ZNF808 Assessment

Ethan de Villiers

2024-09-04

Assessment time start: 5:34pm Wednesday 4th September

Previous steps / analysis for differential expression:
* Compared control vs ZNF808 KO cells (in vitro)
* Measured abundance (non-neg $_{\text{normalised}}$ STANDARDISED count) of genes
> normalised refers to value-min / range = values between 0 - 1
> this mean it was standardised? Unlikely Z score, values too high
* Differential expression = gene is significantly expressed in control or KO cells
* Stratified by pancreatic differentiation (S0 - S4)
> Allows cascaded effects of KO and gene expression to be viewed across differentiation stages

Inherited data:
* Already done the KO vs Control = log2FoldChange, lfcSE, stat, pvalue, padj
* I get differential expression data (KO vs Control)

Task 1 specification:
* Differentially expressed = padj < 0.05
* ACTIVATED = differentially_expressed + log2FoldChange > 0
* REPRESSED = differentially_expressed + log2FoldChange < 0

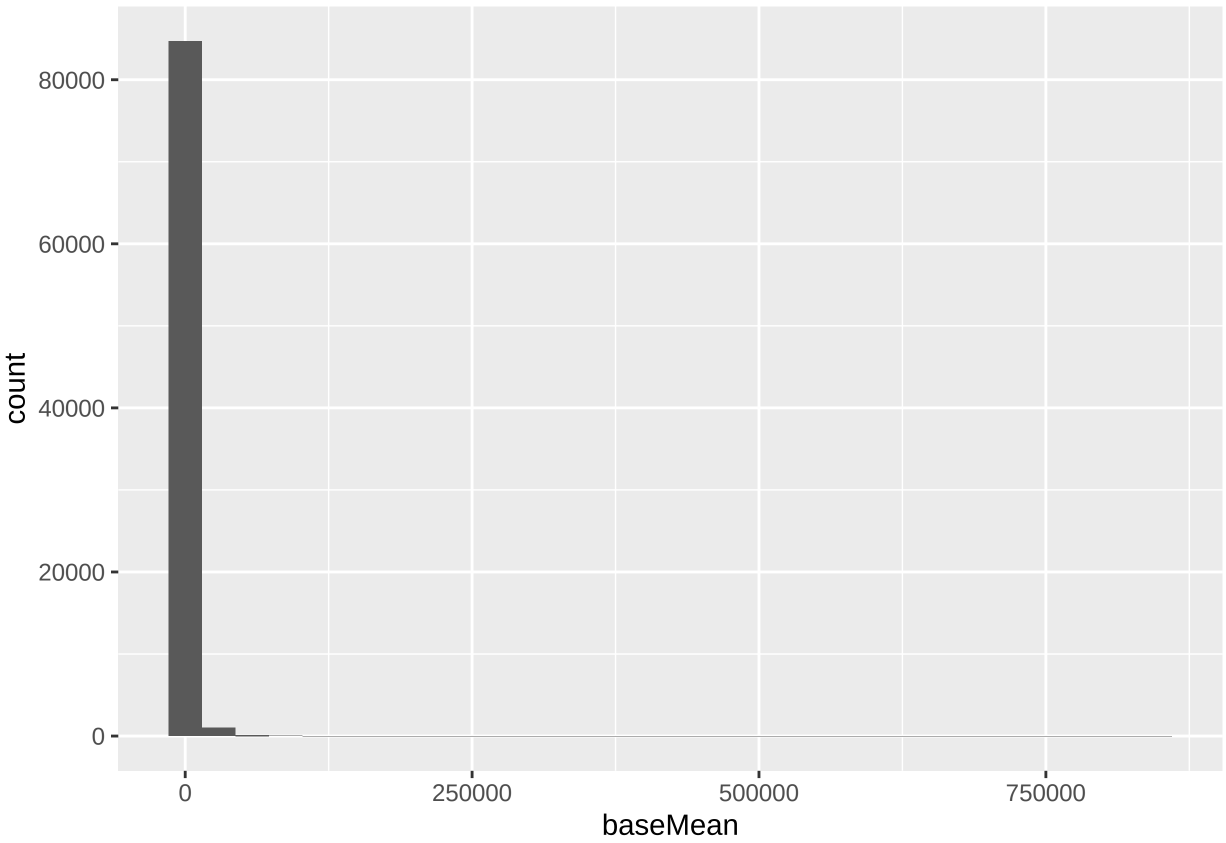
Task 1 my tasks:
* create column differentially_expressed (binary) if padj < 0.05 = 1
* create column: activated (binary) = if differentially_expressed && log2FoldChange > 0
* create column: repressed (binary) = if differentially_expressed && log2FoldChange < 0

```r
library(tidyverse)
library(ggplot2)
library(ggridges)

# LOAD DATA
df = read_tsv('znf808_degene_data_task.tsv')
df %>% summary()
```

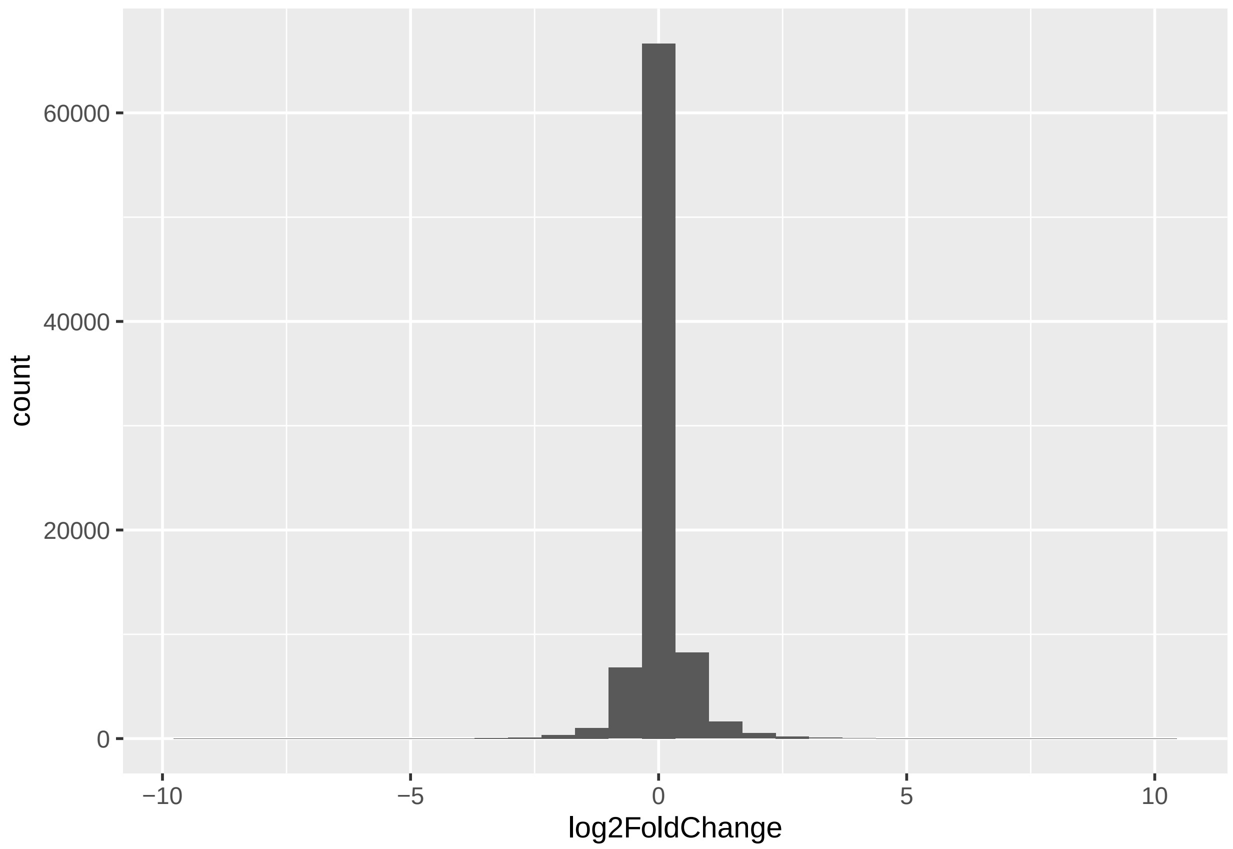
```
##      Gene              baseMean        log2FoldChange         lfcSE
##  Length:85931       Min.   :     2.3   Min.   :-9.480538   Min.   :0.03093
##  Class :character   1st Qu.:    81.3   1st Qu.:-0.118749   1st Qu.:0.09302
##  Mode  :character   Median :   508.6   Median :-0.002058   Median :0.13852
##                     Mean   :  1667.6   Mean   : 0.021276   Mean   :0.25058
##                     3rd Qu.:  1484.2   3rd Qu.: 0.132858   3rd Qu.:0.30547
##                     Max.   :845517.6   Max.   :10.076077   Max.   :5.26790
##
##       stat               pvalue            padj           Stage
##  Min.   :-26.43337   Min.   :0.0000   Min.   :0.0000   Length:85931
##  1st Qu.: -0.82896   1st Qu.:0.1286   1st Qu.:0.5100   Class :character
##  Median : -0.01522   Median :0.4097   Median :0.8463   Mode  :character
##  Mean   :  0.01344   Mean   :0.4296   Mean   :0.7064
##  3rd Qu.:  0.82053   3rd Qu.:0.7064   3rd Qu.:0.9745
##  Max.   : 30.56751   Max.   :1.0000   Max.   :1.0000
##
##    GeneName            chrom                TSS                strand
##  Length:85931       Length:85931       Min.   :      648   Length:85931
##  Class :character   Class :character   1st Qu.: 31067781   Class :character
##  Mode  :character   Mode  :character   Median : 58533994   Mode  :character
##                                        Mean   : 73336569
##                                        3rd Qu.:109258242
##                                        Max.   :249200434
##
##  DistanceNearestMER11
##  Min.   :     663
##  1st Qu.: 1936879
##  Median : 5092475
##  Mean   : 8889678
##  3rd Qu.:11432375
##  Max.   :67949433
##  NA’s   :1058
```

```r
# using summary() shows Gene, Stage, chrom, strand are poorly typed as chars not factors
df = df %>% mutate(
  Gene = as.factor(Gene),
  Stage = as.factor(Stage),
  chrom = as.factor(chrom),
  strand = as.factor(strand)
)

# Plot data of interest:
cols_of_interest = c("baseMean", "log2FoldChange", "padj")

for (col in cols_of_interest) {
  plot = ggplot(df, aes(x=!!sym(col))) +
    geom_histogram()
  print(plot)
}
```

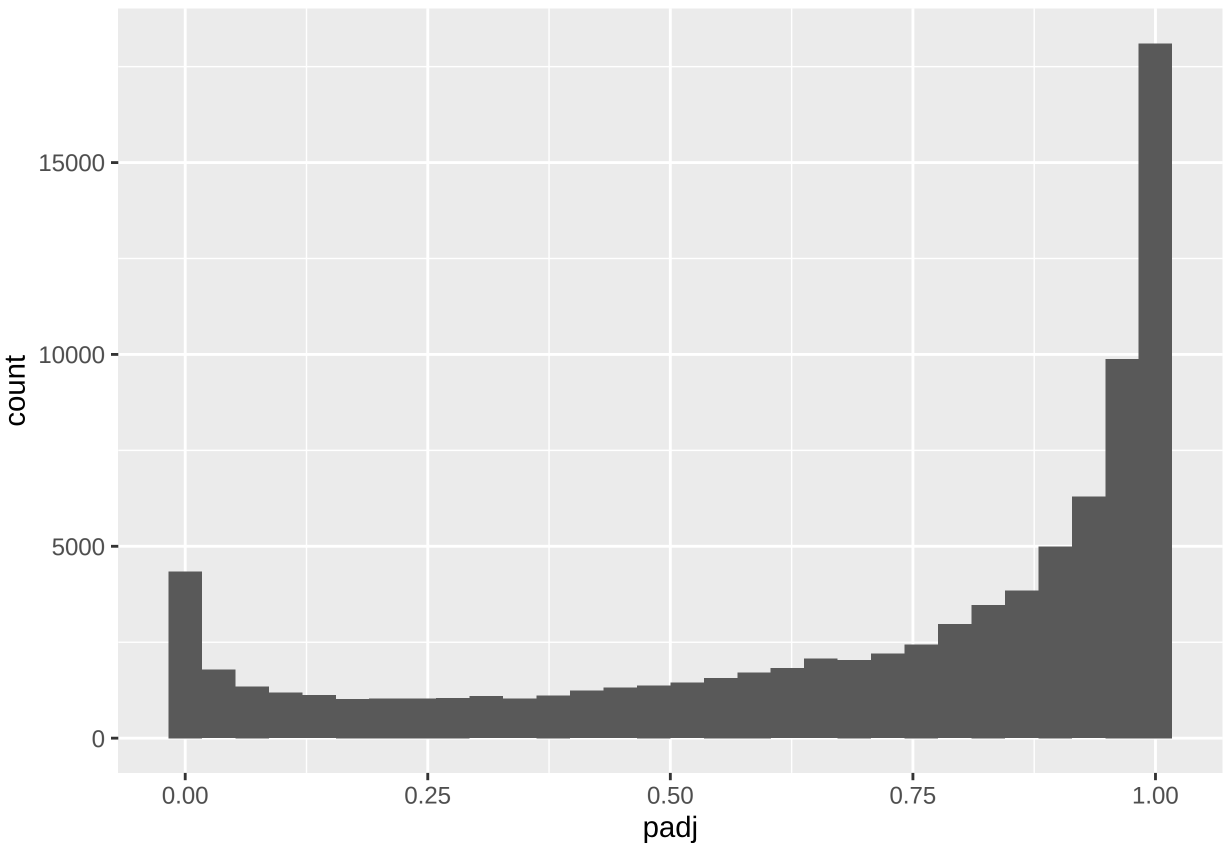
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

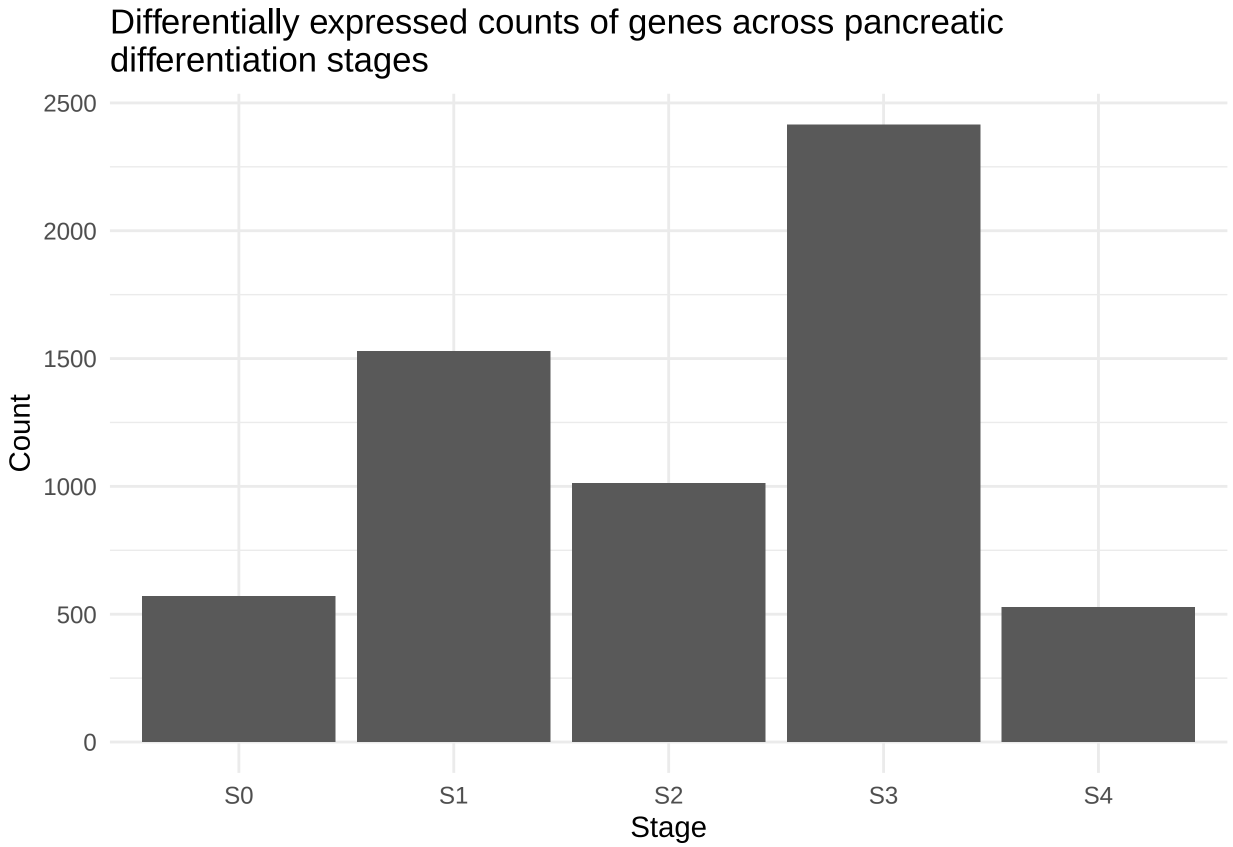


Can visually see from plots that baseMean has extreme values, but assuming it is correct since we inherited the dataset. Otherwise would truncate rows where baseMean > 500000 (only 13 rows). Alternatively can choose +- 3 or 4 StDev to truncate values, depends on if prediction models are underfit.

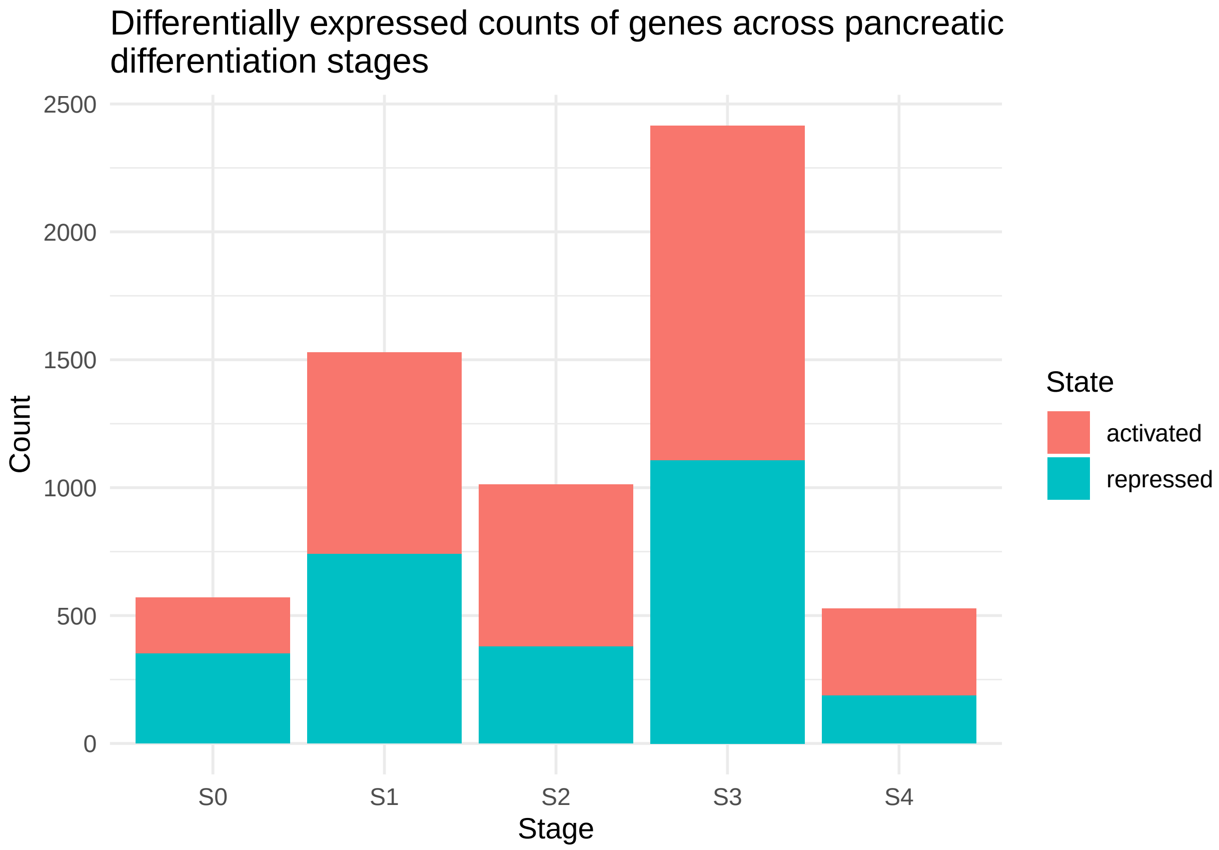
```r
# setting binary classification columns
df = df %>%
  mutate(
    differentially_expressed = ifelse(padj < 0.05, 1, 0),
    activated = ifelse(differentially_expressed == 1 & log2FoldChange > 0,1,0),
    repressed = ifelse(differentially_expressed == 1 & log2FoldChange < 0,1,0)
  )

# Visualisation:
df_summary = df %>%
  filter(differentially_expressed == 1) %>%
  group_by(Stage) %>%
  summarise(
    differentially_expressed = n(),  # Count of differentially expressed genes
    activated = sum(activated),      # Sum of activated genes
    repressed = sum(repressed)       # Sum of repressed genes
  )

print(df_summary)
```

```
## # A tibble: 5 x 4
##   Stage differentially_expressed activated repressed
##   <fct>                    <int>     <dbl>     <dbl>
## 1 S0                         570       218       352
## 2 S1                        1529       788       741
## 3 S2                        1012       633       379
## 4 S3                        2415      1308      1107
## 5 S4                         528       341       187
```

```r
visual_df = df_summary %>%
  pivot_longer(
    cols = colnames(df_summary)[2:length(colnames(df_summary))], # take all col names but first (Stage)
    names_to = "State",
    values_to = "Frequency"
  )

# Plotting plain differential expression across stages
ggplot(
  visual_df %>% filter(State == "differentially_expressed"),
  aes(x = Stage, y = Frequency)
) +
  geom_bar(stat = "identity") +
  labs(
    title = str_wrap(
      "Differentially expressed counts of genes across pancreatic differentiation stages",
      60
    ),
    y = "Count",
  ) +
  theme_minimal()
```


Differentially expressed counts of genes across pancreatic differentiation stages
2500
2000
1500
1000
500
0
Count
S0
S1
S2
S3
S4
Stage

```r
# Plotting activation vs repressed genes across stages
ggplot(
  visual_df %>% filter(State != "differentially_expressed"),
  aes(x = Stage, y = Frequency, fill=State)
) +
  geom_bar(stat = "identity") +
  labs(
    title = str_wrap(
      "Differentially expressed counts of genes across pancreatic differentiation stages",
      60
    ),
    y = "Count",
  ) +
  theme_minimal()
```



Task 2:
* DistanceNearestMER11 = distance between gene + MER11
* Predictor = Distance (continuous) –> outcome = activation / repressed

```r
# creating dataframe
task2_df = df %>%
  filter(activated == 1 | repressed == 1) %>% # only take repressed or activated
  mutate(state = ifelse(activated == 1, 1, 0)) %>% # when not activated, must be repressed based on fil
  select(DistanceNearestMER11, state, Stage)

# Iterate through stages, generate GLM logreg model, print findings.
for (stagei in unique(task2_df$Stage)){
  stage_data = task2_df %>% filter(Stage == stagei) # filter to this stage's data

  # create logreg model for binary outcome prediction
  model = glm(state ~ DistanceNearestMER11, data = stage_data, family = binomial)

  # print model statistics.
  print(summary(model))
  p_value = coef(summary(model))[2, 4]
  if (p_value < 0.05){
    print(paste(
      "PValue = ",
      p_value
    ))
    print(paste(
      "indicates Distance to Nearest MER11 IS statistically significant in Stage ",
      stagei
    ))
  } else {

    print(paste(
      "PValue = ",
      p_value
    ))
    print(paste(
      "indicates Distance to Nearest MER11 is NOT statistically significant in Stage ",
      stagei
    ))
  }
  print("################################")
  print("################################")
  print("################################")
}
```

```
##
## Call:
## glm(formula = state ~ DistanceNearestMER11, family = binomial,
##     data = stage_data)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -3.282e-01  1.098e-01  -2.988   0.0028 **
## DistanceNearestMER11 -1.962e-08  9.018e-09  -2.175   0.0296 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 744.92  on 559  degrees of freedom
## Residual deviance: 739.72  on 558  degrees of freedom
##   (10 observations deleted due to missingness)
## AIC: 743.72
##
## Number of Fisher Scoring iterations: 4
##
## [1] "PValue =  0.0296212530885538"
## [1] "indicates Distance to Nearest MER11 IS statistically significant in Stage  S0"
## [1] "#################################"
## [1] "#################################"
## [1] "#################################"
##
## Call:
## glm(formula = state ~ DistanceNearestMER11, family = binomial,
##     data = stage_data)
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.611e-01  6.669e-02   2.415  0.01573 *
## DistanceNearestMER11 -1.257e-08  4.825e-09  -2.605  0.00919 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2084.0  on 1503  degrees of freedom
## Residual deviance: 2077.1  on 1502  degrees of freedom
##   (25 observations deleted due to missingness)
## AIC: 2081.1
##
## Number of Fisher Scoring iterations: 3
##
## [1] "PValue =  0.00918745292340015"
## [1] "indicates Distance to Nearest MER11 IS statistically significant in Stage  S1"
## [1] "#################################"
## [1] "#################################"
## [1] "#################################"
##
## Call:
## glm(formula = state ~ DistanceNearestMER11, family = binomial,
##     data = stage_data)
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           6.330e-01  8.624e-02   7.340 2.13e-13 ***
## DistanceNearestMER11 -1.514e-08  6.971e-09  -2.172   0.0299 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1324.1  on 1000  degrees of freedom
## Residual deviance: 1319.4  on  999  degrees of freedom
##   (11 observations deleted due to missingness)
## AIC: 1323.4
##
## Number of Fisher Scoring iterations: 4
##
## [1] "PValue =  0.029884265533222"
## [1] "indicates Distance to Nearest MER11 IS statistically significant in Stage  S2"
## [1] "################################"
## [1] "################################"
## [1] "################################"
##
## Call:
## glm(formula = state ~ DistanceNearestMER11, family = binomial,
##     data = stage_data)
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           3.083e-01  5.341e-02   5.773 7.78e-09 ***
## DistanceNearestMER11 -1.695e-08  4.096e-09  -4.137 3.52e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3298.0  on 2390  degrees of freedom
## Residual deviance: 3280.4  on 2389  degrees of freedom
##   (24 observations deleted due to missingness)
## AIC: 3284.4
##
## Number of Fisher Scoring iterations: 4
##
## [1] "PValue =  3.52091960442641e-05"
## [1] "indicates Distance to Nearest MER11 IS statistically significant in Stage  S3"
## [1] "################################"
## [1] "################################"
## [1] "################################"
##
## Call:
## glm(formula = state ~ DistanceNearestMER11, family = binomial,
##     data = stage_data)
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           7.381e-01  1.232e-01   5.992 2.07e-09 ***
## DistanceNearestMER11 -1.828e-08  1.009e-08  -1.812   0.0699 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 679.93  on 521  degrees of freedom
## Residual deviance: 676.67  on 520  degrees of freedom
##   (6 observations deleted due to missingness)
## AIC: 680.67
##
## Number of Fisher Scoring iterations: 4
##
## [1] "PValue =  0.0699475904931409"
## [1] "indicates Distance to Nearest MER11 is NOT statistically significant in Stage  S4"
## [1] "################################"
## [1] "################################"
## [1] "################################"
```

Discussion:
MER11 is a nuclear protein involved in general DNA repair and maintenance, including DNA recombination, telomere length maintenance, and DNA double-strand break repair. (https://www.genecards.org/cgi-bin/carddisp.pl?gene=MRE11) During Stages S0-S3, the distance between gene TSS and the most proximal MER11 element was seen to significantly impact whether the gene was to be repressed or activated in differentially expressed genes relative to control vs ZNF808 KO mice. As the logstic regression coefficient was seen to associate increased distance from MER11 elements to increased risk of repression, it can be proposed that MER11 is able to mitigate the effects of agenesis caused by ZNF808 KO, and protect against repression. Conversely, no signficance was seen in pancreatic differentiation stage S4, possibly suggesting that genetic/cell profile is determined before this state, thereby mitigating the protective effects of MER11.