STAT30850 Homework 1

Sarah Adilijiang

Problem 1

(a)

Using Bonferroni correction at level α, we have:

$$\begin{aligned} P(\text{reject } H_{0,global}) &= P(\min_i p_i \le \frac{\alpha}{n}) \;=\; 1 - P(\min_i p_i > \frac{\alpha}{n}) \\ &= 1 - P(p_1 > \frac{\alpha}{n}, \dots, p_n > \frac{\alpha}{n}) \;=\; 1 - \prod_{i=1}^{n} P(p_i > \frac{\alpha}{n}) \\ &= 1 - \prod_{i=1}^{n} \Big[1 - P(p_i \le \frac{\alpha}{n})\Big] \\ &= 1 - (1 - \frac{\alpha}{n\tau})^{n\pi} (1 - \frac{\alpha}{n})^{n(1-\pi)} \end{aligned}$$

(b)

Using Fisher's test, under the global null where all $p_i \sim \text{Unif}[0,1]$, the test statistics:

$$F = -2\sum_{i=1}^{n} \log p_i = \chi^2_{2n} \approx N(2n, 4n)$$

We reject the global null if $F \ge c$, i.e. $\Phi(\frac{c-2n}{\sqrt{4n}}) = 1 - \alpha$, so we get that:

$$c = 2n + \sqrt{4n}\Phi^{-1}(1-\alpha) = 2n - 2\sqrt{n}\Phi^{-1}(\alpha)$$

Therefore, the probability of rejecting the global null is:

$$\begin{aligned} P(\text{reject } H_{0,global}) &= P(-2\sum_{i=1}^{n} \log p_i \ge c) \\ &= P(-2\sum_{i=1}^{n\pi} [\log \frac{p_i}{\tau} + \log \tau] - 2 \sum_{i=n\pi+1}^{n} \log p_i \;\ge c) \\ &= P(\chi^2_{2n} - 2n\pi \log \tau \;\ge c) \;=\; P(\chi^2_{2n} \;\ge c + 2n\pi \log \tau) \\ &\approx 1 - \Phi\left(\frac{c + 2n\pi \log \tau - 2n}{\sqrt{4n}}\right) = 1 - \Phi(-\Phi^{-1}(\alpha) + \sqrt{n}\pi \log \tau) \\ &= \Phi(\Phi^{-1}(\alpha) - \sqrt{n}\pi \log \tau) \end{aligned}$$

(c)

Now let $\pi = n^{-1/3}$, $\tau = 1/2$, and we have assumed that $\tau > \alpha/n$.

For Bonferroni's test:

$$
\begin{aligned}
P(\text{reject } H_{0,global}) &= P(\min_i p_i \le \frac{\alpha}{n}) \le \sum_{i=1}^{n} P(p_i \le \frac{\alpha}{n}) \\
&= \frac{\alpha}{n\tau} n\pi + \frac{\alpha}{n} n(1-\pi) = \frac{\alpha\pi}{\tau} + \alpha(1-\pi) \\
&= \alpha(1 + n^{-1/3}) \\
&\to \alpha \quad (\text{as n} \to \infty)
\end{aligned}
$$

For Fisher's test:

$$
\begin{aligned}
P(\text{reject } H_{0,global}) &\approx \Phi(\Phi^{-1}(\alpha) - \sqrt{n\pi}\log\tau) \\
&= \Phi(\Phi^{-1}(\alpha) + n^{1/6}\log 2) \\
&\to \Phi(\infty) = 1 \quad (\text{as n} \to \infty)
\end{aligned}
$$

(d)

Now let $\pi = n^{-2/3}$, $\tau = 1/n$, and we have assumed that $\tau > \alpha/n$.

For Bonferroni's test:

$$
\begin{aligned}
P(\text{reject } H_{0,global}) &= 1 - (1 - \frac{\alpha}{n\tau})^{n\pi}(1 - \frac{\alpha}{n})^{n(1-\pi)} \\
&= 1 - (1-\alpha)^{n^{1/3}}(1 - \frac{\alpha}{n})^{n - n^{1/3}} \\
&\to 1 - 0 \times e^{-\alpha} = 1 \quad (\text{as n} \to \infty)
\end{aligned}
$$

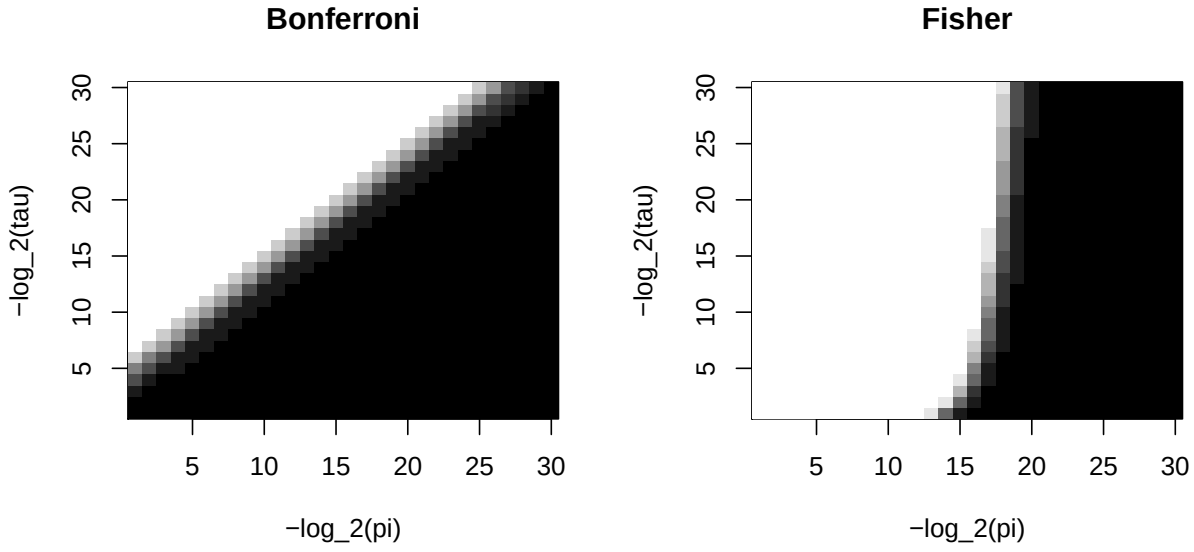
For Fisher's test:

$$
\begin{aligned}
P(\text{reject } H_{0,global}) &\approx \Phi(\Phi^{-1}(\alpha) - \sqrt{n\pi}\log\tau) \\
&= \Phi(\Phi^{-1}(\alpha) + n^{-1/6}\log n) \\
&\to \Phi(\Phi^{-1}(\alpha)) = \alpha \quad (\text{as n} \to \infty)
\end{aligned}
$$

Problem 2

```
alpha = 0.05
n = 2^30
Bonferroni = Fisher = matrix(0,30,30)
for (i in 1:30) {
    pi = 2^(-i)
    for (j in 1:30) {
        tau = 2^(-j)
        Bonferroni[i,j] = 1 - (1-alpha/n/tau)^(n*pi) * (1-alpha/n)^(n-n*pi)
        val = qnorm(alpha, lower.tail=TRUE) - sqrt(n)*pi*log(tau)
        Fisher[i,j] = pnorm(val, lower.tail=TRUE)
    }
}

par(mfrow=c(1,2))
image(1:30, 1:30, Bonferroni, xlab="-log_2(pi)", ylab="-log_2(tau)",
      main="Bonferroni", col=gray((0:10)/10))
image(1:30, 1:30, Fisher, xlab="-log_2(pi)", ylab="-log_2(tau)",
      main="Fisher", col=gray((0:10)/10))
```



The above "phase transition diagrams" shows a sharp transition from a high chance of success to a high chance of failure. The grayscale corresponds to the chance of rejection: white means $P(\text{reject } H_{0,global}) = 100\%$, and black means $P(\text{reject } H_{0,global}) = 0\%$.

We can see that the higher values of π (lower i) and lower values of τ (higher j) improves the probability of rejecting the $H_{0,global}$. However, the regions of successfully rejecting the $H_{0,global}$ are different for the two methods. Also, the phase transition boundaries from success to failure are different for the two methods. These phenomena can be explained by using their exact expressions of the probabilities.

For Bonferroni's test:

$$\begin{aligned} P(\text{reject } H_{0,global}) &= 1 - (1 - \frac{\alpha}{n\tau})^{n\pi}(1 - \frac{\alpha}{n})^{n-n\pi} \\ &\to 1 - e^{-\frac{n\pi}{n\tau/\alpha}} e^{-\frac{n-n\pi}{n/\alpha}} \quad (\text{as n} \to \infty) \\ &= 1 - e^{-\alpha(\pi/\tau + 1 - \pi)} \\ &= 1 - e^{-\alpha\ [\ 1 + (2^j - 1)/2^i\ ]} \end{aligned}$$

It's obvious that lower i values and/or higher j values will increase the probability of rejecting the global null. Also, we can see that the changes caused by i and j are at the same speed level. Therefore, the transition boundary is a straight line close to $y = x$. This indicates that in Bonferroni's test, when the proportion of true signal decreases, it needs stronger true signals to successfully reject the global null.

For Fisher's test:

$$\begin{aligned} P(\text{reject } H_{0,global}) &\approx \Phi(\Phi^{-1}(\alpha) - \sqrt{n}\pi \log \tau) \\ &= \Phi(\Phi^{-1}(\alpha) + 2^{-i} j\ \sqrt{n} \log 2) \end{aligned}$$

Again, it's obvious that lower i values and/or higher j values will increase the probability of rejecting the global null. Also, we can see that the change caused by i has much higher speed than that caused by j. Therefore, the transition boundary is close to a nearly vertical line at some value of i. This indicates that the Fisher's test relies much more on the proportion of true signal than the strength of the true signals to successfully reject the global null.

Problem 3

```
#download.file("ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE71nnn/GSE71220/matrix/GSE71220_series_matrix.t

# gene expression data: 617 people * 12381 genes
gene_expr = t(read.table("GSE71220_series_matrix.txt.gz",skip=66,nrows=12381)[,-1])
nsample = dim(gene_expr)[1]
n = dim(gene_expr)[2]

# which patients take statins
statin = read.table("GSE71220_series_matrix.txt.gz",skip=26,nrows=1)[-1]
statin = (strsplit(toString(unlist(statin)),"_")[[1]][2+3*(0:616)] == "statin")

# patients with COPD disease or healthy patients
disease = read.table("GSE71220_series_matrix.txt.gz",skip=37,nrows=1)[-1]
disease = (unlist(strsplit(strsplit(toString(unlist(disease)),": ")[[1]],","))[2*(1:nsample)] == "COPD"

# age of patient
age = read.table("GSE71220_series_matrix.txt.gz",skip=38,nrows=1)[-1]
age = as.numeric(unlist(strsplit(strsplit(toString(unlist(age)),": ")[[1]],","))[2*(1:nsample)])

# sex of patient (M or F)
sex = read.table("GSE71220_series_matrix.txt.gz",skip=39,nrows=1)[-1]
sex = (unlist(strsplit(strsplit(toString(unlist(sex)),": ")[[1]],","))[2*(1:nsample)])
```

```
# take the first 200 genes to save computation time
# and only perform tests for COPD patients
genes_d = gene_expr[disease==TRUE, 1:200]
statin_d = statin[disease==TRUE]
# X[i,j]: #i person, #j gene
```

(1) Permutation test (500 times)

First, we calculate the original statistic T without permutaion.

```
compute_T_stat = function(genes_data, statin_data) {
    corr = rep(0,200)
    for (i in 1:200) {
        corr[i] = cor(genes_data[,i], statin_data)
    }
    T_stat = sum(corr^2)
    return(T_stat)
}

T_stat = compute_T_stat(genes_d, statin_d); T_stat
```

```
## [1] 0.4660923
```

Then, we permute the data using the scheme (a) and calculate the p_value of the permutation test.

```
scheme_a = function(genes_data, statin_data, T_stat, per_num) {
    T_perms = rep(0,per_num)
    for (k in 1:per_num) {
        corr_perm = rep(0,200)
        for (i in 1:200) {
            genes_perm = sample(genes_data[,i], replace=FALSE)
            corr_perm[i] = cor(genes_perm, statin_data)
```

```
            }
        T_perms[k] = sum(corr_perm^2)
    }
    p_val = (1+sum(T_perms>=T_stat))/(1+per_num)
    return(p_val)
}

set.seed(123)
scheme_a(genes_d, statin_d, T_stat, per_num=500)
```

```
## [1] 0.003992016
```

Next, we permute the data using the scheme (b) and calculate the p_value of the permutation test.

```
scheme_b = function(genes_data, statin_data, T_stat, per_num){
    T_perms = rep(0,per_num)
    for (k in 1:per_num) {
        statin_perm = sample(statin_data, replace=FALSE)
        corr_perm = rep(0,200)
        for (i in 1:200) {
            corr_perm[i] = cor(genes_data[,i], statin_perm)
            }
        T_perms[k] = sum(corr_perm^2)
    }
    p_val = (1+sum(T_perms>=T_stat))/(1+per_num)
    return(p_val)
}

set.seed(123)
scheme_b(genes_d, statin_d, T_stat, per_num=500)
```

```
## [1] 0.1516966
```

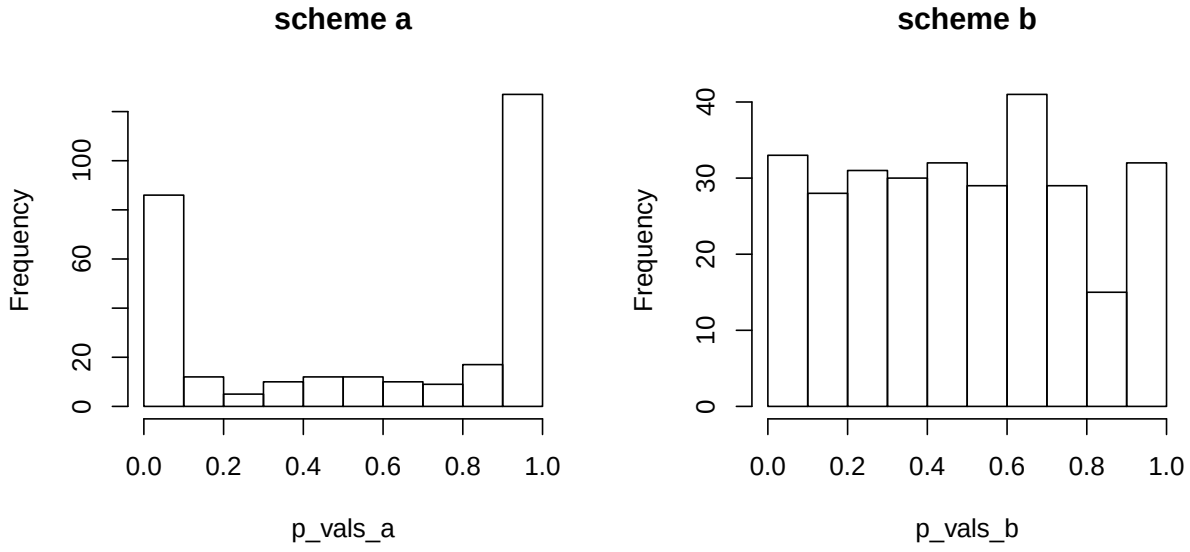
(2) Simulation (300 times): Sample Y randomly + Permutation test (100 times)

Now we first sample the vector Y (statins or no statins) randomly, then perform the permutation test using scheme (a) & (b).

We repeat this simulation process 300 times and check the histograms of p_values.

```
set.seed(123)
p_vals_a = rep(0,300)
p_vals_b = rep(0,300)
for (i in 1:300) {
    statin_sim = sample(c(0,1), length(statin_d) , replace=TRUE)
    T_stat = compute_T_stat(genes_d, statin_sim)
    p_vals_a[i] = scheme_a(genes_d, statin_sim, T_stat, per_num=100)
    p_vals_b[i] = scheme_b(genes_d, statin_sim, T_stat, per_num=100)
}
```

```
# plot histograms of p_values
par(mfrow=c(1,2))
hist(p_vals_a, main="scheme a")
hist(p_vals_b, main="scheme b")
```



Results & Discussions:

Part (2):

In the above simulation, we sampled the vector Y (statins or no statins) randomly. In this case, the simulated vector Y should not have any correlation with the genes expression levels, which means that if the permutation test is valid, then the p_values would have an approximately uniform distribution.

From the resulting histograms, we can see that the distribution of p_values in scheme (b) is approximately a uniform distribution, while that in scheme (a) is obviously not. Therefore, the scheme (b) is a more valid way of testing the global null hypothesis (i.e. testing if there is any association between statin use and gene expression levels).

Scheme (b) is a more valid way because in this way the correlations between different genes in one person are not destroyed by the permutation test.

This is important since the genes within one person are correlated with each other and they together present a gene expression pattern under different conditions. Especially, the genes that are being affected directly or indirectly by statin use would be highly correlated with each other in this experiment thus should be considered as a whole picture. Therefore, when we are testing the association between statin use and gene expression levels, we should not permute a single gene out of the whole genome, instead, should keep the gene structure as it is.

Part (1):

As a result, looking back at the permutation test results in part (1), the p_value of the permutation test in scheme (b) is 0.1516966, which means there is no significant evidence against the null hypothesis.

Therefore, we may conclude that, only looking at the first 200 genes, there is no association between statin use and gene expression levels.