=cell, shape=condition)) +
#   geom_point(size=3) +
#   xlab(paste0("PC1: ",round(100*percent.var[1]),"% variance")) +
#   ylab(paste0("PC2: ",round(100*percent.var[2]),"% variance")) +
#   geom_text_repel(aes(label=condition)) +
#   labs(title="diff.PCA plot") +
#   theme(plot.title = element_text(hjust = 0.5)) +
#   coord_fixed()
#
#
#
# diffexpvalues.drop.zero.count <- diffexpvalues[rowSums(diffexpvalues[])>0,]
# t.diffexpvalues.drop.zero.count <- t(diffexpvalues.drop.zero.count)
#
# calculate distance and clustering
# diffexpvalues.cor <- cor(t.diffexpvalues.drop.zero.count)
# diffexpvalues.dist <- as.dist(1-diffexpvalues.cor)
# diffexp.clust <- hclust(diffexpvalues.dist, method='average')
#
#
# check width distr to determine best k value
# library(cluster)
# avg.sil.values=numeric()
# avg.sil.values[1]=0
# for (i in 2:20) {
#   temp.clusters = cutree(diffexp.clust, k=i)
#   silhouette(temp.clusters, dist=diffexpvalues.dist)-> temp.cluster.sil
#   avg.sil.values[i]=mean(temp.cluster.sil[, "sil_width"])
# }
#
# plot average sil values
# plot(avg.sil.values,main="Average Silhouette Values")
#
#
# diffexp.clust.groups <- cutree(diffexp.clust,k=9)
# diffexp.clust.groups.df <- data.frame(cluster = diffexp.clust.groups)
# diffexp.clust.groups.df$cluster
#
# diffexp.clust.groups.df <- data.frame(cluster = as.character
# (diffexp.clust.groups.df$cluster))
# rownames(diffexp.clust.groups.df) <- diffexpgenes
#
# table(diffexp.clust.groups)
#
#
# library(cluster)
# diffexp.clust.sil = silhouette(diffexp.clust.groups, diffexpvalues.dist)

```

```

#
# plot(difffexp.clust.sil,main="silhouette plot", col="blue", border=NA)
#
#
# group1names = names(which(difffexp.clust.groups==1))
# summary(group1names)
# group2names = names(which(difffexp.clust.groups==2))
# summary(group2names)
# group3names = names(which(difffexp.clust.groups==3))
# summary(group3names)
# group4names = names(which(difffexp.clust.groups==4))
# summary(group4names)
# group5names = names(which(difffexp.clust.groups==5))
# summary(group5names)
# group6names = names(which(difffexp.clust.groups==6))
# summary(group6names)
# group7names = names(which(difffexp.clust.groups==7))
# summary(group7names)
# group8names = names(which(difffexp.clust.groups==8))
# summary(group8names)
# group9names = names(which(difffexp.clust.groups==9))
# summary(group9names)
#
#
# library(pheatmap)
# my_sample_col <- expdesign
# my_sample_col$sample = NULL
# rownames(my_sample_col) <- colnames(difffexpvalues)
#
# my_sample_col
#
# create a complete heatmap using cutree_rows=9
# heatmap <- pheatmap(difffexpvalues.drop.zero.count, cluster_rows=difffexp.clust,
#scale = "row",cutree_rows = 9, cutree_cols = 3,
# annotation_col=my_sample_col, annotation_row=difffexp.clust.groups.df,
# main="complete heatmap",
# color = colorRampPalette(c("blue", "white", "red"))(50), angle_col = 45)
#
# save_pheatmap_pdf <- function(x, filename, width=20, height=20) {
#   stopifnot(!missing(x))
#   stopifnot(!missing(filename))
#   pdf(filename, width=width, height=height)
#   grid::grid.newpage()
#   grid::grid.draw(x$gtable)
#   dev.off()
# }
# save_pheatmap_pdf(heatmap, "complete_heatmap.int.pdf")
#
#
# Hierarchical clustering
# diff.cor <- cor(difffexpvalues[rowSums(difffexpvalues[])>0,])
#
# diff.dist <- as.dist(1-diff.cor)

```

```

#
# diff.clust <- hclust(diff.dist, method='average')
#
# expdesign vs. cell plot
# diff.clust$labels <- expdesign$cell
# ## samples cluster together according to cell type
# plot(diff.clust,main="expdesign vs. cell cluster dendogram")
#
#
# expdesign vs. condition plot
# diff.clust$labels <- expdesign$condition
## some samples according to condition DONOT cluster together
# plot(diff.clust,main="expdesign vs. condition cluster dendogram")
#
# expdesign vs. sample plot
# diff.clust$labels <- expdesign$sample
# plot(diff.clust, main="expdesign vs. sample cluster dendogram")
#
# using clusterprofiler to do GO analysis
# library(clusterProfiler)
# library(org.Hs.eg.db)
#
# ego <- enrichGO(gene      =diffexpvalues.matched.meta.table.drop.na$gene_id,
#                 universe  = raw.count.meta.table.drop.dup$gene_id,
#                 OrgDb     = org.Hs.eg.db,
#                 ont       = "ALL",
#                 pAdjustMethod = "BH",
#                 pvalueCutoff = 0.01,
#                 readable   = TRUE)
# go.res <- as.data.frame(ego@result)
#
# plot top 10 GO
# ggplot(go.res.top.10, aes(x=Description, y=-log10(p.adjust), fill=ONTOLOGY))+
#   geom_bar(stat="identity") +
#   coord_flip()+
#   scale_x_discrete(limits=rev(go.res.top.10$Description))+
#   theme_classic()+
#   ggtitle("Top 10 GO terms")+
#   theme(
#     text = element_text(size=20),
#     axis.title.y=element_blank(),
#     axis.title.x=element_text(size=15)
#   )
#
#
# For group 1:
# heatmap for group1
# grp1.heatmap = pheatmap(diffexpvalues.drop.
#                         zero.count[names(which(diffexp.clust.groups==1)),],
# scale = "row", annotation_col=my_sample_col,
# annotation_row=diffexp.clust.groups.df,
# clustering_distance_rows = "correlation",main="grp1 heatmap",
# color = colorRampPalette(c("blue", "white", "red"))(50), angle_col = 45)

```

```

#
# save_pheatmap_pdf <- function(x, filename, width=20, height=20) {
#   stopifnot(!missing(x))
#   stopifnot(!missing(filename))
#   pdf(filename, width=width, height=height)
#   grid::grid.newpage()
#   grid::grid.draw(x$gtable)
#   dev.off()
# }
# save_pheatmap_pdf(grp1.heatmap, "grp1.heatmap.pdf")
#
#
# cluster1genes = diffexpvalues[diffexp.clust.groups==1, ]
# cluster1match.diffexpvalues.matched.meta.table.drop.na <-na.omit(diffexpvalues.
# matched.meta.table.drop.na[match(rownames(cluster1genes),
# diffexpvalues.matched.meta.table.drop.na$Transcript.stable.ID.version),])
#
# GO terms for group1:
# ego.grp1 <- enrichGO(gene      = cluster1match.diffexpvalues.
#                       matched.meta.table.drop.na$gene_id,
#                       universe  = raw.count.meta.table.drop.dup$gene_id,
#                       OrgDb     = org.Hs.eg.db,
#                       ont       = "ALL",
#                       pAdjustMethod = "BH",
#                       pvalueCutoff = 0.01,
#                       readable    = TRUE)
# go.res.grp1 <- as.data.frame(ego.grp1@result)
#
#
# barplot(ego.grp1, split="ONTOLOGY",showCategory = 10) +
# facet_grid(ONTOLOGY~., scale="free")
#
#
# For group 2:
# heatmap for group2
# grp2.heatmap = pheatmap(diffexpvalues.drop.zero.count[names(which(diffexp.
# clust.groups==2)),],
# scale = "row", annotation_col=my_sample_col,
# annotation_row=diffexp.clust.groups.df,
# clustering_distance_rows = "correlation", main="grp2 heatmap",
# color = colorRampPalette(c("blue", "white", "red"))(50), angle_col = 45)
#
# save_pheatmap_pdf <- function(x, filename, width=20, height=20) {
#   stopifnot(!missing(x))
#   stopifnot(!missing(filename))
#   pdf(filename, width=width, height=height)
#   grid::grid.newpage()
#   grid::grid.draw(x$gtable)
#   dev.off()
# }
# save_pheatmap_pdf(grp2.heatmap, "grp2.heatmap.pdf")
#
#

```



```

# cluster2genes = diffexpvalues[diffexp.clust.groups==2, ]
# cluster2match.diffexpvalues.matched.meta.table.drop.na <- #na.omit
# (diffexpvalues.matched.meta.table.drop.na[match(rownames(cluster2genes),
# diffexpvalues.matched.meta.table.drop.na$Transcript.stable.ID.version),])
#
# GO terms for group2:
# ego.grp2 <- enrichGO(gene      = cluster2match.diffexpvalues.matched.meta.
#                       table.drop.na$gene_id,
#                       universe  = raw.count.meta.table.drop.dup$gene_id,
#                       OrgDb     = org.Hs.eg.db,
#                       ont       = "ALL",
#                       pAdjustMethod = "BH",
#                       pvalueCutoff = 0.01,
#                       readable    = TRUE)
# go.res.grp2 <- as.data.frame(ego.grp2@result)
#
# barplot(ego.grp2, split="ONTOLOGY",showCategory = 10) + facet_grid(ONTOLOGY~.,
# scale="free")
#
#
#
# cluster3genes = diffexpvalues[diffexp.clust.groups==3, ]
# cluster3match.diffexpvalues.matched.meta.table.drop.na <-na.omit(diffexpvalues.
# matched.meta.table.drop.na[match(rownames(cluster3genes),
# diffexpvalues.matched.meta.table.drop.na$Transcript.stable.ID.version),])
#
# GO terms for group3:
# ego.grp3 <- enrichGO(gene      = cluster3match.diffexpvalues.matched.
#                       meta.table.drop.na$gene_id,
#                       universe  = raw.count.meta.table.drop.dup$gene_id,
#                       OrgDb     = org.Hs.eg.db,
#                       ont       = "ALL",
#                       pAdjustMethod = "BH",
#                       pvalueCutoff = 0.01,
#                       readable    = TRUE)
# go.res.grp3 <- as.data.frame(ego.grp3@result)
#
#
# barplot(ego.grp3, split="ONTOLOGY",showCategory = 10) + facet_grid(ONTOLOGY~.,
# scale="free")
#
#
#
# cluster4genes = diffexpvalues[diffexp.clust.groups==4, ]
# cluster4match.diffexpvalues.matched.meta.table.drop.na <-na.omit(diffexpvalues.
# matched.meta.table.drop.na[match(rownames(cluster4genes),
# diffexpvalues.matched.meta.table.drop.na$Transcript.stable.ID.version),])
#
# GO terms for group4:
# ego.grp4 <- enrichGO(gene      = cluster4match.diffexpvalues.
#                       matched.meta.table.drop.na$gene_id,
#                       universe  = raw.count.meta.table.drop.dup$gene_id,
#                       OrgDb     = org.Hs.eg.db,

```

```

#               ont           = "ALL",
#               pAdjustMethod = "BH",
#               pvalueCutoff  = 0.01,
#               readable      = TRUE)
# go.res.grp4 <- as.data.frame(ego.grp4@result)
#
# create barplot
# barplot(ego.grp4, split="ONTOLOGY",showCategory = 10) +
# facet_grid(ONTOLOGY~., scale="free")
#
#
#
# cluster5genes = diffexpvalues[diffexp.clust.groups==5, ]
# cluster5match.diffexpvalues.matched.meta.table.drop.na <- na.omit
# (diffexpvalues.matched.meta.table.drop.na[match(rownames(cluster5genes),
# diffexpvalues.matched.meta.table.drop.na$Transcript.stable.ID.version),])
#
# GO terms for group5:
# ego.grp5 <- enrichGO(gene      = cluster5match.diffexpvalues.matched.
#                       meta.table.drop.na$gene_id,
#                       universe  = raw.count.meta.table.drop.dup$gene_id,
#                       OrgDb     = org.Hs.eg.db,
#                       ont       = "ALL",
#                       pAdjustMethod = "BH",
#                       pvalueCutoff  = 0.01,
#                       readable     = TRUE)
# go.res.grp5 <- as.data.frame(ego.grp5@result)
#
# create barplot
# barplot(ego.grp5, split="ONTOLOGY",showCategory = 10) + facet_grid(ONTOLOGY~.,
# scale="free")
#
#
# cluster6genes = diffexpvalues[diffexp.clust.groups==6, ]
# cluster6match.diffexpvalues.matched.meta.table.drop.na <-na.omit(diffexpvalues.
# matched.meta.table.drop.na[match(rownames(cluster6genes),
# diffexpvalues.matched.meta.table.drop.na$Transcript.stable.ID.version),])
#
# GO terms for group6:
# ego.grp6 <- enrichGO(gene      = cluster6match.diffexpvalues.matched.meta.table
#                       .drop.na$gene_id,
#                       universe  = raw.count.meta.table.drop.dup$gene_id,
#                       OrgDb     = org.Hs.eg.db,
#                       ont       = "ALL",
#                       pAdjustMethod = "BH",
#                       pvalueCutoff  = 0.01,
#                       readable     = TRUE)
# go.res.grp6 <- as.data.frame(ego.grp6@result)
#
# create barplot
# barplot(ego.grp6, tile("group6 GO plot"),split="ONTOLOGY",showCategory = 10)
# + facet_grid(ONTOLOGY~., scale="free"))
#

```

```

#
# cluster7genes = diffexpvalues[diffexp.clust.groups==7, ]
# cluster7match.diffexpvalues.matched.meta.table.drop.na <- na.omit
# (diffexpvalues.matched.meta.table.drop.na[match(rownames(cluster7genes),
# diffexpvalues.matched.meta.table.drop.na$Transcript.stable.ID.version),])
#
# GO terms for group6:
# ego.grp7 <- enrichGO(gene      = cluster7match.diffexpvalues.matched.meta.
#                       table.drop.na$gene_id,
#                       universe  = raw.count.meta.table.drop.dup$gene_id,
#                       OrgDb     = org.Hs.eg.db,
#                       ont       = "ALL",
#                       pAdjustMethod = "BH",
#                       pvalueCutoff = 0.01,
#                       readable   = TRUE)
# go.res.grp7 <- as.data.frame(ego.grp7@result)
#
# create barplot
# barplot(ego.grp7, split="ONTOLOGY",showCategory = 10) + facet_grid(ONTOLOGY~.,
# scale="free")
#
#
#
# cluster8genes = diffexpvalues[diffexp.clust.groups==8, ]
# cluster8match.diffexpvalues.matched.meta.table.drop.na <-na.omit(diffexpvalues.
# matched.meta.table.drop.na[match(rownames(cluster8genes),
# diffexpvalues.matched.meta.table.drop.na$Transcript.stable.ID.version),])
#
# GO terms for group8:
# ego.grp8 <- enrichGO(gene      = cluster8match.diffexpvalues.matched.meta.
#                       table.drop.na$gene_id,
#                       universe  = raw.count.meta.table.drop.dup$gene_id,
#                       OrgDb     = org.Hs.eg.db,
#                       ont       = "ALL",
#                       pAdjustMethod = "BH",
#                       pvalueCutoff = 0.01,
#                       readable   = TRUE)
# go.res.grp8 <- as.data.frame(ego.grp8@result)
#
# create barplot
# barplot(ego.grp8, split="ONTOLOGY",showCategory = 10
# facet_grid(ONTOLOGY~., scale="free")
#
#
# cluster9genes = diffexpvalues[diffexp.clust.groups==9, ]
# cluster9match.diffexpvalues.matched.meta.table.drop.na <- na.omit(diffexpvalues.
# matched.meta.table.drop.na[match(rownames(cluster9genes),
# diffexpvalues.matched.meta.table.drop.na$Transcript.stable.ID.version),])
#
# GO terms for group9:
# ego.grp9 <- enrichGO(gene      = cluster9match.diffexpvalues.matched.
#                       meta.table.drop.na$gene_id,

```

```

#           universe      = raw.count.meta.table.drop.dup$gene_id,
#           OrgDb          = org.Hs.eg.db,
#           ont            = "ALL",
#           pAdjustMethod = "BH",
#           pvalueCutoff   = 0.01,
#           readable       = TRUE)
# go.res.grp9 <- as.data.frame(ego.grp9@result)
#
#
# barplot(ego.grp9, split="ONTOLOGY", showCategory = 10) +
# facet_grid(ONTOLOGY~., scale="free")

```

References

1. Myelopoiesis | Pathology Made Simple. (n.d.). Retrieved August 22, 2021, from <http://ilovepathology.com/myelopoiesis/>
2. Reticular dysgenesis | Genetic and Rare Diseases Information Center (GARD) – an NCATS Program. (n.d.). Rarediseases.info.nih.gov. Retrieved August 22, 2021, from <https://rarediseases.info.nih.gov/diseases/8625/reticular-dysgenesis>
3. Wenqing Wang, Avni Awani, Andrew Devilbiss, Thomas Mathews, Daniel Thomas, Daniel P. Dever, Matthew Porteus, Sean Morrison, Katja G Weinacht, Adenylate Kinase 2 Links Energy Metabolism and Cell Fate in Hematopoietic Stem and Progenitor Cells, Blood, Volume 134, Supplement 1, 2019, Page 3705, ISSN 0006-4971, <https://doi.org/10.1182/blood-2019-129424>. (<https://www.sciencedirect.com/science/article/pii/S0006497118616337>)
4. Adenylate Kinase 2 deficiency causes NAD⁺ depletion and impaired purine metabolism during myelopoiesis Wenqing Wang, Andrew DeVilbiss, Martin Arreola, Thomas Mathews, Misty Martin-Sandoval, Zhiyu Zhao, Avni Awani, Daniel Dever, Waleed Al-Herz, Luigi Noratangelo, Matthew H. Porteus, Sean J. Morrison, Katja G. Weinacht bioRxiv 2021.07.05.450633; doi: <https://doi.org/10.1101/2021.07.05.450633>