Cairo University,

Faculty of engineering,

Systems and Biomedical Engineering Department.

SBE304_Biostatistics

# Gene Expression (GE) Data analysis for Lung Squamous carcinoma (LUSC)

Prepared by:

Hassan Hosni, Mahmoud Khaled,

Mikhail Nady and Yousof Mahmoud

Submitted to:

Prof. Ibrahim Mohamed Youssef

Eng. Eslam Adel

# 1.Introduction

The disease of altered gene expression can be identified as the cancer. Many proteins (gene activation or gene silencing) are switched on or off which alter the cell's overall activity dramatically. A gene not usually expressed in the cell may be activated and expressed at high levels. This is due to gene mutation or changes in any gene regulatory stage (epigenetic, transcription, post-transcription, translation, or post-translation). In this Paper we conduct a study to analyze gene expression (GE) data for the cancer type Lung Squamous Cell Carcinoma (LUSC).

We have two files for the GE data for this cancer type:
1. GE data for tissues with cancer,
2. GE data for tissues in a healthy case.

Each row in the two files represent a gene, and the columns represent the expression level of this gene in different samples.

The Data which we have are paired meaning that a healthy sample and a diseased sample are taken from the same subject. So, both GE files have the same number of cases with the same order.

We discuss how to Compute the correlation between the normal samples and the diseased samples for each gene, rank genes based on their correlation coefficient (CC), report the highest positive CC and the lowest negative CC and the names of these two genes and Plot the expression levels of the above two genes. We also Infer the differentially expressed genes (DEGs); the genes whose expression level differ from one condition (healthy) to another (diseased).

Apply the appropriate test statistic for the following two pairing cases:

1. Samples are paired,
2. Samples are independent. Apply the FDR multiple tests correction method.

Report the set of DEGs before and after the FDR correction for each of the above two pairing cases. Compare the two DEGs sets (paired and independent) after the FDR correction in terms of the common and distinct genes.

# 2. Methods

## I. Correlation Coefficient

When two or more random variables are described in a probability space, how they differ together is helpful to describe; in other words, measuring the relation of the variables is useful. Covariance is a common measure of the relation between two random variables

The covariance between the random variables X and Y , denoted as cov (X, Y ) or $\sigma_{XY}$ , is

$$\sigma_{XY} = E[(\ X - \ \mu_x\ )\ (\ Y - \ \mu_Y\ )] \\ = E(\ XY\ ) - \ \mu_X \mu_Y \qquad (1)$$

we measure the relationship between two random variables by using correlation, because it is often easier to interpret than the covariance.

### I.1 Pearsonr Correlation

The correlation between random variables X and Y, denoted as $\rho_{XY}$ , is

$$\rho_{XY} = \frac{cov\ (\ X\ .\ Y\ )}{\sqrt{V(\ X\ )\ V(Y)}} = \frac{\sigma_{XY}}{\sigma_X\ \sigma_y} \qquad (\ 2)$$

For any two random variables X and Y,

$$-\ 1 \leq \rho_{XY} \leq +1$$

The correlation just scales the covariance by the product of the standard deviation of each variable. Consequently, the correlation

is a dimensionless quantity that can be used to compare the linear relationships between pairs of variables in different units.

For our study let (x: random variable represents gene expression for tissue in a healthy case) and (Y: random variable represents gene expression for tissue with cancer)

We Compute the correlation between the normal samples and the diseased samples for each gene by using Python code as follow:

*the first step:* We upload and read the data form (*"lusc-rsem-fpkm-tcga_paired.txt"*) file which contains GE data for tissues in a healthy case as in *fig1*

```
In [6]: #printing data of normal data for show not more and getting the number of rows
        df1
Out[6]:
```

| Hugo_Symbol | TCGA-43-7657 | TCGA-58-8386 | TCGA-22-5478 | TCGA-22-5472 | TCGA-43-5670 | TCGA-60-2709 | TCGA-22-5489 | TCGA-77-8007 | TCGA-22-5471 | TCGA-22-4609 | ... | TCGA-85-7710 | TCGA-56-7580 | TCGA-43-6647 | TCGA-90-6837 | TCGA-56-8083 | TCGA-51-4079 | TCGA-56-7222 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HIST3H2A | 62.12 | 130.60 | 33.06 | 35.50 | 73.03 | 60.39 | 92.05 | 66.65 | 54.33 | 15.56 | ... | 90.77 | 59.55 | 40.07 | 22.92 | 29.91 | 82.29 | 4.70 |
| LIN7B | 185.11 | 283.05 | 119.26 | 169.07 | 165.57 | 161.02 | 131.51 | 198.47 | 175.07 | 147.06 | ... | 185.11 | 119.26 | 102.97 | 123.50 | 264.03 | 194.36 | 166.73 |
| LXN | 909.17 | 819.30 | 412.00 | 743.43 | 1340.84 | 607.87 | 1709.26 | 1709.26 | 603.67 | 555.41 | ... | 813.63 | 2400.97 | 543.96 | 2193.99 | 540.19 | 521.76 | 253.23 |
| CNKSR2 | 41.81 | 18.29 | 40.93 | 67.12 | 54.72 | 29.27 | 20.26 | 23.76 | 28.04 | 39.22 | ... | 34.51 | 70.01 | 57.49 | 57.89 | 67.12 | 34.51 | 22.10 |
| SCML1 | 133.36 | 214.27 | 108.14 | 109.66 | 190.34 | 211.31 | 96.01 | 208.38 | 120.10 | 239.52 | ... | 251.48 | 209.84 | 120.10 | 109.66 | 155.50 | 162.14 | 277.20 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| HAVCR2 | 423.61 | 529.06 | 660.68 | 620.67 | 518.15 | 848.22 | 366.09 | 1073.91 | 363.56 | 366.09 | ... | 559.28 | 579.04 | 420.68 | 366.09 | 334.46 | 258.57 | 329.84 |
| RP1-66C13.4 | 0.00 | 0.00 | 1.79 | 3.32 | 0.00 | 0.00 | 1.79 | 0.00 | 0.00 | 0.00 | ... | 0.00 | 4.86 | 0.00 | 0.00 | 2.81 | 2.64 | 0.00 |
| C3orf79 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.16 | ... | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| CTD-2116N17.1 | 0.00 | 0.00 | 0.99 | 3.59 | 2.23 | 3.35 | 0.00 | 7.34 | 1.41 | 0.00 | ... | 6.94 | 1.00 | 0.00 | 4.58 | 0.96 | 3.06 | 1.83 |
| FUT2 | 64.34 | 101.54 | 14.45 | 69.52 | 110.43 | 36.53 | 60.82 | 116.78 | 66.18 | 25.54 | ... | 38.12 | 17.64 | 14.56 | 14.35 | 5.77 | 24.28 | 21.78 |

19648 rows × 50 columns

*Figure 1 GE data for Normal tissues*

and data from (*"lusc-rsem-fpkm-tcga-t_paired.txt")* file which contains GE data for tissues with cancer as in *fig 2*

| Hugo_Symbol | TCGA-43-7657 | TCGA-58-8386 | TCGA-22-5478 | TCGA-22-5472 | TCGA-43-5670 | TCGA-60-2709 | TCGA-22-5489 | TCGA-77-8007 | TCGA-22-5471 | TCGA-22-4609 | ... | TCGA-85-7710 | TCGA-56-7580 | TCGA-43-6647 | TCGA-90-6837 | TCGA-56-8083 | TCGA-51-4079 | TCGA-56-7222 | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HIST3H2A | 336.79 | 500.46 | 703.28 | 287.01 | 486.75 | 70.51 | 145.02 | 14.03 | 397.93 | 318.57 | ... | 3.06 | 420.68 | 109.66 | 106.63 | 1233.75 | 172.65 | 303.44 | 2 |
| LIN7B | 105.15 | 212.78 | 102.25 | 212.78 | 172.65 | 244.57 | 105.89 | 152.28 | 258.57 | 218.79 | ... | 135.24 | 135.24 | 151.22 | 395.18 | 295.11 | 120.94 | 114.36 | |
| LXN | 848.22 | 236.21 | 271.48 | 759.08 | 61.25 | 620.67 | 329.84 | 599.49 | 587.13 | 638.15 | ... | 688.78 | 204.07 | 438.59 | 503.95 | 3039.30 | 607.87 | 106.63 | 5 |
| CNKSR2 | 32.59 | 8.51 | 45.85 | 6.16 | 49.21 | 11.91 | 12.27 | 15.00 | 1.38 | 8.71 | ... | 1.38 | 6.62 | 6.11 | 1.66 | 33.54 | 3.11 | 0.82 | |
| SCML1 | 84.63 | 74.58 | 67.12 | 57.89 | 102.97 | 132.44 | 66.65 | 57.08 | 336.79 | 171.45 | ... | 165.57 | 119.26 | 87.65 | 53.57 | 232.94 | 67.12 | 64.80 | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| HAVCR2 | 74.58 | 432.53 | 128.79 | 208.38 | 13.93 | 633.73 | 348.71 | 420.68 | 283.05 | 432.53 | ... | 150.17 | 57.49 | 236.21 | 353.59 | 29.48 | 202.66 | 33.78 | |
| RP1-66C13.4 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 7.69 | 0.00 | 0.00 | 0.00 | ... | 0.00 | 31.00 | 1.83 | 0.00 | 3.32 | 5.73 | 0.00 | |
| C3orf79 | 2.27 | 1.66 | 0.00 | 1.22 | 0.00 | 0.00 | 0.00 | 0.00 | 3.44 | 0.00 | ... | 0.00 | 5.50 | 0.00 | 0.00 | 0.00 | 2.86 | 0.00 | |
| CTD-2116N17.1 | 6.89 | 10.79 | 8.51 | 6.84 | 9.13 | 9.93 | 14.78 | 15.56 | 8.13 | 0.97 | ... | 8.85 | 6.21 | 4.94 | 17.64 | 28.65 | 14.78 | 27.64 | |
| FUT2 | 105.15 | 58.30 | 39.79 | 1059.11 | 366.09 | 97.36 | 46.50 | 101.54 | 336.79 | 414.87 | ... | 106.63 | 35.50 | 55.49 | 738.29 | 232.94 | 244.57 | 71.50 | |

19648 rows × 50 columns

*Figure 2 GE data for tissues with cancer .*

From fig.1 and fig.2 , note that :

1- We have 19648 rows which represent 19648 genes for 50 samples which equal the number of columns after we delete the column which represent gene ID.

2- There are many genes which have many zeroes in their samples which mean that when we compute the variance for these samples, it will be zero so the correlation will be nan value.

*The second step*: we will delete the genes (from normal and cancerous data) which have 50% of their samples have zeroes by using python codes as in *fig 3* and *fig 4*

```
In [10]: #get the number or genes that will be deleted in stage 1
         len (oo)
Out[10]: 1973
```

*Figure 3 number of genes in normal data which 50% of their samples have zero value*

```
In [14]: #get the number or genes that will be deleted in stage 2
         len (oo2)
Out[14]: 338
```

*Figure 4 number of genes in cancer data which 50% of their samples have zero value*

From (fig.3 and fig.4) after we have deleted these genes from normal and cancerous data, we will compute Pearson correlation coefficient for 17337 genes instead of 19648 genes because we find (1973 + 338 = 2311) genes which variance is zero so the Pearson correlation coefficient for them will be nan value.

*The final step*: we will compute Pearson correlation coefficient for each gene using equation (1) by using python code as it is difficult to repeat these calculations for 17337 genes.

```
#calculate the pearsonr 's correlation between two variables deu to the linearity of data
from scipy.stats import pearsonr
#make array to take the correlation coeff
r=[]
for i in range (17337):
    # prepare data
    data_hy = datah[i]
    data_ca = datac[i]
    # calculate Pearson's correlation
    corr ,_=pearsonr (data_hy,data_ca)
    r.append(corr)
```

*Figure 5 python code to compute Pearson correlation coefficient for each gene for each gene*

```
In [20]: #sorting the correlation array
         sortcorr=ny.sort(r)
         print(sortcorr)
```

*Figure 6 Values for Pearson correlation coefficient for each gene after sorting*

## 1.2 - Pearson correlation Vs. Spearman correlation

Why do we use Pearson correlation and don't use Spearman correlation?

The Pearson correlation evaluates the linear relationship between two continuous variables. ... The Spearman correlation coefficient is based on the ranked values for each variable rather than the raw data. Spearman correlation is often used to evaluate relationships involving ordinal variables. so we use Pearson correlation

# II. Hypothesis Test

As we all know that in case of analyzing any biological data we need to make some assumptions or so called “Hypothesis", and to make sure of your assumption you need to test your hypothesis, so in this section we are going to see how did we test our data and a discussion about the results.

Like making the statistic test for the input data to determine the either you reject or fail to reject it, that’s by modeling your data as T distribution and then defining your confidence level, Critical regions – in case of two tailed curve-, the critical value and the accumulative probability of the critical region α in our case =0.05 .

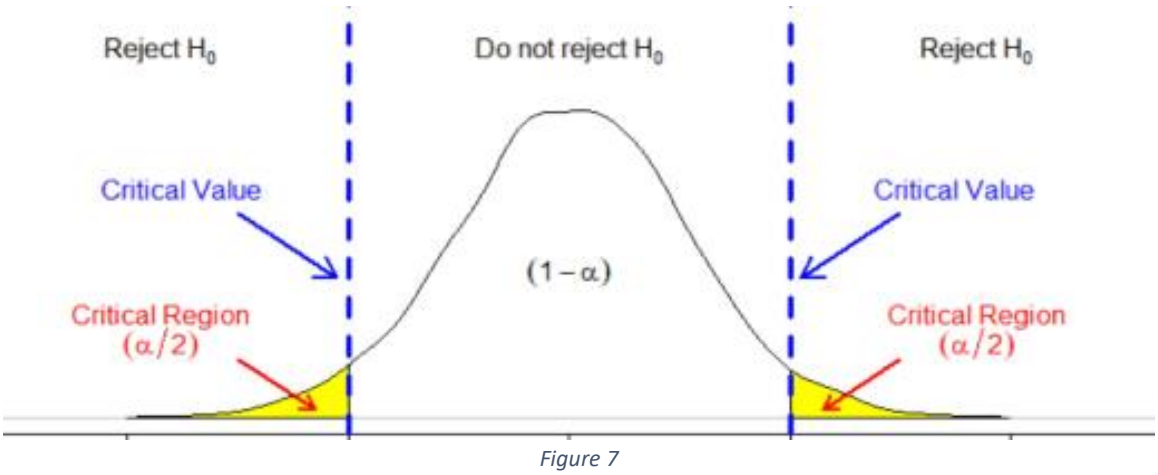


*Figure 7*

As illustrated in *fig.7* We have 2 spaces:

1- The first where we cannot reject the null Hypothesis with a probability of (1-α).
2- The second where the Null Hypothesis is rejected or the critical regions P=( α )

## For the code:

So we started the code by including some libraries we will use further, then we have read the files data both the healthy and the diseased without the gene ID as we are not using it in this stage

To deal with the data easily we have put them into arrays and started filtrating them by calculating the rows with more than 50% Zeros and then append them as we are not using these data in our following procedures, like in *fig 8*

```
In [7]: new_datah=ny. delete(datah, oo , 0)
        new_datac=ny. delete(datac, oo, 0)

In [8]: oo2=[]
        for i in range (17675):
            t=0
            ss=new_datac[i]
            for x in range (50):
                if ss[x]==0 :
                    t=t+1

            if t>25:
                oo2.append(i)

In [9]: datah2=ny. delete(new_datah, oo2, 0)
        datac2=ny. delete(new_datac, oo2, 0)

In [10]: datah =ny.array(datah2)
         datac =ny.array(datac2)
```

*Figure 8*

Here we got 2 cases the first one is by assuming our Paired data to be independent and make an Independent T test for it, and the second case is dealing with it as Dependent T test as in *fig 9*, which is the real case in this experiment because these data are paired, so in the first case we have run the Independent T test with

*the critical value = .05*
*Confidence level =95%*

to get the number of genes that have changed the gene expression and the genes that have not changed in GE , for the 2 case of Independent and Dependent data.

```
In [45]: #importing ttest_ind (independant t test)
from scipy.stats import ttest_ind
#creat 8 arrayes
inacc=[]
indeaccH0=[]
inre=[]
inderefH0=[]
pvalue1=[]
for i in range (17337):
    data_hy=datah[i]
    data_ca=datac[i]
    stat,p=ttest_ind(data_hy,data_ca)
    pvalue1.append(p)
    #accept the null hypothesis if p>0.05
    if p > 0.05:
        #appending the p value to inacc array
        inacc.append(p)
        #appending the gene to indeaccH0
        indeaccH0.append(datah[i])

    #reject the null hypothesis
    else:
        # appending the p value to inre array
        inre.append(p)
        #appending the gene to inderefH0
        inderefH0.append(datah[i])
```

*Figure 9*

For the last part and to correct the high error we are going to do the False Discovery Rate (FDR).
So as in ***fig10*** we run The FDR for the Independent and he dependent tests, to get to know the False and true positives or negatives ,and also to determine the common genes that will change in gene expression.

```
In [53]: import statsmodels.stats.multitest as multi
#do FDR for indepandant probably
q_fdr1=multi.multipletests(pvalue1,method='fdr_bh')[1]
#make new array a that will contane the genes we expect are significant GE for independant
a=[]
for i in range (len (q_fdr1)):
    if q_fdr1 [i]<=0.05 :
        a.append(q_fdr1 [i])
print ('the gene that is independant  and that will reject null hyposthesis are : '+str(len(a)))
```

```
the gene that is independant  and that will reject null hyposthesis are : 12320
```

```
In [54]: import statsmodels.stats.multitest as multi
#do FDR for indepandant probably
q_fdr2=multi.multipletests(pvalue2,method='fdr_bh')[1]
#make new array a that will contane the genes we expect are significant GE for dependant
b=[]
for i in range (len (q_fdr2)):
    if q_fdr2 [i]<=0.05 :
        b.append(q_fdr2 [i])
print ('the gene that is independant  and that will reject null hyposthesis are : '+str(len(b)))
```

```
the gene that is independant  and that will reject null hyposthesis are : 12410
```

```
In [55]: print ('for independant t test the comman genes that will change in GE ='+str(len(a))+' genes')
print ('for independant t test the  genes that will no  change in GE after FDR ='+str(len(inre)-len(a))+' genes')
print ('for dependant t test the comman genes that will change in GE ='+str(len(b))+' genes')
print ('for dependant t test the  genes that will no change in GE after FDR  ='+str(len(dre)-len(b))+' genes')
```

```
for independant t test the comman genes that will change in GE =12320 genes
for independant t test the  genes that will no  change in GE after FDR =311 genes
for dependant t test the comman genes that will change in GE =12410 genes
for dependant t test the  genes that will no change in GE after FDR  =314 genes
```

*Figure 10*

# III. Results

In this section we will show and discuss the output and results of the code and the meaning of these data.

At first we have filtered the input data, so after the filtration we got only 17337 genes left from 19648 genes.

## For the first part of Correlation

We had sorted the correlation values to arrange them to easily get the minimum negative Correlation coefficient" min –ve CC' and the maximum positive correlation coefficient "max +ve CC"

As in *fig.11*

*the min –ve CC = -0.45280727852470837*

*the max +ve CC= 0.9690441442970708*

```
In [21]: #min -ve cc
         mincorr=min(r)
         print ('min -ve cc : '+str(mincorr))

         min -ve cc : -0.45280727852470837

In [22]: #max +ve cc
         maxcorr=max(r)
         print("max +ve cc : "+str(maxcorr))

         max +ve cc : 0.9690441442970708
```

*Figure 11 the Min –ve CC & the Max +ve CC*

from the index we got to know the genes itself as in *Fig.12 and 13*

```
In [26]: #printing the highest  +ve cc in normal data
         data11[maxindex:maxindex+1]
```

Out[26]:

| Hugo_Symbol | TCGA-43-7657 | TCGA-58-8386 | TCGA-22-5478 | TCGA-22-5472 | TCGA-43-5670 | TCGA-60-2709 | TCGA-22-5489 | TCGA-77-8007 | TCGA-22-5471 | TCGA-22-4609 | ... | TCGA-85-7710 | TCGA-56-7580 | TCGA-43-6647 | TCGA-90-6837 | TCGA-56-8083 | TCGA-51-4079 | TCGA-56-7222 | TC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AREGB | 23.76 | 89.51 | 17.0 | 7.75 | 12.55 | 39.22 | 33.78 | 4.62 | 29.48 | 96.01 | ... | 0.0 | 10.16 | 34.26 | 10.31 | 33.54 | 65.26 | 43.02 | 2 |

1 rows × 50 columns

*Figure 12 the Max Gene*

```
In [28]: #printing the lowest -ve cc in normal data
         data11[minindex:minindex+1]
```

Out[28]:

| Hugo_Symbol | TCGA-43-7657 | TCGA-58-8386 | TCGA-22-5478 | TCGA-22-5472 | TCGA-43-5670 | TCGA-60-2709 | TCGA-22-5489 | TCGA-77-8007 | TCGA-22-5471 | TCGA-22-4609 | ... | TCGA-85-7710 | TCGA-56-7580 | TCGA-43-6647 | TCGA-90-6837 | TCGA-56-8083 | TCGA-51-4079 | TCGA-56-7222 | TC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FAM222B | 285.03 | 214.27 | 336.79 | 295.11 | 312.0 | 206.94 | 295.11 | 214.27 | 283.05 | 281.09 | ... | 249.73 | 287.01 | 297.17 | 339.14 | 248.0 | 307.69 | 248.0 | 19 |

1 rows × 50 columns

*Figure 13 The Min Gene*

and the following is the plots for the min –ve CC and the max +ve CC in *fig.14* and *fig.15* respectfully.

```
In [31]: #plotting the expression level for the lowest -ve cc
         #importing pyplot
         from matplotlib import pyplot
         # prepare data
         data1 = datahh[minindex]
         data2 = datacc[minindex]
         # plot
         pyplot.scatter(data1, data2)
         pyplot.show()
```

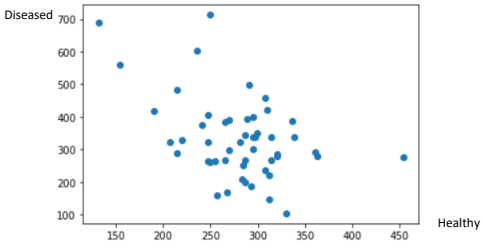


*Figure 14 Plot for min –ve CC*

```
In [30]: #plotting the expression level for the highest +ve cc
         #importing pyplot
         from matplotlib import pyplot
         # prepare data
         data1 = datahh[maxindex]
         data2 = datacc[maxindex]
         # plot
         pyplot.scatter(data1, data2)
         pyplot.show()
```

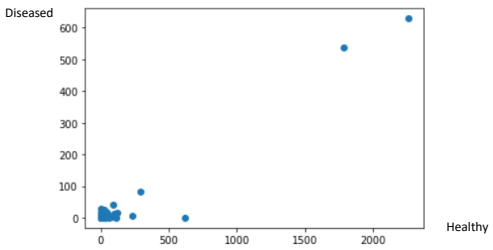


*Figure 15 Plot for max +ve CC*

## For the Second part of Hypothesis testing

we used the independent and Paired T distribution to calculate the Null Hypothesis and Alternative Hypothesis

$H_0$ : there is no GE due to LUSC

$H_1$: there is GE due to LUSC

As we illustrate in *Table1* the values of both $H_0$ and $H_1$

*Table 1 H0 and H1 values*

| | $H_0$ | $H_1$ |
|---|---|---|
| Paired Data | **4613** | **12724** |
| Independent | **4706** | **12631** |
| FDR for Paired Data | **4927** | **12410** |
| FDR for Independent Data | **5017** | **12320** |

And in *table 2* we have the values of the common and differ genes

*Table 2 For Common and differ Values*

| | $H_1$ |
|---|---|
| Common genes for Paired Data after and before FDR | **12410** |
| Genes with no GE after FDR for Paired Data | **314** |
| Common genes for independent Data after and before FDR | **12320** |
| Genes with no GE after FDR for Independent Data | **311** |

So we concluded that the impact of the FDR is cutting 97.5% of the original $H_1$ to get the common genes for the Data after and before FDR.

And the impact of the FDR is cutting 2.5% of the original $H_1$ to get the distinct genes for the Data after and before FDR.

## The Contribution for each members

We all had an online meeting at the first of the task to discuss the idea and to split the task into small ones and the following is how it has been done

Each two members had a requirement of the task and worked on it to make the code and the paper draft and then we got to make an online meeting to collect the data and make the final Paper.

| Tasks | Hassan Hosni | Mahmoud Khaled | Mikhail Nady | Yousof Mahmoud |
|---|---|---|---|---|
| **Method of Importing Data** | 40% | 60% | - | - |
| **Method of filtering Data** | 60% | 20% | 10% | 10% |
| **Correlation Function** | 30% | 50% | 20% | - |
| **Correlation Documentation** | - | - | 50% | 50% |
| **Hypothesis Function** | 50% | 10% | 10% | 30% |
| **Hypothesis Documentation** | 30% | - | 10% | 60% |
| **Gathering the Code** | %70 | 30% | - | - |
| **Gathering the Documentation** | - | - | 40% | 60% |