STAT416 Assignment 3

Ashish Jain

Question 2:

```
data <- read.csv("https://raw.githubusercontent.com/ashishjain1988/STAT416/master/HW3/hwk3_2.csv")
mseMethod1 <- sum((data[, 2] - mean(data[, 2]))^2)/nrow(data)
mseMethod2 <- sum((data[, 3] - mean(data[, 3]))^2)/nrow(data)
print(paste0("MSE of the first method is ", mseMethod1))
```

```
## [1] "MSE of the first method is 0.232096696903414"
```

```
print(paste0("MSE of the second method is ", mseMethod2))
```

```
## [1] "MSE of the second method is 0.362899131818149"
```

From the results, it is seen that the Mean Square error of the first method is less than the second method, thus the first method is better in terms of estimatig the log-fold-change.

Question 3:

a).

For each gene with length L and having m reads, the probability to observe those reads from the random mapping of the background level is the probability to observe at least m $P(Y \geq m)$ reads based on $Y \sim Poisson(L * p0)$ where $p0$ is the background rate or null probability to observe a hit on one base pair.

```
data1 <- read.csv("https://raw.githubusercontent.com/ashishjain1988/STAT416/master/HW3/hwk3_31.csv")
data2 <- read.csv("https://raw.githubusercontent.com/ashishjain1988/STAT416/master/HW3/hwk3_32.csv")
p0 <- 5e-06
m <- 3
L <- 1000
ppois(m, L * p0, lower.tail = FALSE) + dpois(m, L * p0)
```

```
## [1] 2.075536e-08
```

As the probability of $P(Y \geq m)$ for this gene is very less (less than 0.05), so we can say that these reads are true signals and not due to background noise or random mapping.

b).

```
## Upper Quartile
Cij1 <- calcNormFactors(data1, method = "upperquartile") * apply(data1,
    2, sum)
print("Normalization factor Cij for dataset 1")
```

```
## [1] "Normalization factor Cij for dataset 1"
```

```
print(Cij1)
```

```
## sample1.1 sample1.2 sample1.3 sample2.1 sample2.2 sample2.3
##   2251447   2446659   2300900   2217610   2186376   1967738
```

```
Cij2 <- calcNormFactors(data2, method = "upperquartile") * apply(data2,
    2, sum)
print("Normalization factor Cij for dataset 2")
```

```
## [1] "Normalization factor Cij for dataset 2"
```

```
print(Cij2)
```

```
## sample1.1 sample1.2 sample1.3 sample2.1 sample2.2 sample2.3
##   2367122   2443153   2169439   1926137   2311365   1997100
```

c).

```
## DESeq
Cij1 <- calcNormFactors(data1, method = "RLE") * apply(data1,
    2, sum)
print("Normalization factor Cij for dataset 1")
```

```
## [1] "Normalization factor Cij for dataset 1"
```

```
print(Cij1)
```

```
## sample1.1 sample1.2 sample1.3 sample2.1 sample2.2 sample2.3
##   2230626   2442040   2236071   2230681   2219150   2005492
```

```
Cij2 <- calcNormFactors(data2, method = "RLE") * apply(data2,
    2, sum)
print("Normalization factor Cij for dataset 2")
```

```
## [1] "Normalization factor Cij for dataset 2"
```

```
print(Cij2)
```

```
## sample1.1 sample1.2 sample1.3 sample2.1 sample2.2 sample2.3
##   2270336   2260806   2052058   2060756   2466185   2083911
```

d).

```
## TMM
Cij1 <- calcNormFactors(data1, method = "TMM") * apply(data1,
    2, sum)
print("Normalization factor Cij for dataset 1")
```

```
## [1] "Normalization factor Cij for dataset 1"
```

```
print(Cij1)
```

```
## sample1.1 sample1.2 sample1.3 sample2.1 sample2.2 sample2.3
##   2233553   2434171   2249305   2215057   2218728   2011989
```

```
Cij2 <- calcNormFactors(data2, method = "TMM") * apply(data2,
    2, sum)
print("Normalization factor Cij for dataset2")
```

```
## [1] "Normalization factor Cij for dataset2"
```

```
print(Cij2)
```

```
## sample1.1 sample1.2 sample1.3 sample2.1 sample2.2 sample2.3
##   2277655   2238957   2057230   2068171   2469354   2082033
```

Question 4:

a).

```
y <- c(53, 72, 37, 135, 157, 189)
Cij <- c(123, 236, 195, 208, 164, 171)
log.75q <- log(Cij)
trt = as.factor(c(1, 1, 1, 2, 2, 2))
# cbind(trt, y, log.75q)
o = glm(y ~ trt, family = poisson(link = log), offset = log.75q)
summary(o)
```

```
##
## Call:
## glm(formula = y ~ trt, family = poisson(link = log), offset = log.75q)
##
## Deviance Residuals:
##       1        2        3        4        5        6
##  2.6512   0.3573  -2.8350  -3.8112   0.9602   2.9345
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.22957    0.07857  -15.65   <2e-16 ***
## trt2         1.10833    0.09084   12.20   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 211.205  on 5  degrees of freedom
## Residual deviance:  39.252  on 4  degrees of freedom
## AIC: 81.354
##
## Number of Fisher Scoring iterations: 4
```

```
a = anova(o, test = "Chisq")
print(a)
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: y
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                     5    211.205
## trt   1   171.95         4     39.252 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
phihat = deviance(o)/df.residual(o)
print(phihat)
```

```
## [1] 9.81302
```

```
1 - pchisq(deviance(o), df.residual(o))
```

```
## [1] 6.179242e-08
```

The phihat in this case is greater than 1. This is also confirmed by the chi-square test that the data is over-dispersed with significant p-value. This means that the data is over-dispersed compare to poission model.

b).

As, the data is over-dispersed we need to use F-test based on quasi-likelihood method to calculate the differential expression analysis instead of Likelihood ratio test.

```
Fstat = a[2, 2]/a[2, 1]/phihat
print(Fstat)
```

```
## [1] 17.52294
```

```r
pvalue = 1 - pf(Fstat, a[2, 1], a[2, 3])
print(pvalue)
```

```
## [1] 0.01385064
```

As, the p-value is less than 0.05, we can reject the null hypothesis and say that this gene is differentially expressed in the two genotypes.