Cairo University

Systems & Biomedical Engineering Department

Biostatistics

Presented to: DR/ Ibrahim Mohamed Ibrahim

Presented by	**Sec**	**BN**
Mariam Ashraf Mohamed	2	23
Marwa AdbelAal Ali	2	22
Nada Ezzat Shawky	2	37
Noura Mahmoud Sayed	2	38

Introduction:

In this paper we want to analyze gene expression (GE) data for the cancer type Lung Squamous Cell Carcinoma (LUSC) applying correlation and hypothesis test concepts.

Correlation concept:-

The correlation coefficient is a statistical measure of the strength of the relationship between the relative movements of two variables. The values range between -1.0 and 1.0. A calculated number greater than 1.0 or less than -1.0 means that there was an error in the correlation measurement. So, we want to know How likely is the relationship between the two data frames of healthy and cancerous genes to be linear and which gene expressions that associate highly in the precedence of the disease and how strong their relationship is and it's direction (+ve or -ve), So we're going to calculate and plot all CC's to answer these questions.

Hypothesis test concept:-

Thus, we are going to use hypothesis test assuming that null hypothesis (H0) is gene's expression doesn't affect having cancer and the alternative hypothesis (H1) represents gene's expression affects on existence of cancer. We will have this study on two parallel cases: independent and paired samples. We aim to reduce the error using FDR correction method.

Methods:-

Software packages:

- pandas: to use data frame and read files
- scipy.stats: to use functions that applies t-test statistic and calculate p-values in case of independence and pairing samples and comparing them to confidence level of 95%
- statsmodels.stats.multitest: to do (FDR) correction process
- matplotlib.pyplot : it's a collection of functions that make matplotlib work like MATLAB. we use its functions to plot pearsons correlation coefficient like plot.scatter(x,y), plot.show().

Steps:

1-filtration:

- At first, we imported the files of both healthy and cancerous genes and stored them in data frame to be like tables with rows and columns
- As we don't need any gene which has more than 25 zeros in its expression level samples so we should get rid of all rows of these unwanted genes of the healthy data from both healthy and cancerous data
- Similarly, we remove all rows of unwanted genes of the cancerous data from both healthy and cancerous data. Now, we have both data with only wanted genes to do processes we want

Correlation steps:

1. We iterate over the filtered data frames to calculate each r (correlation coefficient) of every gene in our data.
2. We store these coefficients in a list, then we add this list as a column in the filtered data frames.
3. We now sort these data frames according to the column (r or CC).
4. We then get the highest positive CC and the lowest negative CC and the names of these two genes and this satisfies the first requirement.
5. Then we plot the expression levels of the above two genes in 2 graphs one for each gene in healthy and cancer data frames using our matplotlib package.

Hypothesis test steps:

1- We iterate on all rows to calculate the p-values in case of independent and paired samples
2- We assume that confidence level of 95%, so the significance level (alpha) will be 0.05
3- Each p-value is compared to the significance level so that, if p-value is greater than or equal to 0.05 then it's failed to reject null hypothesis. Hence, p-value is smaller than 0.05 then it rejects the null hypothesis.

Now, we know the genes which lie in rejection region in addition to their p-values for both independence and paired samples

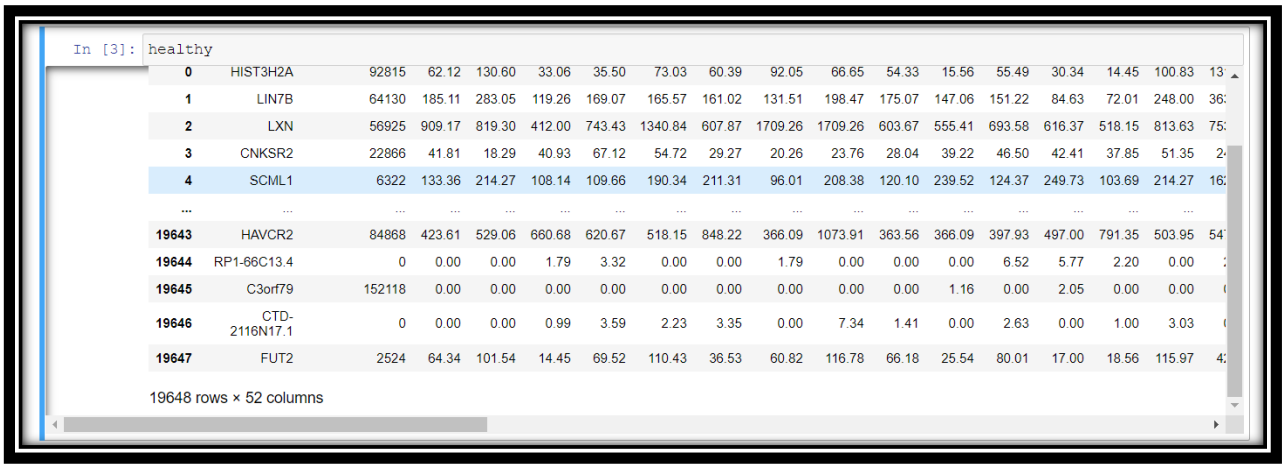
4- Then we need to reduce error, so we use the FDR correction method, so apply it on all p-values of both independent and paired samples
5- We loop on all rows again to specify the genes and their p-values which lie in rejection modes after correction
6- Finally, we apply conversion on the lists of corrected p-values in rejection region in the two pairing cases to be in set form, so that we can get the common and distinct genes.

Results and Discussion:

Correlation:

1. We first read the data of healthy and cancerous genes and convert them to data frames using pandas package.

2. We start by filtering our data frames from all the rows that has more than 25 zeros as these values doesn't make sense and give us unwanted or weird results. We do this by dropping all the rows we find in the healthy genes from both data frames health and cancer, then we drop the rows that has the same condition we find in the cancerous one from both data frames health and cancer.



In [3]: healthy

0	HIST3H2A	92815	62.12	130.60	33.06	35.50	73.03	60.39	92.05	66.65	54.33	15.56	55.49	30.34	14.45	100.83	13
1	LIN7B	64130	185.11	283.05	119.26	169.07	165.57	161.02	131.51	198.47	175.07	147.06	151.22	84.63	72.01	248.00	36
2	LXN	56925	909.17	819.30	412.00	743.43	1340.84	607.87	1709.26	1709.26	603.67	555.41	693.58	616.37	518.15	813.63	75
3	CNKSR2	22866	41.81	18.29	40.93	67.12	54.72	29.27	20.26	23.76	28.04	39.22	46.50	42.41	37.85	51.35	2
4	SCML1	6322	133.36	214.27	108.14	109.66	190.34	211.31	96.01	208.38	120.10	239.52	124.37	249.73	103.69	214.27	16
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
19643	HAVCR2	84868	423.61	529.06	660.68	620.67	518.15	848.22	366.09	1073.91	363.56	366.09	397.93	497.00	791.35	503.95	54
19644	RP1-66C13.4	0	0.00	0.00	1.79	3.32	0.00	0.00	1.79	0.00	0.00	0.00	6.52	5.77	2.20	0.00	
19645	C3orf79	152118	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.16	0.00	2.05	0.00	0.00	
19646	CTD-2116N17.1	0	0.00	0.00	0.99	3.59	2.23	3.35	0.00	7.34	1.41	0.00	2.63	0.00	1.00	3.03	
19647	FUT2	2524	64.34	101.54	14.45	69.52	110.43	36.53	60.82	116.78	66.18	25.54	80.01	17.00	18.56	115.97	4

19648 rows × 52 columns

Original Health Genes Data frame

In here we have 19648 rows × 52 columns

```
In [5]: fhealthy = healthy[healthy.astype(bool).sum(1)>27]

In [6]: fcancer = cancer[healthy.astype(bool).sum(1)>27]

In [7]: fhealthy
```

Out[7]:

	Hugo_Symbol	Entrez_Gene_Id	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-22-5482	TCGA-56-8082	TCGA-22-5483	T
0	HIST3H2A	92815	62.12	130.60	33.06	35.50	73.03	60.39	92.05	66.65	54.33	15.56	55.49	30.34	14.45	1
1	LIN7B	64130	185.11	283.05	119.26	169.07	165.57	161.02	131.51	198.47	175.07	147.06	151.22	84.63	72.01	2
2	LXN	56925	909.17	819.30	412.00	743.43	1340.84	607.87	1709.26	1709.26	603.67	555.41	693.58	616.37	518.15	8
3	CNKSR2	22866	41.81	18.29	40.93	67.12	54.72	29.27	20.26	23.76	28.04	39.22	46.50	42.41	37.85	
4	SCML1	6322	133.36	214.27	108.14	109.66	190.34	211.31	96.01	208.38	120.10	239.52	124.37	249.73	103.69	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
19641	ZNF521	25925	215.77	148.09	83.45	232.94	167.90	80.01	161.02	104.42	81.14	113.56	220.32	81.71	85.22	1
19642	SPINT2	10653	6792.79	5441.30	5831.91	5329.30	5711.87	6164.49	8134.41	6516.03	8598.28	5633.22	5219.60	4937.99	6792.79	67
19643	HAVCR2	84868	423.61	529.06	660.68	620.67	518.15	848.22	366.09	1073.91	363.56	366.09	397.93	497.00	791.35	5
19646	CTD-2116N17.1	0	0.00	0.00	0.99	3.59	2.23	3.35	0.00	7.34	1.41	0.00	2.63	0.00	1.00	
19647	FUT2	2524	64.34	101.54	14.45	69.52	110.43	36.53	60.82	116.78	66.18	25.54	80.01	17.00	18.56	1

17626 rows × 52 columns

First filtration of Health Genes Data frame

After this stage we can see we now have 17626 rows × 52 columns

```
In [8]: fhealthy = fhealthy[cancer.astype(bool).sum(1)>27]

<ipython-input-8-0d5660575906>:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  fhealthy = fhealthy[cancer.astype(bool).sum(1)>27]

In [9]: fcancer = fcancer[cancer.astype(bool).sum(1)>27]

<ipython-input-9-86a562e1fbaa>:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  fcancer = fcancer[cancer.astype(bool).sum(1)>27]

In [10]: fhealthy
```

Out[10]:

	Hugo_Symbol	Entrez_Gene_Id	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-22-5482	TCGA-56-8082	TCGA-22-5483	T
0	HIST3H2A	92815	62.12	130.60	33.06	35.50	73.03	60.39	92.05	66.65	54.33	15.56	55.49	30.34	14.45	1
1	LIN7B	64130	185.11	283.05	119.26	169.07	165.57	161.02	131.51	198.47	175.07	147.06	151.22	84.63	72.01	2
2	LXN	56925	909.17	819.30	412.00	743.43	1340.84	607.87	1709.26	1709.26	603.67	555.41	693.58	616.37	518.15	8
3	CNKSR2	22866	41.81	18.29	40.93	67.12	54.72	29.27	20.26	23.76	28.04	39.22	46.50	42.41	37.85	
4	SCML1	6322	133.36	214.27	108.14	109.66	190.34	211.31	96.01	208.38	120.10	239.52	124.37	249.73	103.69	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
19641	ZNF521	25925	215.77	148.09	83.45	232.94	167.90	80.01	161.02	104.42	81.14	113.56	220.32	81.71	85.22	1
19642	SPINT2	10653	6792.79	5441.30	5831.91	5329.30	5711.87	6164.49	8134.41	6516.03	8598.28	5633.22	5219.60	4937.99	6792.79	67
19643	HAVCR2	84868	423.61	529.06	660.68	620.67	518.15	848.22	366.09	1073.91	363.56	366.09	397.93	497.00	791.35	5
19646	CTD-2116N17.1	0	0.00	0.00	0.99	3.59	2.23	3.35	0.00	7.34	1.41	0.00	2.63	0.00	1.00	
19647	FUT2	2524	64.34	101.54	14.45	69.52	110.43	36.53	60.82	116.78	66.18	25.54	80.01	17.00	18.56	1

17275 rows × 52 columns

Second filtration of Health Genes Data frame

- After this stage we can see we now have 17275 rows × 52 columns. So now we don't have any row that has more than 25 zeros and so we expect a more accurate calculation. This function returns in the index of the row false if there was more than 25 zeros and true if otherwise like we see next.

```
In [46]: healthy.astype(bool).sum(1)>27

Out[46]: 0         True
         1         True
         2         True
         3         True
         4         True
                  ...
         19643     True
         19644    False
         19645    False
         19646     True
         19647     True
         Length: 19648, dtype: bool
```

3. Now we want to figure out the relation between every gene expression in the 2 cases (healthy, cancer) so we want to calculate the CC (correlation coefficient) for each gene to know how each gene contribute in this cancer disease. So we make an empty list and fill it with every CC of each gene.

```
In [11]: CCo = []

In [12]: from scipy.stats import pearsonr
         for i in range (17275):
             G_h = fhealthy.iloc[i, 2:]
             G_c = fcancer.iloc[i, 2:]
             r, _ = pearsonr(G_h, G_c)
             CCo.append(r)

In [13]: CCo

Out[13]: [0.010398534580493182,
          0.10931300371692468,
          -0.07122138604098166,
          -0.010463207301235164,
          -0.12255635276175522,
          -0.19589472948698555,
          0.18725713692716683,
          0.10572392181655951,
          0.369500145408585,
          0.3887225251812886,
          -0.2512978545317218,
          0.13244390569390352,
          0.15857899701921213,
          0.2725131203669697,
          0.24537293600944343,
          -0.030734670252134173,
          0.2799049313122258,
          -0.19103085374626624,
          -0.04384717721672306,
          0.06105224727963827
```

4. We want to rank genes based on their correlation coefficient (CC) , so we add CC list as a column in both data frames the we sort our data frames ascendingly according to CC column.

```
In [47]: fhealthy.insert(2, "CCo", CCo , True)

In [48]: fhealthy
```

Out[48]:

	Hugo_Symbol	Entrez_Gene_Id	CCo	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-22-5482	TCGA-56-8082
0	HIST3H2A	92815	0.010399	62.12	130.60	33.06	35.50	73.03	60.39	92.05	66.65	54.33	15.56	55.49	30.34
1	LIN7B	64130	0.109313	185.11	283.05	119.26	169.07	165.57	161.02	131.51	198.47	175.07	147.06	151.22	84.63
2	LXN	56925	-0.071221	909.17	819.30	412.00	743.43	1340.84	607.87	1709.26	1709.26	603.67	555.41	693.58	616.37
3	CNKSR2	22866	-0.010463	41.81	18.29	40.93	67.12	54.72	29.27	20.26	23.76	28.04	39.22	46.50	42.41
4	SCML1	6322	-0.122556	133.36	214.27	108.14	109.66	190.34	211.31	96.01	208.38	120.10	239.52	124.37	249.73
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
17270	ZNF521	25925	0.059694	215.77	148.09	83.45	232.94	167.90	80.01	161.02	104.42	81.14	113.56	220.32	81.71
17271	SPINT2	10653	0.099033	6792.79	5441.30	5831.91	5329.30	5711.87	6164.49	8134.41	6516.03	8598.28	5633.22	5219.60	4937.99
17272	HAVCR2	84868	0.169032	423.61	529.06	660.68	620.67	518.15	848.22	366.09	1073.91	363.56	366.09	397.93	497.00
17273	CTD-2116N17.1	0	0.135677	0.00	0.00	0.99	3.59	2.23	3.35	0.00	7.34	1.41	0.00	2.63	0.00
17274	FUT2	2524	-0.047345	64.34	101.54	14.45	69.52	110.43	36.53	60.82	116.78	66.18	25.54	80.01	17.00

17275 rows × 53 columns

```
In [49]: fhealthy.sort_values('CCo')
```

Out[49]:

	Hugo_Symbol	Entrez_Gene_Id	CCo	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-22-548
12926	FAM222B	55731	-0.452807	285.03	214.27	336.79	295.11	312.00	206.94	295.11	214.27	283.05	281.09	329.8
13516	PTPRJ	5795	-0.424345	226.54	206.94	323.03	339.14	232.94	145.02	190.34	170.25	205.50	249.73	256.7
11353	ZFYVE20	64145	-0.418618	260.38	220.32	343.89	329.84	281.09	198.47	273.37	191.67	253.23	283.05	316.3
12324	VPRBP	9730	-0.416206	387.02	320.80	503.95	543.96	438.59	262.20	376.41	231.32	417.77	450.94	429.5
3789	S100A6	6277	-0.402969	59063.35	67377.47	69753.56	69271.73	55107.99	131982.68	80683.28	73730.83	79022.82	49322.93	59474.1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
14020	OVCH1-AS1	0	0.819990	14.67	117.60	3.14	85.22	12.09	0.73	5.15	8.65	30.56	4.06	7.7
16927	NUTM2E	0	0.826948	0.00	1.64	35.50	0.00	1.93	2.58	6.67	5.92	3.89	0.00	0.7
6565	MTRNR2L2	100462981	0.847577	4.31	7.00	7.00	1.04	3.72	56.28	4.10	4.62	47.17	8.51	2.1
5357	OR7D2	162998	0.930574	0.51	0.53	0.51	0.52	0.49	0.00	0.00	0.89	0.51	0.00	1.7
10790	AREGB	374	0.969044	23.76	89.51	17.00	7.75	12.55	39.22	33.78	4.62	29.48	96.01	10.6

This data frame can be found in ranked_healthy.csv

5. Then we can detect the highest positive CC and the lowest negative CC and the names of these two genes.

```
In [61]: Max_CCo = max(CCo)
         Max_CCo
Out[61]: 0.9690441442970706

In [62]: Min_CCo = min(CCo)
         Min_CCo
Out[62]: -0.4528072785247083
```

```
In [14]: Max_CCo = max(CCo)

In [15]: Min_CCo = min(CCo)

In [16]: max_index = CCo.index(Max_CCo)

In [17]: min_index = CCo.index(Min_CCo)

In [18]: max_index
Out[18]: 10790

In [19]: min_index
Out[19]: 12926

In [20]: len(CCo)
Out[20]: 17275
```

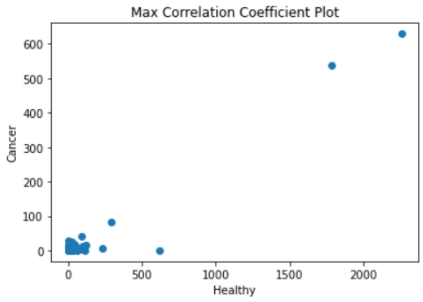
```
In [30]: fhealthy.iloc[ Max_index , : ]
Out[30]: Hugo_Symbol          AREGB
         Entrez_Gene_Id         374
         TCGA-43-7657         23.76
         TCGA-58-8386         89.51
         TCGA-22-5478            17
         TCGA-22-5472          7.75
         TCGA-43-5670         12.55
         TCGA-60-2709         39.22
         TCGA-22-5489         33.78
         TCGA-77-8007          4.62
         TCGA-22-5471         29.48
         TCGA-22-4609         96.01
         TCGA-22-5482         10.63
         TCGA-56-8082        111.99
         TCGA-22-5483          9.41
         TCGA-56-8623         45.21
         TCGA-33-4587        231.32
         TCGA-56-7579         11.82
         TCGA-43-3394         19.25
         TCGA-34-8454         13.42
         TCGA-77-7338          1.08
         TCGA-43-6143          8.85
         TCGA-43-6773         24.46
         TCGA-51-4080         13.03
         TCGA-34-7107         15.68
         TCGA-39-5040        2255.7
         TCGA-43-6771        102.97
         TCGA-92-7340        291.04
         TCGA-77-7138         23.08
         TCGA-77-7142          1.51
         TCGA-56-7823        620.67
         TCGA-22-5491          2.78
```

```
In [31]: fhealthy.iloc[ Min_index , : ]

Out[31]: Hugo_Symbol        FAM222B
         Entrez_Gene_Id       55731
         TCGA-43-7657        285.03
         TCGA-58-8386        214.27
         TCGA-22-5478        336.79
         TCGA-22-5472        295.11
         TCGA-43-5670           312
         TCGA-60-2709        206.94
         TCGA-22-5489        295.11
         TCGA-77-8007        214.27
         TCGA-22-5471        283.05
         TCGA-22-4609        281.09
         TCGA-22-5482        329.84
         TCGA-56-8082        220.32
         TCGA-22-5483        299.25
         TCGA-56-8623           248
         TCGA-33-4587        154.42
         TCGA-56-7579        267.73
         TCGA-43-3394        454.09
         TCGA-34-8454        256.78
         TCGA-77-7338         320.8
         TCGA-43-6143         269.6
         TCGA-43-6773        363.56
         TCGA-51-4080         320.8
         TCGA-34-7107        314.17
         TCGA-39-5040        131.51
         TCGA-43-6771        309.83
         TCGA-92-7340        314.17
         TCGA-77-7138         269.6
         TCGA-77-7142        289.02
```

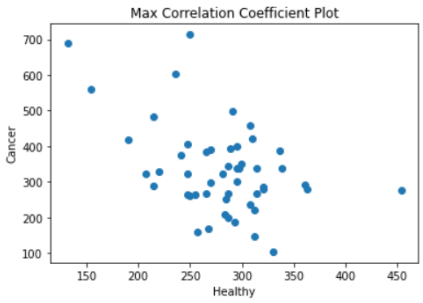
6. From here we can figure out that highest +ve CC = 0.969044 & lowest -ve CC = -0.452807 with Genes names as : AREGB , FAM222B Consecutively.
7. Now we want to plot these 2 genes in 2 graphs. We first take the gene that has the highest +ve CC between its Healthy gene expressions on the X-axis and cancer gene expressions on the y-axis and plot this graph

```
In [65]: plt.title('Max Correlation Coefficient Plot') #giving titles and labels specific names
         plt.xlabel('Healthy')
         plt.ylabel('Cancer')
         plt.scatter(fhealthy.iloc[Max_index, 3:] , fcancer.iloc[Max_index, 3:])
         plt.show()
```



using plot.scatter ,then we do the same thing for the gene that has the lowest -ve CC.

```
In [60]: plt.title('Min Correlation Coefficient Plot') #giving titles and labels specific names
         plt.xlabel('Healthy')
         plt.ylabel('Cancer')
         plt.scatter(fhealthy.iloc[Min_index, 3:] , fcancer.iloc[Min_index, 3:])
         plt.show()
```



Hypothesis test:

Before filtration: 1948*52

healthy

	Hugo_Symbol	Entrez_Gene_Id	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-22-5482	TCGA-56-8082	TCGA-22-5483	TCGA-56-8623	TCG
0	HIST3H2A	92815	62.12	130.60	33.06	35.50	73.03	60.39	92.05	66.65	54.33	15.56	55.49	30.34	14.45	100.83	131
1	LIN7B	64130	185.11	283.05	119.26	169.07	165.57	161.02	131.51	198.47	175.07	147.06	151.22	84.63	72.01	248.00	363
2	LXN	56925	909.17	819.30	412.00	743.43	1340.84	607.87	1709.26	1709.26	603.67	555.41	693.58	616.37	518.15	813.63	753
3	CNKSR2	22866	41.81	18.29	40.93	67.12	54.72	29.27	20.26	23.76	28.04	39.22	46.50	42.41	37.85	51.35	24
4	SCML1	6322	133.36	214.27	108.14	109.66	190.34	211.31	96.01	208.38	120.10	239.52	124.37	249.73	103.69	214.27	162
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19643	HAVCR2	84868	423.61	529.06	660.68	620.67	518.15	848.22	366.09	1073.91	363.56	366.09	397.93	497.00	791.35	503.95	547
19644	RP1-66C13.4	0	0.00	0.00	1.79	3.32	0.00	0.00	1.79	0.00	0.00	0.00	6.52	5.77	2.20	0.00	2
19645	C3orf79	152118	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.16	0.00	2.05	0.00	0.00	0
19646	CTD-2116N17.1	0	0.00	0.00	0.99	3.59	2.23	3.35	0.00	7.34	1.41	0.00	2.63	0.00	1.00	3.03	0
19647	FUT2	2524	64.34	101.54	14.45	69.52	110.43	36.53	60.82	116.78	66.18	25.54	80.01	17.00	18.56	115.97	42

19648 rows × 52 columns

After filtration:17275*52

healthy2

	Hugo_Symbol	Entrez_Gene_Id	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-22-5482	TCGA-56-8082	TCGA-22-5483	T
0	HIST3H2A	92815	62.12	130.60	33.06	35.50	73.03	60.39	92.05	66.65	54.33	15.56	55.49	30.34	14.45	1
1	LIN7B	64130	185.11	283.05	119.26	169.07	165.57	161.02	131.51	198.47	175.07	147.06	151.22	84.63	72.01	2
2	LXN	56925	909.17	819.30	412.00	743.43	1340.84	607.87	1709.26	1709.26	603.67	555.41	693.58	616.37	518.15	8
3	CNKSR2	22866	41.81	18.29	40.93	67.12	54.72	29.27	20.26	23.76	28.04	39.22	46.50	42.41	37.85	
4	SCML1	6322	133.36	214.27	108.14	109.66	190.34	211.31	96.01	208.38	120.10	239.52	124.37	249.73	103.69	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
19641	ZNF521	25925	215.77	148.09	83.45	232.94	167.90	80.01	161.02	104.42	81.14	113.56	220.32	81.71	85.22	1
19642	SPINT2	10653	6792.79	5441.30	5831.91	5329.30	5711.87	6164.49	8134.41	6516.03	8598.28	5633.22	5219.60	4937.99	6792.79	67
19643	HAVCR2	84868	423.61	529.06	660.68	620.67	518.15	848.22	366.09	1073.91	363.56	366.09	397.93	497.00	791.35	5
19646	CTD-2116N17.1	0	0.00	0.00	0.99	3.59	2.23	3.35	0.00	7.34	1.41	0.00	2.63	0.00	1.00	
19647	FUT2	2524	64.34	101.54	14.45	69.52	110.43	36.53	60.82	116.78	66.18	25.54	80.01	17.00	18.56	1

17275 rows × 52 columns

We observed that after applying FDR correction, p-values increase which reduce type 1 error (false positive), as the value may exceed significance level after correction however being in rejection region before it.

significance_genes_i

	p-values-independent	p-values_i_fdr
0	3.607140e-09	1.373146e-08
1	3.138295e-01	3.634380e-01
2	8.164044e-05	1.718039e-04
3	6.374652e-15	5.016953e-14
4	4.726590e-02	6.500943e-02
...	...	...
17270	2.273493e-06	5.975140e-06
17271	5.250215e-08	1.714832e-07
17272	1.228186e-14	9.305663e-14
17273	1.068283e-12	6.277074e-12
17274	2.133666e-07	6.428162e-07

17275 rows × 2 columns

significance_genes_r

	p-values-paired	p-values_r_fdr
0	4.043607e-08	1.448337e-07
1	2.891646e-01	3.361135e-01
2	2.322367e-04	4.579260e-04
3	3.420577e-12	2.445798e-11
4	6.251346e-02	8.352696e-02
...	...	...
17270	4.142164e-06	1.081558e-05
17271	2.452619e-07	7.755627e-07
17272	2.435125e-13	2.159486e-12
17273	4.129496e-11	2.418205e-10
17274	1.166719e-06	3.324822e-06

17275 rows × 2 columns

And here are some examples of p-values that probably lie in acceptance (fail to reject) region instead of rejection after FDR correction.

```
significance_genes_i['significance:p_vlaue_i'] = significance_genes_i['p-values-independent'].apply(lambda x: x < 0.05)
significance_genes_i['significance:p_vlaue_i_fdr'] = significance_genes_i['p-values_i_fdr'].apply(lambda x: x < 0.05)
significance_genes_i
```

	p-values-independent	p-values_i_fdr	significance:p_vlaue_i	significance:p_vlaue_i_fdr
0	3.607140e-09	1.373146e-08	True	True
1	3.138295e-01	3.634380e-01	False	False
2	8.164044e-05	1.718039e-04	True	True
3	6.374652e-15	5.016953e-14	True	True
4	4.726590e-02	6.500943e-02	True	False
...	...	...	...	...
17270	2.273493e-06	5.975140e-06	True	True
17271	5.250215e-08	1.714832e-07	True	True
17272	1.228186e-14	9.305663e-14	True	True
17273	1.068283e-12	6.277074e-12	True	True
17274	2.133666e-07	6.428162e-07	True	True

17275 rows × 4 columns

```
significance_genes_i['significance:p_vlaue_i'] = significance_genes_i['p-values-independent'].apply(lambda x: x < 0.05)
significance_genes_i['significance:p_vlaue_i_fdr'] = significance_genes_i['p-values_i_fdr'].apply(lambda x: x < 0.05)
significance_genes_i
```

	p-values-independent	p-values_i_fdr	significance:p_vlaue_i	significance:p_vlaue_i_fdr
0	3.607140e-09	1.373146e-08	True	True
1	3.138295e-01	3.634380e-01	False	False
2	8.164044e-05	1.718039e-04	True	True
3	6.374652e-15	5.016953e-14	True	True
4	4.726590e-02	6.500943e-02	True	False
...	...	...	...	...
17270	2.273493e-06	5.975140e-06	True	True
17271	5.250215e-08	1.714832e-07	True	True
17272	1.228186e-14	9.305663e-14	True	True
17273	1.068283e-12	6.277074e-12	True	True
17274	2.133666e-07	6.428162e-07	True	True

17275 rows × 4 columns

Then number of genes in rejection region **decrease** after FDR correction. So, the genes which remained in rejection region (significance = true) are the actual affected genes

```
diffrentially_genes_i = significance_genes_i[significance_genes_i['significance:p_vlaue_i_fdr']== True]
diffrentially_genes_i
```

	p-values-independent	p-values_i_fdr	significance:p_vlaue_i	significance:p_vlaue_i_fdr
0	3.607140e-09	1.373146e-08	True	True
2	8.164044e-05	1.718039e-04	True	True
3	6.374652e-15	5.016953e-14	True	True
6	5.344289e-06	1.333949e-05	True	True
7	7.857877e-06	1.917029e-05	True	True
...	...	...	...	...
17270	2.273493e-06	5.975140e-06	True	True
17271	5.250215e-08	1.714832e-07	True	True
17272	1.228186e-14	9.305663e-14	True	True
17273	1.068283e-12	6.277074e-12	True	True
17274	2.133666e-07	6.428162e-07	True	True

12290 rows × 4 columns

```
diffrentially_genes_r = significance_genes_r[significance_genes_r['significance:p_vlaue_r_fdr']== True]
diffrentially_genes_r
```

	p-values-paired	p-values_r_fdr	significance:p_vlaue_r	significance:p_vlaue_r_fdr
0	4.043607e-08	1.448337e-07	True	True
2	2.322367e-04	4.579260e-04	True	True
3	3.420577e-12	2.445798e-11	True	True
6	3.041721e-06	8.115171e-06	True	True
7	1.938575e-05	4.547030e-05	True	True
...	...	...	...	...
17270	4.142164e-06	1.081558e-05	True	True
17271	2.452619e-07	7.755627e-07	True	True
17272	2.435125e-13	2.159486e-12	True	True
17273	4.129496e-11	2.418205e-10	True	True
17274	1.166719e-06	3.324822e-06	True	True

12380 rows × 4 columns

After the correction besides determining common genes between independent and paired samples, we notice that there are a large number of genes are placed in RR.

	intersected
0	IFIT5
1	FGF18
2	PLXNA3
3	SLC16A2
4	CHPF2
...	...
12206	DSC2
12207	TCTEX1D1
12208	MRC2
12209	NFAM1
12210	ZMAT4

12211 rows × 1 columns

For distinct genes, paired ones are more than independent ones.

```
diff_ri = list(diff_ri)
diff_ri_genes = pd.DataFrame({'diff_ri':diff_ri})
diff_ri_genes
```

	diff_ri
0	DHRS4-AS1
1	RMND1
2	CAPN10
3	IKZF2
4	KLHDC9
...	...
164	PLGLB2
165	RABEP2
166	C20orf201
167	ADAM28
168	HEY1

169 rows × 1 columns

```
diff_ir = list(diff_ir)
diff_ir_genes = pd.DataFrame({'diff_ir':diff_ir})
diff_ir_genes
```

	diff_ir
0	NR2C2
1	SLC22A14
2	EVPL
3	RPS6KA6
4	LAMC1
...	...
74	SLC29A1
75	PGPEP1L
76	SHISA9
77	GM2A
78	GJA9

79 rows × 1 columns

Conclusion:

From Correlation:-

Correlation coefficient formulas are used to find how strong a relationship is between each gene in data frame It shows the linear relationship between two sets of data. In simple terms, it answers the question; *can I draw a line graph to represent the data?* The formulas return a value between -1 and 1, where:

1 indicates a strong positive relationship.
-1 indicates a strong negative relationship.
A result of zero indicates no relationship at all.
So AREGB gene has the strongest positive relationship which means that it's not responsible for the cancer disease, and FAM222B has the lowest negative relationship which means that it highly contributes in the presence of this disease.

We can also figure out this conclusion from the graphs we plotted. In the plot of the highest +ve CC we can find that the best fit line would have a positive slope which means we have strong positive relationship (CC between 0 and 1) between health and cancerous gene, also we can see that all the data of gene expressions are so close to the fit line more than any other gene and not too scattered and this is why this gene has the highest CC because as we know the more the scattering along y-axis compared to scattering along x-axis the stronger the relationship is.

In the plot of the highest -ve CC we can find that the best fit line would have a negative slope which means we have strong negative relationship (CC between 0 and -1) between health and cancerous gene, also we can see that all the data of gene expressions are scattered around the fit line more than any other gene and not too close and this is why this gene has the lowest CC because as we know the more the scattering along the x-axis compared to scattering along y-axis the inversely stronger the relationship is.

In both cases we drew the healthy gene expressions (as it's considered the predictor) along x-axis and the cancerous gene expressions (the response) along y-axis.

From Hypothesis test:-

Through this paper we deduced that sometimes a value is considered in rejection region (unusual event) although if we were more precise it should be located in fail to reject region (usual event) and this situation could be fixed by applying a proper correction method like FDR.

Contribution

task	Mariam Ashraf	Marwa AdbelAal	Nada Ezzat	Noura Mahmoud
Filtration method	**50%**	**----**	**----**	**50%**
Correlation method	**100%**	**----**	**----**	**----**
Correlation documentation	**100%**	**----**	**----**	**----**
Hypothesis test	**----**	**35%**	**30%**	**35%**
FDR correction	**----**	**35%**	**35%**	**30%**
Hypothesis test_ documentation	**----**	**30%**	**35%**	**35%**
Gathering _ documentation	**50%**	**50%**	**---**	**---**