Hyun-Joon Yang

BISC 478

yanghyun@usc.edu

Final Exam

In [1]:
```
from IPython.display import Image
from IPython.core.display import HTML

HTML("""
<style>
.output_png {
    display: table-cell;
    text-align: center;
    vertical-align: middle;
}
.output_jpeg {
    display: table-cell;
    text-align: center;
    vertical-align: middle;
}
</style>
""")
```
Out[1]:

In [2]:
```
import numpy as np
import pandas as pd
import scipy.stats as stats
```

1. Below are the un-normalized counts for a fictional RNA-seq experiment with only four genes

In [3]:
```
def TPM(df):
    df = df.copy()
    for i in range(df.shape[0]):
        df.iloc[i,1:] = df.iloc[i,1:] / df.iloc[i,0]
#     print(df)
    for i in range(df.shape[1]-1):
        i += 1
        total = df.iloc[:,i].sum()
#         print(total)
        factor = total / 1000000
#         print(factor)
        df.iloc[:,i] = df.iloc[:,i] / factor
    return df
```

In [4]:
```
d = {
    'Size': [1, 20, 5, 10],
    'Condition1-A': [10000, 800000, 100000, 0],
    'Condition1-B': [50000, 4000000, 500000, 0],
    'Condition2-A': [100000, 8000000, 1000000, 7000000],
    'Condition2-B': [25000, 2000000, 250000, 1750000]
}
df = pd.DataFrame(data=d)
df.index = ['A', 'B', 'C', 'D']
df
```
Out[4]:

	Size	Condition1-A	Condition1-B	Condition2-A	Condition2-B
A	1	10000	50000	100000	25000
B	20	800000	4000000	8000000	2000000
C	5	100000	500000	1000000	250000
D	10	0	0	7000000	1750000

a. (5 pts) Compute TPM for the data on just genes A, B, and C

In [5]:
```
TPM(df.iloc[0:3,:])
```
Out[5]:

	Size	Condition1-A	Condition1-B	Condition2-A	Condition2-B
A	1	142857.142857	142857.142857	142857.142857	142857.142857
B	20	571428.571429	571428.571429	571428.571429	571428.571429
C	5	285714.285714	285714.285714	285714.285714	285714.285714

b. (5 pts) Compute TPM for the data on all four genes

In [6]:
```
TPM(df)
```
Out[6]:

	Size	Condition1-A	Condition1-B	Condition2-A	Condition2-B
A	1	142857.142857	142857.142857	71428.571429	71428.571429
B	20	571428.571429	571428.571429	285714.285714	285714.285714
C	5	285714.285714	285714.285714	142857.142857	142857.142857
D	10	0.000000	0.000000	500000.000000	500000.000000

c. (5 pts) Considering the ideas behind TMM (which is different than TPM), which gene(s) are differentially expressed between conditions 1 and 2? Explain. (To answer this question, you do not need to do a statistical test.)

Under the assumption behind TMM that only a small number of genes will be highly differentially expressed in the different samples, only gene D seems to be differentially expressed especially considering the proportions of genes A, B, and C are the same in the two conditions. This is also evident in the first TPM table where the normalized counts are the same across conditions for genes A, B, C.

2. In this problem we are going to consider two different RNA-seq experiments. For both experiments we test for differential gene expression between mutant and wild-type. The first experiment (a) tests only 200 genes, while the second experiment (b) tests 20,000 genes. For both experiments, the three smallest un-adjusted p-values are: 10^{-10}, 10^{-7} , and 10^{-5}

In [7]:
```
def FDR(pvs):
    pvs = pd.Series(np.sort(pvs), name='p-value')
    rank = pd.Series(np.arange(1,len(pvs)+1), name='rank')
    df = pd.concat([pvs, rank], axis=1)
    df['pre'] = df['p-value'] * df.shape[0] / df['rank']
    adjusted = np.empty(df.shape[0])
    adjusted[:] = np.nan
    adjusted = pd.Series(adjusted, name='adjusted')
    adjusted.iloc[-1] = df['pre'].iloc[-1]
    for i in range(df.shape[0], 1, -1):
        i -= 2
        adjusted.iloc[i] = min(adjusted.iloc[i+1], df['pre'].iloc[i])
    df = pd.concat([df, adjusted], axis=1)
    return df
```

a. (5 pts) Complete the nine missing entries in the table below to compute the adjusted p-values by the False Discovery Rate (Benjamini-Hochberg) procedure for the three smallest p-values. The "…" represent the 196 p-values greater than 0.003. How many adjusted p-values are less than 0.05?

In [8]:
```
pvs = np.zeros([200]) + 100
pvs[0:4] = [10**-10, 10**-7, 10**-5, 0.003]
print(pvs.shape)
pvs[0:10]
```
```
(200,)
```
Out[8]:
```
array([1.e-10, 1.e-07, 1.e-05, 3.e-03, 1.e+02, 1.e+02, 1.e+02, 1.e+02,
       1.e+02, 1.e+02])
```

In [9]:
```
FDR(pvs).iloc[0:4]
```
Out[9]:

	p-value	rank	pre	adjusted
0	1.000000e-10	1	2.000000e-08	2.000000e-08
1	1.000000e-07	2	1.000000e-05	1.000000e-05
2	1.000000e-05	3	6.666667e-04	6.666667e-04
3	3.000000e-03	4	1.500000e-01	1.500000e-01

3 p-values are less than 0.05

b. (5 pts) Complete the nine missing entries in the table below to compute the adjusted p-values by the False Discovery Rate (Benjamini-Hochberg) procedure for the three smallest p-values. The "…" represent the 19,996 p-values greater than 0.00003. How many adjusted p-values are less than 0.05?

In [10]:
```
pvs = np.zeros([20000]) + 100
pvs[0:4] = [10**-10, 10**-7, 10**-5, 0.00003]
print(pvs.shape)
pvs[0:10]
```
```
(20000,)
```
Out[10]:
```
array([1.e-10, 1.e-07, 1.e-05, 3.e-05, 1.e+02, 1.e+02, 1.e+02, 1.e+02,
       1.e+02, 1.e+02])
```

In [11]:
```
FDR(pvs).iloc[0:4]
```
Out[11]:

	p-value	rank	pre	adjusted
0	1.000000e-10	1	0.000002	0.000002
1	1.000000e-07	2	0.001000	0.001000
2	1.000000e-05	3	0.066667	0.066667
3	3.000000e-05	4	0.150000	0.150000

2 p-values are less than 0.05

3. (5 pts) Explain one advantage of using the newer longer-read sequencing technologies instead of short-read Illumina sequencing technology in RNA-seq experiments.

Longer-read sequencing is better for discovering splice junctions as each read will be more likely to cover splice junctions.

4. (5 pts) Propose a scientific question that you could address with a single cell RNA-seq experiment that you could not answer with a bulk RNA-seq experiment.

How does a condition affect the gene expression of a specific type of cell as opposed to a collection of different cells? Ex. How does a certain virus affect the gene expression of the infected cell?

5.

a. (5 pts) Explain one difference between Principal Component Analysis and t-SNE.

One difference between PCA and t-SNE is that t-SNE has a random component so that each trial may lead to different projection.

b. (5 pts) When using the t-SNE method for dimension reduction for a single cell RNA-seq experiment, you have to select a value for the perplexity parameter P. Explain the problem if either too small or too large a value is selected for this parameter.

If P is too small, there will be too many resulting clusters for them to be useful as only a small number of neighbors can attract each other. On the other hand, if P is too large, there may not be any clustering at all.

6. In this problem we are going to consider two different GWAS on congenital heart disease. The first study has 1,000 cases and controls while the second study has 100,000 cases and controls. Other than the difference in sample sizes, the two studies are similar: they investigate the same disease, sample similar populations, and use identical SNP chips.

Below are tables for the same SNP from the two different studies (G is the risk allele). For each table, compute the p-value for the Cochran-Armitage trend test (with genomic control inflation factor $\lambda = 1$, so no adjustment), the odds ratio, and the 95% confidence interval for the odds ratio. Similar to HW #5, you may copy and paste the R code from GWAS lecture 4, which is posted on Blackboard, and change the numbers for the matrix m and the inflation factor λ. If you do not have R already installed on your computer, use the website mentioned in lecture: https://rdrr.io/snippets/

In [12]:
```
def trend_test(m, l):
    assert m.shape == (2,3)
    # calculate Y2
    N = sum(m.sum())
    R = np.sum(m.iloc[1,:])
    r1 = m.iloc[0,1]
    r2 = m.iloc[0,2]
    n1 = np.sum(m.iloc[:,1])
    n2 = np.sum(m.iloc[:,2])
    num = N*(N*r1+N*2*r2-R*n1-R*2*n2)**2
    den = (N-R)*R*(N*n1+N*4*n2-(n1+2*n2)**2)
    Y2 = num/den
    # calculate p-value
    pval = 1-stats.chi2.cdf(Y2/l,df=1)
    return [Y2, pval]
```

In [13]:
```
def odds_ratio(m):
    assert m.shape == (2,3)
    mall = np.zeros((2,2))
    print(mall[0,:])
    mall[0,:] = [2*m.iloc[0,0] + m.iloc[0,1], m.iloc[0,1] + 2*m.iloc[0,2]]
    mall[1,:] = [2*m.iloc[1,0] + m.iloc[1,1], m.iloc[1,1] + 2*m.iloc[1,2]]
    oddsratio = (mall[0,1]/mall[1,1])/(mall[0,0]/mall[1,0])
    print(oddsratio)
    s = np.sqrt(1/mall[0,0]+1/mall[0,1]+1/mall[1,0]+1/mall[1,1])
    conf = oddsratio*np.array([np.exp(-2*s),np.exp(2*s)], dtype='float') # 95% conf int
    print(conf)
    return oddsratio, conf
```

a. (5 pts)

In [14]:
```
d = np.reshape([614, 338, 48, 626, 334, 40], (2,3))
m = pd.DataFrame(data=d, columns=['TT', 'TG', 'GG'])
m.index = ['Cases', 'Controls']
m
```
Out[14]:

	TT	TG	GG
Cases	614	338	48
Controls	626	334	40

In [15]:
```
trend_test(m, 1)
```
Out[15]:
```
[0.602003467539973, 0.43781463184776115]
```

- p-value = 0.4378

In [16]:
```
odds_ratio(m)
```
```
[0. 0.]
1.061697546288584
[0.90946005 1.23941858]
```
Out[16]:
```
(1.061697546288584, array([0.90946005, 1.23941858]))
```

- odds-ratio = 1.0617
- 95% confidence interval = $[0.9095, 1.2394]$

b. (5 pts)

In [17]:
```
d = np.reshape([61400, 33800, 4800, 62600, 33400, 4000], (2,3))
m = pd.DataFrame(data=d, columns=['TT', 'TG', 'GG'])
m.index = ['Cases', 'Controls']
m
```
Out[17]:

	TT	TG	GG
Cases	61400	33800	4800
Controls	62600	33400	4000

In [18]:
```
# trend_test(m, 1)
```

- p-value = 8.5487×10^{-15}

In [19]:
```
odds_ratio(m)
```
```
[0. 0.]
1.061697546288584
[1.04539181 1.07825761]
```
Out[19]:
```
(1.061697546288584, array([1.04539181, 1.07825761]))
```

- odds-ratio = 1.0617
- 95% confidence interval = $[1.0454, 1.0783]$

c. (5 pts) The standard GWAS multiple test corrected significance value is 5×10^{-8} (SNPs with p-values below this threshold are significant, SNPs with p-values above this threshold are not significant). Is this value corrected for the number of SNPs or the number of individuals? Do the two studies agree on whether the (same) SNP in the two tables is significant? Explain why or why not. In your answer make sure to discuss the similarity in the counts in the two tables

It seems like the value is corrected for the number of SNPs not the number of individuals. In fact, despite the fact that the proportions of the counts in the 2 tables are the same for the same SNP, the SNP in the first table is considered not significant whereas the SNP in the second table is considered significant.

7. (5 pts) Explain one difference between linkage analysis and GWAS.

One difference between linkage analysis and GWAS is that linkage analysis uses pedigrees and genetic markers to determine regions of interest whereas GWAS uses SNPs of many different individuals.

8. (5 pts) For each of the five labeled regions in the figure below (Common variants with small effects, Intermediate frequency variants with moderate effects, etc.) state whether or not GWAS has been successful in finding SNPs in significant associations with phenotypes. For those regions that GWAS has not been successful, explain why not.

- Highly penetrant mutations: successful
- Intermediate frequency variants with moderate effects: successful
- Common variants with small effects: successful
- Rare variants with small effects: unsuccessful
- Common variants with large effects: unsuccessful

GWAS has not been successful for the last two categories as it operates under the common disease common variant hypothesis. For common variants with large effects, they are naturally limited by evolution. For rare variants with small effects, the rare SNPs are usually not genotyped on the SNP chips.

9. (10 pts) Explain one advantage of using whole genome sequencing instead of SNP chips in GWAS. Explain how this advantage relates to two of the possible solutions to the missing heritability problem.

Whole genome sequencing sequences the whole genome rather than specific SNPs (which only cover a very small portion of the genome) allowing it to not only detect genetic variations other than SNPs (eg indels), but also account for rare variants that are overlooked by SNP chips which usually only cover SNPs with relatively high frequency.

10. Imagine you do a GWAS to study the quantitative trait of height on an island population. You believe that the population is relatively homogeneous so you do not do genomic control or any other adjustment for population structure. After adjusting for multiple tests, you find 750 different SNPs that are significantly associated with height.

You learn later that everyone on the island belongs to of one of two tribes, and members of these two tribes have never inter-bred. Your GWAS sample included members of both tribes. Further, it turns out that one tribe is on average taller than the other tribe.

a. (4 pts) Given the information about the two tribes, do you expect all of the 750 significant loci to be related to the genetic architecture of height? Explain why or why not.

Not all 750 significant loci would be expected to be related to genetic architecture of height. Rather, some or most of them might just be loci related to the difference in population structure especially given that the two tribes never inter-bred and thus may diverge drastically genetically.

b. (3 pts) Assume that you do not have information on which members of your sample belong to which tribe. Explain a change you would make in analyzing the data that has already been collected.

In analyzing the data, we could use some form of genetic control like calculating inflation factor lambda to obtain the p-value or using EIGENSTRAT when finding significant SNPs.

c. (3 pts) Now assume that you have been able to determine which members of your sample belong to which tribe. Explain a change in analyzing the data that is different than what you described in (b).

If we know which members of the sample belong to which tribe, we can do 2 separate studies (one for each tribe) to see how different SNPs affect height in different populations. For each study, we can use an additive model and use regression to see whether a certain SNP is significant (whether / by how much parameter B_1 is significantly different than 0 in $y = B_0 + B_1 x$ where x is number of minor alleles). The trait model, then, would be the sum of all SNP models that were significant.

11. For both (a) and (b) below, assume you have access to any resources and currently developed technologies that you need. Explain the questions you are trying to address, the experiments you would do, and any challenges you foresee. (There are multiple correct answers to (a) and (b).)

a. (5 pts) Propose a GWAS experiment to study coronavirus

We could conduct an experiment to see whether there is a part of the genome that is associated with coronavirus deaths vs survival. We could get the genotypes of SNPs of all the individuals that have contracted the virus and whether they have died from the virus or not. We could then use genomic control like Cochran-Armitage trend test using inflation factor to find SNPs that are significantly associated with coronavirus deaths. With the SNPs, further analyses can be done to study the relationship between the SNP and the virus like the genotype odds ratio. However, some challenges that must be accounted for are multiple testing, confounding factors (eg pre-existing condition), and time scale (ie perhaps a subject has contracted the virus and may die soon, just not at the time data was collected).

b. (5 pts) Propose an RNA-seq experiment to study coronavirus.

We could conduct an experiment to see how the coronavirus affects the gene expression in red blood cells. Using single cell RNA-seq on many RBC samples from people that have contracted the virus vs people that have not, we can obtain the data for gene expression. We can then normalize the data using methods like TMM or TPM as well as correct for multiple testing using Benjamini-Hochberg Procedure to identify which genes are differentially expressed. Further studies can be done to examine how exactly the differentially expressed genes have an impact on the diseased body. However, some challenges that must be accounted for are batch effects, sequencing quality control, and PCR duplicates. Replication may help in finding better insight.