BIOS 7659 Homework 6

Tim Vigers

31 October 2020

data("montgomery.subset")  
data("uCovar")

# 1. Next Generation Sequencing: Differential Expression

## a) Calculate RPKM and perform a t test

RPKM stands for reads per kilobases per million reads and is calculated by

lg = uCovar$length / 1000  
t = sum(unlist(montgomery.subset))  
rpkm = montgomery.subset/(lg\*t)

Do a standard t test between the groups for each gene:

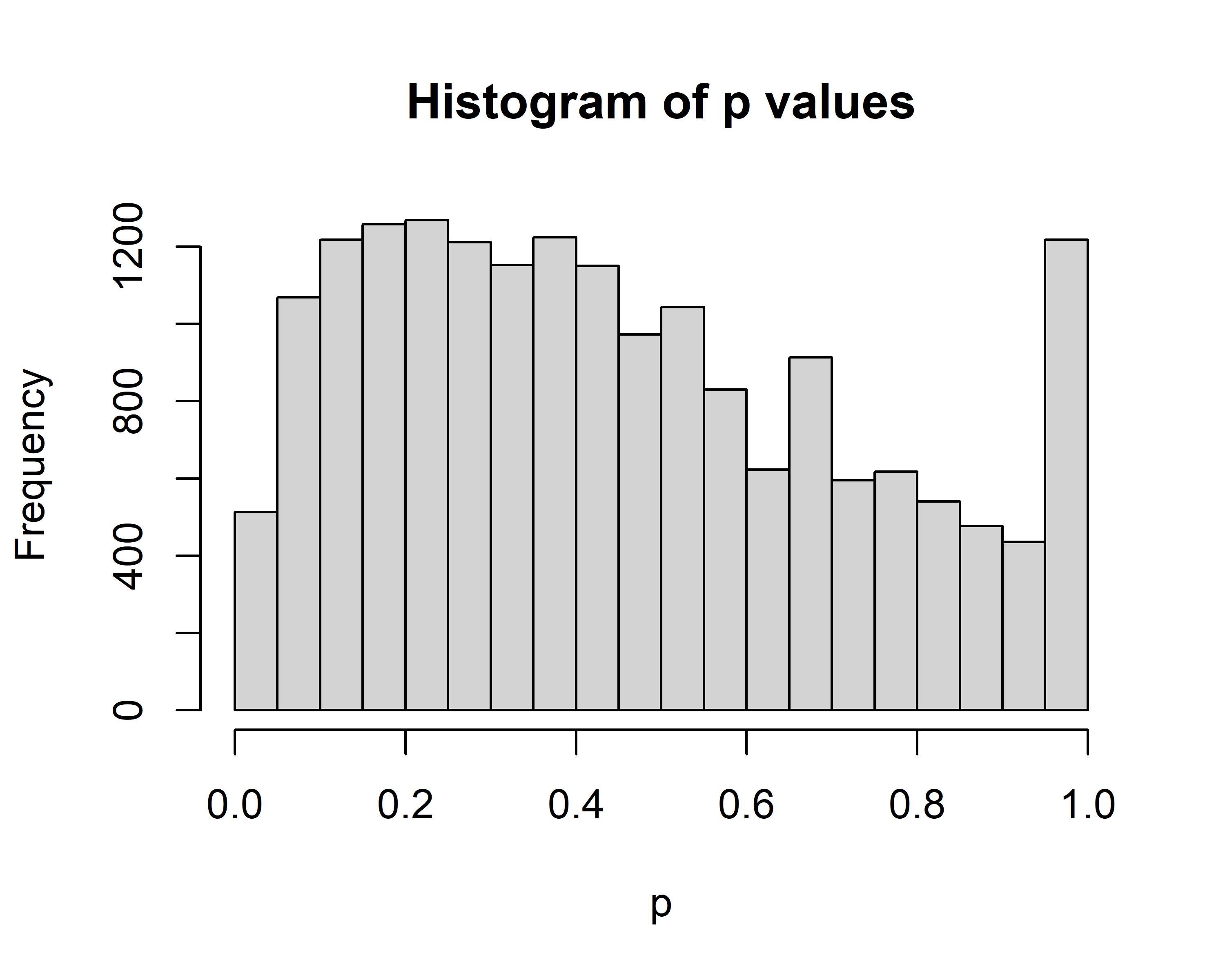
# T test for each row  
t\_tests = apply(rpkm,1,function(x){  
 group1 = as.numeric(x[1:5])  
 group2 = as.numeric(x[6:10])  
 if(var(group1)==0 | var(group2)==0){ # Skip those with constant values   
 return(c(NA,NA))  
 } else {  
 t <- t.test(group1,group2)  
 return(c(t$statistic,t$p.value))  
 }  
})  
# Format and print  
t\_tests = as.data.frame(t(t\_tests))  
t\_tests = t\_tests %>% rownames\_to\_column() %>%   
 set\_names(c("Gene","T","p")) %>% arrange(desc(abs(T)))   
  
t\_tests %>% head(10) %>% flextable(.) %>%   
 set\_caption("Top 10 Genes by t-statistic") %>% autofit(.)

Top 10 Genes by t-statistic

| Gene | T | p |
| --- | --- | --- |
| ENSG00000245208 | 5.498052 | 0.0005766918 |
| ENSG00000228109 | 5.258951 | 0.0011172910 |
| ENSG00000158985 | -5.146613 | 0.0053740327 |
| ENSG00000175274 | 4.826005 | 0.0023624029 |
| ENSG00000154710 | 4.810702 | 0.0021405930 |
| ENSG00000197747 | 4.714045 | 0.0018518080 |
| ENSG00000170802 | -4.686417 | 0.0028152695 |
| ENSG00000125629 | -4.624472 | 0.0035333403 |
| ENSG00000185947 | -4.575696 | 0.0085521476 |
| ENSG00000005187 | -4.495972 | 0.0026741415 |

## b) Plot the histogram of p-values

hist(t\_tests$p,main = "Histogram of p values",xlab = "p")



Normally we would expect a uniform distribution of p values, but this distribution appears to have a peak at around 0.3 and another at p = 1. My guess is that this is because we have only filtered genes with all 0 counts, but kept other genes with counts so low that they are effectively 0.

## c) Filter genes by total counts

# DOESN’T DGELIST AUTOMATICALLY INCLUDE lib.size

with the total reads per subject?

# Remove genes with low counts  
filtered = montgomery.subset[rowSums(montgomery.subset)>=10,]  
# Create edgeR object  
filtered\_dge = DGEList(filtered,group = c(rep(1,5),rep(2,5)))

## d) Calculate TMM normalization factors

norm = calcNormFactors(filtered\_dge)  
autofit(flextable(norm$samples))

| group | lib.size | norm.factors |
| --- | --- | --- |
| 1 | 2817342 | 0.9420009 |
| 1 | 2168185 | 1.0077134 |
| 1 | 2782270 | 0.9975042 |
| 1 | 2551303 | 0.9790312 |
| 1 | 4259582 | 0.9025138 |
| 2 | 1461731 | 1.1474002 |
| 2 | 5013427 | 1.0833630 |
| 2 | 3623044 | 0.9307130 |
| 2 | 6332824 | 0.9784469 |
| 2 | 3786172 | 1.0558556 |

The effective library sizes are generally similar to the column sums because the normalization factors are fairly close to 1. A normalization factor > 1 increases the library size, which is similar to downscaling the counts (and vice versa for factors < 1). So, the effective library size for sample 6 is increased by about 115%. Conversely, the effective library size for sample 5 is decreased by approximately 90%, which suggests that there are a small number of high-count sequences that need to be adjusted for.

## e) Use the estimateDisp() function to calculate the common, trended and tagwise dispersions

e\_disp = estimateDisp(filtered\_dge)

## Design matrix not provided. Switch to the classic mode.