Gene KO Study

The following work was put together for a manuscript that is due for submission Spring 2020. This work was an example of using singular value decomposition (SVD - a type of factorization analysis used previously for gene expression), to identify genes with different knockout phenotype behaviors in select backgrounds. Whole-genome CRISPR screening was conducted in identical cells under wild-type and gene specific knockouts. The purpose of this project was to identify genetic depedency differences between these backgrounds to identify genetic interactions of tested genes. Further plots, data, and descriptions will be made available once the manuscript is published.

```
### Loading in Functions
cat("\014")
```

```
'%!in%' <- function(x, y)
  ! ('%in%'(x, y))
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse
1.2.1 ──
```

```
## ✔ ggplot2 3.2.1     ✔ purrr   0.3.2
## ✔ tibble  2.1.3     ✔ dplyr   0.8.3
## ✔ tidyr   0.8.3     ✔ stringr 1.4.0
## ✔ readr   1.3.1     ✔ forcats 0.4.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_confl
icts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(cowplot)
```

```
##
## ************************************************************
```

```
## Note: As of version 1.0.0, cowplot does not change the
```

```
##   default ggplot2 theme anymore. To recover the previous
```

```
##   behavior, execute:
##   theme_set(theme_cowplot())
```

```
## ************************************************************
```

```
library(preprocessCore)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(ggrepel)

theme_Publication <- function(base_size=20, base_family="helvetica") {
  library(grid)
  library(ggthemes)
  (theme_foundation(base_size=base_size)#, base_family=base_family)
    + theme(plot.title = element_text(face = "bold",
                                      size = rel(1.2), hjust = 0.5),
            text = element_text(),
            panel.background = element_rect(colour = NA),
            plot.background = element_rect(colour = NA),
            panel.border = element_rect(colour = NA),
            axis.title = element_text(face = "bold",size = rel(1)),
            axis.title.y = element_text(angle=90,vjust =2),
            axis.title.x = element_text(vjust = -0.2),
            axis.text = element_text(),
            axis.line = element_line(colour="black"),
            axis.ticks = element_line(),
            panel.grid.major = element_line(colour="#f0f0f0"),
            panel.grid.minor = element_blank(),
            legend.key = element_rect(colour = NA),
            legend.key.size= unit(0.2, "cm"),
            legend.margin = unit(0, "cm"),
            legend.title = element_text(face="italic"),
            plot.margin=unit(c(10,5,5,5),"mm"),
            strip.background=element_rect(colour="#f0f0f0",fill="#f0f0f0"),
            strip.text = element_text(face="bold")
    ))

}
#setwd("")
set.seed(1)
color1 = "#f247f5"
color2 = "#d6d6d6"
color3 = "#1ccceb"
```

Loading in necessary plotting data - hidden due to hide gene names.

Quantile normalizing data in order to fix data distributions. Next performing SVD analysis to identify differential behavior of genes.

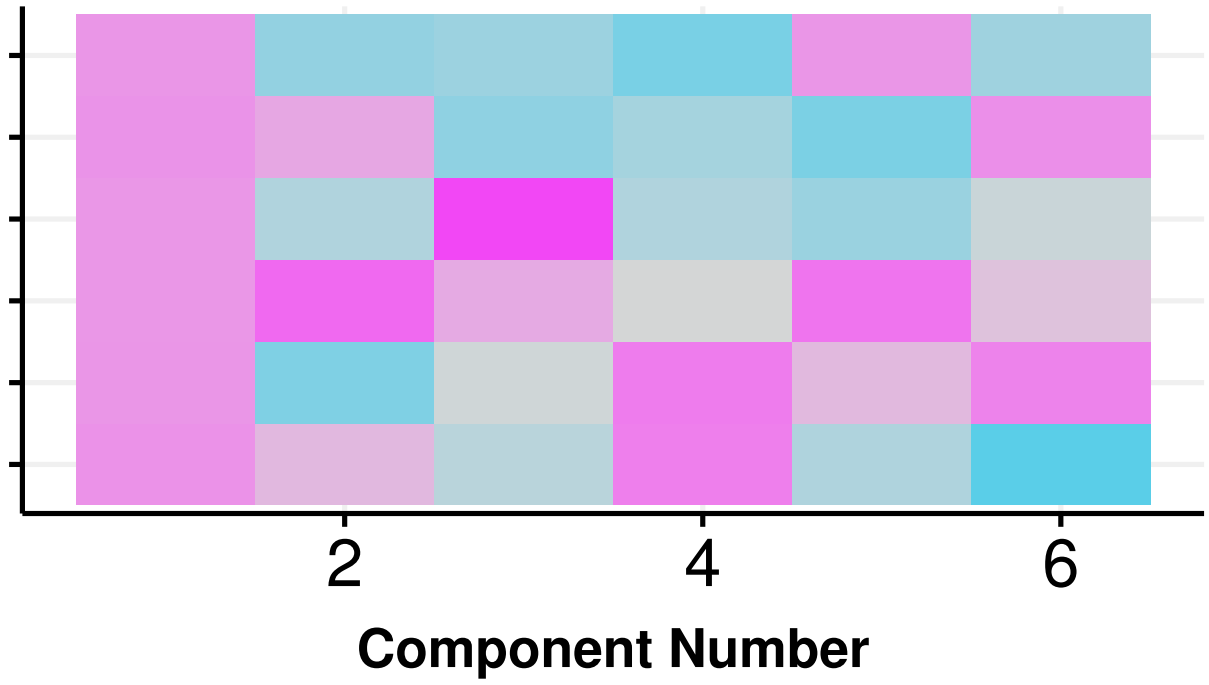
```
qnormed_dat <- normalize.quantiles(as.matrix(bagel_dat))
rownames(qnormed_dat) = rownames(bagel_dat)
colnames(qnormed_dat) = order
qnormed_dat_svd <- svd(qnormed_dat)

u <- qnormed_dat_svd$u
v <- t(qnormed_dat_svd$v)
d <- diag(qnormed_dat_svd$d) # amount of variation due to component

colnames(v) = order

rownames(u) = rownames(bagel_dat)

v_melt <- melt(v)
```

```
ggplot(v_melt, aes(Var1, Var2, fill = value)) + geom_tile() +
  scale_fill_gradient2(high = color1, low = color3, mid = color2) +
  xlab("Component Number") + ylab("") + theme_Publication() +
  ggtitle("Singular Value Decomposition \nTransformed V Matrix") +
  theme(
    legend.position = "none",
    plot.title = element_text(size = 30),
    axis.text.x = element_text(size = 25),
    axis.text.y = element_text(size = 0) #hiding gene names
  )
```

```
##
## Attaching package: 'ggthemes'
```

```
## The following object is masked from 'package:cowplot':
##
##     theme_map
```

```
## Warning: `legend.margin` must be specified using `margin()`. For the old
## behavior use legend.spacing
```

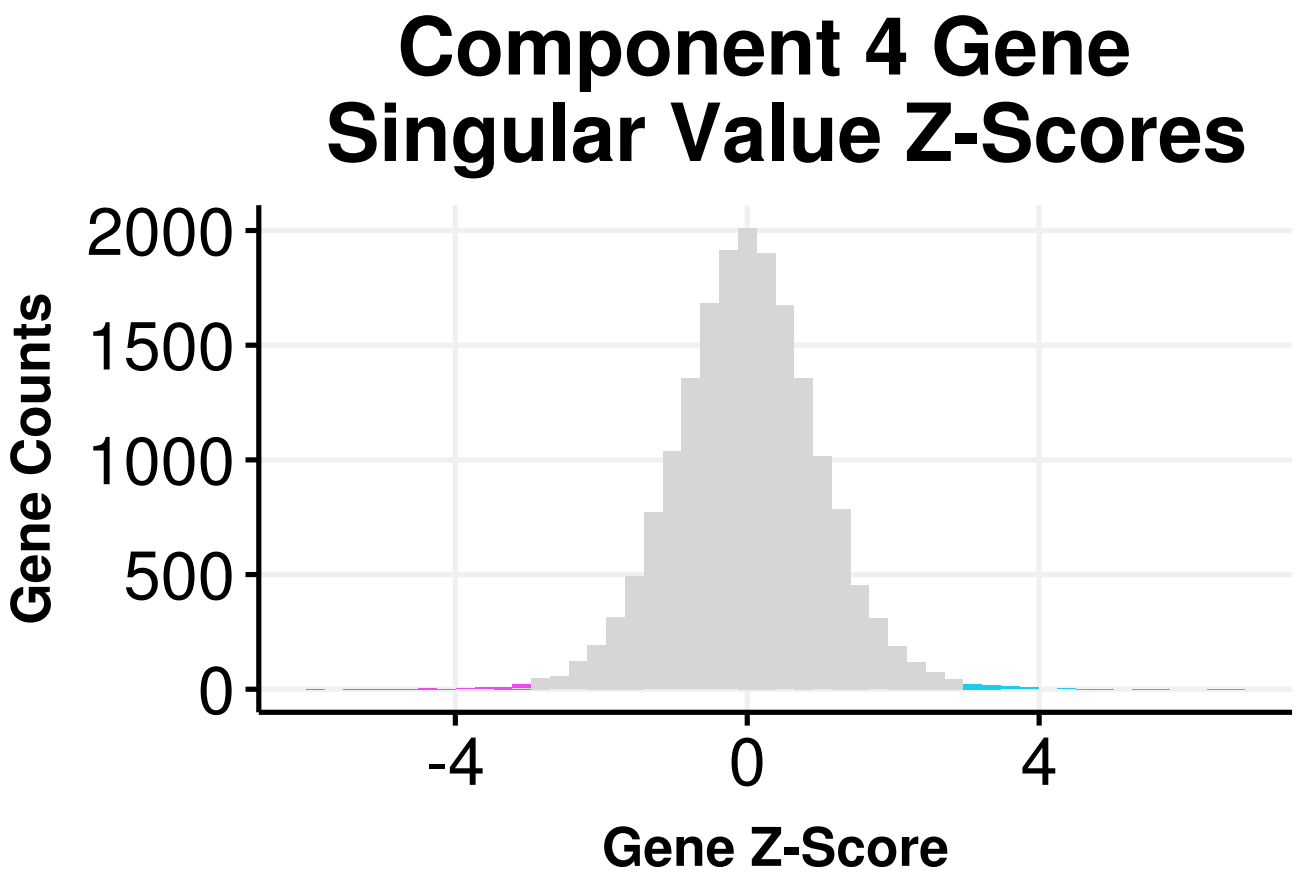


Coloring the tail ends of the Z-Score distribution to examine genes with distinct behaviors between backgrounds.

```
u_dat <- as.data.frame(u)
u_dat$V4 = scale(u_dat$V4)
u_dat$color = color2
u_dat$color[u_dat$V4 < -3] = color3
u_dat$color[u_dat$V4 > 3] = color1
```

```
ggplot(u_dat, aes(V4)) +
  geom_histogram(data = subset(u_dat, color == color3),
                 fill = color1,
                 bins = 50) +
  geom_histogram(data = subset(u_dat, color == color2),
                 fill = color2,
                 bins = 50) +
  geom_histogram(data = subset(u_dat, color == color1),
                 fill = color3,
                 bins = 50) +
  ylab("Gene Counts") + xlab("Gene Z-Score") +
  theme_Publication() +
  ggtitle("Component 4 Gene \n Singular Value Z-Scores") +
  theme(
    legend.position = "none",
    plot.title = element_text(size = 30),
    axis.text.x = element_text(size = 25),
    axis.text.y = element_text(size = 25)
  )
```

```
## Warning: `legend.margin` must be specified using `margin()`. For the old
## behavior use legend.spacing
```

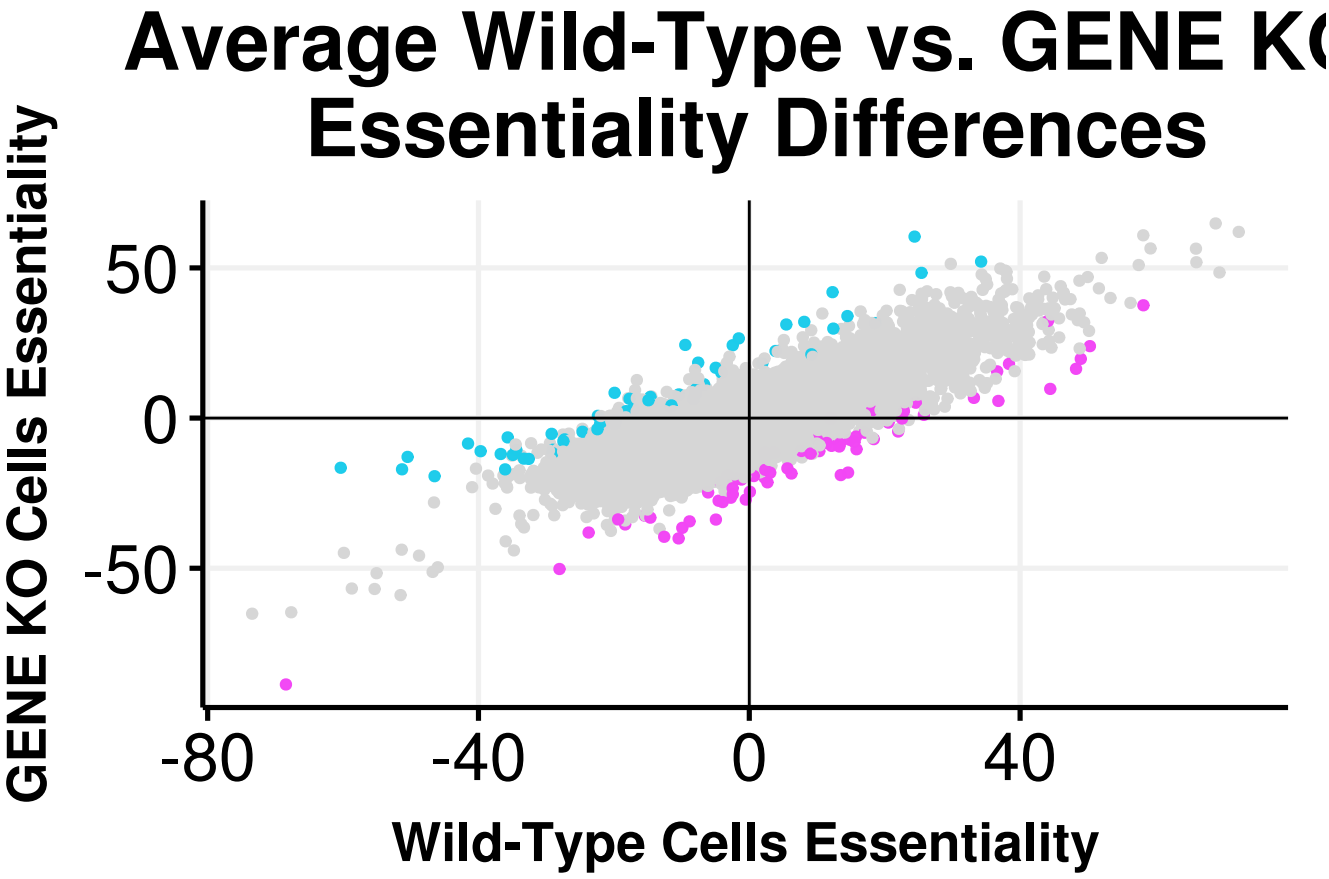


Code block hidden due to gene name. Manipulating data and merging into a single data frame.

Final plot demonstrating effective use of factorization techniques.

```
plot_data_agg <- plot_data %>% dplyr::select(GENE, BF, WT_BF, V4, color)
plot_data_agg_WT <-
  aggregate(plot_data_agg$WT_BF,
            by = list(plot_data_agg$GENE),
            FUN = mean)
plot_data_agg_KO <-
  aggregate(plot_data_agg$BF,
            by = list(plot_data_agg$GENE),
            FUN = mean)
plot_data_agg <- plot_data_agg %>% dplyr::select(GENE, V4, color) %>% unique()

colnames(plot_data_agg_WT) = c("GENE", "WT_BF")
colnames(plot_data_agg_KO) = c("GENE", "KO_BF")
plot_data_agg <- merge(plot_data_agg, plot_data_agg_WT)
plot_data_agg <- merge(plot_data_agg, plot_data_agg_KO)

plot_data_agg  %>%
  ggplot(aes(x = WT_BF, y = KO_BF)) + geom_point(alpha = 0.99,
                                                 size = 1.5,
                                                 color = plot_data_agg$color) + ggtitle(
"Average Wild-Type vs. GENE KO\n Essentiality Differences") +
  scale_color_gradient2(high = color1, low = color3, mid = color2)  + geom_vline(aes(xin
tercept = 0)) + geom_hline(aes(yintercept = 0)) + theme_Publication() +
  theme(
    legend.position = "none",
    plot.title = element_text(size = 30),
    axis.text.x = element_text(size = 25),
    axis.text.y = element_text(size = 25)
  ) +
  #xlim(c(-50,75)) + ylim(c(-50,75)) +
  xlab("Wild-Type Cells Essentiality") + ylab("GENE KO Cells Essentiality")
```

```
## Warning: `legend.margin` must be specified using `margin()`. For the old
## behavior use legend.spacing
```



This plot shows differential gene response between wild-type and gene KO backgrounds. Colored points are particularly interesting as they represent distinct data pattern differences. Several of these genes are within the same pathway of the knocked out gene suggesting genetic dependency changes between the two backgrounds.