Using q-value to Find Significance in Gene Expression

Dave Rogers

March 16, 2021

Introduction

It has been proven that stress has an impact on our health, but how does stress effect the unborn and how does it effect the development of the newly born? Are genes related to development, passed through the umbilical chord to the fetus, over or underexpressed in mothers that have PTSD or depression or both versus mothers without diagnosed stress? Somewhere around 3% of all pregnancies result in a child with genetic abnormalities. What role does stress play in this percentage, if any? In this study we will examine the expression rates of genes in the umbilical blood. Our goal is to identify a number of those genes for further study that may point to links between a mother's stress and a child's development.

The standard for determining significance has been the use of p-values at p equals 0.1, 0.05 or 0.01 for many years. This provides an expected Type 1 error rate, but does not control those errors. In this project q-value will be used in determining significance. In this way we can set the expected false discovery rate to coincide with the data.

Methods and Analysis

Principal Components Analysis

The data to be examined are the gene expressions of 13,405 genes obtained from the umbilical cord blood samples from 149 neonates. The mothers had differing levels of stress during pregnancy, ranging from PTSD (n = 20), depression (n = 31), PTSD with comorbid depression (n = 13), trauma exposed as part of the control group (n = 23 and the control group (n = 62). The gene expressions have been normalized [1].

After organizing the data into a data frame with 149 observations of 13,405 genes, each observation had it's stress category added to the data frame. From here principal components analysis was performed to try and remove any stratification. This was only partially successful. The table on the next page shows the individual variance and the cumulative variance for the first ten principal components. It shows that only 50% of the total variance is captured by the first 6 pc's.

Var	Prop	Cum.Prop
339	17.956	17.956
226	11.959	29.915
138	7.320	37.235
94	4.963	42.199
84	4.469	46.667
72	3.783	50.450
46	2.431	52.881
44	2.310	55.191
34	1.815	57.006
33	1.753	58.759

Finding Significance

It was initially decided that principal components would not be used in the analysis and linear regression models were run using the stress category as the only predictor of gene expression. P-values for category as a predictor for each gene were calculated and fed to the qvalue function from the qvalue package for R [2]. When attempts were made to compute q-value, the distribution of p-values was found to be too uniform (Figure 1). This led to all q-values being approximately the same and no genes being found to be significant. See Figure 2. for the qplot.

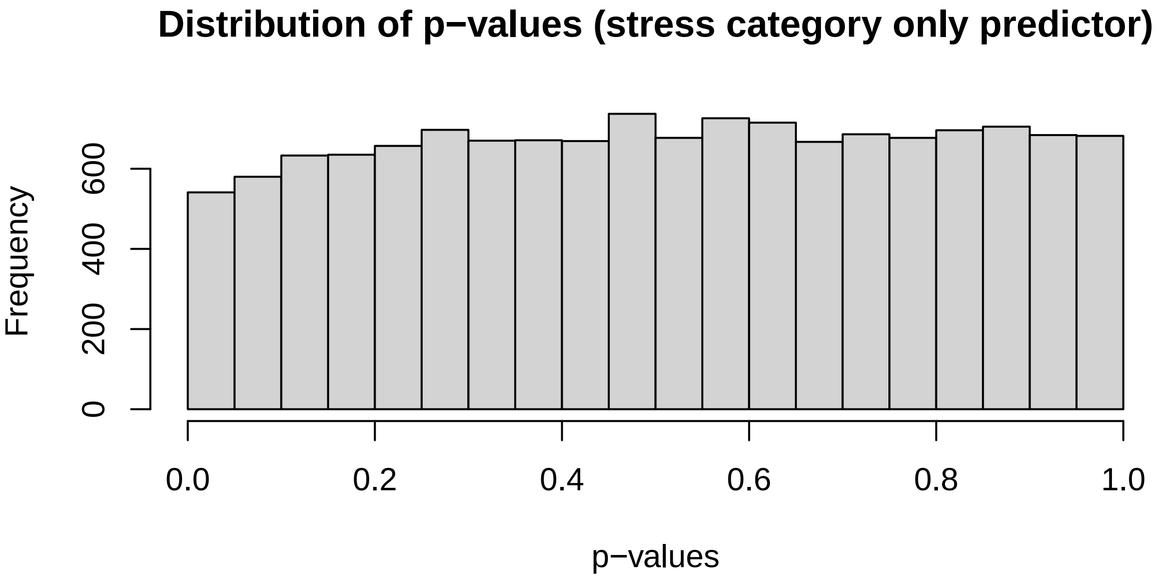

Figure 1.

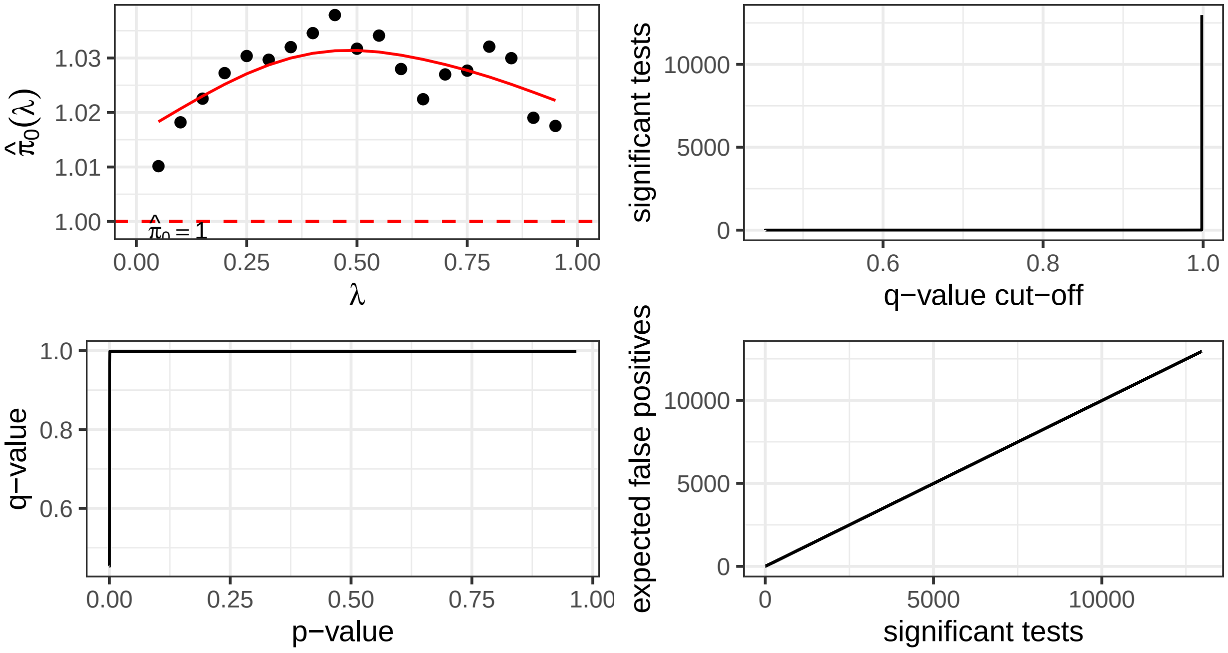

Figure 2. Qplot w/ category Only Predictor

This left the options of using only p-value to determine significance or to add principal components to the model. Using p-values at this point would result in 541 being considered significant and an unknown portion of these being false positives. With 671 expected false positives (13,405 * 0.05) it was decided that the first 6 principal component which account for 50% of the cumulative variation in the data would be used.

This led to a markedly different distribution of p-values from the model (Figure 3) and when we look at the qplot (Figure 4) we see that there is now significance in the q-values.

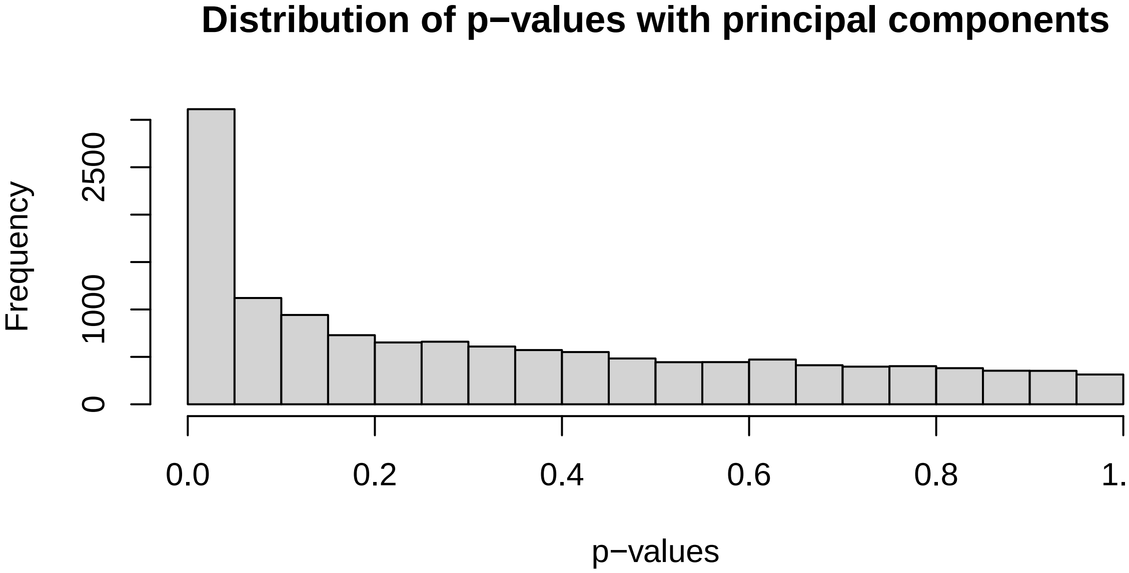

Figure 3.

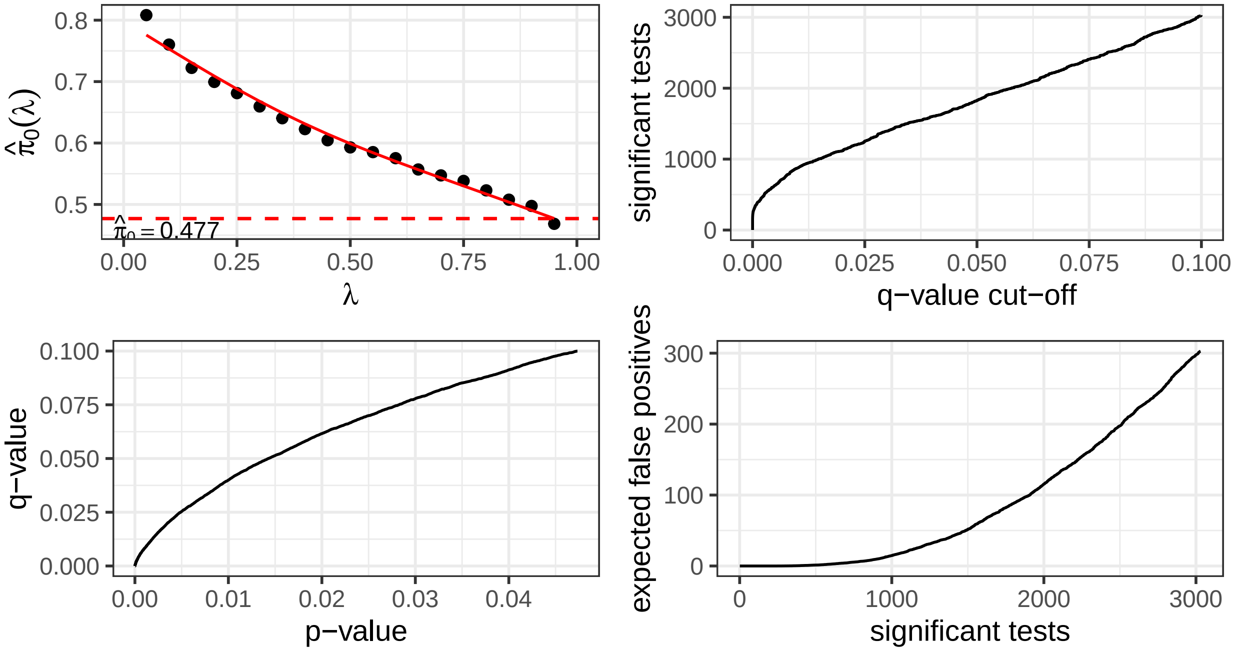

Figure 4. Qplot w/ category Only Predictor

Looking at the upper right pain in Figure 4. we see that the number of significant tests rises rapidly early before becoming approximately linear after q equals approximately 0.01. Looking at the plots in the bottom row we notice nothing of significance in these curves. For these reasons we will choose a qvalue of 0.01 to determine significance in this data. This results in 872 significant genes with an estimated number of false positives under 25.

Once differently expressed genes were identified, we needed to determine whether they were overexpressed or underexpressed when comparing one of the control categories to stress categories. A data frame was created that included each gene, the mean gene expression for each of the five categories and comparisons of the two control categories versus the three stress categories.

Gene Names

The data was encoded using illuminaHumanv4.db giving the gene names to this point no meaning. The next step was to convert the illumnina ID to one more easily recognizable. The illuminaHumanv4SYMBOL from the illuminaHumanv4.db package for R [3] was used to convert the illumina ID's to abbreviations consistent with the National Center for Biotechnical Information.

After conversion, it was found that 146 of the illumnia ID's did not convert to a known gene name. For this reason they were removed from the data at this point. The information has been kept though for possible further study. It was also found that at least 1 gene has multiple illumina ID's as there were 623 remaining distinct gene names, but 726 illumina ID's. This is possibly due to different transcripts of the same gene. For this reason the duplicates were left in the data.

The table below shows the first six rows of data after converting the gene names and removing unidentified genes. Please note that "over" indicates that the gene is overexpressed in samples from mothers with a stressed condition and "under' indicates that the gene is underexpressed in those samples. As an example, the first row indicates that the gene SLC38A2 is over expressed in all of the diagnosed stress categories versus the control group and the control with traumatic event group. This particular gene is a protein encoding gene and is associated with Adult Syndrome and Persistent Fetal Circulation Syndrome. This gene functions as a sodium-dependent amino acid transporter [4]. The full .csv file can be found at https://github.com/daverogers68/ST592

gene	CvDep	CvPTSD	CvPTSDDep	CTEvDep	CTEvPTSD	CTEvPTSDDEP
SLC38A2	Over	Over	Over	Over	Over	Over
BASP1	Under	Under	Under	Under	Under	Under
SUCLG2	Over	Under	Over	Over	Over	Over
PIK3AP1	Over	Under	Over	Over	Over	Over
RPS27	Over	Over	Over	Over	Under	Under
RBIS	Over	Over	Over	Over	Under	Under

Summary

This project has identified 726 genes (of these up to 103 may be duplicates due to different transcripts of the same gene) for possible further study in childhood development where a mother is diagnosed as having depression, PTSD or PTSD with depression during pregnancy. Five different stress categories were identified with two control groups one of which was a control group that had experienced a traumatic event but had not been diagnosed with one of the three diagnoses in the study. Linear regression models were utilized with stress category and six principal components used as predictors of gene expression for each gene individually. A qvalue cut off of 0.01 was chosen based on the resulting p-value for stress category from the Analysis of Variance of each model. Initially of 800 genes were found significant, but after identifying genes that could not be given a recognized name and those that were duplicated most likely due to different transcripts of the same gene we were left with 726. Of these 726 less than 25 are thought to be false positives.

Limitations

This project had the express goal of using q-value rather than p-value in determining significance in gene expression between the differing categories of maternal stress. Using PCA was not a primary concern, but when q-values could not be calculated without including PCA as a predictor, PCA was included. A differing number of principal components would have certainly provided different results. Using only two principal components would have yielded a q cutoff of 0.05 and resulted in approximately 225 genes being considered significant. That being said, two principal components would have accounted for less than 30% of the total variance in the data.

Originally the project was intended to go one step further and identify which category of the 5 was significantly different from which of the other categories. It was decided that due to time constraints this feature would not be added to the project. This is an area for further study at a later date.

It is not known if any of the genes removed from the final data frame due to a lack of gene name should be considered further. It is beyond the scope of this project to identify those genes that could not be identified by their illumina ID.

This project was only intended to provide a list of genes that were over or underexpressed between the control groups and the stressed groups and might be of interest in further study as to the effect on childhood development. It was beyond the scope of this project to look at each gene found and identify its role in childhood development.

References

1. Breen, M. S., Wingo, A. P., Koen, N., Donald, K. A., Nicol, M., Zar, H. J., Ressler, K. J., Buxbaum, J. D., & Stein, D. J. (2018). Gene expression in cord blood links genetic risk for neurodevelopmental disorders with maternal psychological distress and adverse childhood outcomes. *Brain, behavior, and immunity, 73*, 320–330. https://doi.org/10.1016/j.bbi.2018.05.016

2. Storey JD, Bass AJ, Dabney A, Robinson D (2020). *qvalue: Q-value estimation for false discovery rate control.* R package version 2.22.0, http://github.com/jdstorey/qvalue.

3. Dunning M, Lynch A, Eldridge M (2015). *illuminaHumanv4.db: Illumina HumanHT12v4 annotation data (chip illuminaHumanv4).* R package version 1.26.0.

4. Database, G. C. H. G. *SLC38A2 gene (PROTEIN CODING).* https://www.genecards.org/cgi-bin/carddisp.pl?gene=SLC38A2.

Appendix A

R code

```
knitr::opts_chunk$set(echo = FALSE, warning = F, message = F)
#libraries that will be needed
library(tidyverse)
library(kableExtra)
library(qvalue)
library(illuminaHumanv4.db)
#read data
fp_dat <- read.table("GSE114852_NormExprs.txt", header = TRUE)
#reorder data to natural numeric order by samples
fp_dat3 <- fp_dat[, c(1:3, 124:125, 146:147, 168:169, 190:191, 212:213, 234:235, 256:257
                      4:5, 26:27, 48:49, 70:71, 92:93, 114:123, 126:145, 148:167, 170:18
                      192:211, 214:233, 236:255, 258:277, 280:299, 6:25, 28:47, 50:69, 7
                      94:113)]
#list to be  used when removing p-values
c_list <- seq(2, 298, 2)
#remove p-values
fp_dat3 <- fp_dat3[, c(1,c_list)]
#collect row names to be used as column names
n <- fp_dat3$Row.names
#transpose data to put samples in rows and genes in columns
fp_dat4 <- as.data.frame(t(fp_dat3[,-1]))
#add back column names
colnames(fp_dat4) <- n
#convert row names to sample variable
fp_dat4$sample <- row.names(fp_dat4)
#define the 5 categories
```

```
cats <- c(rep("Depression", 31), rep("PTSDDep", 13), rep("PTSD", 20), rep("ControlTE", 2
          rep("Control", 62))
#add categories to data
fp_dat4$cat <- cats
#remove the row names
row.names(fp_dat4) <- NULL
#reorder to put sample and cat in front of data for personal ease of viewing
fp_dat4 <- fp_dat4[,c(13406, 13407, 1:13405)]
#create principal compnents
fp_fit <- prcomp(fp_dat4[, 3:13407])

#create principal components table
variance.table <- data.frame(Var = round(fp_fit$sdev^2),
                             Prop = fp_fit$sdev^2/sum(fp_fit$sdev^2)*100,
                             Cum.Prop = cumsum(fp_fit$sdev^2/sum(fp_fit$sdev^2)*100))
#view principal tomponents table
kable(round(head(variance.table, n = 10), digits = 3))
#data frame with only gene expressions
fp_dat5 <- fp_dat4[, 3:13407]
#stor p-values from model
fp_pvalues <- rep(0,  ncol(fp_dat5))
#number of itierations in for loop
n <- ncol(fp_dat5)

#linear regression model with only category as predictor.
for(i in 1:n) {
  fp_m1 <- lm(fp_dat5[,i] ~ factor(fp_dat4$cat))
  fp_anova_m1 <- anova(fp_m1)
  fp_pvalues[i] <- fp_anova_m1$`Pr(>F)`[1]
```

```r
}
alpha <- 0.05


fp_qobj <- qvalue(fp_pvalues)
hist(fp_pvalues, main = "Distribution of p-values (stress category only predictor)",
     xlab = "p-values")
title(sub = "Figure 1.", adj = 0, family = "serif", cex = 0.95)
plot(fp_qobj)
#add first 6 principal components to table
fp_dat4$pc1 <- fp_fit$x[, 1]
fp_dat4$pc2 <- fp_fit$x[, 2]
fp_dat4$pc3 <- fp_fit$x[, 3]
fp_dat4$pc4 <- fp_fit$x[, 4]
fp_dat4$pc5 <- fp_fit$x[, 5]
fp_dat4$pc6 <- fp_fit$x[, 6]
#reset
fp_pvalues <- rep(0,  ncol(fp_dat5))
#reset
n <- ncol(fp_dat5)
for(i in 1:n) {
  fp_m1 <- lm(fp_dat5[,i] ~ factor(cat) + pc1 + pc2 + pc3 + pc4 + pc5 + pc6, data = fp_d
  fp_anova_m1 <- anova(fp_m1)
  fp_pvalues[i] <- fp_anova_m1$`Pr(>F)`[1]
}
#create q object
fp_qobj2 <- qvalue(fp_pvalues)
#histogram of p-values
hist(fp_pvalues, main = "Distribution of p-values with principal components",
     xlab = "p-values")
```

```
title(sub = "Figure 3.", adj = 0, family = "serif", cex = 0.95)
#plot q object
plot(fp_qobj2)
fp_names <- names(fp_dat5)[which(fp_qobj2$qvalues <= 0.01)]
fp_index <- which(fp_qobj2$qvalues <= 0.01)
fp_qvals <- fp_qobj2$qvalues[fp_index]
fp_pvalues2 <- fp_qobj2$pvalues[fp_index]

fp_dat6 <- data.frame("IllumID" = fp_names,
                      "qval" = fp_qvals,
                      "pval" = fp_pvalues2)
#five empty lists to hold the mean per category for each gene
fp_dep <- rep(0,length(fp_names))
fp_con <- rep(0,length(fp_names))
fp_cte <- rep(0,length(fp_names))
fp_pts <- rep(0,length(fp_names))
fp_ptd <- rep(0,length(fp_names))

n <- length(fp_names)
#create a dataframe to hold only the sig genes and the categories
fp_dat_red <- fp_dat4[,fp_names]
fp_dat_red <- cbind(fp_dat4$cat, fp_dat_red)
colnames(fp_dat_red) <- c('cat', fp_names)

# find each category mean for each signiifcant gene
for(i in 1:n) {
  mean_column <- data.frame("cat" = fp_dat_red$cat, "value" = fp_dat_red[, i+1])
  fp_dep[i] <- mean(mean_column$value[which(mean_column$cat == "Depression")])
  fp_con[i] <- mean(mean_column$value[which(mean_column$cat == "Control")])
  fp_cte[i] <- mean(mean_column$value[which(mean_column$cat == "ControlTE")])
  fp_pts[i] <- mean(mean_column$value[which(mean_column$cat == "PTSD")])
  fp_ptd[i] <- mean(mean_column$value[which(mean_column$cat == "PTSDDep")])
}
#add category means to data frame
fp_dat6 <- fp_dat6 %>% mutate("Control_mean" = fp_con,
                              "ControlTE_mean" = fp_cte,
                              "Depress_mean" = fp_dep,
                              "PTSD_mean" = fp_pts,
                              "PTSDDep_mean" = fp_ptd)
fp_dat6 <- fp_dat6 %>%
  rowwise() %>%
  mutate("CvDep" = ifelse(Control_mean > Depress_mean, "Under", "Over"),
         "CvPTSD" = ifelse(Control_mean > PTSD_mean, "Under", "Over"),
         "CvPTSDDep" = ifelse(Control_mean > PTSDDep_mean, "Under", "Over"),
         "CTEvDep" = ifelse(ControlTE_mean > Depress_mean, "Under", "Over"),
         "CTEvPTSD" = ifelse(ControlTE_mean > PTSD_mean, "Under", "Over"),
         "CTEvPTSDDEP" = ifelse(ControlTE_mean > PTSDDep_mean, "Under", "Over"))
fp_gene <- unlist(mget(x = fp_names,envir = illuminaHumanv4SYMBOL))
names(fp_gene) <- NULL
fp_dat6$gene <- fp_gene


fp_dat7 <- fp_dat6 %>%
  filter(!is.na(gene))
fp_dat7 <-fp_dat7[,c(1,15,2:14)]

fp_dat8 <- fp_dat7[c(2,10:15)]
fp_dat8[1:6,]
```

```r
write.csv(fp_dat8, "final_project.csv")
```