Statistical Methods in Genomics - BIOS 7649 Homework 1

Dominic Adducci

Question 1

Part A

Is it possible to avoid the effects of the experimental factors?

No, it is not possible to avoid the effects of the experimental factors. Different confounding factors including time-of-day and stress from handling can have an effect on the expression of genes. Additionally, even between mice there can be large variation, and because most experiments are based on single samples from each subject (in this case mice) this variable is often impossible to correct for.

Part B

If not, how should we perform the modeling and interpret the results in light of these effects?

ANOVA can be used to control for noise which results from confounding factors, but this leads to the question of whether the differentially expressed genes selected by ANOVA are truly the result of experimental factors or confounders. The authors suggested that the one-gene-at-a-time method may not sufficiently capture expression differences, and a more systematic approach which analyzes groups of genes would be better. Interpretation should be done carefully, as differentially expressed genes may be the result of uncontrollable confounders, such as time of day, handling stress, and variability between mice.

Part C

Suppose that the microarray experiments were performed over several batches. What happens if all the samples relevant to any of these factors (variables of interest or the other experimental factors) are entirely contained in on of the experimental batches? How can this be avoided?

In addition to experimental confounders mentioned above there are also effects from running different batches. If all samples relevent to a specific factor are contained in one batch it would be impossible to try and control for the differences between batches. This can be avoided by spreading out these factors between batches, allowing observations into differences.

Question 2

Background:

- A two-fold difference between the treated and untreated mice would correspond to a $\delta = 1$, assuming a base 2 logarithm is used.
- Recommended α and σ values from *Design of DNA Microarray Experiment* will be used, which corresponds to $\alpha = 0.001$ and $\sigma = 0.5$ (when using base 2 logarithms).
- Sample size equation is as follows:

$$n = \frac{4(z_{\alpha/2} + z_{\beta})^2}{(\delta/\sigma)^2}$$

Where $\alpha = 0.001$ fixes $z_{\alpha/2} = -3.29$.

These steps are performed in the *pwr.t.test()* from the *pwr* package.

Table 1: Sample Size and Cost of Power Levels

Power	Sample Size	Cost ($)
0.80	24	24000
0.81	24	24000
0.82	24	24000
0.83	24	24000
0.84	24	24000
0.85	26	26000
0.86	26	26000
0.87	26	26000
0.88	26	26000
0.89	28	28000
0.90	28	28000
0.91	28	28000
0.92	28	28000
0.93	30	30000
0.94	30	30000
0.95	30	30000

Table 1 illustrates the relationship between increasing power and sample size and cost. Assumptions include that kidney gene expressions between the treated and non-treated mice will be compared by level of expression of each gene. The value $\delta = 1$ represents the difference in means for each gene, where in this study a base 2 logarithm is used meaning that $\delta = 1$ corresponds to a twofold difference in gene expression in the kidneys of treated and non-treated mice. The significance level, α, is set at $\alpha = 0.001$ to limit the number of false discoveries. Power level varies from 0.80 to 0.95. The σ value can be approximated as 0.5 for the Affymetrix GeneChips, and the d parameter is δ/σ.

The null and alternative hypothesis for each gene are as follows:

$$H_0 : \delta = 0$$

$$H_1 : \delta \neq 0$$

The null hypothesis states that there is no meaningful difference between the mean expression of a particular gene, and the alternative is that there is a significant difference between the mean expression of the gene is both groups. A two sample t-test was used as there are two different groups, treated and un-treated, and a two sided test was applied as expression for a particular gene may potentially be higher in either group. 20,000 probe sets corresponds to 20,000 genes, and with a significance level of 0.001 this mean an expected number of false positives is 20. R version 4.3.1 was used in this analysis, and sample size calculations were performed using the *pwr.t.test* function from the *pwr* package in R.

Because sample size cannot be a fraction there are multiple power levels which overlap after rounding up. When considering the relationship between power, cost, and sample size the largest power for each sample size should be considered.

Question 3

Part A

What is the sample size needed based on $\alpha = 0.001$, fold change of 2 ($\delta = 1$ in log_2) and standard deviation of 0.5 to achieve power of at least 0.8 or 0.95? Use *pwr.t.test()* in the *pwr* package. Summarize your findings. Note: The d option in *pwr.t.test()* is δ/sd.

Table 2: Sample size Calculations Using pwr.t.test

Power	Sample Size
0.80	24
0.95	30

For $\delta = 1$, $\sigma = 0.5$, $\alpha = 0.001$, and a two-sample two-sided t-test the sample size needed for a power of 0.80 is 24, and for a power of 0.95 the sample size is 30. This means there are only 6 more subjects required to raise the power from 0.80 to 0.95.

Part B

As in part a), determine the sample size needed, but with a FDR of 0.05 instead. Use *power.t.test.FDR()* in the *ssize* package. Explain π_0 and summarize your findings.

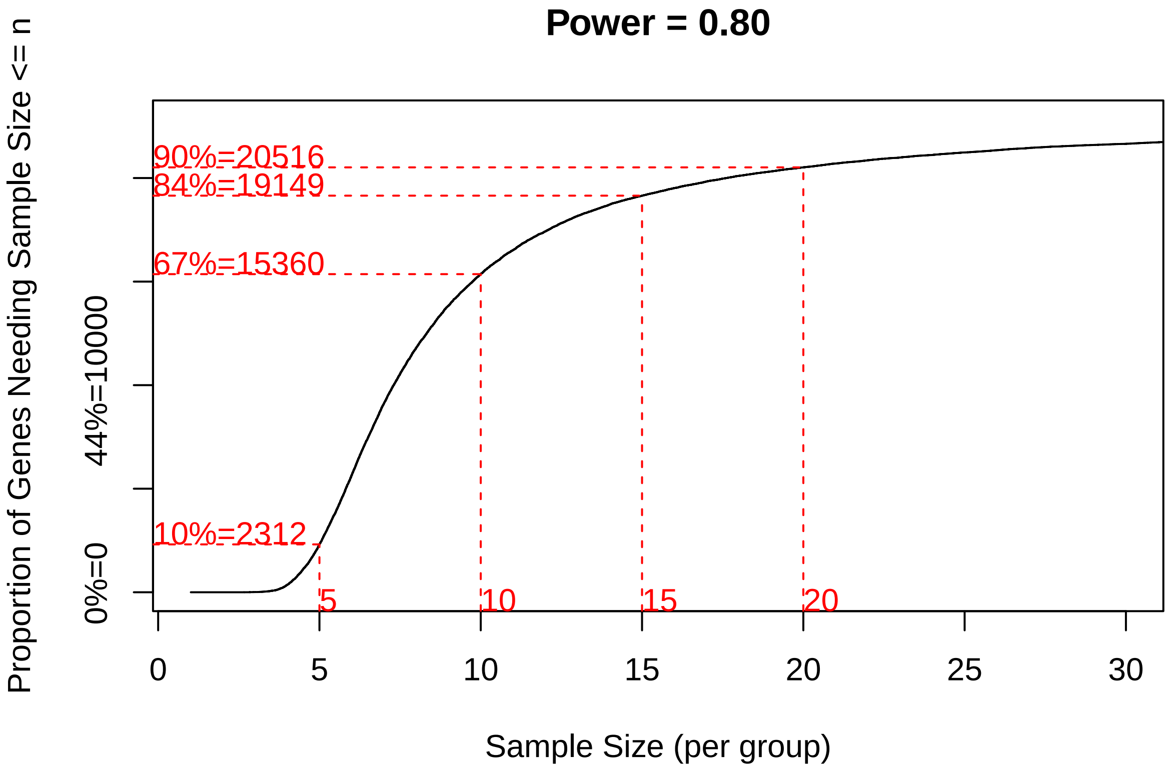
Table 3: Sample Size Calculation Using power.t.test.FDR

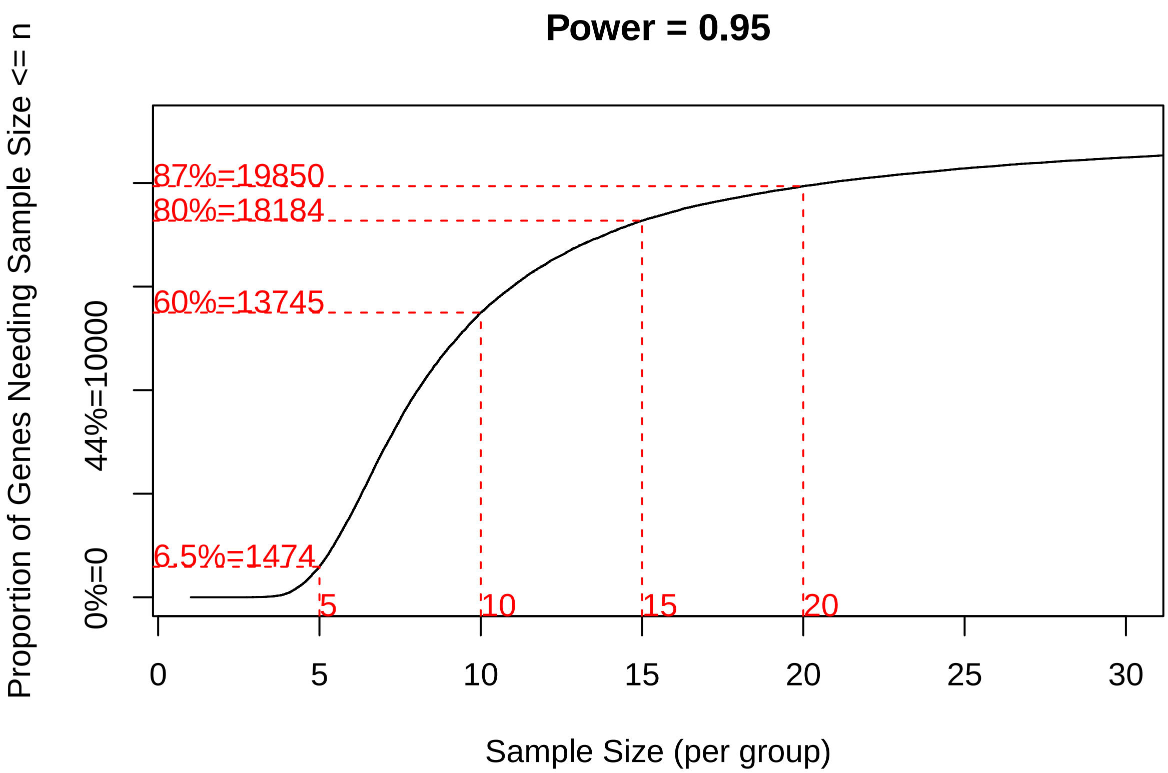
Power	Sample Size
0.80	16
0.95	22

For $\delta = 1$, $\sigma = 0.5$, $FDR = 0.05$, $\pi_0 = 0.80$, and a two-sample two-sided t-test the sample size needed for a power of 0.80 is 16, and for a power of 0.95 the sample size required is 22. This means 6 more subjects are required to raise the power from 0.80 to 0.95. π_0 is the proportion of true null hypothesis. In other words, this is the proportion of genes which are not differentially expressed.

Part C

The attached file **sdvalues.txt** contains pooled standard deviations (for the two groups) for the example data set. Read this file and plot the density (or histogram) of the standard deviations. Use *ssize()* and *ssize.plot()* in the *ssize* package to examine the sample size based on these standard deviations. Use *sig.level*, *delta* and *power* as in part a). What do you conclude from the plots?

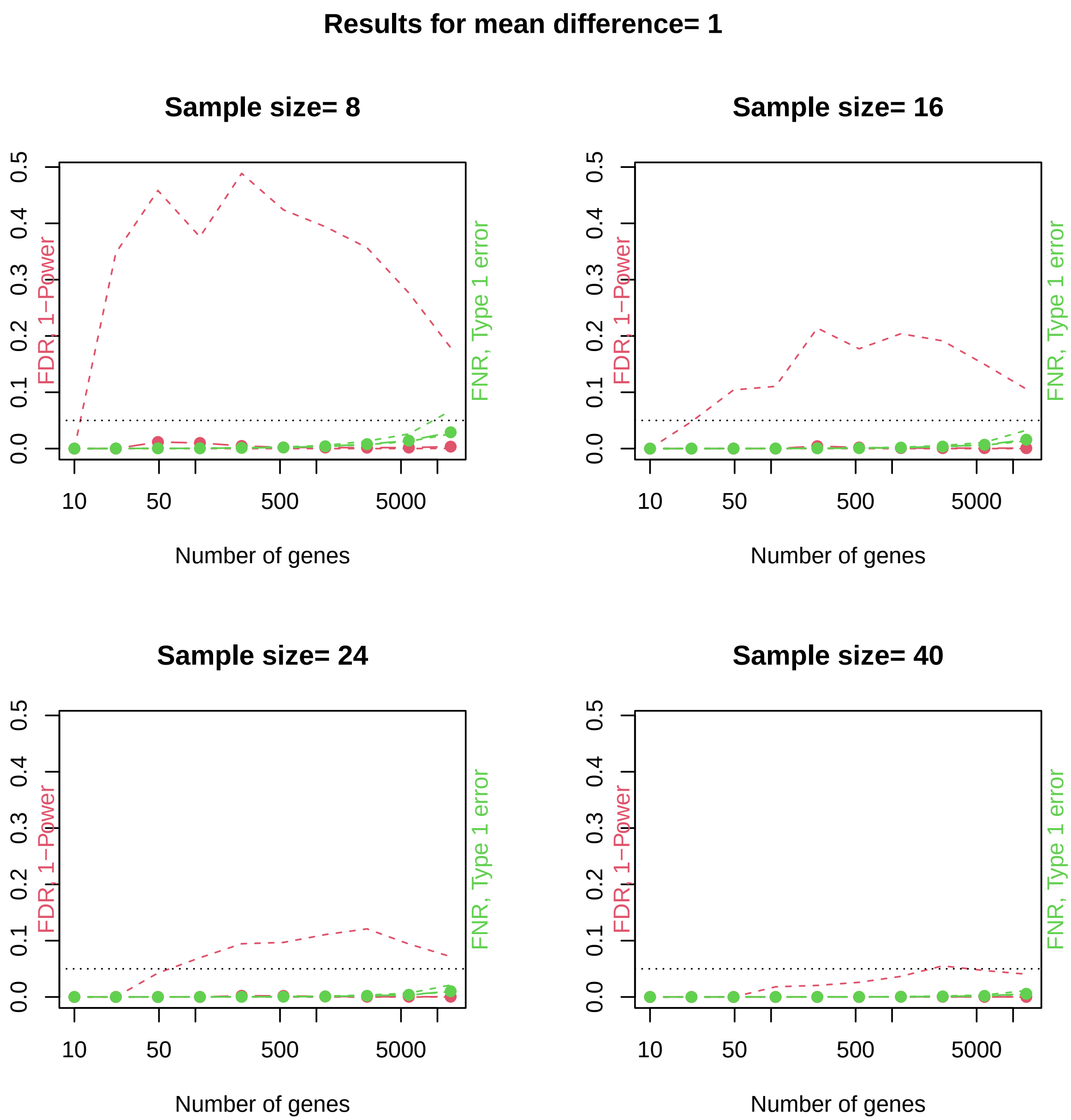

Power = 0.80
90%=20516
84%=19149
67%=15360
10%=2312
5
10
15
20
0%=0
44%=10000
Proportion of Genes Needing Sample Size <= n
0
5
10
15
20
25
30
Sample Size (per group)



When the power increases from 0.80 to 0.95 the proportion of genes needing a sample size $\leq$ n reduces. In other words a higher power test results in a more sensitive test.

Part D

The attached file **arraydata.txt** contains the data for each sample. Use **samr.assess.samplesize()** in the **samr** package, which implements the method from journal club (Tibshirani,2006), to examine the sample size based on these data. What plots are displayed with **samr.assess.samplesize.plot()**? What do you conclude from the plots?




```
## NULL
```

There are 4 different plots displayed with different sample sizes illustrated in each plot. The solid red line represents the False Discovery Rate (FDR) and the solid green line represents the False Negative Rate (FNR). The dotted lines represent the 10 and 90th percentiles for the FDR and FNR. From these plots the FNR stays relatively low across all sample sizes, with the 90th percentile extending slightly above 0.05 after reaching 5000 genes. The FDR 90th percentile varies significantly as sample size changes, with increasing sample size reducing the range of the 10 and 90th percentiles. Predictably, larger sample size has better properties.

Part E

Describe the different methods and then compare and contrast the sample size calculations in parts a)-d).

The method in part A is based on the **pwr.t.test** function, which requires specification of the significance level in addition to power, d (δ/sd), and power. This is different from the method in part B, where instead of specifying the significance level the FDR level is specified. Method B results in smaller sample size estimates for the specified power.

Method C illustrates the effects of power, sample size, and number of genes. The results from method C are relatively close to those in method B. Method D plots the relationship between FDR/FNR versus the number of genes for different sample sizes. This allows the statistician to weigh the effects of increasing sample size on the FDR and FNR depending on the number of genes of interest.

Each method provides different interpretations, and the best method may depend on the parameters for a specific problem.

CODE

```r
library(tidyverse)
library(pwr)
library(kableExtra)
library(ssize)
library(impute)
library(samr)


### START QUESTION 2 CODE ###

# Making a function to calculate sample size and cost for specific power level.
size_cost <- function(power_level){
  # power_level: Refers to the desired power level (will span 0.80 to 0.95).

  power_output <- pwr.t.test(n = NULL,d = 1/0.5,sig.level = 0.001,
                             power = power_level,type = "two.sample",
                             alternative = "two.sided")

  # Calculating the total sample size. n from output is the number in each
  # group, meaning we have to multiply by 2 for total sample size.
  sample_size <- ceiling(power_output$n) * 2

  cost <- 1000 * sample_size

  return(c(sample_size,cost))

}


power_vector <- seq(0.80,0.95,by = 0.01) # Vector of different power levels.

# Calculating sample size and cost for various power levels.
cost_size_matrix <- t(sapply(power_vector, function(x) size_cost(x)))

# Combining vector of power levels with cost and sample size results.
cost_size_power <- data.frame(cbind(power_vector,cost_size_matrix))
colnames(cost_size_power) <- c("Power","Sample Size","Cost ($)")

cost_size_power_tbl <- kbl(cost_size_power,
                           caption = "Sample Size and Cost of Power Levels",
                           booktabs = TRUE, align = "lcc") %>%
  kable_styling(latex_options = "HOLD_position")

cost_size_power_tbl

### FINISH QUESTION 2 CODE ###


### START QUESTION 3 CODE ###

# Loading in data for question 3.
```

```
# Laptop location:"C:/Biostatistics Masters Program/pring 2024/SMG-BIOS7659/Homework 1/arraydata.txt"
# Desktop location:"C:/Users/domin/Documents/Biostatistics Masters Program/Spring 2024/SMG-BIOS7659/Hom
arraydata <- read.table("C:/Users/domin/Documents/Biostatistics Masters Program/Spring 2024/SMG-BIOS765


# Laptop location:"C:/Biostatistics Masters Program/Spring 2024/SMG-BIOS7659/Homework 1/sdvalues.txt"
# Desktop location:"C:/Users/domin/Documents/Biostatistics Masters Program/Spring 2024/SMG-BIOS7659/Hom
sdvalues <- read.table("C:/Users/domin/Documents/Biostatistics Masters Program/Spring 2024/SMG-BIOS7659


## START QUESTION 3 PART A CODE ##

# Calculating sample size for power = 0.80.
lower_power_a <- pwr.t.test(n = NULL,d = 1/0.5,sig.level = 0.001,
                            power = 0.8,type = "two.sample",
                            alternative = "two.sided")

# Calculating sample size for power = 0.95.
upper_power_a <- pwr.t.test(n = NULL,d = 1/0.5,sig.level = 0.001,
                            power = 0.95,type = "two.sample",
                            alternative = "two.sided")

# Extracting sample size for each group, rounding up, and multiplying by 2
# in order to get the full sample size.
lower_size_a <- ceiling(lower_power_a$n) * 2
upper_size_a <- ceiling(upper_power_a$n) * 2

pa_size_df <- data.frame(power = c(0.80,0.95),
                         samplesize = c(lower_size_a,upper_size_a))

pa_size_tbl <- kbl(pa_size_df,
                   caption = "Sample size Calculations Using pwr.t.test",
                   col.names = c("Power","Sample Size"),
                   booktabs = TRUE, align = "cc") %>%
  kable_styling(latex_options = "HOLD_position")

pa_size_tbl

## FINISH QUESTION 3 PART A CODE ##


## START QUESTION 3 PART B CODE ##

# Calculating sample size using specified FDR level, power = 0.80.
lower_power_b <- power.t.test.FDR(sd = 0.5,n = NULL,delta = 1,
                                  FDR.level = 0.05,pi0 = 0.80,power = 0.80,
                                  type = "two.sample",alternative = "two.sided")

# Calculating sample size using specified FDR level, power = 0.95.
upper_power_b <- power.t.test.FDR(sd = 0.5,n = NULL,delta = 1,
                                  FDR.level = 0.05,pi0 = 0.80,power = 0.95,
                                  type = "two.sample",alternative = "two.sided")
```

```r
# Extracting sample size for each group, rounding up, and multiplying by 2
# in order to get the full sample size.
lower_size_b <- ceiling(lower_power_b$n) * 2
upper_size_b <- ceiling(upper_power_b$n) * 2

pb_size_df <- data.frame(power = c(0.80,0.95),
                         samplesize = c(lower_size_b,upper_size_b))

pb_size_tbl <- kbl(pb_size_df,
                   caption = "Sample Size Calculation Using power.t.test.FDR",
                   col.names = c("Power","Sample Size"),
                   booktabs = TRUE, align = "cc") %>%
  kable_styling(latex_options = "HOLD_position")

pb_size_tbl

## FINISH QUESTION 3 PART B CODE ##


## START QUESTION 3 PART C CODE ##

# Using ssize function to make a vector of powers. From the prompt n = 4 per
# group, alpha = 0.001, delta = 1, and sd comes from the provided file.
ssize_vec_lower <- ssize(sdvalues$V2,delta = 1,sig.level = 0.001,
                         power = 0.80,alpha.correct = "Bonferonni")

ssize_vec_upper <- ssize(sdvalues$V2,delta = 1,sig.level = 0.001,
                         power = 0.95,alpha.correct = "Bonferonni")


ssize.plot(ssize_vec_lower,xlim = c(1,30),marks = c(5,10,15,20),
           main = "Power = 0.80")
ssize.plot(ssize_vec_upper,xlim = c(1,30),marks = c(5,10,15,20),
           main = "Power = 0.95")

## FINISH QUESTION 3 PART C CODE ##


## START QUESTION 3 PART D CODE ##

set.seed(303)

# Setting up samr object
data_matrix <- as.matrix(arraydata)

data = list(x = data_matrix,y = c(rep(1,4),rep(2,4)),
            geneid = row.names(data_matrix),genenames = row.names(data_matrix),
            logged2 = TRUE)

samr.obj = samr(data,resp.type = "Two class unpaired",nperms = 500,
                assay.type = "array")

# using samr.assess.samplesize function to estimate sample size. dif is set
```

```r
# to 1.
sample_size_3d <- samr.assess.samplesize(samr.obj,data = data,dif = 1)


# Plotting the sample size, FDR, and FNR.
samr.assess.samplesize.plot(sample_size_3d)

## FINISH QUESTION 3 PART D CODE ##

### FINISH QUESTION 3 CODE ###
```