Cairo university
Biomedical engineering biostatistics research

Abstract- The upcoming research article indicates both the correlation coefficient of a sample of genes found In human bodies which was found by Hugo gene and will be used to get the correlation of the dataset and its hypothesis test

Index Terms- About four key words or phrases in alphabetical order, separated by commas. Keywords are used to retrieve documents in an information system such as correlation, hypothesis, genes.

I. Introduction

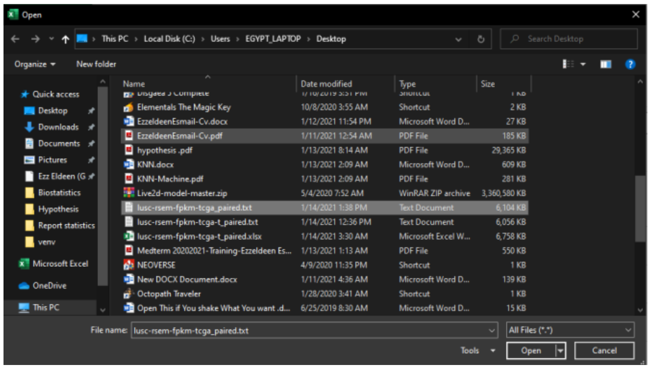
This article guides a stepwise walkthrough by biomedical engineering Cairo university for finding the correlation coefficient of a sample. In addition to getting the hypothesis test of that sample of genes. Now, here we enlist the proven steps to publish the research paper in a journal.

Identify the constructs of a Research – Essentially a Research consists of three major sections. The import of the data into the python files program used as Vs code or PyCharm. The correlation coefficient, the hypothesis test

II. Importing the data into text editor [correlation]

A. *implementing the data into excel file*

First the two samples were downloaded then opened as excel files



Open

This PC › Local Disk (C:) › Users › EGYPT_LAPTOP › Desktop

Name	Date modified	Type	Size
Disgaea 5 Complete	7/10/2019 3:51 PM	Shortcut	1 KB
Elementals The Magic Key	10/8/2020 3:55 AM	Shortcut	2 KB
EzzeldeenEsmail-Cv.docx	1/12/2021 11:54 PM	Microsoft Word D...	27 KB
EzzeldeenEsmail-Cv.pdf	1/11/2021 12:54 AM	PDF File	185 KB
hypothesis .pdf	1/13/2021 8:14 AM	PDF File	29,365 KB
KNN.docx	1/13/2021 2:09 AM	Microsoft Word D...	609 KB
KNN-Machine.pdf	1/13/2021 2:09 AM	PDF File	281 KB
Live2d-model-master.zip	5/4/2020 7:52 AM	WinRAR ZIP archive	3,360,580 KB
lusc-rsem-fpkm-tcga_paired.txt	1/14/2021 1:38 PM	Text Document	6,104 KB
lusc-rsem-fpkm-tcga-t_paired.txt	1/14/2021 12:36 PM	Text Document	6,056 KB
lusc-rsem-fpkm-tcga-t_paired.xlsx	1/14/2021 3:30 AM	Microsoft Excel W...	6,758 KB
Medterm 20202021-Training-Ezzeldeen Es...	1/13/2021 1:13 AM	PDF File	550 KB
NEOVERSE	4/9/2020 11:35 PM	Shortcut	1 KB
New DOCX Document.docx	1/11/2021 4:36 AM	Microsoft Word D...	139 KB
Octopath Traveler	1/28/2020 3:41 AM	Shortcut	1 KB
Open This if You shake What You want .d...	6/25/2019 8:30 AM	Microsoft Word D...	15 KB

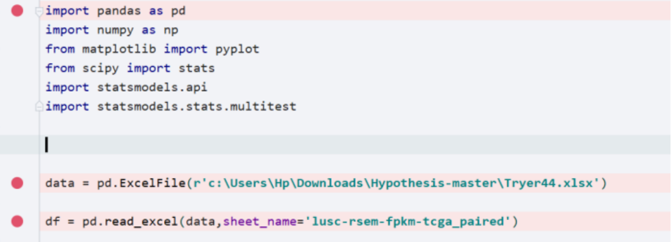
File name: lusc-rsem-fpkm-tcga_paired.txt

All Files (*.*)

After that we had to go through some research of how to implement the data into PyCharm or Vs code so we found a similar question in Stack overflow that uses two libraries pandas or Xlsx writer, and we decided to go with the pandas library.

B. *coding for reading the data*

We then used the pandas to read the excel file that was made

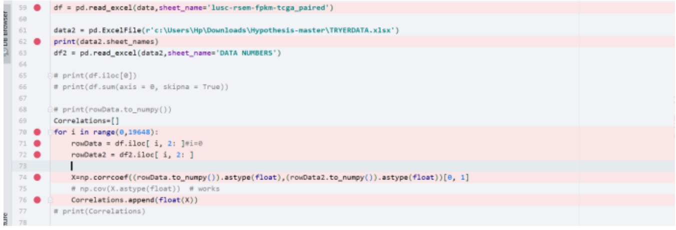


```python
import pandas as pd
import numpy as np
from matplotlib import pyplot
from scipy import stats
import statsmodels.api
import statsmodels.stats.multitest


data = pd.ExcelFile(r'c:\Users\Hp\Downloads\Hypothesis-master\Tryer44.xlsx')

df = pd.read_excel(data,sheet_name='lusc-rsem-fpkm-tcga_paired')
```

Were data and df is the reader of the excel files and done the same thing for the two excel files.

C. *Calculating the correlation*

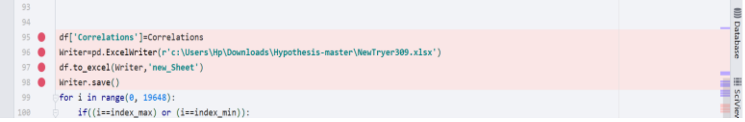
We searched and found out we could use two methods for calculating the correlation either Pearson or we could use spearman and for simplified usage we used NumPy library which is embedded as Pearson and by using a for loop that iterates over all rows of both excel files we could calculate the correlation



```python
df = pd.read_excel(data,sheet_name='lusc-rsem-fpkm-tcga_paired')

data2 = pd.ExcelFile(r'c:\Users\Hp\Downloads\Hypothesis-master\TRYERDATA.xlsx')
print(data2.sheet_names)
df2 = pd.read_excel(data2,sheet_name='DATA NUMBERS')

# print(df.iloc[0])
# print(df.sum(axis = 0, skipna = True))

# print(rowData.to_numpy())
Correlations=[]
for i in range(0,19648):
    rowData = df.iloc[ i, 2: ]#i=0
    rowData2 = df2.iloc[ i, 2: ]

    X=np.corrcoef((rowData.to_numpy()).astype(float),(rowData2.to_numpy()).astype(float))[0, 1]
    # np.cov(X.astype(float))  # works
    Correlations.append(float(X))
# print(Correlations)
```

D. *Writing new excel*

And by the writer function implemented in pandas we could save a new file with correlation of each gene.

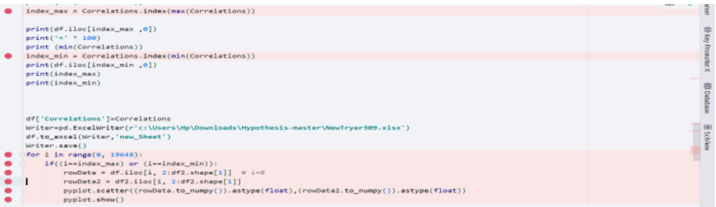


```python


df['Correlations']=Correlations
Writer=pd.ExcelWriter(r'c:\Users\Hp\Downloads\Hypothesis-master\NewTryer309.xlsx')
df.to_excel(Writer,'new_Sheet')
Writer.save()
for i in range(0, 19648):
    if((i==index_max) or (i==index_min)):
```

A. *Plotting of the max and min*

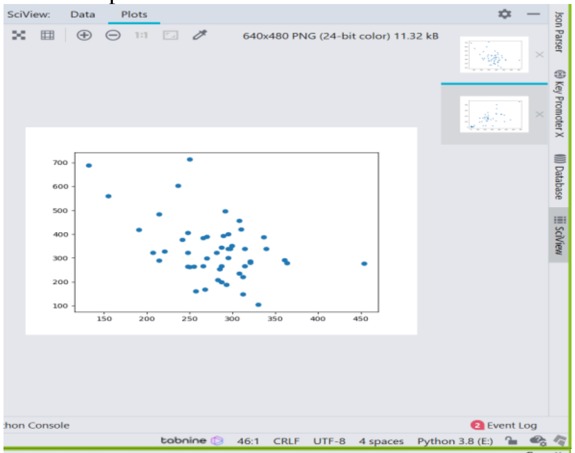
The plotting was used as the graph library pyplot which was imported then used for the max and min which was detected by its functions

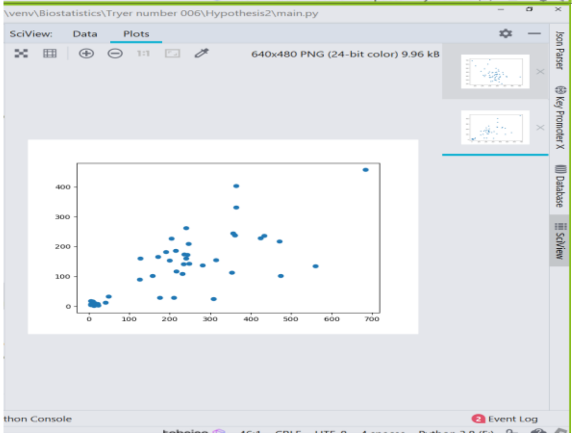
.



```python
index_max = Correlations.index(max(Correlations))

print(df.iloc[index_max ,0])
print('=' * 100)
print (min(Correlations))
index_min = Correlations.index(min(Correlations))
print(df.iloc[index_min ,0])
print(index_max)
print(index_min)



df['Correlations']=Correlations
Writer=pd.ExcelWriter(r'c:\Users\Hp\Downloads\Hypothesis-master\NewTryer309.xlsx')
df.to_excel(Writer,'new_Sheet')
Writer.save()
for i in range(0, 19648):
    if((i==index_max) or (i==index_min)):
        rowData = df.iloc[i, 2:df2.shape[1]]  # i=0
        rowData2 = df2.iloc[i, 2:df2.shape[1]]
        pyplot.scatter((rowData.to_numpy()).astype(float),(rowData2.to_numpy()).astype(float))
        pyplot.show()
```

Which its plotting can be seen here
Note :the plot was done after the filter

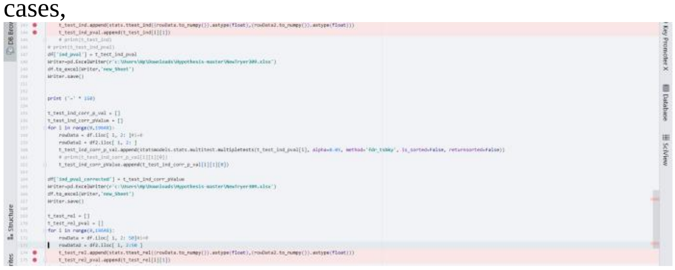




III. hypothesis

A. *t-test in both cases [independent-paired]*

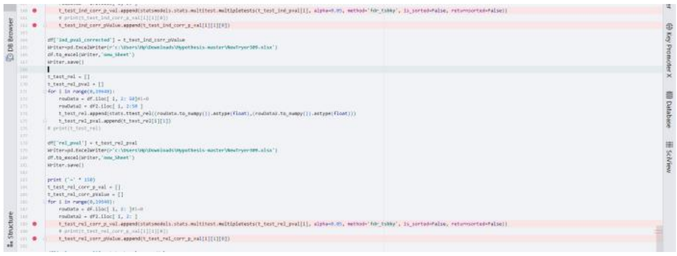
we have two cases first if the data is independent means they have nothing to do with each other or if the data itself is not independent and we are actually having the same data but after a change for example [Drug effects on mice] so we may have 2 different mice types [independent] or we have the same mice type and we check the drug effects on the same sample.in both cases,



```
    t_test_ind.append(stats.ttest_ind((rowData.to_numpy()).astype(float),(rowData2.to_numpy()).astype(float)))
    t_test_ind_pval.append(t_test_ind[i][1])
    # print(t_test_ind)
# print(t_test_ind_pval)
df['ind_pval'] = t_test_ind_pval
Writer=pd.ExcelWriter(r'c:\Users\Hp\Downloads\Hypothesis-master\NewTryer309.xlsx')
df.to_excel(Writer,'new_Sheet')
Writer.save()


print ('-' * 150)

t_test_ind_corr_p_val = []
t_test_ind_corr_pValue = []
for i in range(0,19648):
    rowData = df.iloc[ i, 2: ]#i=0
    rowData2 = df2.iloc[ i, 2: ]
    t_test_ind_corr_p_val.append(statsmodels.stats.multitest.multipletests(t_test_ind_pval[i], alpha=0.05, method='fdr_tsbky', is_sorted=False, returnsorted=False))
    # print(t_test_ind_corr_p_val[i][1][0])
    t_test_ind_corr_pValue.append(t_test_ind_corr_p_val[i][1][0])

df['ind_pval_corrected'] = t_test_ind_corr_pValue
Writer=pd.ExcelWriter(r'c:\Users\Hp\Downloads\Hypothesis-master\NewTryer309.xlsx')
df.to_excel(Writer,'new_Sheet')
Writer.save()

t_test_rel = []
t_test_rel_pval = []
for i in range(0,19648):
    rowData = df.iloc[ i, 2: 50]#i=0
    rowData2 = df2.iloc[ i, 2:50 ]
    t_test_rel.append(stats.ttest_rel((rowData.to_numpy()).astype(float),(rowData2.to_numpy()).astype(float)))
    t_test_rel_pval.append(t_test_rel[i][1])
```

B. *FDR regulation*

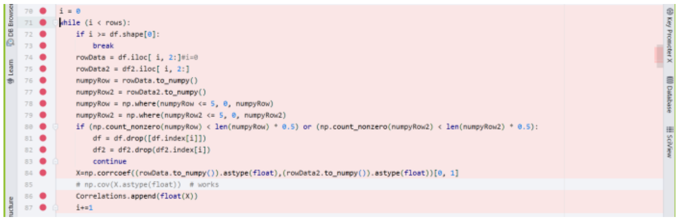
but we had to regulate our P-values because of false positive numbers[the out numbers that is way out of mean that caused the problem itself and rejected the null hypothesis]so we applied the FDR regulation method in both cases which can be seen here



```
    t_test_ind_corr_p_val.append(statsmodels.stats.multitest.multipletests(t_test_ind_pval[i], alpha=0.05, method='fdr_tsbky', is_sorted=False, returnsorted=False))
    # print(t_test_ind_corr_p_val[i][1][0])
    t_test_ind_corr_pValue.append(t_test_ind_corr_p_val[i][1][0])

df['ind_pval_corrected'] = t_test_ind_corr_pValue
Writer=pd.ExcelWriter(r'c:\Users\Hp\Downloads\Hypothesis-master\NewTryer309.xlsx')
df.to_excel(Writer,'new_Sheet')
Writer.save()

t_test_rel = []
t_test_rel_pval = []
for i in range(0,19648):
    rowData = df.iloc[ i, 2: 50]#i=0
    rowData2 = df2.iloc[ i, 2:50 ]
    t_test_rel.append(stats.ttest_rel((rowData.to_numpy()).astype(float),(rowData2.to_numpy()).astype(float)))
    t_test_rel_pval.append(t_test_rel[i][1])
# print(t_test_rel)

df['rel_pval'] = t_test_rel_pval
Writer=pd.ExcelWriter(r'c:\Users\Hp\Downloads\Hypothesis-master\NewTryer309.xlsx')
df.to_excel(Writer,'new_Sheet')
Writer.save()

print ('-' * 150)
t_test_rel_corr_p_val = []
t_test_rel_corr_pValue = []
for i in range(0,19648):
    rowData = df.iloc[ i, 2: ]#i=0
    rowData2 = df2.iloc[ i, 2: ]
    t_test_rel_corr_p_val.append(statsmodels.stats.multitest.multipletests(t_test_rel_pval[i], alpha=0.05, method='fdr_tsbky', is_sorted=False, returnsorted=False))
    # print(t_test_rel_corr_p_val[i][1][0])
    t_test_rel_corr_pValue.append(t_test_rel_corr_p_val[i][1][0])
```

IV. Filter

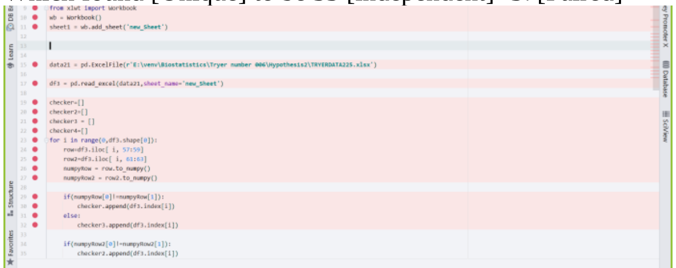
As it was seen from the data , that Some of its rows aren't accurate in addition to having a lot of zeros as well so it was seen after the plotting that there was error in the max and min plot so a filter was needed for the zeros and low data values <5 So a filter was made based on these principles which can be seen here.



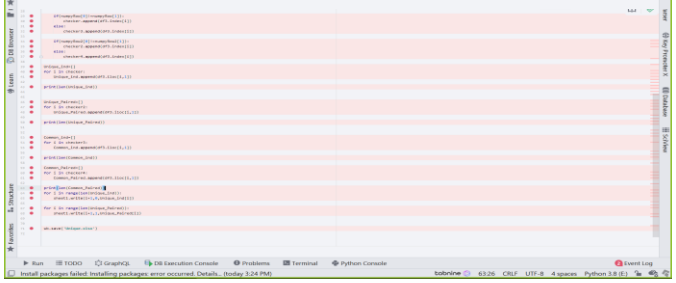
```
i = 0
while (i < rows):
    if i >= df.shape[0]:
        break
    rowData = df.iloc[ i, 2:]#i=0
    rowData2 = df2.iloc[ i, 2:]
    numpyRow = rowData.to_numpy()
    numpyRow2 = rowData2.to_numpy()
    numpyRow = np.where(numpyRow <= 5, 0, numpyRow)
    numpyRow2 = np.where(numpyRow2 <= 5, 0, numpyRow2)
    if (np.count_nonzero(numpyRow) < len(numpyRow) * 0.5) or (np.count_nonzero(numpyRow2) < len(numpyRow2) * 0.5):
        df = df.drop([df.index[i]])
        df2 = df2.drop(df2.index[i])
        continue
    X=np.corrcoef((rowData.to_numpy()).astype(float),(rowData2.to_numpy()).astype(float))[0, 1]
    # np.cov(X.astype(float))  # works
    Correlations.append(float(X))
    i+=1
```

V. unique [distinct] and common genes

Our main file that we made , had all the data that of p-value as a value and the rejection value [True/False] so we found it better to make another python file [main2.py] which will store this [True/false] data and make a counter that calculate their numbers , then made the if statement to identify our unique genes from the common ones, and in the end we stored the data in another excel file which we called [Unique.xlsx]
Which found [Unique] to be 33 [independent]- 37[Paired]



```
from xlwt import Workbook
wb = Workbook()
sheet1 = wb.add_sheet('new_Sheet')



data21 = pd.ExcelFile(r'E:\venv\Biostatistics\Tryer number 006\Hypothesis2\TRYERDATA225.xlsx')

df3 = pd.read_excel(data21,sheet_name='new_Sheet')

checker=[]
checker2=[]
checker3 = []
checker4=[]
for i in range(0,df3.shape[0]):
    row=df3.iloc[ i, 57:59]
    row2=df3.iloc[ i, 61:63]
    numpyRow = row.to_numpy()
    numpyRow2 = row2.to_numpy()

    if(numpyRow[0]!=numpyRow[1]):
        checker.append(df3.index[i])
    else:
        checker3.append(df3.index[i])

    if(numpyRow2[0]!=numpyRow2[1]):
        checker2.append(df3.index[i])
```



```
    if(numpyRow[0]!=numpyRow[1]):
        checker.append(df3.index[i])
    else:
        checker3.append(df3.index[i])

    if(numpyRow2[0]!=numpyRow2[1]):
        checker2.append(df3.index[i])
    else:
        checker4.append(df3.index[i])

Unique_Ind=[]
for i in checker:
    Unique_Ind.append(df3.iloc[i,1])

print(len(Unique_Ind))


Unique_Paired=[]
for i in checker2:
    Unique_Paired.append(df3.iloc[i,1])

print(len(Unique_Paired))


Common_Ind=[]
for i in checker3:
    Common_Ind.append(df3.iloc[i,1])

print(len(Common_Ind))

Common_Paired=[]
for i in checker4:
    Common_Paired.append(df3.iloc[i,1])

print(len(Common_Paired))
for i in range(len(Unique_Ind)):
    sheet1.write(i+1,0,Unique_Ind[i])

for i in range(len(Unique_Paired)):
    sheet1.write(i+1,1,Unique_Paired[i])


wb.save('Unique.xlsx')
```

Install packages failed: Installing packages: error occurred. Details... (today 3:24 PM)

IV. Some Common Mistakes

- The first mistake was the Filter the last research numbers weren’t wrong but not accurate as there were some data that needed filtration these data which was used for coding and excel files were all before filtration so the team had to calculate all the assumptions from the start using the filtered data which was about 4000 less in numbers than last data as these 4000 number had more 0 than 50 % than the normal data
- Another mistake was the FDR regulation the regulations had different types [methods] which could affect P in different ways based on there principles.
- Another mistake was the difference between our assumption [I] value and the df.index[I] as all our rows index was first shifted.
- The filter numbers had different index than the rows as the rows drops its max number decreases so we calculated them and made a break function for it when it reaches its max I after the drop it will stop the loop.

References[used]

1) https://www.genenames.org/

2) https://pandas.pydata.org/

3) https://stackabuse.com/calculating-pearson-correlation-coefficient-in-python-with-numpy/#:~:text=The%20Pearson%20Correlation%20coefficient%20can,sample%20of%20n%20random%20variables

4) https://numpy.org/

5) https://www.w3schools.com/python/python_ml_scatterplot.asp

6) https://stackoverflow.com/questions/57648069/choosing-pandas-over-xlsxwriter-when-working-with-excel-data

7) https://www.statsmodels.org/dev/generated/statsmodels.stats.multitest.multipletests.html

8) https://stackoverflow.com/questions/11788900/importerror-no-module-named-statsmodels

9) https://stackoverflow.com/questions/49814258/statsmodel-attributeerror-module-scipy-stats-has-no-attribute-chisqprob

10) https://stackoverflow.com/questions/24808043/importerror-no-module-named-scipy

11) https://stackoverflow.com/questions/45670487/numpy-cov-exception-float-object-has-no-attribute-shape

12) https://www.geeksforgeeks.org/writing-excel-sheet-using-python/

13) https://www.statsmodels.org/dev/generated/statsmodels.stats.multitest.multipletests.html

14) https://stackoverflow.com/questions/8364674/how-to-count-the-number-of-true-elements-in-a-numpy-bool-array

15) https://www.javaer101.com/en/article/931010.html [0,1]

16) https://www.shanelynn.ie/select-pandas-dataframe-rows-and-columns-using-iloc-loc-and-ix/ iloc

17) https://stackoverflow.com/questions/45670487/numpy-cov-exception-float-object-has-no-attribute-shape astype[‘float’]

18) https://www.w3schools.com/python/numpy_array_sort.asp sort

19) https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html

20) https://stackoverflow.com/questions/24808043/importerror-no-module-named-scipy

21) https://stackoverflow.com/questions/49814258/statsmodel-attributeerror-module-scipy-stats-has-no-attribute-chisqprob

22) https://stackoverflow.com/questions/11788900/importerror-no-module-named-statsmodels

23) https://www.statsmodels.org/dev/generated/statsmodels.stats.multitest.multipletests.html

group names:

Name	Section	BN
Ezzeldden esmail ezzeldeen	**1**	**50**
Yehia osama mohammed	**2**	**47**
Abdallah tamer reda	**1**	**48**
Ahmed Hossam mohammed	**1**	**2**

Participation	
Ezzeldeen esmail	Import +correlation
Abdallah tamer	Filter + plotting
Yehia and Ahmed Hossam	Hypothesis
all team	Distinct + common genes