COLOMBOS DB E. Coli expression exploratory data analysis

Emanuel Becerra Soto

June 05, 2019

```
suppressMessages(library("tidyverse"))
library(ggthemes)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(ggrepel)
library(NbClust)
library(forcats)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:viridis':
##
##     viridis_pal
```

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

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
# For reproducibility
set.seed(243014)

quantile_plot <- function(data, feature){
  library(tidyverse)
  library(ggrepel)
  n <- nrow(data)
  percentiles <- (1:n - 0.5) / n
  values <- quantile( x = data[[ feature ]] , probs = percentiles)
  quantiles <- quantile(data[[ feature ]])
  IQ1 <- quantiles[2]
  Median <- quantiles[3]
  Q3 <- quantiles[4]
  normal_values <- qnorm(seq(0.01,1,0.01), mean = mean(data[[ feature ]]), sd = sd(data[[ feature ]]))
  unif_values <- qunif(seq(0.01,1,0.01), max = max(data[[ feature ]]), min = min(data[[ feature ]]))
  percentiles2 <- c(percentiles, 0.25, 0.50, 0.75)
  values2 <- c(values, IQ1, Median, Q3)
  to_plot <- tibble(percentiles2, values2)
  names(to_plot) <- c('percentiles', 'values')
  theoretical <- tibble(perc =  seq(0.01,1,0.01) , normal_values, unif_values)
  last_three <- (nrow(to_plot) - 2) : nrow(to_plot)
  theoretical_tidy <- theoretical %>%
```

```
    gather(normal_values, unif_values, key = 'distribution', value = 'values' )
  ggplot(to_plot, aes(x = percentiles, y = values))+
    geom_point(size  = 2)+
    geom_line(data = theoretical_tidy,
              aes(x = perc, y=values, color = distribution),
              size  = 1.5,
              alpha = 0.6)+
    geom_point(data = to_plot[ last_three, ],
               fill = 'red', color = 'red', shape = 23, size = 2.5 )+
    geom_text_repel( data = to_plot[ last_three, ],
                     aes(label = c('Q1','Median','Q3')),
                     vjust        = -5,
                     segment.size = 0.2)+
    ggtitle(paste(feature,'Quantile Plot'))+
    scale_color_brewer(type = 'qual', palette = 6, name = 'Distribution', labels=c('Normal', 'Uniform'))
}
```

Introduction

Exploratory thesis

What is COLOMBOS DB? Integrates transcriptomics data for several prokaryotic model-organisms

Micro-array and RNA-Seq

The motivation of the project is to address the challenge of novel biological discoveries derived from the future collection of ChIP-seq and RNA-seq experiments for every single transcription factor in E. coli, using machine learning approaches. More precisely we will focus on searching for correlation of labels of co-expressed/co-regulated genes using the COLOMBOS collection as a first step. This will prepare us to use the future ChIP-seq and RNA-seq experiments to propose new biological knowledge in E. coli.

Reading the data

```
expr_file <- 'ecoli_expr.tsv'

# Reading the data
ecoli_expr <- read_tsv(expr_file)
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   LocusTag = col_character(),
##   `Gene name` = col_character(),
##   `Geneid/Contrast_id` = col_integer()
## )
```

```
## See spec(...) for full column specifications.
```

```
dim(ecoli_expr)
```

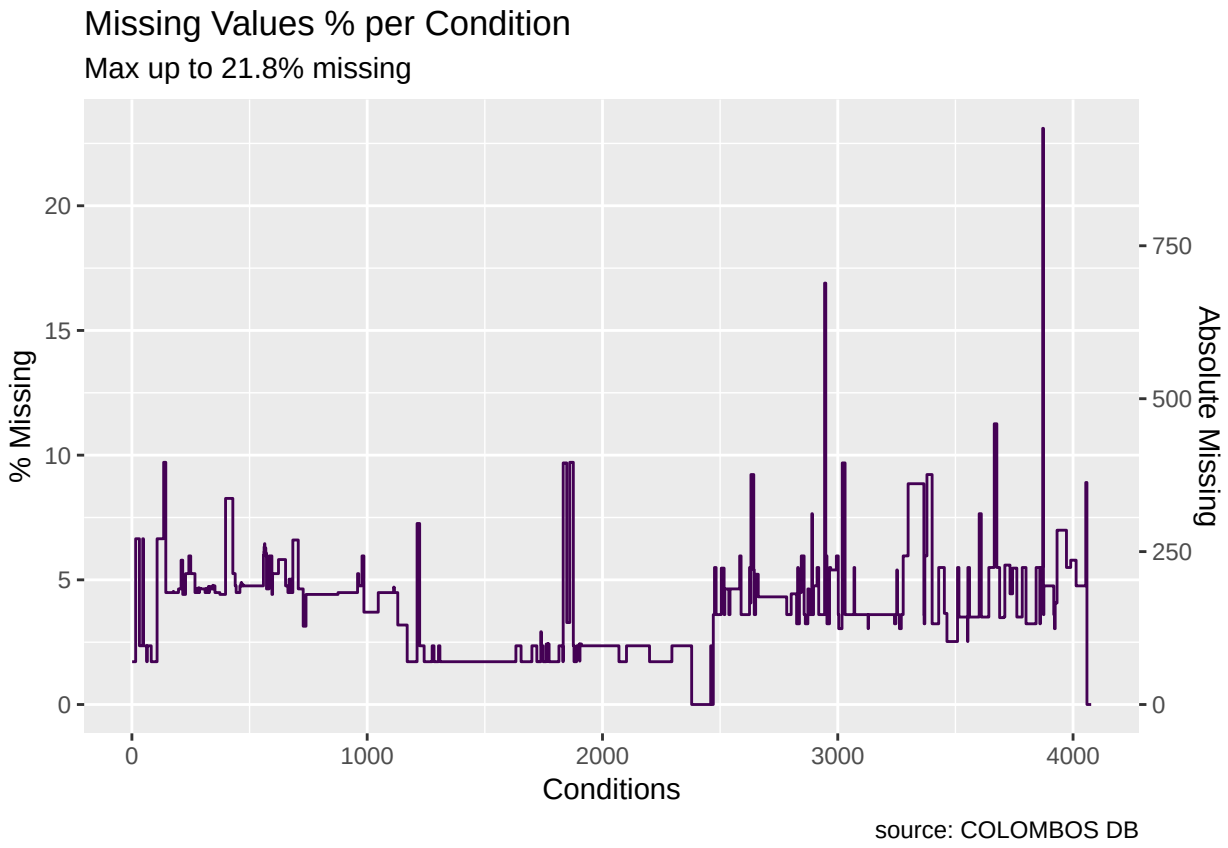
```
## [1] 4321 4080
```

```
head(ecoli_expr)
```

```
## # A tibble: 6 x 4,080
##   LocusTag `Gene name` `Geneid/Contras~     `1`     `2`     `3`     `4`
##   <chr>    <chr>                  <int>   <dbl>   <dbl>   <dbl>  <dbl>
## 1 b0001    thrL                       1 -0.473   0.698   0.0927  0.182
## 2 b0002    thrA                       2 -0.266   1.58   -0.0381  0.236
## 3 b0003    thrB                       3 -0.292   0.800  -0.108  -0.157
## 4 b0004    thrC                       4 -0.0312  1.18   -0.0184  0.459
## 5 b0005    yaaX                       5  0.106   0.0671  0.196   0.200
## 6 b0006    yaaA                       6 -0.158  -0.307  -0.406  -0.183
## # ... with 4,073 more variables: `5` <dbl>, `6` <dbl>, `7` <dbl>,
## #   `8` <dbl>, `9` <dbl>, `10` <dbl>, `11` <dbl>, `12` <dbl>, `13` <dbl>,
## #   `14` <dbl>, `15` <dbl>, `16` <dbl>, `17` <dbl>, `18` <dbl>,
## #   `19` <dbl>, `20` <dbl>, `21` <dbl>, `22` <dbl>, `23` <dbl>,
## #   `24` <dbl>, `25` <dbl>, `26` <dbl>, `27` <dbl>, `28` <dbl>,
## #   `29` <dbl>, `30` <dbl>, `31` <dbl>, `32` <dbl>, `33` <dbl>,
## #   `34` <dbl>, `35` <dbl>, `36` <dbl>, `37` <dbl>, `38` <dbl>,
## #   `39` <dbl>, `40` <dbl>, `41` <dbl>, `42` <dbl>, `43` <dbl>,
## #   `44` <dbl>, `45` <dbl>, `46` <dbl>, `47` <dbl>, `48` <dbl>,
## #   `49` <dbl>, `50` <dbl>, `51` <dbl>, `52` <dbl>, `53` <dbl>,
## #   `54` <dbl>, `55` <dbl>, `56` <dbl>, `57` <dbl>, `58` <dbl>,
## #   `59` <dbl>, `60` <dbl>, `61` <dbl>, `62` <dbl>, `63` <dbl>,
## #   `64` <dbl>, `65` <dbl>, `66` <dbl>, `67` <dbl>, `68` <dbl>,
## #   `69` <dbl>, `70` <dbl>, `71` <dbl>, `72` <dbl>, `73` <dbl>,
## #   `74` <dbl>, `75` <dbl>, `76` <dbl>, `77` <dbl>, `78` <dbl>,
## #   `79` <dbl>, `80` <dbl>, `81` <dbl>, `82` <dbl>, `83` <dbl>,
## #   `84` <dbl>, `85` <dbl>, `86` <dbl>, `87` <dbl>, `88` <dbl>,
## #   `89` <dbl>, `90` <dbl>, `91` <dbl>, `92` <dbl>, `93` <dbl>,
## #   `94` <dbl>, `95` <dbl>, `96` <dbl>, `97` <dbl>, `98` <dbl>,
## #   `99` <dbl>, `100` <dbl>, `101` <dbl>, `102` <dbl>, `103` <dbl>,
## #   `104` <dbl>, ...
```

```r
# Conditions
n_cond <- 4077
n_gene <- nrow(ecoli_expr)
```

```r
# Counting NA's per column
na_count_per_condition <- sapply(ecoli_expr[4:4080], function(col) {
  sum(is.na(col))
})

miss_cond <- tibble(cond = 1:n_cond,
                    abs_miss = na_count_per_condition,
                    rel_miss = na_count_per_condition/n_cond)

# Plotting missing conditions per gene
ggplot(miss_cond, aes(x = cond, y = rel_miss*100))+
  geom_step(color = viridis_pal()(1))+
  scale_y_continuous(
    sec.axis = sec_axis(~ . * nrow(miss_cond)/100, name = "Absolute Missing"))+
  labs(title="Missing Values % per Condition",
       subtitle="Max up to 21.8% missing",
       x = "Conditions",
       y = "% Missing",
       caption="source: COLOMBOS DB")
```



Removing missing values

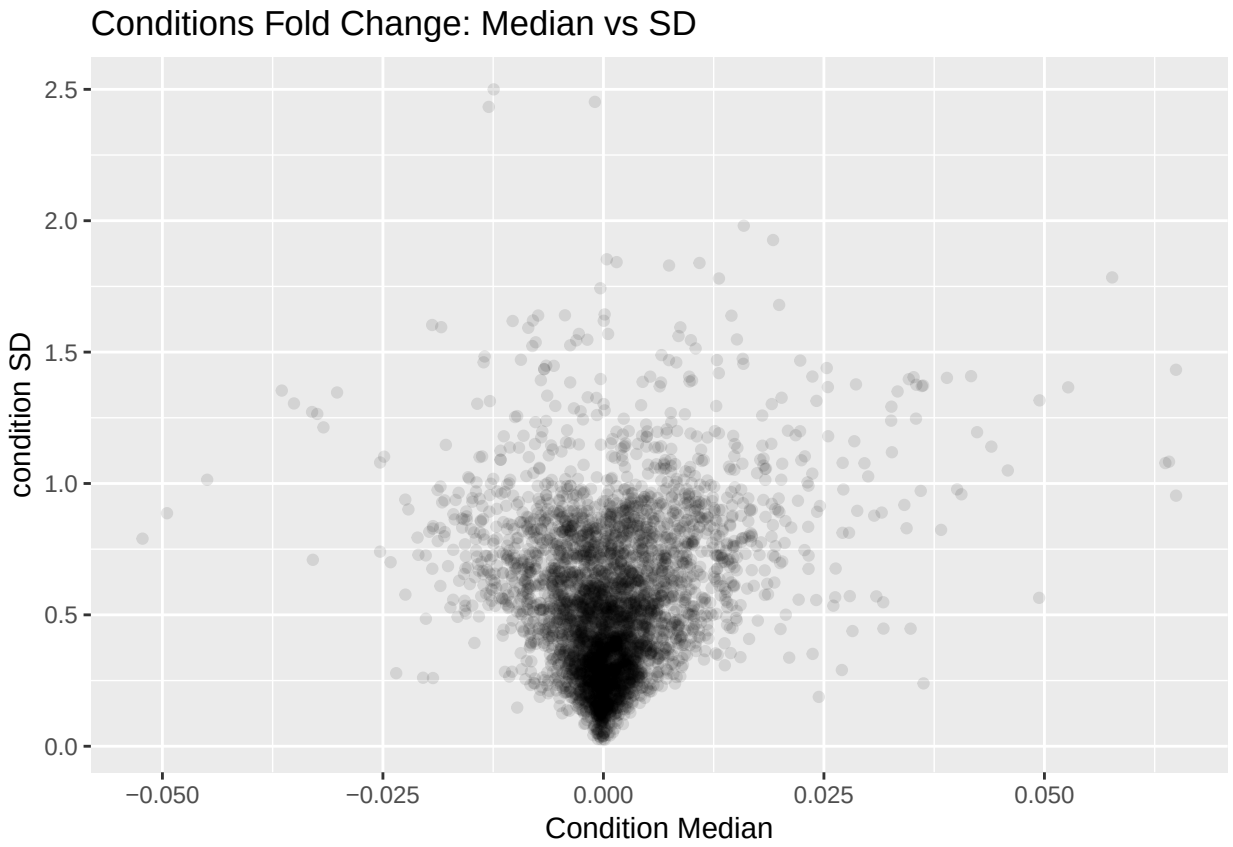
```
# Removing missing values
# Substituting by the median value of the condition
NA2median <- function(x) replace(x, is.na(x), median(x, na.rm = TRUE))
ecoli_expr <- replace(ecoli_expr, TRUE, lapply(ecoli_expr, NA2median))
```

Summary of conditions

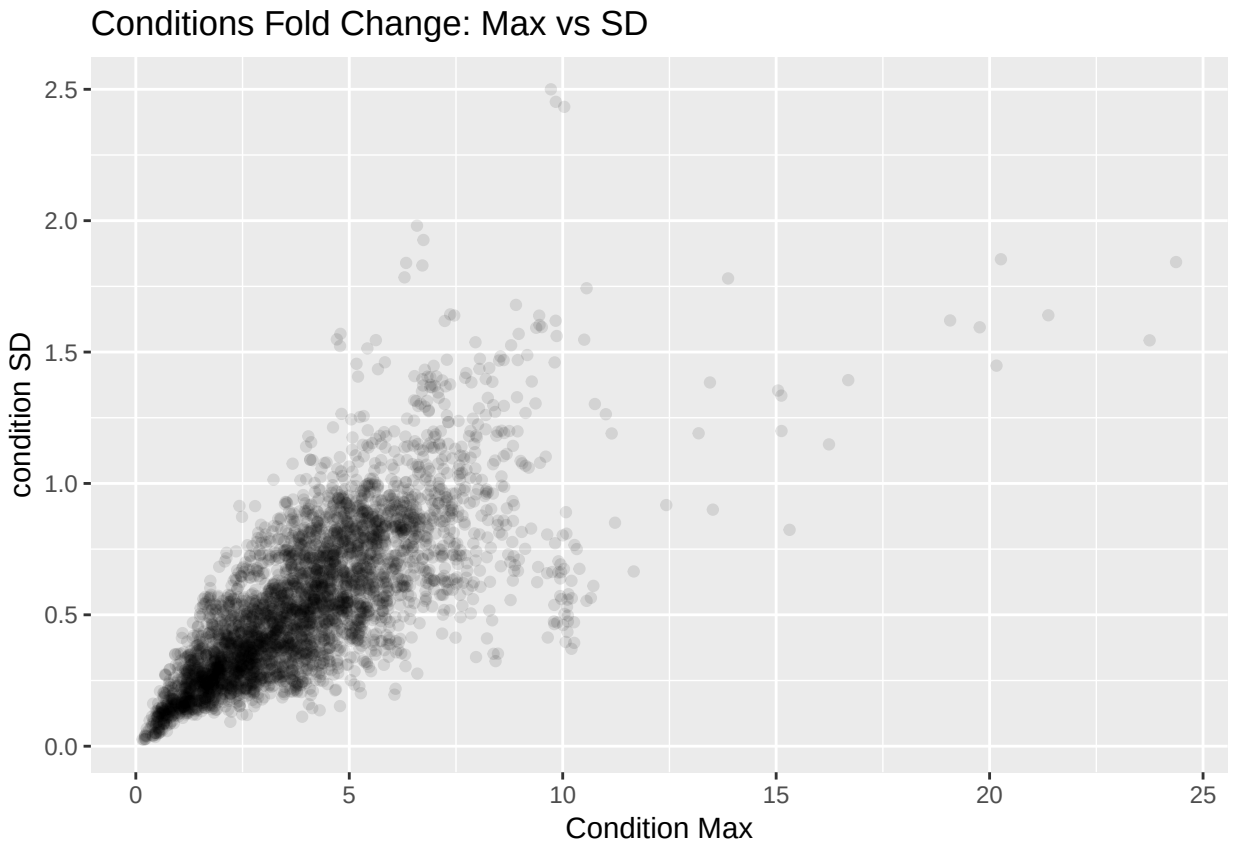
```
# Summary of first 20 columns
# summary(ecoli_expr[1:20])
```

```
numeric_ecoli_expr <- ecoli_expr[4:ncol(ecoli_expr)]

max_e <- sapply(numeric_ecoli_expr, max)
min_e <- sapply(numeric_ecoli_expr, min)
med_e <- sapply(numeric_ecoli_expr, median)
sd_e <- sapply(numeric_ecoli_expr, sd)

# Summary per condition
ecoli_expr_summary <- tibble( condition = 1:length(med_e), med_fold = med_e,
                          sd_fold = sd_e, max_fold = max_e, min_fold = min_e )
```

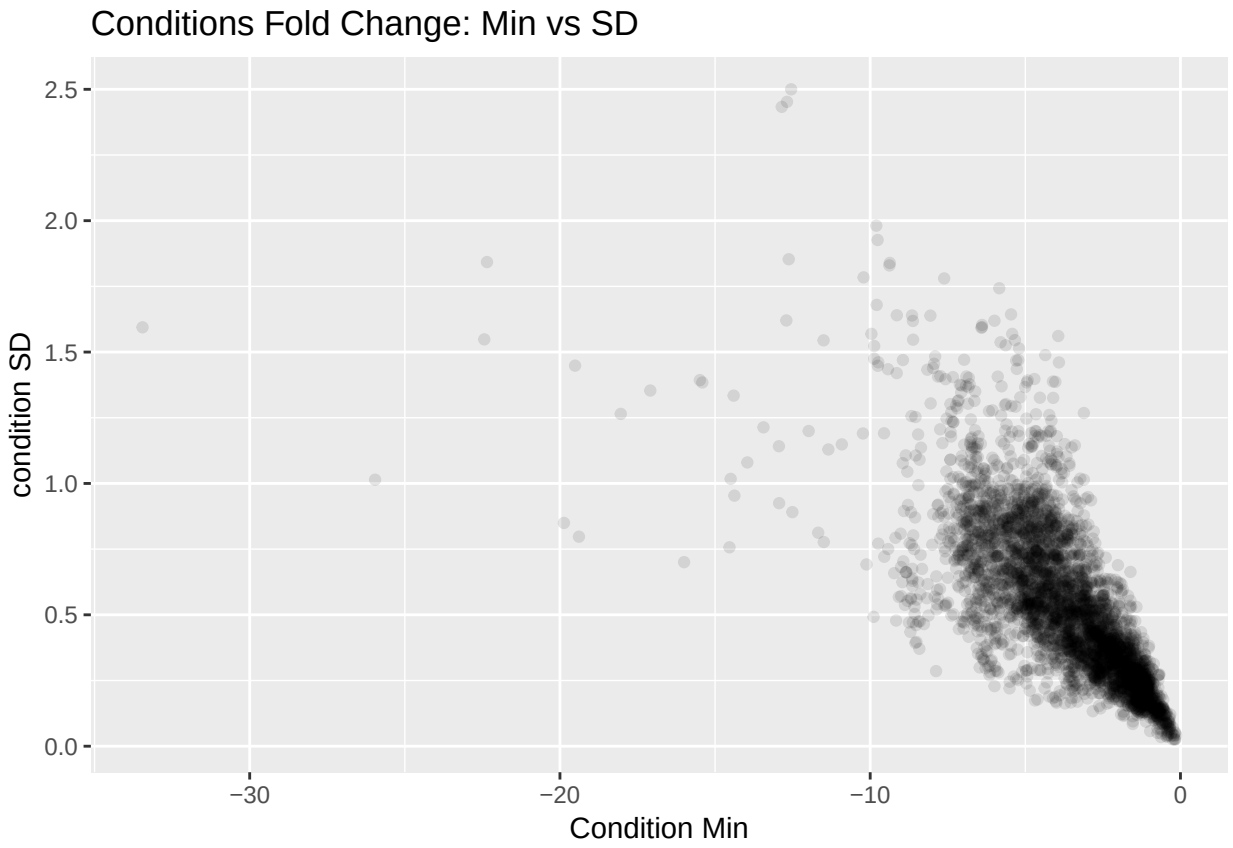
```
ggplot(ecoli_expr_summary)+
  geom_point( aes(med_fold, sd_fold), alpha = 1/10 )+
  xlab('Condition Median')+
  ylab('condition SD')+
  ggtitle('Conditions Fold Change: Median vs SD')
```



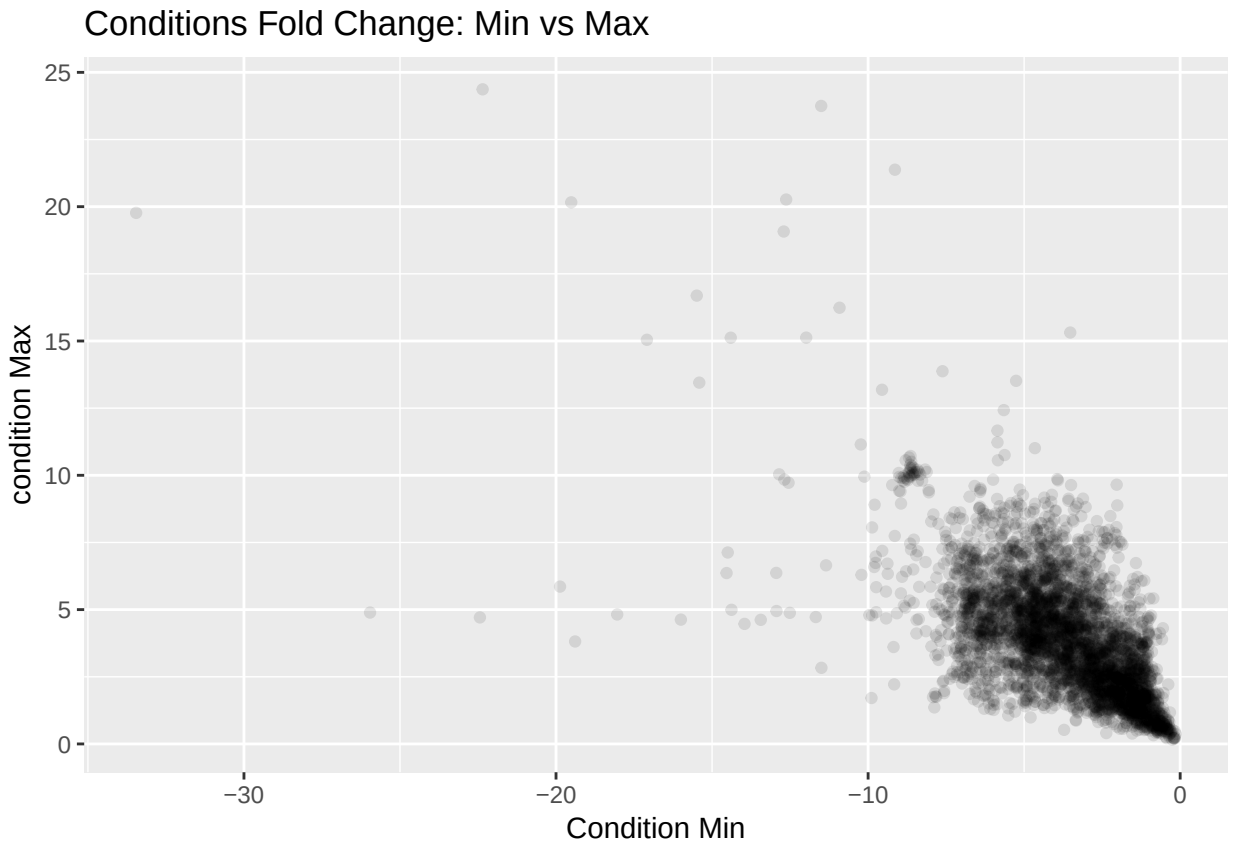

```
ggplot(ecoli_expr_summary)+
  geom_point( aes(max_fold, sd_fold), alpha = 1/10 )+
  xlab('Condition Max')+
  ylab('condition SD')+
  ggtitle('Conditions Fold Change: Max vs SD')
```




```
ggplot(ecoli_expr_summary)+
  geom_point( aes(min_fold, sd_fold), alpha = 1/10 )+
  xlab('Condition Min')+
  ylab('condition SD')+
  ggtitle('Conditions Fold Change: Min vs SD')
```




```
ggplot(ecoli_expr_summary)+
  geom_point( aes(min_fold, max_fold), alpha = 1/10 )+
  xlab('Condition Min')+
  ylab('condition Max')+
  ggtitle('Conditions Fold Change: Min vs Max')
```

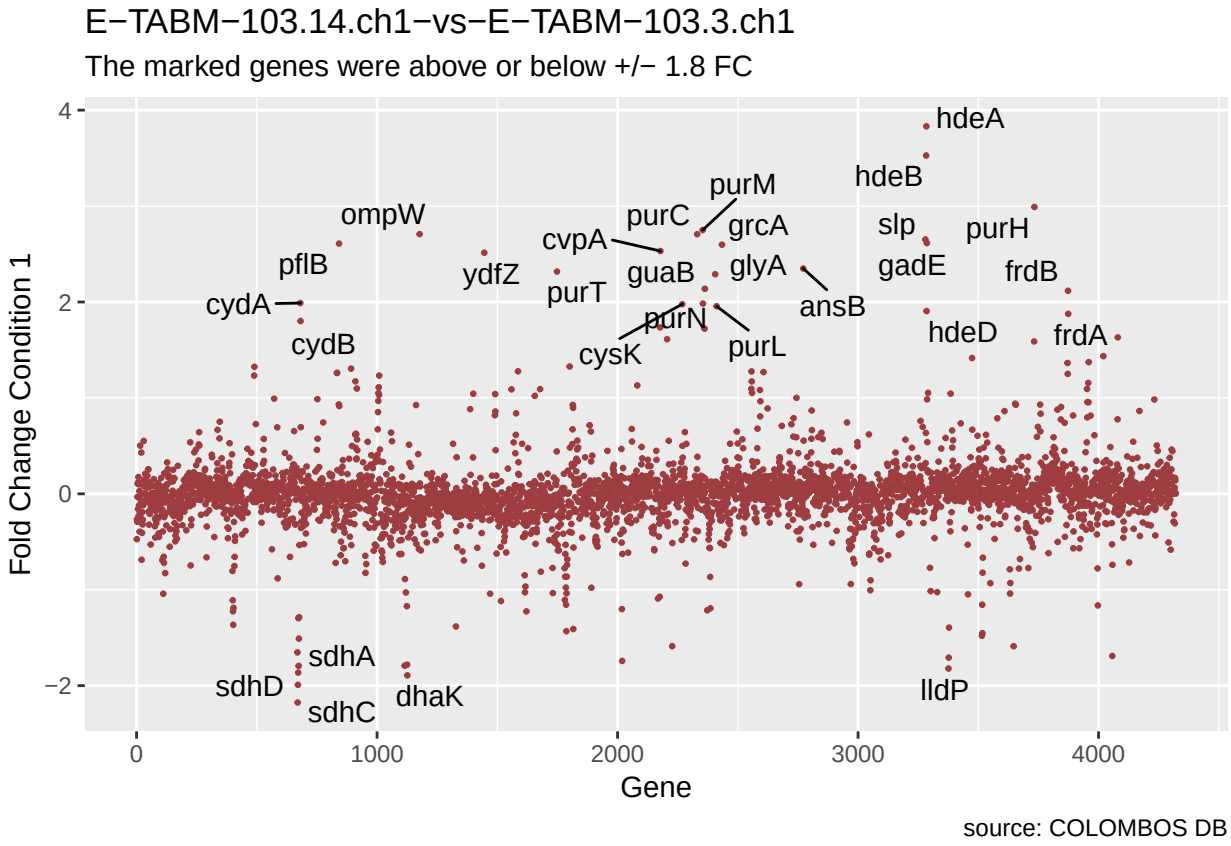


Example Conditions

Number 1 E-TABM-103.14.ch1-vs-E-TABM-103.3.ch1

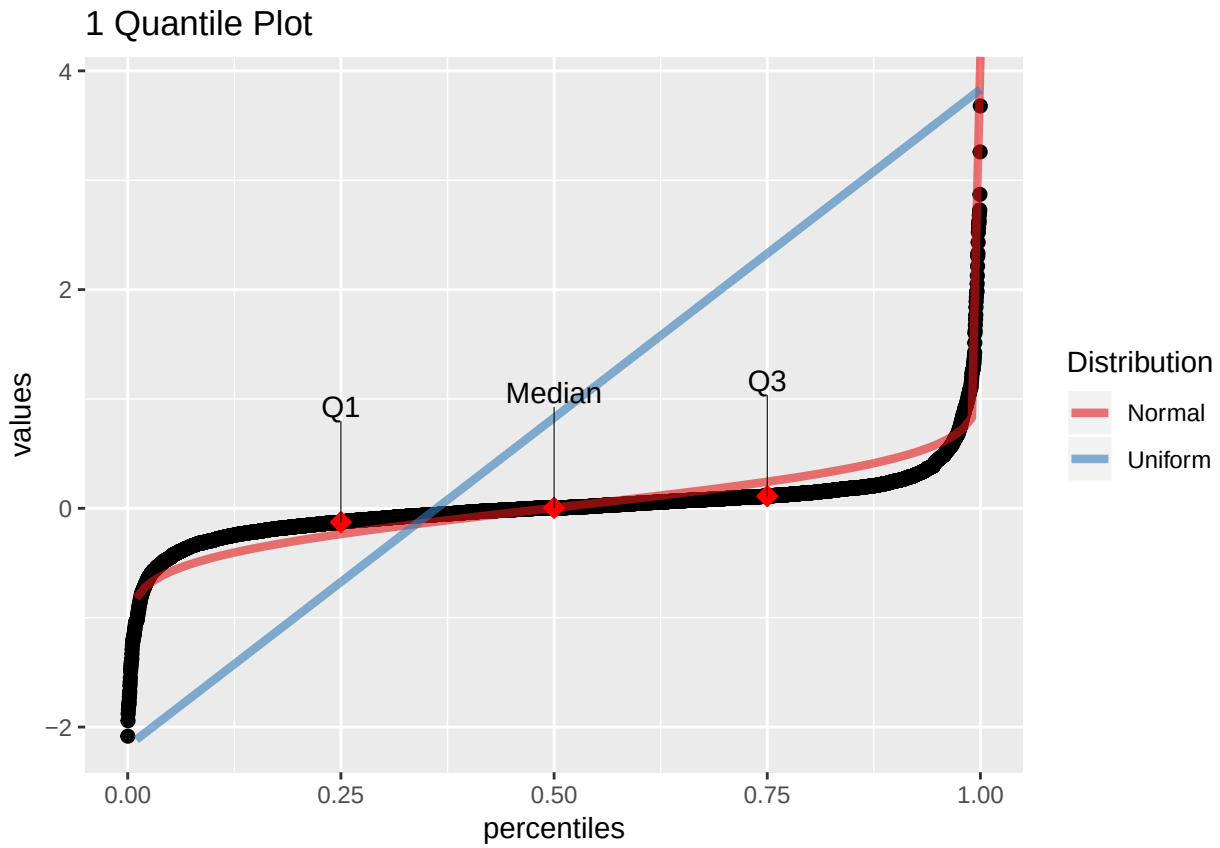
MEDIUM.LB:1 TIME:186min GROWTH.EXPONENTIAL:1 STRAIN.RP437:1

```
ggplot(ecoli_expr, aes(x = 1:nrow(ecoli_expr), y = `1`))+
  geom_point(size = 0.5, color = '#9e3d40')+
  geom_text_repel( aes(label = ifelse(`1` >= 1.8, `Gene name`, '')) )+
  geom_text_repel( aes(label = ifelse(`1` <= -1.8, `Gene name`, '')) )+
  labs(title="E-TABM-103.14.ch1-vs-E-TABM-103.3.ch1",
     subtitle="The marked genes were above or below +/- 1.8 FC",
     x = "Gene",
     y = "Fold Change Condition 1",
     caption="source: COLOMBOS DB")
```



source: COLOMBOS DB

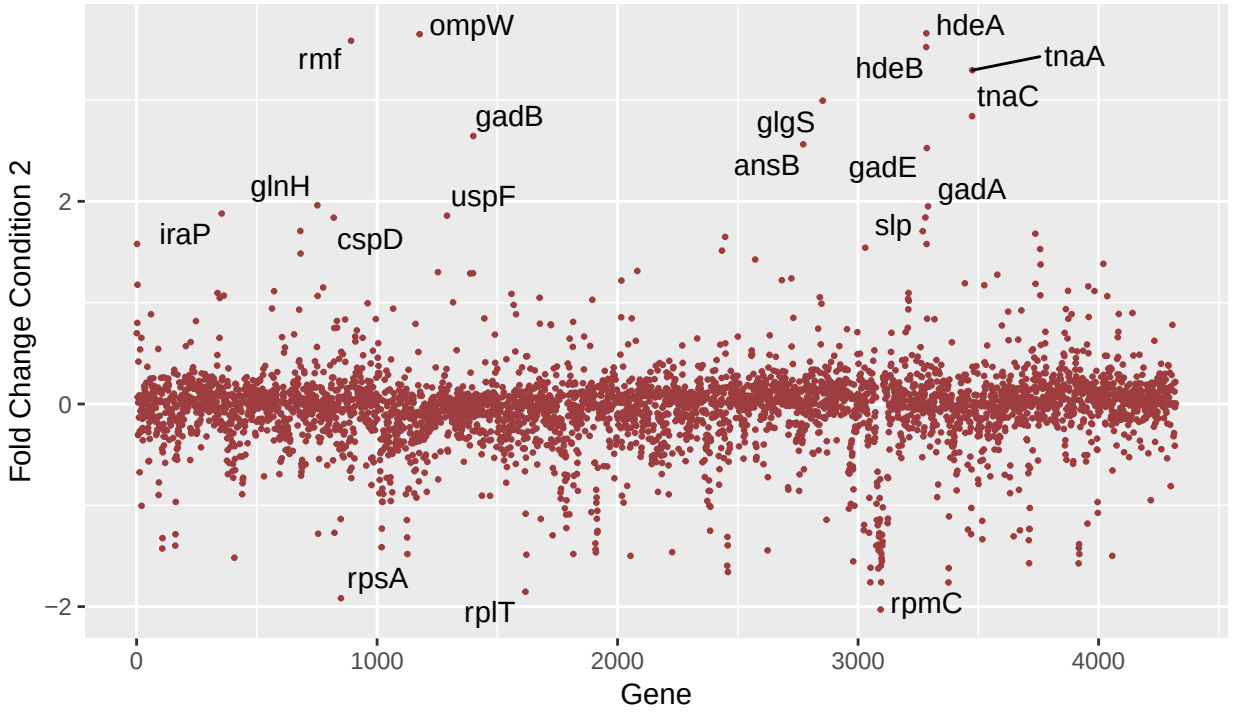
```
# Quantile plot
quantile_plot(ecoli_expr, '1')
```



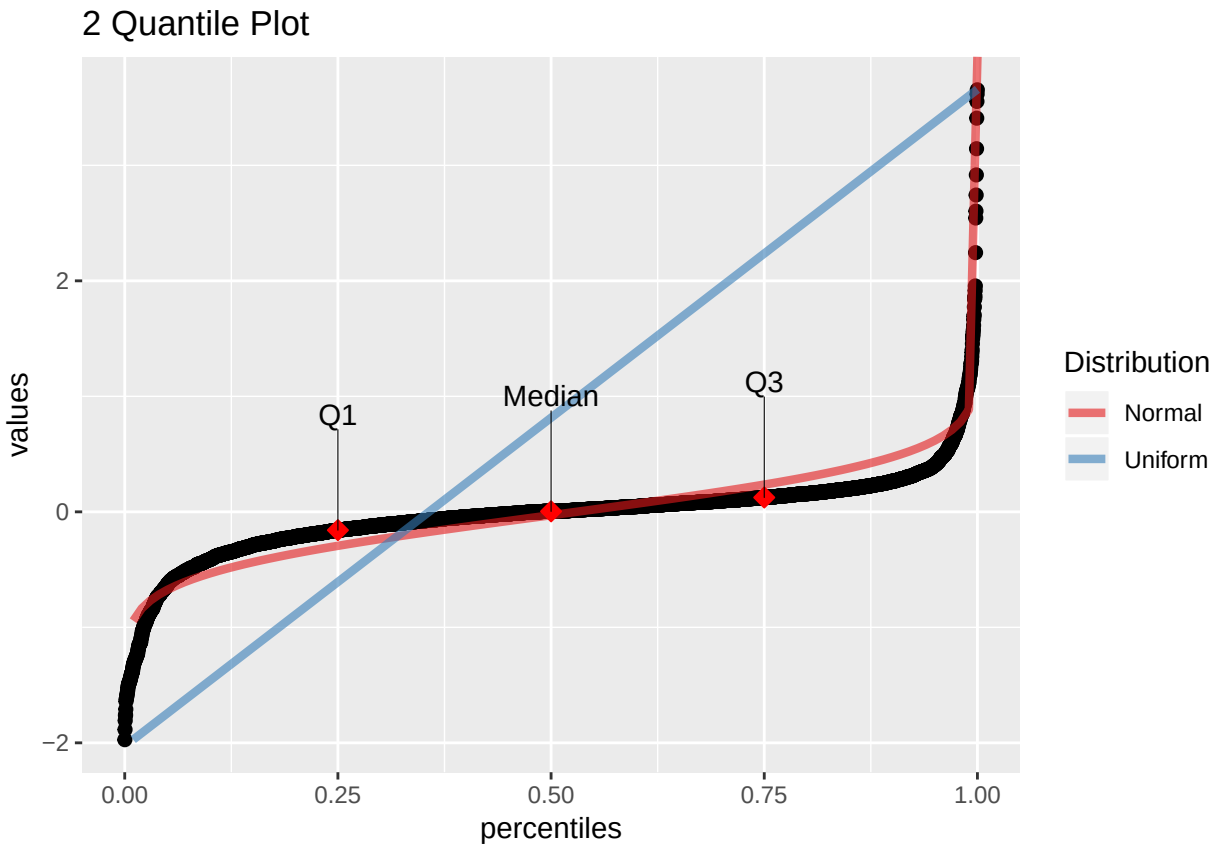
Number 2 E-TABM-103.19.ch1-vs-E-TABM-103.3.ch1

MEDIUM.LB:1 TIME:186min GROWTH.EXPONENTIAL:1 STRAIN.RP437:1

```
ggplot(ecoli_expr, aes(x = 1:nrow(ecoli_expr), y = `2`))+
  geom_point(size = 0.5, color = '#9e3d40')+
  geom_text_repel( aes(label = ifelse(`2` >= 1.8, `Gene name`, '')) )+
  geom_text_repel( aes(label = ifelse(`2` <= -1.8, `Gene name`, '')) )+
  labs(title="E-TABM-103.19.ch1-vs-E-TABM-103.3.ch1",
   subtitle="The marked genes were above or below +/- 1.8 FC",
   x = "Gene",
   y = "Fold Change Condition 2",
   caption="source: COLOMBOS DB")
```




```
# Quantile plot
quantile_plot(ecoli_expr, '2')
```



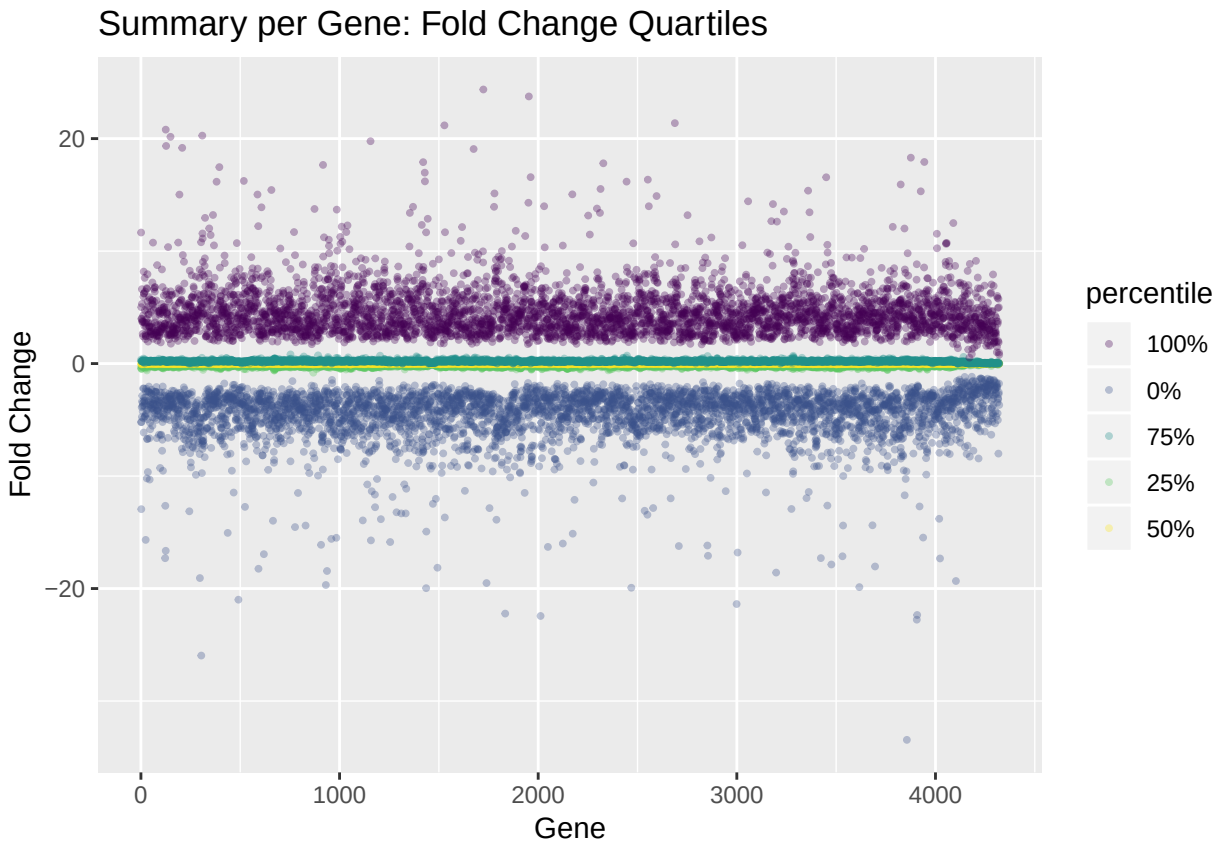
Quartiles of fold change per gene

```
# Gene quartiles
gene_quartiles <- apply(numeric_ecoli_expr, MARGIN = 1, quantile) %>%
  t() %>%
  as_tibble() %>%
  mutate(gene = ecoli_expr$`Gene name`, num_gene = 1:n_gene) %>%
  gather(`0%`, `25%`, `50%`, `75%`, `100%`,
       key = 'percentile', value = 'fold_change')

# Percentil as a factor
gene_quartiles$percentile <- factor(gene_quartiles$percentile)
# Reorder Levels to get a better plot
gene_quartiles$percentile <- factor(gene_quartiles$percentile,
       levels = levels(gene_quartiles$percentile)[c(2,1,5,3,4)])
head(gene_quartiles)
```

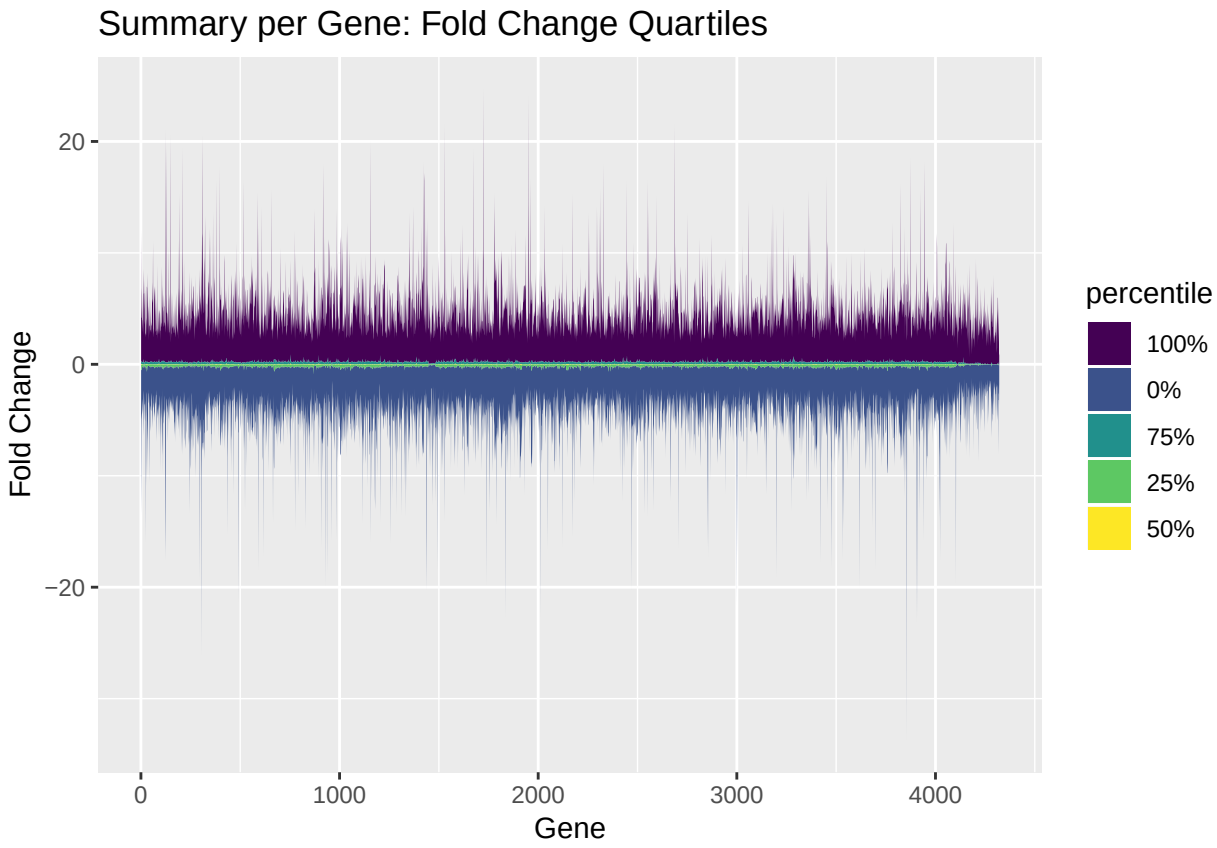
```
## # A tibble: 6 x 4
##   gene  num_gene percentile fold_change
##   <chr>    <int> <fct>            <dbl>
## 1 thrL         1 0%               -5.21
## 2 thrA         2 0%              -12.9
## 3 thrB         3 0%               -5.20
## 4 thrC         4 0%               -3.88
## 5 yaaX         5 0%               -4.75
```

```
## 6 yaaA          6 0%                -3.37
```

```
# Geom Line
ggplot(gene_quartiles, aes(x = 1:n_gene))+
  geom_point(aes(x = num_gene, y = fold_change, color = percentile), alpha = 1/3, size = 0.7)+
  scale_color_viridis_d()+
  labs(title = 'Summary per Gene: Fold Change Quartiles',
       x = 'Gene',
       y = 'Fold Change')
```

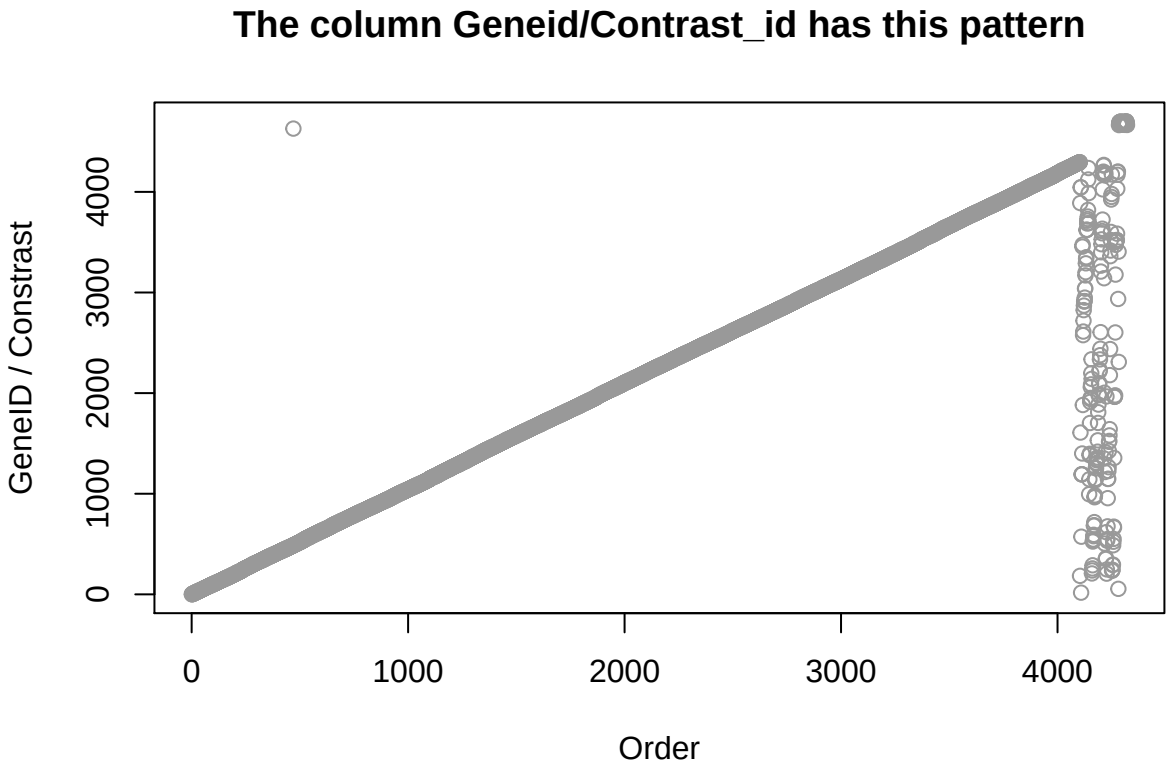



```
# Area Fill
ggplot(gene_quartiles, aes(x = 1:n_gene))+
  geom_area(aes(x = num_gene, y = fold_change, fill = percentile) )+
  scale_fill_viridis_d()+
  labs(title = 'Summary per Gene: Fold Change Quartiles',
       x = 'Gene',
       y = 'Fold Change')
```




```
# Area Fill Zoom
ggplot(gene_quartiles, aes(x = 1:n_gene))+
  geom_area(aes(x = num_gene, y = fold_change, fill = percentile) )+
  coord_cartesian(ylim = c(-1,1))+
  scale_fill_viridis_d()+
  labs(title = 'Summary per Gene: Fold Change Quartiles',
       x = 'Gene',
       y = 'Fold Change')
```


```
# What Geneid/Constrast_id are?
plot(ecoli_expr$`Geneid/Contrast_id`,
     xlab = 'Order',
     ylab = 'GeneID / Constrast',
     main = 'The column Geneid/Contrast_id has this pattern',
     col = 'gray60'
)
```



Saving tables

```
# Saving the table
# Inverting to take the conditions as observations
cond_gene <- as_tibble( t( numeric_ecoli_expr ) )
names(cond_gene) <- ecoli_expr$`Gene name`

head(cond_gene)
```

```
## # A tibble: 6 x 4,321
##      thrL    thrA   thrB    thrC     yaaX   yaaA    yaaJ    talB    mog
##     <dbl>   <dbl>  <dbl>   <dbl>    <dbl>  <dbl>   <dbl>   <dbl>  <dbl>
## 1 -0.473  -0.266  -0.292 -0.0312  0.106   -0.158  0.121  0.176   -0.230
## 2  0.698   1.58    0.800  1.18    0.0671  -0.307  0.0137 0.418   -0.296
## 3  0.0927 -0.0381 -0.108 -0.0184  0.196   -0.406  0.0707 0.272   -0.440
## 4  0.182   0.236  -0.157  0.459   0.200   -0.183  0.0971 0.297   -0.294
## 5  0.0269 -0.575  -0.543 -0.121  -0.00258 -0.520 -0.291  0.00734 -0.664
## 6  0.293  -0.166  -0.276  0.115   0.231   -0.302  0.157  0.00156 -0.452
## # ... with 4,312 more variables: yaaH <dbl>, yaaW <dbl>, yaaI <dbl>,
## #   dnaK <dbl>, dnaJ <dbl>, insL <dbl>, mokC <dbl>, nhaA <dbl>,
## #   nhaR <dbl>, insB <dbl>, insA <dbl>, rpsT <dbl>, yaaY <dbl>,
## #   ribF <dbl>, ileS <dbl>, lspA <dbl>, fkpB <dbl>, ispH <dbl>,
## #   rihC <dbl>, dapB <dbl>, carA <dbl>, carB <dbl>, caiF <dbl>,
## #   caiE <dbl>, caiD <dbl>, caiC <dbl>, caiB <dbl>, caiA <dbl>,
## #   caiT <dbl>, fixA <dbl>, fixB <dbl>, fixC <dbl>, fixX <dbl>,
## #   yaaU <dbl>, kefF <dbl>, kefC <dbl>, folA <dbl>, apaH <dbl>,
## #   apaG <dbl>, ksgA <dbl>, pdxA <dbl>, surA <dbl>, imp <dbl>, djlA <dbl>,
## #   rluA <dbl>, hepA <dbl>, polB <dbl>, araD <dbl>, araA <dbl>,
```

```
## #   araB <dbl>, araC <dbl>, yabI <dbl>, thiQ <dbl>, thiP <dbl>,
## #   tbpA <dbl>, sgrR <dbl>, setA <dbl>, leuD <dbl>, leuC <dbl>,
## #   leuB <dbl>, leuA <dbl>, leuL <dbl>, leuO <dbl>, ilvI <dbl>,
## #   ilvH <dbl>, cra <dbl>, mraZ <dbl>, mraW <dbl>, ftsL <dbl>, ftsI <dbl>,
## #   murE <dbl>, murF <dbl>, mraY <dbl>, murD <dbl>, ftsW <dbl>,
## #   murG <dbl>, murC <dbl>, ddlB <dbl>, ftsQ <dbl>, ftsA <dbl>,
## #   ftsZ <dbl>, lpxC <dbl>, secM <dbl>, secA <dbl>, mutT <dbl>,
## #   yacG <dbl>, yacF <dbl>, coaE <dbl>, guaC <dbl>, hofC <dbl>,
## #   hofB <dbl>, ppdD <dbl>, nadC <dbl>, ampD <dbl>, ampE <dbl>,
## #   aroP <dbl>, pdhR <dbl>, aceE <dbl>, aceF <dbl>, lpd <dbl>, ...
```

```
# Saving to disk
# saveRDS(cond_gene, file = 'conditions_vs_genes.RDS')

# Creating gene vs condition table
# gene_cond <- as_tibble(t(cond_gene))
# colnames(gene_cond) <- paste('C', 1:ncol(gene_cond), sep = '')
# gene_cond$gene <- names(cond_gene)
# gene_cond <- gene_cond[ , c(ncol(gene_cond),1:ncol(gene_cond)-1) ]
# gene_cond

# Saving gene vs condition table
# saveRDS(gene_cond , 'genes_vs_conditions.RDS')

# Loading tables
# condition_gene <- readRDS('genes_vs_conditions.RDS')
gene_condition <- readRDS('genes_vs_conditions.RDS')

head(gene_condition)
```

```
## # A tibble: 6 x 4,078
##   gene       C1      C2      C3     C4       C5     C6      C7      C8
##   <chr>   <dbl>   <dbl>   <dbl>  <dbl>    <dbl>  <dbl>   <dbl>   <dbl>
## 1 thrL  -0.473   0.698   0.0927  0.182  0.0269   0.293  0.148  -0.471
## 2 thrA  -0.266   1.58   -0.0381  0.236 -0.575   -0.166  0.0280 -0.299
## 3 thrB  -0.292   0.800  -0.108  -0.157 -0.543   -0.276  0.328  -0.218
## 4 thrC  -0.0312  1.18   -0.0184  0.459 -0.121    0.115 -0.127  -0.131
## 5 yaaX   0.106   0.0671  0.196   0.200 -0.00258  0.231  0.173  -0.0208
## 6 yaaA  -0.158  -0.307  -0.406  -0.183 -0.520   -0.302  0.0686 -0.314
## # ... with 4,069 more variables: C9 <dbl>, C10 <dbl>, C11 <dbl>,
## #   C12 <dbl>, C13 <dbl>, C14 <dbl>, C15 <dbl>, C16 <dbl>, C17 <dbl>,
## #   C18 <dbl>, C19 <dbl>, C20 <dbl>, C21 <dbl>, C22 <dbl>, C23 <dbl>,
## #   C24 <dbl>, C25 <dbl>, C26 <dbl>, C27 <dbl>, C28 <dbl>, C29 <dbl>,
## #   C30 <dbl>, C31 <dbl>, C32 <dbl>, C33 <dbl>, C34 <dbl>, C35 <dbl>,
## #   C36 <dbl>, C37 <dbl>, C38 <dbl>, C39 <dbl>, C40 <dbl>, C41 <dbl>,
## #   C42 <dbl>, C43 <dbl>, C44 <dbl>, C45 <dbl>, C46 <dbl>, C47 <dbl>,
## #   C48 <dbl>, C49 <dbl>, C50 <dbl>, C51 <dbl>, C52 <dbl>, C53 <dbl>,
## #   C54 <dbl>, C55 <dbl>, C56 <dbl>, C57 <dbl>, C58 <dbl>, C59 <dbl>,
## #   C60 <dbl>, C61 <dbl>, C62 <dbl>, C63 <dbl>, C64 <dbl>, C65 <dbl>,
## #   C66 <dbl>, C67 <dbl>, C68 <dbl>, C69 <dbl>, C70 <dbl>, C71 <dbl>,
## #   C72 <dbl>, C73 <dbl>, C74 <dbl>, C75 <dbl>, C76 <dbl>, C77 <dbl>,
## #   C78 <dbl>, C79 <dbl>, C80 <dbl>, C81 <dbl>, C82 <dbl>, C83 <dbl>,
## #   C84 <dbl>, C85 <dbl>, C86 <dbl>, C87 <dbl>, C88 <dbl>, C89 <dbl>,
## #   C90 <dbl>, C91 <dbl>, C92 <dbl>, C93 <dbl>, C94 <dbl>, C95 <dbl>,
## #   C96 <dbl>, C97 <dbl>, C98 <dbl>, C99 <dbl>, C100 <dbl>, C101 <dbl>,
```

```
## #   C102 <dbl>, C103 <dbl>, C104 <dbl>, C105 <dbl>, C106 <dbl>,
## #   C107 <dbl>, C108 <dbl>, ...
```

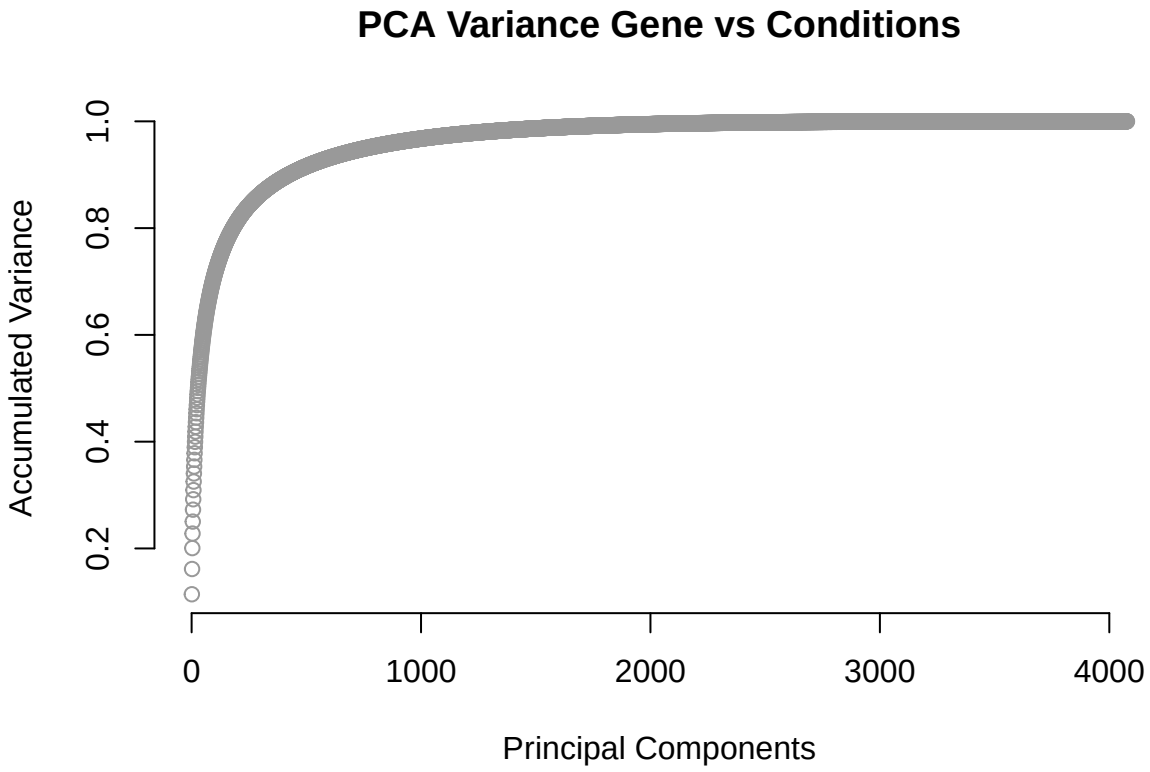
PCA and Clustering

```
PCA_reduce_dimension <- function(data, perct_tot_var = 0.95){
  pca_re <- prcomp(data)
  # Loadings
  var_per_comp <- pca_re$sdev^2
  # Cutoff for dim reduction
  var_cumsum <- cumsum(var_per_comp / sum(var_per_comp))
  # plot(var_cumsum)
  cut_idx <- which(var_cumsum >= perct_tot_var)[1]
  dim_reduc <- pca_re$x[,1:cut_idx]
  print(paste('Your data was reduced from',
              ncol(data), 'columns', 'to', cut_idx,
              'capturing', perct_tot_var * 100, 'percent of the variance'))
  return(list(reduced = dim_reduc, pca = pca_re))
}

# Running PCA to reduce dimensions
pca_results <- PCA_reduce_dimension(gene_condition[2:ncol(gene_condition)],
                                    perct_tot_var = 0.80)
```

```
## [1] "Your data was reduced from 4077 columns to 181 capturing 80 percent of the variance"
```

```
# Reduced table
reduced_gene_cond <- as_tibble(pca_results$reduced)
```

```
var_pca <- pca_results$pca$sdev ^ 2
total_var_pca <- sum(var_pca)
cum_var_pca <- cumsum(var_pca / total_var_pca)

plot(cum_var_pca,
     xlab = 'Principal Components',
     ylab = 'Accumulated Variance',
     main = 'PCA Variance Gene vs Conditions',
     col = 'gray60',
     frame.plot = FALSE # Remove the frame
)
```




```
# Variance captured by the first two components
(PCA1_explained <- round( var_pca[1] / total_var_pca * 100, 2 ))
```

```
## [1] 11.42
```

```
(PCA2_explained <- round( var_pca[2] / total_var_pca * 100, 2 ))
```

```
## [1] 4.73
```

Using NbClust to test for the number of gene clusters thus getting the best values for k (number of clusters) in k-means algorithm

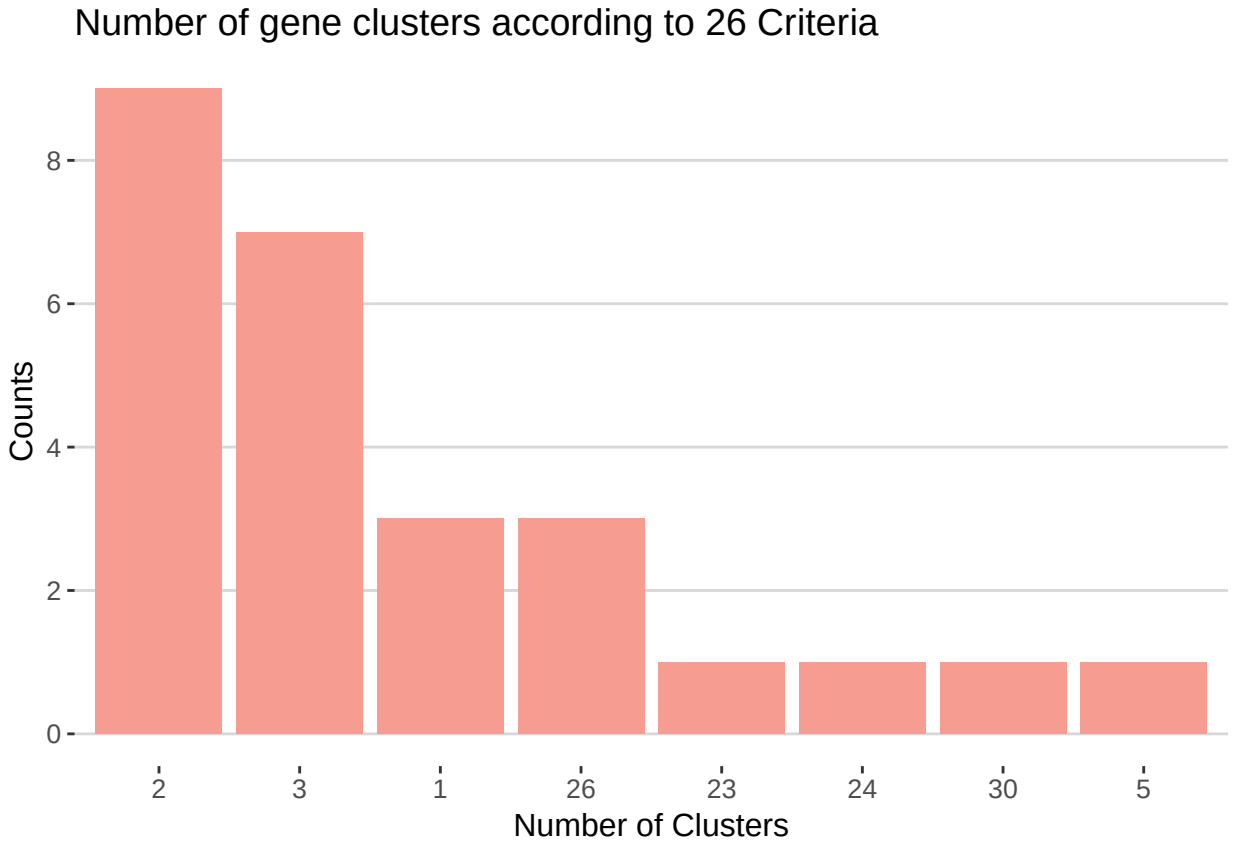
```
# Heavy computational line
# Time to run aprox 1 hour
# nc_ALL <- NbClust(reduced_gene_cond, min.nc=2, max.nc=30, method="kmeans", index = 'all')
# Saving the results
# saveRDS(nc_ALL, 'number_clusters_ALL_gene_cond.RDS')

# Loading the results
nc_ALL <- readRDS('number_clusters_ALL_gene_cond.RDS')
```

```
nc_gene <- as_tibble(t(nc_ALL$Best.nc))
nc_gene$Method <- colnames(nc_ALL$Best.nc)
nc_gene <- select(nc_gene, Method, everything()) %>%
  arrange(Number_clusters)
nc_gene[1:3, 2] <- 1
nc_gene
```

```
## # A tibble: 26 x 3
##    Method     Number_clusters Value_Index
##    <chr>                <dbl>       <dbl>
##  1 Scott                    1    -Inf
```

```
##  2 Dunn                    1       0
##  3 SDindex                 1       0
##  4 CH                      2     344.
##  5 Cindex                  2       2.76
##  6 DB                      2       0.205
##  7 Silhouette              2       1.03
##  8 Duda                    2    -117.
##  9 PseudoT2                2      -3.89
## 10 PtBiserial              2       2.02
## # ... with 16 more rows
```

```
ggplot(nc_gene)+
  geom_bar(aes(x = fct_infreq(as.character(Number_clusters))),
           size = 0.4,
           fill = '#F79C91')+
  scale_y_continuous(breaks= pretty_breaks())+
  labs(title="Number of gene clusters according to 26 Criteria",
     x = "Number of Clusters",
     y = "Counts")+
  theme_hc()
```



K-means

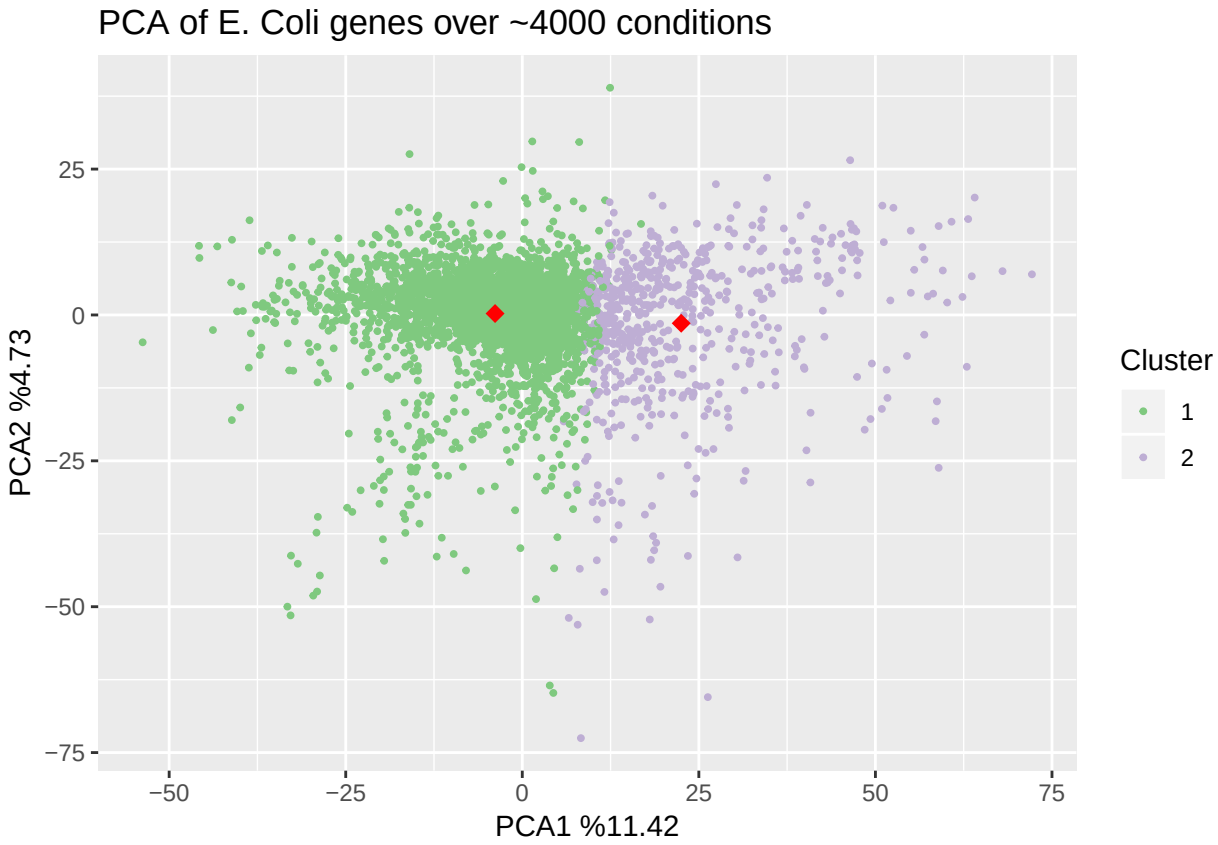
```
# K means with 2, 3 and 26 clusters
clust2 <- kmeans(reduced_gene_cond, centers = 2, nstart = 50, iter.max = 30)
clust3 <- kmeans(reduced_gene_cond, centers = 3, nstart = 50, iter.max = 30)
```

```
clust26 <- kmeans(reduced_gene_cond, centers = 26, nstart = 50, iter.max = 30)
```

```
gene_cond_clusters <- mutate(reduced_gene_cond,
       gene = ecoli_expr$`Gene name`,
       group2 = clust2$cluster,
       group3 = clust3$cluster,
       group26 = clust26$cluster) %>%
  select(gene, group2, group3, group26, everything())
```

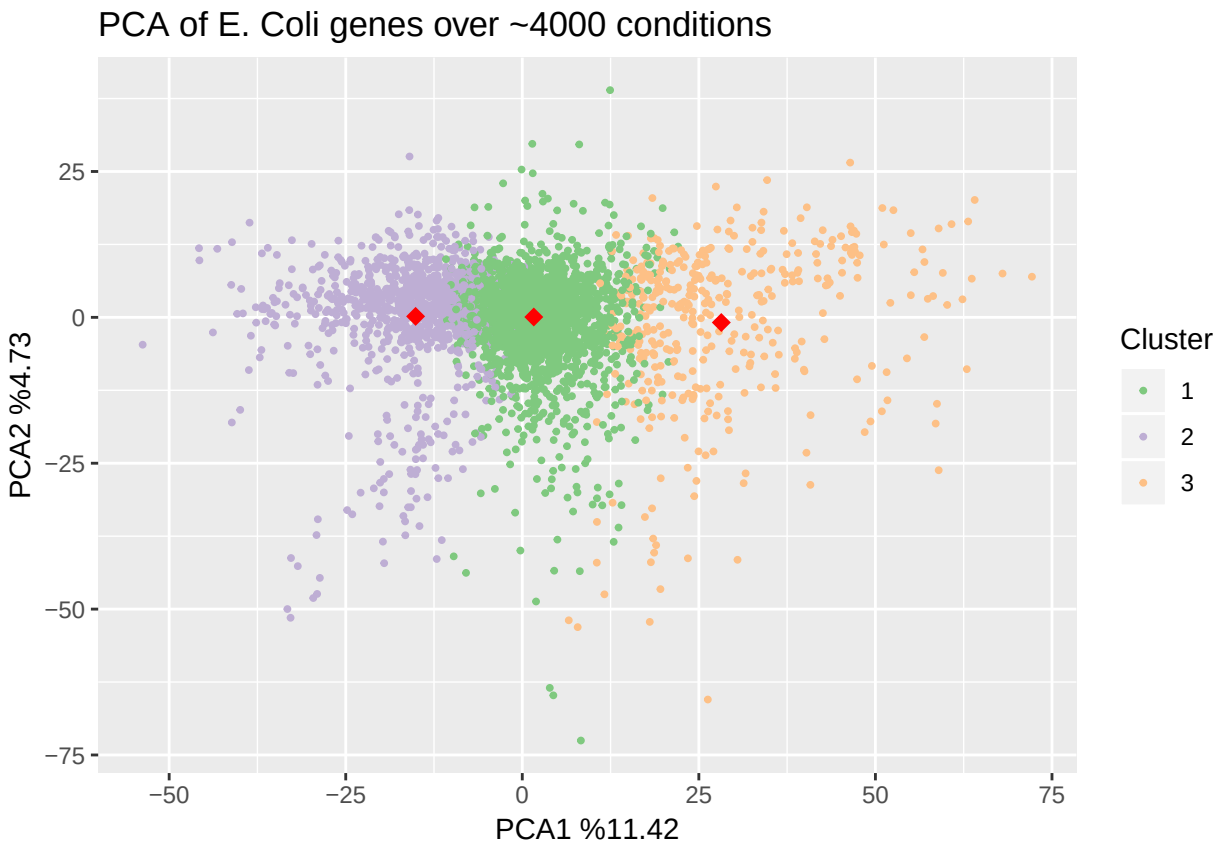
Plots PCA and Clustering

```
ggplot(gene_cond_clusters)+
  geom_point(aes(x = PC1, y = PC2, color = as.character(group2)),
             size = 0.7)+
  xlab(paste('PCA1 ', '%', PCA1_explained, sep = ''))+
  ylab(paste('PCA2 ', '%', PCA2_explained, sep = ''))+
  scale_color_brewer(type = 'qual')+
  labs(title = 'PCA of E. Coli genes over ~4000 conditions',
       color = 'Cluster')+
  geom_point(data = as_tibble(clust2$centers), aes(x = PC1, y = PC2),
             color = 'red', size = 3, shape = 18)
```

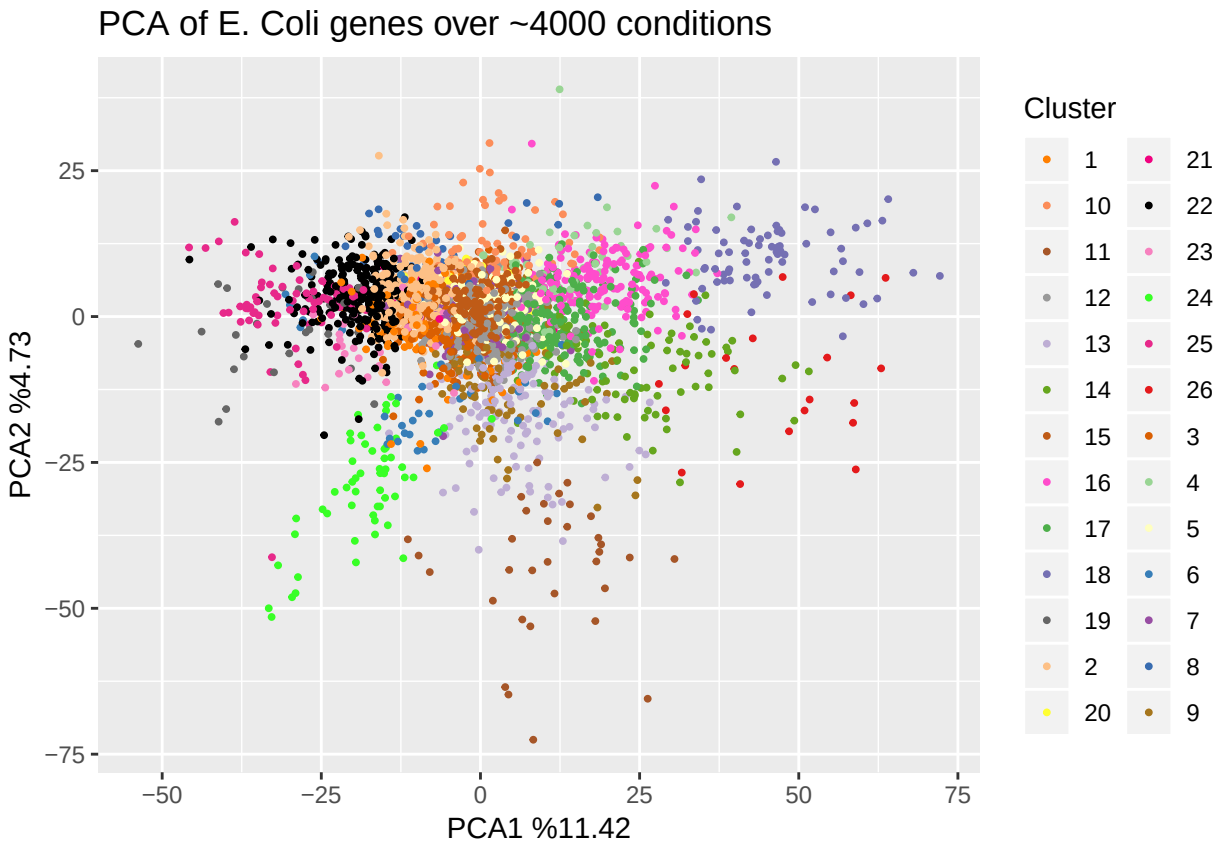



```
ggplot(gene_cond_clusters)+
  geom_point(aes(x = PC1, y = PC2, color = as.character(group3)),
             size = 0.7)+
```

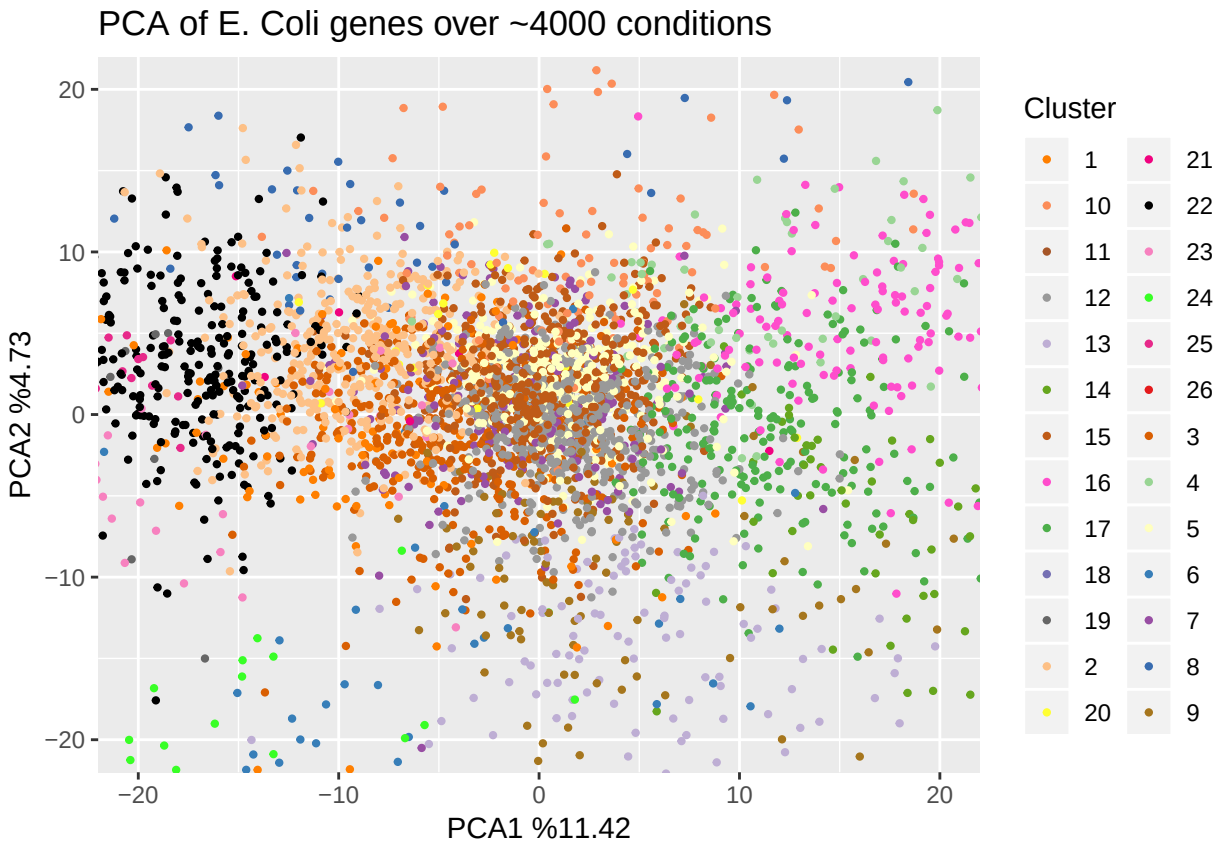
```
  xlab(paste('PCA1 ', '%', PCA1_explained, sep = ''))+
  ylab(paste('PCA2 ', '%', PCA2_explained, sep = ''))+
  scale_color_brewer(type = 'qual')+
  labs(title = 'PCA of E. Coli genes over ~4000 conditions',
       color = 'Cluster')+
  geom_point(data = as_tibble(clust3$centers), aes(x = PC1, y = PC2),
             color = 'red', size = 3, shape = 18)
```



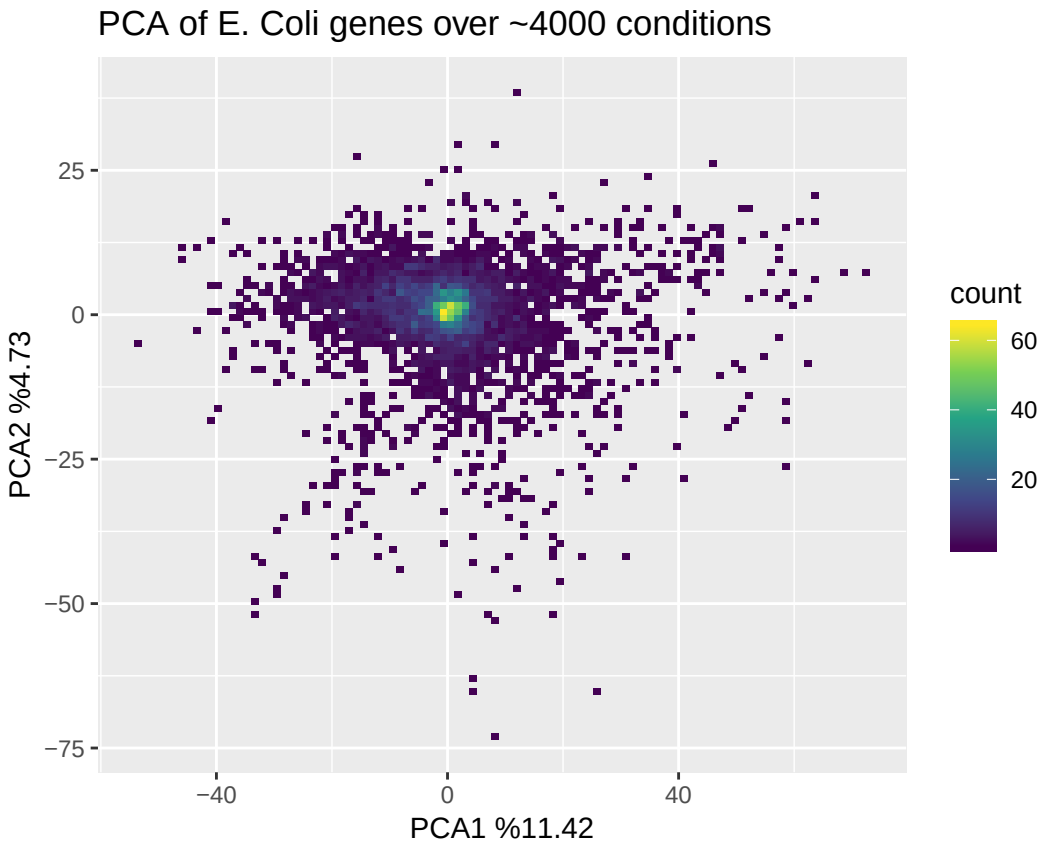

```
custom26 <- c("#FF7F00", "#FC8D59", "#A65628", "#999999",
              "#BEAED4", "#66A61E", "#BF5B17", "#FF4DCC",
              "#4DAF4A", "#7570B3", "#666666", "#FDC086",
              "#FFFF33", "#F0027F", "#000000", "#F781BF",
              "#38FF26", "#E7298A", "#E41A1C", "#D95F02",
              "#99D594", "#FFFFBF", "#377EB8", "#984EA3",
              "#386CB0", "#A6761D")

ggplot(gene_cond_clusters)+
  geom_point(aes(x = PC1, y = PC2, color = as.character(group26)),
             size = 0.7)+
  xlab(paste('PCA1 ', '%', PCA1_explained, sep = ''))+
  ylab(paste('PCA2 ', '%', PCA2_explained, sep = ''))+
  scale_color_manual(values = custom26)+
  labs(title = 'PCA of E. Coli genes over ~4000 conditions',
       color = 'Cluster')
```




```
ggplot(gene_cond_clusters)+
  geom_point(aes(x = PC1, y = PC2, color = as.character(group26)),
             size = 0.8)+
  xlab(paste('PCA1 ', '%', PCA1_explained, sep = ''))+
  ylab(paste('PCA2 ', '%', PCA2_explained, sep = ''))+
  scale_color_manual(values = custom26)+
  labs(title = 'PCA of E. Coli genes over ~4000 conditions',
       color = 'Cluster')+
  coord_cartesian(xlim = c(-20,20), ylim = c(-20,20))
```

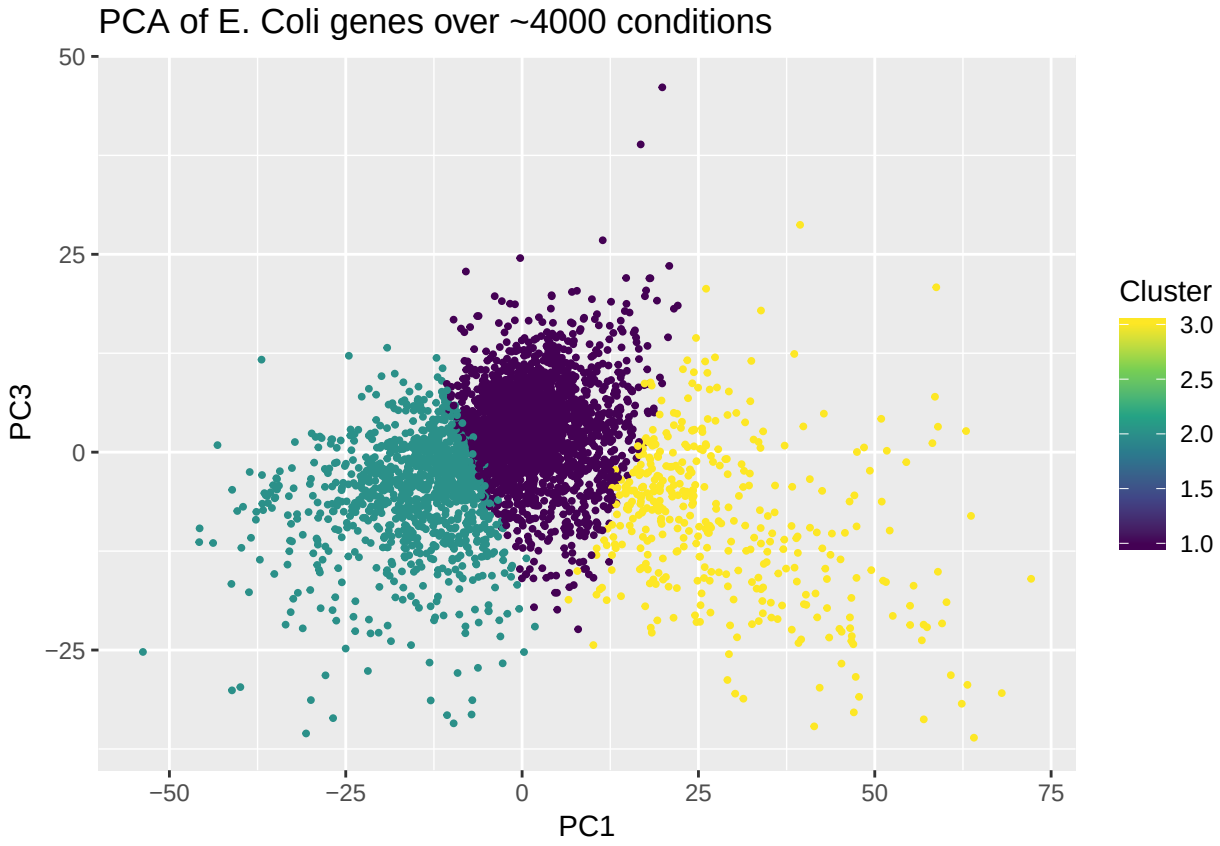



```
ggplot(gene_cond_clusters)+
  geom_bin2d(aes(x = PC1, y = PC2),
             bins = 100)+
  xlab(paste('PCA1 ', '%', PCA1_explained, sep = ''))+
  ylab(paste('PCA2 ', '%', PCA2_explained, sep = ''))+
  scale_fill_viridis_c()+
  labs(title = 'PCA of E. Coli genes over ~4000 conditions')+
  coord_equal(ratio=1)
```

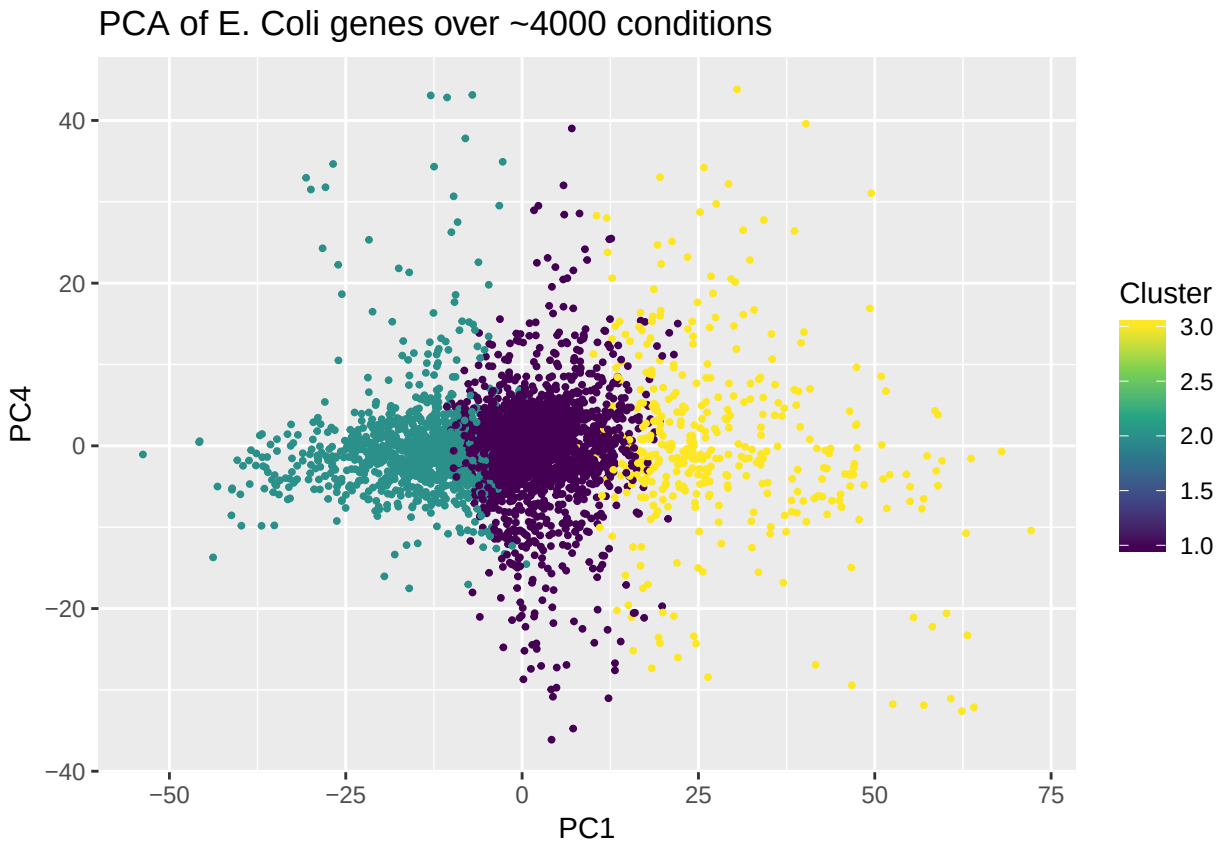



```
explore_pcs3 <- function(pcx, pcy){
  ggplot(gene_cond_clusters)+
  geom_point(aes_string(x = pcx, y = pcy, color = 'group3'),
             size = 0.7)+
  scale_color_viridis_c()+
  labs(title = 'PCA of E. Coli genes over ~4000 conditions',
       color = 'Cluster')
  }
```

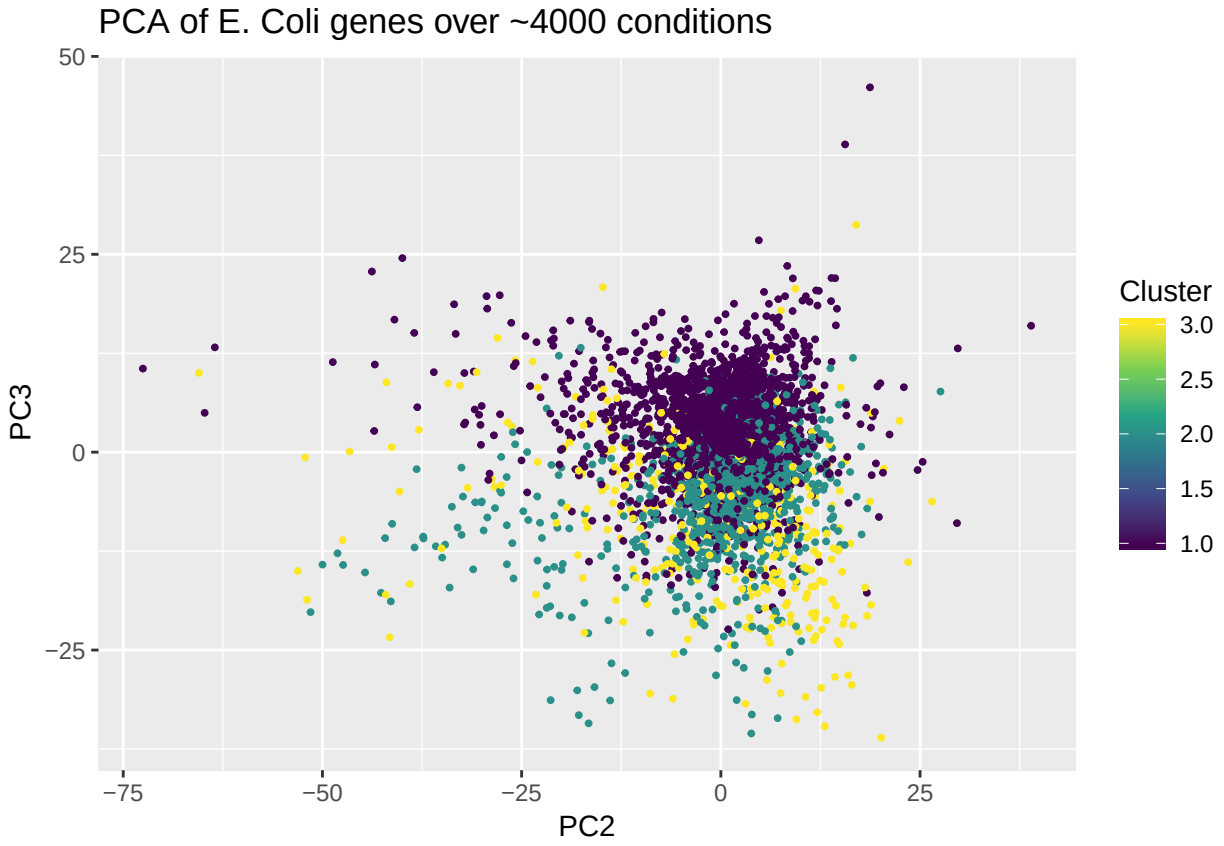
```
explore_pcs3('PC1', 'PC3')
```



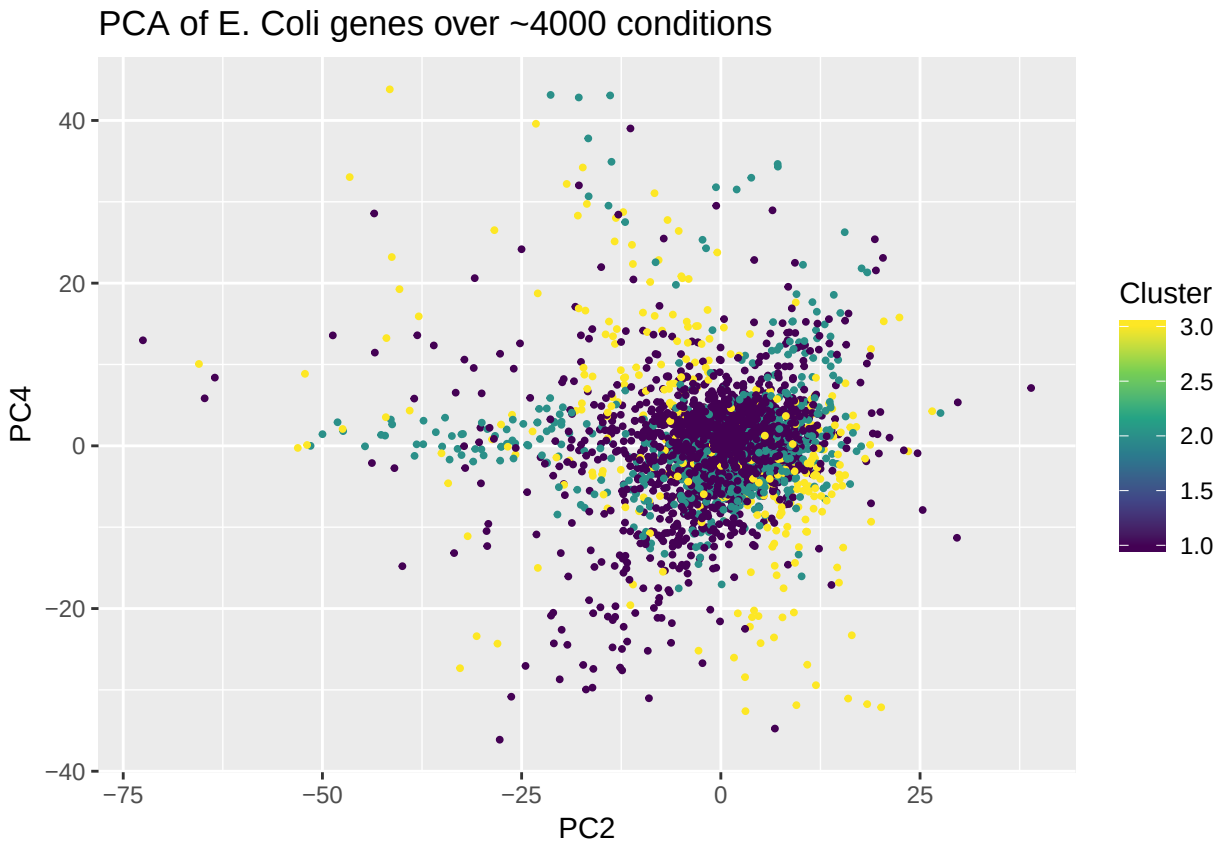

```
explore_pcs3('PC1', 'PC4')
```




```
explore_pcs3('PC2', 'PC3')
```




```
explore_pcs3('PC2', 'PC4')
```


PCA of E. Coli genes over ~4000 conditions
PC4
PC2
40
20
0
−20
−40
−75
−50
−25
0
25
Cluster
3.0
2.5
2.0
1.5
1.0