

Python Script 1

Python Script 1.0 (By Jing)

```
In [ ]: fin = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/EB++AdjustPANCAN_IlluminaHiSeq_RNASeqV2.geneExp.xena', 'r')
fout = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_1/sample_t.gct', 'w')

counter =0

K = 10000 #100, 1000, 10,000
columnN = K
rowN = 0
all_values =[]

while 1:
    line = fin.readline()
    counter = counter +1

    #process line
    values = line.split('\t')

    values = values[0:K+1]
    gene = values[0]
    values.insert(1, gene)
    all_values.append(values)

    if line == '':
        break

rowN = counter -1

#write to output
fout.write("#1.2\n")
fout.write(str(rowN)+'\t'+str(columnN)+'\n')
fout.write('Name\tDescription\t'+ '\t'.join(all_values[0][2:]) + '\n')
for list in all_values[1:]:
    fout.write('\t'.join(list) + '\n' )

fin.close()
fout.close()
```

Python Script 1.1

Creating .gct file

```
In [3]: fin = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/EB++AdjustPANCAN_IlluminaHiSeq_RNASeqV2.geneExp.xena', 'r')
fout = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_1/sample_10000.gct', 'w')

all_values = []
K = 10000 #100 500 1000 2000 10000
index = 0
for pos, value in enumerate(fin):
    values = value.split()
    values = values[0:K+1]
    gene = values[0]
    values.insert(1, gene)
    all_values.append(values)
    index += 1

rowN = index - 1
fout.write("#1.2\n")
fout.write(str(rowN)+'\t'+str(K)+'\n')
fout.write('Name\tDescription\t'+ '\t'.join(all_values[0][2:]) + '\n')
for list in all_values[1:]:
    fout.write('\t'.join(list) + '\n' )

print(rowN)
fin.close()
fout.close()
```

20531

Creating .cls file

In [4]:

```
def read(fin,a,b): # a, b represents the proportion of phenotypes
    index,pointer = 0 , 0 #index to read the third line in the file
    for x in fin:
        index += 1
        if index == 3:
            value = x.split() #process the tab delimited line
            value1 = value[2:]
            length = len(value1)
            for x in range(len(value1)):
                if x < (length * (a/100)):
                    value1[x] = 0
                else:
                    value1[x] = 1
            string_value = ""
            for pos,x in enumerate(value1):
                if pos != len(value1) - 1: #updated line
                    string_value += (str(x) + '\t')
                else: string_value += str(x) #updated line
            return string_value

fin = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_1/sample_10000.gct', 'r')
fout = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_1/sample_10000.cls', 'w')
sample = read(fin,50,50)
No_class = 2;
fout.write(str(K)+'\t'+ str(No_class)+'\t'+ str(1)+'\n')
fout.write('#'+'\t'+ 'Phen_A'+'\t'+ 'Phen_B'+'\n')
fout.write(sample)

fin.close()
fout.close()
```

Python Script 2

Creating .gct file

In [11]:

```
file1_content = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/TCGA.BRCA.sampleMap_HiSeqV2",'r')
file2_content = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/BasalLum.tsv", 'r')
value1 = file1_content.readline(-1)
sample = value1.split()[1:] # fist line of the expression_file
sample1 = [] #contains all the gene sample data
for pos,value in enumerate(file2_content):
    if pos > 0:
        val = value.split()
        sample1.append(val[0])
common_sample = []
common_sample_column = []
index = 0
for x in range(len(sample)):
    for y in range(len(sample1)):
        if sample1[y] == sample[x]:
            common_sample.append(sample[x])
            common_sample_column.append(x)

gene_value = ""
row = 0

for pos,value in enumerate(file1_content):
    if pos >= 0:
        row += 1
        gene_name = value.split()[0]
        gene_value += (gene_name + '\t' + gene_name+ '\t')
        value_x = value.split()[1:]
        for pos,valx in enumerate(common_sample_column):
            if pos == len(common_sample_column) - 1: #updated
                gene_value += (value_x[valx])
            else: gene_value += (value_x[valx] + '\t')
        if row != 20530:
            gene_value += '\n'

common_sample_final = ""
for pos,value in enumerate(common_sample):
    if pos != len(common_sample) - 1: #updated
        common_sample_final += value + '\t'
    else:
        common_sample_final += value

b = str(len(common_sample_column))

fout = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_2/sample_500.gct', 'w')
fout.write("#1.2\n")
fout.write(str(row) + '\t'+ b +'\n')
fout.write('Name\tDescription\t'+ common_sample_final+ '\n')
fout.write(gene_value)
#print(row)
#print(b)
file1_content.close()
fout.close()
```

Creating .cls file

In [12]:

```
fin = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_2/sample_500.gct', 'r')
tsv_file = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/BasalLum.tsv', 'r')

classN = []
sample_phenotype = [x.split() for pos,x in enumerate(tsv_file) if pos > 0]
class_counter = 0
for pos,value in enumerate(fin):
    if pos == 2:
        sample_value = value.split()[2:]
        index = 0
        for x in sample_value:
            for y in sample_phenotype:
                if x == y[0] and index == 0:
                    classN.append((y[1],class_counter))
                    index += 1
                if x == y[0]:
                    if y[1] not in [i[0] for i in classN]:
                        class_counter += 1
                        classN.append((y[1],class_counter))
for x in sample_value:
    for y in sample_phenotype:
        if x == y[0]:
            for lum_basal in classN:
                if lum_basal[0] == y[1]:
                    valuex = [lum_basal[0],lum_basal[1]]
                    sample_value[sample_value.index(x)] = [x, valuex[0], valuex[1]]

listx = [x[2] for x in sample_value]
class_label = ""
for pos,x in enumerate(listx):
    if pos != len(listx) - 1: #updated
        class_label += str(x) + '\t'
    else: class_label += str(x)

print(classN[0][0], classN[1][0])

fout = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_2/sample_500.cls', 'w')

fout.write(str(len(sample_value))+'\t'+ str(len(classN))+'\t'+ str(1)+'\n')
fout.write('#'+'\t'+str(classN[0][0]) +'\t'+str(classN[1][0]) +'\n')
fout.write(class_label)

fin.close()
fout.close()
```

Lum Basal

Python Script 3

Creating .gct file

In [13]:

```

file1_content = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/TCGA.BRCA.sampleMap_HiSeqV2", 'r')
file2_content = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/ESR1.tsv", 'r')
value1 = file1_content.readline(-1)
sample = value1.split()[1:] # fist line of the expression_file
sample1 = [] #contains all the gene sample data
for pos,value in enumerate(file2_content):
    if pos > 0:
        val = value.split()
        sample1.append(val[0])
common_sample = []
common_sample_column = []
index = 0
for x in range(len(sample)):
    for y in range(len(sample1)):
        if sample1[y] == sample[x]:
            common_sample.append(sample[x])
            common_sample_column.append(x)
print(common_sample)

gene_value = ""
row = 0

for pos,value in enumerate(file1_content):
    if pos >= 0:
        row += 1
        gene_name = value.split()[0]
        gene_value += (gene_name + '\t' + gene_name + '\t')
        value_x = value.split()[1:]
        for pos,valx in enumerate(common_sample_column): #updated
            if pos != len(common_sample_column) - 1:
                gene_value += (value_x[valx] + '\t')
            else: gene_value += (value_x[valx])
        if row != 20530:
            gene_value += '\n'

common_sample_final = ""
for pos,value in enumerate(common_sample):
    if pos != len(common_sample) - 1: #updated
        common_sample_final += (value + '\t')
    else: common_sample_final += (value)

b = str(len(common_sample_column))

fout = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_3/sample_500.gct', 'w')
fout.write("#1.2\n")
fout.write(str(row) + '\t' + b + '\n')
fout.write('Name\tDescription\t' + common_sample_final + '\n')
fout.write(gene_value)
#print(row)
#print(b)
file1_content.close()
fout.close()

```

['TCGA-BH-A0BQ-01', 'TCGA-BH-A0BT-01', 'TCGA-BH-A18J-01', 'TCGA-BH-A0W7-01', 'TCGA-BH-A0HA-01', 'TCGA-A7-A0CE-01', 'TCGA-AR-A0U1-01', 'TCGA-EW-A1OZ-01', 'TCGA-A2-A0EY-01', 'TCGA-A8-A09R-01', 'TCGA-E2-A1II-01', 'TCGA-C8-A273-01', 'TCGA-BH-A1EO-01', 'TCGA-AN-A0AS-01', 'TCGA-A2-A1G0-01', 'TCGA-B6-A0I8-01', 'TCGA-AR-A24S-01', 'TCGA-D8-A1XR-01', 'TCGA-E2-A1IJ-01', 'TCGA-BH-A0AY-01', 'TCGA-B6-A0IH-01', 'TCGA-A1-A0SH-01', 'TCGA-AR-A1AQ-01', 'TCGA-A1-A0SN-01', 'TCGA-BH-A18T-01', 'TCGA-E9-A22B-01', 'TCGA-AO-A0J9-01', 'TCGA-A1-A0SE-01', 'TCGA-A2-A04U-01', 'TCGA-D8-A1Y2-01', 'TCGA-E9-A229-01', 'TCGA-BH-A0HK-01', 'TCGA-E2-A14Z-01', 'TCGA-D8-A1Y1-01', 'TCGA-A2-A04P-01', 'TCGA-A7-A13E-01', 'TCGA-A7-A0D9-01', 'TCGA-E2-A14R-01', 'TCGA-BH-A0E2-01', 'TCGA-E2-A1L7-01', 'TCGA-GM-A2DM-01', 'TCGA-A2-A0D4-01', 'TCGA-AO-A12F-01', 'TCGA-AC-A3HN-01', 'TCGA-A2-A25C-01', 'TCGA-E9-A1N4-01', 'TCGA-AQ-A1H3-01', 'TCGA-E9-A295-01', 'TCGA-D8-A1XD-01', 'TCGA-AC-A2FB-01', 'TCGA-BH-A0DQ-01', 'TCGA-BH-A0BW-01', 'TCGA-A8-A0A4-01', 'TCGA-D8-A1JS-01', 'TCGA-AR-A1AJ-01', 'TCGA-AN-A0AK-01', 'TCGA-BH-A0BZ-01', 'TCGA-AR-A24P-01', 'TCGA-BH-A0DI-01', 'TCGA-A2-A0T6-01', 'TCGA-A2-A25B-01', 'TCGA-A2-A0SV-01', 'TCGA-BH-A18G-01', 'TCGA-BH-A1FC-01', 'TCGA-E2-A14W-01', 'TCGA-AO-A124-01', 'TCGA-A8-A075-01', 'TCGA-AQ-A0Y5-01', 'TCGA-A8-A07C-01', 'TCGA-BH-A201-01', 'TCGA-AR-A24H-01', 'TCGA-BH-A0WA-01', 'TCGA-A8-A08P-01', 'TCGA-BH-A2L8-01', 'TCGA-BH-A0DK-01', 'TCGA-A2-A0T2-01', 'TCGA-E2-A1IF-01', 'TCGA-E9-A1N8-01', 'TCGA-EW-A1P0-01', 'TCGA-E9-A244-01', 'TCGA-AO-A0JD-01', 'TCGA-A1-A0SO-01', 'TCGA-A8-A09T-01', 'TCGA-BH-A18I-01', 'TCGA-B6-A0RU-01', 'TCGA-AO-A0JC-01', 'TCGA-E9-A3Q9-01', 'TCGA-C8-A1HL-01', 'TCGA-AR-A24W-01', 'TCGA-D8-A13Y-01', 'TCGA-AR-A2LO-01', 'TCGA-E2-A1LS-01', 'TCGA-AR-A0TS-01', 'TCGA-AR-A0TT-01', 'TCGA-D8-A1JJ-01', 'TCGA-BH-A1ES-01', 'TCGA-C8-A1HJ-01', 'TCGA-BH-A0BR-01', 'TCGA-AO-A12E-01', 'TCGA-A2-A0CM-01', 'TCGA-E9-A1RB-01', 'TCGA-E9-A1R3-01', 'TCGA-D8-A1JK-01', 'TCGA-D8-A27L-01', 'TCGA-C8-A133-01', 'TCGA-A2-A0EO-01', 'TCGA-A2-A0YL-01', 'TCGA-AR-A1AY-01', 'TCGA-AR-A24L-01', 'TCGA-AR-A0U0-01', 'TCGA-BH-A1EW-01', 'TCGA-AN-A0FF-01', 'TCGA-AN-A0FX-01', 'TCGA-BH-A0DZ-01', 'TCGA-E2-A1LI-01', 'TCGA-AR-A255-01', 'TCGA-B6-A0IE-01', 'TCGA-A8-A07F-01', 'TCGA-AR-A1AS-01', 'TCGA-EW-A1PH-01', 'TCGA-BH-A0HP-01', 'TCGA-C8-A1HG-01', 'TCGA-E2-A1L9-01', 'TCGA-BH-A0BA-01', 'TCGA-D8-A1XY-01', 'TCGA-A2-A0D2-01', 'TCGA-AR-A1AL-01', 'TCGA-AO-A0J6-01', 'TCGA-D8-A1J9-01', 'TCGA-BH-A0B5-01', 'TCGA-BH-A0B4-01', 'TCGA-AR-A1AX-01', 'TCGA-B6-A0IN-01', 'TCGA-BH-A1F6-01', 'TCGA-A8-A07U-01', 'TCGA-BH-A18V-01', 'TCGA-AQ-A04L-01', 'TCGA-AR-A1AP-01', 'TCGA-AC-A2QH-01', 'TCGA-BH-A0HO-01', 'TCGA-E2-A1L8-01', 'TCGA-BH-A18H-01', 'TCGA-BH-A0HU-01', 'TCGA-A7-A26J-01', 'TCGA-A2-A0ST-01', 'TCGA-D8-A27W-01', 'TCGA-A2-A0SW-01', 'TCGA-GM-A2DL-01', 'TCGA-E2-A159-01', 'TCGA-BH-A0W4-01', 'TCGA-D8-A1Y3-01', 'TCGA-A2-A0T5-01', 'TCGA-AN-A0XS-01', 'TCGA-D8-A1JB-01', 'TCGA-B6-A0RE-01', 'TCGA-C8-A134-01', 'TCGA-AN-A0AT-01', 'TCGA-GM-A2DN-01', 'TCGA-BH-A0HX-01', 'TCGA-E9-A1NE-01', 'TCGA-BH-A0BP-01', 'TCGA-AR-A2LM-01', 'TCGA-B6-A0I1-01', 'TCGA-A2-A04R-01', 'TCGA-AO-A126-01', 'TCGA-AO-A0JM-01', 'TCGA-EW-A2FS-01', 'TCGA-E2-A1LH-01', 'TCGA-A2-A25D-01', 'TCGA-E9-A243-01', 'TCGA-E9-A1N6-01', 'TCGA-BH-A0W5-01', 'TCGA-E2-A15I-01', 'TCGA-BH-A0H5-01', 'TCGA-E9-A227-01', 'TCGA-EW-A2FV-01', 'TCGA-BH-A0BG-01', 'TCGA-E2-A1LA-01', 'TCGA-A2-A04Y-01', 'TCGA-AR-A240-01', 'TCGA-GM-A2DA-01', 'TCGA-AO-A12A-01', 'TCGA-EW-A1IW-01', 'TCGA-C8-A12K-01', 'TCGA-E9-A1NA-01', 'TCGA-A8-A07E-01', 'TCGA-BH-A18K-01', 'TCGA-A1-A0SJ-01', 'TCGA-D8-A1JL-01', 'TCGA-A8-A097-01', 'TCGA-AN-A0FZ-01', 'TCGA-BH-A0E6-01', 'TCGA-C8-A132-01', 'TCGA-BH-A0H6-01', 'TCGA-D8-A1XS-01', 'TCGA-C8-A12Y-01', 'TCGA-E9-A1QZ-01', 'TCGA-GM-A2D9-01', 'TCGA-AN-A0XU-01', 'TCGA-D8-A142-01', 'TCGA-EW-A1J2-01', 'TCGA-BH-A0HF-01', 'TCGA-AO-A0JG-01', 'TCGA-C8-A131-01', 'TCGA-E2-A1AZ-01', 'TCGA-E9-A3HO-01', 'TCGA-D8-A1JM-01', 'TCGA-BH-A0BF-01', 'TCGA-BH-A1FH-01', 'TCGA-A2-A0SX-01', 'TCGA-E9-A1ND-01', 'TCGA-A8-A084-01', 'TCGA-B6-A0X0-01', 'TCGA-AO-A0J4-01', 'TCGA-E9-A1NG-01', 'TCGA-BH-A18M-01', 'TCGA-E2-A10F-01', 'TCGA-D8-A27H-01', 'TCGA-AN-A0XL-01', 'TCGA-AR-A0U2-01', 'TCGA-E9-A22G-01', 'TCGA-AR-A1AW-01', 'TCGA-A7-A2KD-01', 'TCGA-EW-A1P4-01', 'TCGA-E9-A245-01', 'TCGA-E2-A1IP-01', 'TCGA-C8-A3M7-01', 'TCGA-BH-A0DV-01', 'TCGA-AO-A128-01', 'TCGA-EW-A1J3-01', 'TCGA-AC-A2BM-01', 'TCGA-AN-A0FN-01', 'TCGA-A8-A08O-01', 'TCGA-A8-A09A-01', 'TCGA-A2-A0T3-01', 'TCGA-B6-A0RN-01', 'TCGA-GM-A2DC-01', 'TCGA-AN-A04A-01', 'TCGA-E9-A2JT-01', 'TCGA-E2-A1LL-01', 'TCGA-E9-A1R0-01', 'TCGA-A8-A07B-01', 'TCGA-A8-A06Q-01', 'TCGA-E2-A14Y-01', 'TCGA-AN-A0AJ-01', 'TCGA-A8-A076-01', 'TCGA-A2-A0ES-01', 'TCGA-BH-A1F2-01', 'TCGA-BH-A0B0-01', 'TCGA-A8-A09D-01', 'TCGA-A2-A0YJ-01', 'TCGA-A1-A0SI-01', 'TCGA-BH-A0E0-01', 'TCGA-A7-A3IY-01', 'TCGA-EW-A1P3-01', 'TCGA-AR-A24V-01', 'TCGA-A7-A13F-01', 'TCGA-BH-A1FB-01', 'TCGA-BH-A0BM-01', 'TCGA-AR-A1AU-01', 'TCGA-A7-A26G-01', 'TCGA-B6-A0I5-01', 'TCGA-BH-A0E1-01', 'TCGA-A2-A0EW-01', 'TCGA-BH-A0AU-01', 'TCGA-AO-A03V-01', 'TCGA-E9-A22D-01', 'TCGA-B6-A0I6-01', 'TCGA-D8-A1JH-01', 'TCGA-E2-A1BC-01', 'TCGA-BH-A0C0-01', 'TCGA-AR-A24X-01', 'TCGA-EW-A1OW-01', 'TCGA-D8-A1XF-01', 'TCGA-AR-A0U4-01', 'TCGA-D8-A1XK-01', 'TCGA-E2-A15D-01', 'TCGA-B6-A0IQ-01', 'TCGA-AO-A129-01', 'TCGA-GM-A2DF-01', 'TCGA-D8-A27M-01', 'TCGA-AN-A03X-01', 'TCGA-B6-A0WY-01', 'TCGA-C8-A1HI-01', 'TCGA-A1-A0SF-01', 'TCGA-AR-A2LR-01', 'TCGA-E9-A1RF-01', 'TCGA-B6-A0RT-01', 'TCGA-E9-A1RI-01', 'TCGA-BH-A0DL-01', 'TCGA-AR-A251-01', 'TCGA-E2-A1B5-01', 'TCGA-A2-A0YH-01', 'TCGA-BH-A0DE-01', 'TCGA-A2-A3KD-01', 'TCGA-BH-A0GY-01', 'TCGA-AC-A2FF-01', 'TCGA-BH-A0H3-01', 'TCGA-A2-A25F-01', 'TCGA-A7-A26F-01', 'TCGA-B6-A0WT-01', 'TCGA-BH-A28Q-01', 'TCGA-B6-A1KF-01', 'TCGA-E2-A1LK-01', 'TCGA-BH-A209-01', 'TCGA-GI-A2C8-01', 'TCGA-BH-A0EA-01', 'TCGA-E9-A1R4-01', 'TCGA-A8-A092-01', 'TCGA-BH-A18F-01', 'TCGA-C8-A12V-01', 'TCGA-A7-A13D-01', 'TCGA-D8-A1XQ-01', 'TCGA-AR-A0TW-01', 'TCGA-BH-A0H7-01', 'TCGA-D8-A141-01', 'TCGA-C8-A26V-01', 'TCGA-A2-A0YE-01', 'TCGA-AR-A24K-01', 'TCGA-A1-A0SQ-01', 'TCGA-C8-A27B-01', 'TCGA-D8-A143-01', 'TCGA-A2-A3KC-01', 'TCGA-GM-A2DI-01', 'TCGA-BH-A0DT-01', 'TCGA-A8-A07O-01', 'TCGA-B6-A0IA-01', 'TCGA-AR-A0TZ-01', 'TCGA-E2-A10A-01', 'TCGA-AO-A0J5-01', 'TCGA-B6-A2IU-01', 'TCGA-D8-A27N-01', 'TCGA-D8-A27K-01', 'TCGA-E2-A14X-01', 'TCGA-A2-A1G1-01', 'TCGA-A7-A26I-01', 'TCGA-A7-A0DA-01', 'TCGA-E2-A15E-01', 'TCGA-E2-A1LG-01', 'TCGA-A8-A0A6-01', 'TCGA-E2-A15L-01', 'TCGA-B6-A1KC-01', 'TCGA-E2-A15S-01', 'TCGA-A2-A0YD-01', 'TCGA-E2-A1B1-01', 'TCGA-C8-A27A-01', 'TCGA-A8-A082-01', 'TCGA-D8-A1JI-01', 'TCGA-A2-A0T7-01', 'TCGA-A1-A0SP-01', 'TCGA-E2-A1IG-01', 'TCGA-AR-A24Q-01', 'TCGA-A1-A0SK-01', 'TCGA-D8-A147-01', 'TCGA-A8-A06P-01', 'TCGA-A8-A07J-01', 'TCGA-A2-A04V-01', 'TCGA-A2-A0EV-01', 'TCGA-AO-A12C-01', 'TCGA-D8-A1XM-01', 'TCGA-BH-A0B2-01', 'TCGA-E2-A150-01', 'TCGA-BH-A0B0-01', 'TCGA-BH-A1F0-01', 'TCGA-A2-A1FX-01', 'TCGA-BH-A1EX-01', 'TCGA-HN-A2NL-01', 'TCGA-A2-A0SY-01', 'TCGA-E2-A1IO-01', 'TCGA-E2-A3DX-01', 'TCGA-B6-A0IG-01', 'TCGA-A2-A0CV-01', 'TCGA-AR-A0TU-01', 'TCGA-EW-A1IX-01', 'TCGA-E2-A15K-01', 'TCGA-AR-A1AR-01', 'TCGA-E9-A1R2-01', 'TCGA-E2-A15H-01', 'TCGA-B6-A0WZ-01', 'TCGA-AN-A0XN-01', 'TCGA-AN-A0AL-01', 'TCGA-C8-A12X-01', 'TCGA-GI-A2C9-01', 'TCGA-E9-A226-01', 'TCGA-BH-A0RX-01', 'TCGA-A8-A07R-01', 'TCGA-A2-A0EN-01', 'TCGA-A1-A0SD-01', 'TCGA-BH-A0BJ-01', 'TCGA-BH-A0BD-01', 'TCGA-BH-A18U-01', 'TCGA-A1-A0SM-01', 'TCGA-A8-A09I-01', 'TCGA-E2-A1IH-01', 'TCGA-AO-A03N-01', 'TCGA-C8-A12U-01', 'TCGA-AO-A0JL-01', 'TCGA-AO-A0JF-01', 'TCGA-BH-A0C1-01', 'TCGA-BH-A0AV-01', 'TCGA-D8-A1JD-01', 'TCGA-AR-A2LN-01', 'TCGA-E2-A1BD-01', 'TCGA-AR-A24R-01', 'TCGA-BH-A0DH-01', 'TCGA-EW-A1IY-01', 'TCGA-EW-A1PB-01', 'TCGA-AN-A0FJ-01', 'TCGA-E2-A15M-01', 'TCGA-AN-A04D-01', 'TCGA-BH-A0B3-01', 'TCGA-AO-A12G-01', 'TCGA-AQ-A04J-01', 'TCGA-A2-A0YM-01', 'TCGA-A8-A08Z-01', 'TCGA-D8-A27F-01', 'TCGA-AN-A041-01', 'TCGA-A2-A0D3-01', 'TCGA-GM-A3NY-01', 'TCGA-AC-A2B8-01', 'TCGA-AR-A1AK-01', 'TCGA-E9-A2JS-01', 'TCGA-AN-A0FL-01', 'TCGA-BH-A1FN-01', 'TCGA-A2-A25E-01', 'TCGA-A8-A09V-01', 'TCGA-A2-A0CT-01', 'TCGA-E9-A22E-01', 'TCGA-BH-A18Q-01', 'TCGA-AO-A03L-01', 'TCGA-AN-A0XW-01', 'TCGA-AC-A23G-01', 'TCGA-EW-A1P8-01', 'TCGA-B6-A0I2-01', 'TCGA-B6-A0RI-01', 'TCGA-B6-A0WX-01', 'TCGA-C8-A26W-01', 'TCGA-AO-A03M-01', 'TCGA-EW-A1J5-01', 'TCGA-C8-A12W-01', 'TCGA-AO-A1KR-01', 'TCGA-EW-A1PD-01', 'TCGA-EW-A1PC-01', 'TCGA-AR-A0TQ-01', 'TCGA-A2-A0EX-01', 'TCGA-E2-A14N-01', 'TCGA-AR-A1AN-01', 'TCGA-A2-A04T-01', 'TCGA-B6-A0IJ-01', 'TCGA-AR-A256-01', 'TCGA-A7-A3J1-01', 'TCGA-C8-A12O-01', 'TCGA-E9-A249-01', 'TCGA-A8-A08R-01', 'TCGA-BH-A0AZ-01', 'TCGA-BH-A0H0-01', 'TCGA-D8-A27R-01', 'TCGA-A2-A0T0-01', 'TCGA-D8-A27I-01', 'TCGA-BH-A0EI-01', 'TCGA-E2-A15A-01', 'TCGA-BH-A1EY-01', 'TCGA-AN-A0FT-01', 'TCGA-C8-A130-01', 'TCGA-D8-A1XL-01', 'TCGA-C8-A12N-01', 'TCGA-GM-A2DO-01', 'TCGA-AC-A23E-01', 'TCGA-B6-A1KN-01', 'TCGA-EW-A1OX-01', 'TCGA-BH-A0C3-01', 'TCGA-E9-A22A-01', 'TCGA-AR-A1AI-01', 'TCGA-AC-A2BK-01', 'TCGA-C8-A1HO-01', 'TCGA-E2-A15J-01', 'TCGA-BH-A1FE-01', 'TCGA-D8-A27G-01', 'TCGA-AN-A0FY-01', 'TCGA-D8-A27P-01', 'TCGA-BH-A0DS-01', 'TCGA-E2-A1L6-01', 'TCGA-BH-A0DG-01', 'TCGA-A2-A0CP-01', 'TCGA-A8-A09W-01', 'TCGA-AR-A252-01', 'TCGA-BH-A202-01', 'TCGA-AO-A1KT-01', 'TCGA-D8-A1JU-01', 'TCGA-E9-A1N5-01', 'TCGA-E2-A153-01', 'TCGA-A2-A0YG-01', 'TCGA-BH-A0BL-01', 'TCGA-A2-A04Q-01', 'TCGA-BH-A0HY-01', 'TCGA-AR-A24T-01', 'TCGA-AO-A0JA-01', 'TCGA-D8-A1XZ-01', 'TCGA-B6-A0X7-01', 'TCGA-B6-A0X1-01']

Creating .cls file

In [14]:

```
fin = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_3/sample_500.gct', 'r')
tsv_file = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/ESR1.tsv','r')

sample_phenotype = [x.split()[0] for pos,x in enumerate(tsv_file) if pos > 0]
mid_index = int(len(list(sample_phenotype)) / 2)
mid_val = sample_phenotype.copy()
for pos,value in enumerate(mid_val):
    if pos <= mid_index - 1:
        mid_val[pos] = 'ESR1_low'
    else: mid_val[pos] = 'ESR1_high'
ESR1_class = list(zip(sample_phenotype,mid_val))
classN = []
class_counter = 0
for pos,value in enumerate(fin):
    if pos == 2:
        sample_value = value.split()[2:]
        index = 0
        for x in sample_value:
            for y in ESR1_class:
                if x == y[0] and index == 0:
                    classN.append((y[1],class_counter))
                    index += 1
                if x == y[0]:
                    if y[1] not in [i[0] for i in classN]:
                        class_counter += 1
                        classN.append((y[1],class_counter))

for x in sample_value:
    for y in ESR1_class:
        if x == y[0]:
            for lum_basal in classN:
                if lum_basal[0] == y[1]:
                    valuex = [lum_basal[0],lum_basal[1]]
                    sample_value[sample_value.index(x)] = [x, valuex[0], valuex[1]]

listx = [x[2] for x in sample_value]
class_label = ""
for pos,x in enumerate(listx): #updated
    if pos != len(listx) - 1:
        class_label += str(x) + '\t'
    else : class_label += str(x)

print(classN)

fout = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_3/sample_500.cls', 'w')

fout.write(str(len(sample_value))+'\t'+ str(len(classN))+'\t'+ str(1)+'\n')
fout.write('#'+'\t'+str(classN[0][0]) +'\t'+str(classN[1][0]) +'\n')
fout.write(class_label)

fin.close()
fout.close()
```

[('ESR1_low', 0), ('ESR1_high', 1)]

In []:

In []:

Check 1

Check Output 1 (10000)

```

In [1]:
import random
test1 = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_1/sample_10000.gct",'r')
expression_file = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/EB++AdjustPANCAN_IlluminaHiSeq_RNASeqV2.geneExp.xena",'r')
def merge(list1,list2):
    merge_list = list(zip(list1,list2))
    return merge_list
number_of_random_samples = 200
number_of_random_genes = 200
index = 0
sample_value = [] #has the value of all the (random_sample,random_genes)
for pos,value in enumerate(test1):
    if pos == 1:
        rows = value.split()[0]
        column = value.split()[1]
    if pos == 2:
        random_gene = sorted(random.sample(range(int(rows)),number_of_random_genes))
        value = value.split()[2:]
        random_sample = random.sample(value, number_of_random_samples)
        for a,valuex in enumerate(value):
            if valuex in random_sample:
                random_sample[random_sample.index(valuex)] = (valuex,a)
        combine_list_index = merge(random_sample,random_gene)
    if pos > 2:
        value = value.split()[2:]
        for x in combine_list_index:
            if (pos - 2) == x[1]:
                sample_value.append(value[x[0][1]])

final_val = merge(combine_list_index,sample_value)

#reading the value from the large main expression data file.
indx = 0
expression_file_common_list = []
sample_value_2 = []
for pos,value in enumerate(expression_file):
    if pos == 0:
        value = value.split()[1:]
        for y in final_val:
            for x in range(len(value)):
                if value[x] == y[0][0][0]:
                    expression_file_common_list.append(x)
        combine_expression_index = merge(expression_file_common_list, random_gene)
    if pos > 0:
        value = value.split()[1:]
        for x in combine_expression_index:
            if pos == x[1]:
                a = x[0]
                sample_value_2.append(value[a])
        indx += 1

counter = True
x1 = 0
for x in range(len(sample_value)):
    if sample_value[x] != sample_value_2[x]:
        x1 += 1
        counter = False
print(merge(sample_value, sample_value_2))

print("Comparing all the random values from expression data and .gct file : ",counter)
print(x1)

```



```
[('1.13', '1.13'), ('6.74', '6.74'), ('2.38', '2.38'), ('13.72', '13.72'), ('6.97', '6.97'), ('7.58', '7.58'), ('9.86', '9.86'),
('10.47', '10.47'), ('1.11', '1.11'), ('3.69', '3.69'), ('9.60', '9.60'), ('9.04', '9.04'), ('10.15', '10.15'), ('8.89', '8.89'),
('10.42', '10.42'), ('10.31', '10.31'), ('5.02', '5.02'), ('7.76', '7.76'), ('8.92', '8.92'), ('7.34', '7.34'), ('8.54', '8.54'),
('10.57', '10.57'), ('9.43', '9.43'), ('11.02', '11.02'), ('7.96', '7.96'), ('0.00', '0.00'), ('9.41', '9.41'), ('9.04', '9.04'),
('10.51', '10.51'), ('7.91', '7.91'), ('7.81', '7.81'), ('1.72', '1.72'), ('3.96', '3.96'), ('7.71', '7.71'), ('12.71', '12.71'),
('4.98', '4.98'), ('7.52', '7.52'), ('12.43', '12.43'), ('0.00', '0.00'), ('1.74', '1.74'), ('8.38', '8.38'), ('10.30', '10.30'),
('11.37', '11.37'), ('7.80', '7.80'), ('3.91', '3.91'), ('9.25', '9.25'), ('13.50', '13.50'), ('11.62', '11.62'), ('5.37', '5.37'
), ('0.00', '0.00'), ('8.86', '8.86'), ('4.33', '4.33'), ('11.46', '11.46'), ('4.42', '4.42'), ('2.60', '2.60'), ('8.01', '8.01'
), ('6.05', '6.05'), ('9.10', '9.10'), ('7.47', '7.47'), ('10.61', '10.61'), ('3.38', '3.38'), ('0.00', '0.00'), ('7.08', '7.08'
), ('10.34', '10.34'), ('9.32', '9.32'), ('10.82', '10.82'), ('0.00', '0.00'), ('6.96', '6.96'), ('7.12', '7.12'), ('11.12', '11.12'
), ('11.94', '11.94'), ('11.81', '11.81'), ('3.72', '3.72'), ('9.62', '9.62'), ('0.00', '0.00'), ('6.20', '6.20'), ('0.90', '0.90
'), ('0.00', '0.00'), ('11.62', '11.62'), ('0.00', '0.00'), ('1.76', '1.76'), ('2.57', '2.57'), ('10.86', '10.86'), ('9.56', '9.5
6'), ('7.53', '7.53'), ('5.97', '5.97'), ('10.65', '10.65'), ('7.20', '7.20'), ('9.54', '9.54'), ('0.00', '0.00'), ('8.81', '8.81
'), ('9.58', '9.58'), ('9.31', '9.31'), ('0.00', '0.00'), ('9.73', '9.73'), ('9.40', '9.40'), ('0.00', '0.00'), ('4.42', '4.42'
), ('11.24', '11.24'), ('7.68', '7.68'), ('6.64', '6.64'), ('5.46', '5.46'), ('10.83', '10.83'), ('0.00', '0.00'), ('9.34', '9.34'
), ('8.89', '8.89'), ('9.64', '9.64'), ('8.92', '8.92'), ('8.56', '8.56'), ('9.74', '9.74'), ('10.05', '10.05'), ('10.02', '10.02'
), ('0.00', '0.00'), ('0.00', '0.00'), ('14.26', '14.26'), ('8.89', '8.89'), ('6.13', '6.13'), ('10.21', '10.21'), ('2.75', '2.75'
), ('5.88', '5.88'), ('10.17', '10.17'), ('9.09', '9.09'), ('0.59', '0.59'), ('11.45', '11.45'), ('7.29', '7.29'), ('NA', 'NA'), ('0
.00', '0.00'), ('0.00', '0.00'), ('0.00', '0.00'), ('9.33', '9.33'), ('6.38', '6.38'), ('4.59', '4.59'), ('9.50', '9.50'), ('4.40
', '4.40'), ('5.95', '5.95'), ('7.80', '7.80'), ('10.30', '10.30'), ('10.94', '10.94'), ('0.00', '0.00'), ('0.00', '0.00'), ('10.
32', '10.32'), ('0.00', '0.00'), ('11.65', '11.65'), ('2.55', '2.55'), ('1.26', '1.26'), ('2.47', '2.47'), ('8.79', '8.79'), ('9.
56', '9.56'), ('2.06', '2.06'), ('9.56', '9.56'), ('1.05', '1.05'), ('12.40', '12.40'), ('0.00', '0.00'), ('10.66', '10.66'), ('1
0.32', '10.32'), ('2.42', '2.42'), ('10.71', '10.71'), ('9.00', '9.00'), ('5.03', '5.03'), ('1.44', '1.44'), ('0.66', '0.66'), ('
11.91', '11.91'), ('11.36', '11.36'), ('11.28', '11.28'), ('7.64', '7.64'), ('4.94', '4.94'), ('0.00', '0.00'), ('0.00', '0.00'
), ('0.00', '0.00'), ('0.00', '0.00'), ('10.70', '10.70'), ('9.78', '9.78'), ('6.23', '6.23'), ('10.71', '10.71'), ('10.38', '10.38'
), ('1.38', '1.38'), ('6.72', '6.72'), ('2.53', '2.53'), ('10.20', '10.20'), ('9.70', '9.70'), ('9.35', '8.73'), ('11.12', '13.19
'), ('1.82', '5.87'), ('5.77', '7.31'), ('3.61', '8.51'), ('4.08', '1.63'), ('11.81', '13.07'), ('2.75', '4.67'), ('4.25', '3.91'
), ('13.60', '10.40'), ('0.78', '0.48'), ('10.43', '10.64'), ('9.34', '8.29'), ('4.80', '3.59'), ('5.51', '6.33'), ('7.49', '4.44
'), ('9.32', '8.81'), ('7.45', '6.51'), ('7.41', '7.25'), ('3.29', '0.00')]
```

Comparing all the random values from expression data and .gct file : False

20

Check Output 2

In [5]:

```

import random
test1 = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_2/sample_500.gct",'r')
expression_file = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/TCGA.BRCA.sampleMap_HiSeqV2",'r')
def merge(list1,list2):
    merge_list = list(zip(list1,list2))
    return merge_list
number_of_random_samples = 200
number_of_random_genes = 200
index = 0
sample_value = [] #has the value of all the (random_sample,random_genes)
for pos,value in enumerate(test1):
    if pos == 1:
        rows = value.split()[0]
        column = value.split()[1]
    if pos == 2:
        random_gene = sorted(random.sample(range(int(rows)),number_of_random_genes))
        value = value.split()[2:]
        random_sample = random.sample(value, number_of_random_samples)
        for a,valuex in enumerate(value):
            if valuex in random_sample:
                random_sample[random_sample.index(valuex)] = (valuex,a)
        combine_list_index = merge(random_sample,random_gene)
    if pos > 2:
        value = value.split()[2:]
        for x in combine_list_index:
            if (pos - 2) == x[1]:
                sample_value.append(value[x[0][1]])

final_val = merge(combine_list_index,sample_value)

#reading the value from the large main expression data file.
indx = 0
expression_file_common_list = []
sample_value_2 = []
for pos,value in enumerate(expression_file):
    if pos == 0:
        value = value.split()[1:]
        for y in final_val:
            for x in range(len(value)):
                if value[x] == y[0][0][0]:
                    expression_file_common_list.append(x)
        combine_expression_index = merge(expression_file_common_list, random_gene)
    if pos > 0:
        value = value.split()[1:]
        for x in combine_expression_index:
            if pos == x[1]:
                a = x[0]
                sample_value_2.append(value[a])
        indx += 1

counter = True
x1 = 0
for x in range(len(sample_value)):
    if sample_value[x] != sample_value_2[x]: #updated
        x1 += 1
        counter = False
print(merge(sample_value, sample_value_2))

print("Comparing all the random values from expression data and .gct file : ",counter)
print("Number of False: ",x1)

```

```
[('11.1984', '11.1984'), ('4.0376', '4.0376'), ('0.0000', '0.0000'), ('9.4436', '9.4436'), ('9.2210', '9.2210'), ('6.3075', '6.3075'), ('0.0000', '0.0000'), ('7.6045', '7.6045'), ('1.7517', '1.7517'), ('8.8507', '8.8507'), ('0.0000', '0.0000'), ('11.9812', '11.9812'), ('11.0877', '11.0877'), ('10.8611', '10.8611'), ('8.1451', '8.1451'), ('0.0000', '0.0000'), ('8.7178', '8.7178'), ('4.4296', '4.4296'), ('0.0000', '0.0000'), ('10.4259', '10.4259'), ('9.4719', '9.4719'), ('10.7430', '10.7430'), ('8.9228', '8.9228'), ('6.3307', '6.3307'), ('8.7386', '8.7386'), ('10.4727', '10.4727'), ('0.0000', '0.0000'), ('8.4589', '8.4589'), ('9.4011', '9.4011'), ('0.0000', '0.0000'), ('5.2869', '5.2869'), ('4.8646', '4.8646'), ('11.5647', '11.5647'), ('14.3976', '14.3976'), ('0.5407', '0.5407'), ('10.0688', '10.0688'), ('0.0000', '0.0000'), ('0.8842', '0.8842'), ('10.0262', '10.0262'), ('6.5891', '6.5891'), ('7.5785', '7.5785'), ('0.7710', '0.7710'), ('10.8805', '10.8805'), ('10.7500', '10.7500'), ('0.0000', '0.0000'), ('0.0000', '0.0000'), ('5.9063', '5.9063'), ('4.8346', '4.8346'), ('8.5477', '8.5477'), ('10.6847', '10.6847'), ('6.8166', '6.8166'), ('0.0000', '0.0000'), ('4.4057', '4.4057'), ('11.4899', '11.4899'), ('8.5952', '8.5952'), ('0.2981', '0.2981'), ('10.8442', '10.8442'), ('0.0000', '0.0000'), ('6.7427', '6.7427'), ('10.1146', '10.1146'), ('2.0179', '2.0179'), ('6.6022', '6.6022'), ('8.2691', '8.2691'), ('6.8309', '6.8309'), ('5.6216', '5.6216'), ('0.0000', '0.0000'), ('11.0145', '11.0145'), ('11.2854', '11.2854'), ('7.1581', '7.1581'), ('10.5343', '10.5343'), ('9.1297', '9.1297'), ('0.0000', '0.0000'), ('7.0674', '7.0674'), ('0.0000', '0.0000'), ('8.6531', '8.6531'), ('0.0000', '0.0000'), ('7.1149', '7.1149'), ('10.4217', '10.4217'), ('6.6111', '6.6111'), ('9.5110', '9.5110'), ('8.7054', '8.7054'), ('0.4522', '0.4522'), ('10.4195', '10.4195'), ('8.4864', '8.4864'), ('6.1648', '6.1648'), ('9.3734', '9.3734'), ('9.7353', '9.7353'), ('1.2506', '1.2506'), ('0.5573', '0.5573'), ('11.8800', '11.8800'), ('0.7149', '0.7149'), ('10.8792', '10.8792'), ('10.8905', '10.8905'), ('7.6029', '7.6029'), ('0.9713', '0.9713'), ('8.6808', '8.6808'), ('9.7218', '9.7218'), ('9.2871', '9.2871'), ('8.4618', '8.4618'), ('10.0084', '10.0084'), ('5.1603', '5.1603'), ('9.9966', '9.9966'), ('9.0345', '9.0345'), ('11.1741', '11.1741'), ('0.0000', '0.0000'), ('1.2051', '1.2051'), ('10.2148', '10.2148'), ('7.7850', '7.7850'), ('0.0000', '0.0000'), ('0.6669', '0.6669'), ('7.1825', '7.1825'), ('1.3785', '1.3785'), ('2.4330', '2.4330'), ('7.0992', '7.0992'), ('10.7176', '10.7176'), ('1.4465', '1.4465'), ('8.9665', '8.9665'), ('12.0115', '12.0115'), ('8.1560', '8.1560'), ('7.6533', '7.6533'), ('4.1368', '4.1368'), ('4.0644', '4.0644'), ('7.5855', '7.5855'), ('7.6196', '7.6196'), ('6.7884', '6.7884'), ('10.1325', '10.1325'), ('0.5218', '0.5218'), ('11.1136', '11.1136'), ('11.7726', '11.7726'), ('2.8014', '2.8014'), ('10.9054', '10.9054'), ('10.7491', '10.7491'), ('8.7181', '8.7181'), ('7.5127', '7.5127'), ('3.1533', '3.1533'), ('10.0452', '10.0452'), ('5.5741', '5.5741'), ('9.9204', '9.9204'), ('9.6924', '9.6924'), ('9.1680', '9.1680'), ('2.2221', '2.2221'), ('0.0000', '0.0000'), ('4.2063', '4.2063'), ('13.2196', '13.2196'), ('7.6515', '7.6515'), ('10.1214', '10.1214'), ('10.5750', '10.5750'), ('3.1899', '3.1899'), ('10.0988', '10.0988'), ('9.8474', '9.8474'), ('8.9545', '8.9545'), ('0.0000', '0.0000'), ('14.0317', '14.0317'), ('4.6427', '4.6427'), ('0.0000', '0.0000'), ('10.0054', '10.0054'), ('10.3760', '10.3760'), ('0.3648', '0.3648'), ('6.9743', '6.9743'), ('11.2349', '11.2349'), ('11.8392', '11.8392'), ('0.0000', '0.0000'), ('9.2579', '9.2579'), ('10.4837', '10.4837'), ('10.3143', '10.3143'), ('9.8222', '9.8222'), ('7.7975', '7.7975'), ('4.4153', '4.4153'), ('0.0000', '0.0000'), ('11.0487', '11.0487'), ('0.3873', '0.3873'), ('8.3654', '8.3654'), ('10.0710', '10.0710'), ('8.8159', '8.8159'), ('2.1768', '2.1768'), ('0.0000', '0.0000'), ('0.0000', '0.0000'), ('3.7933', '3.7933'), ('8.9803', '8.9803'), ('7.1272', '7.1272'), ('0.0000', '0.0000'), ('1.2032', '1.2032'), ('7.0223', '7.0223'), ('1.9225', '1.9225'), ('9.3981', '9.3981'), ('10.3936', '10.3936'), ('8.9043', '8.9043'), ('6.2966', '6.2966'), ('10.7933', '10.7933'), ('0.0000', '0.0000'), ('8.3078', '8.3078'), ('2.4637', '2.4637'), ('0.8579', '0.8579'), ('0.0000', '0.0000'), ('8.2968', '8.2968'), ('9.7768', '9.7768'), ('8.6907', '8.6907'), ('0.0000', '0.0000'), ('3.3659', '3.3659'), ('1.7402', '1.7402')]
Comparing all the random values from expression data and .gct file :  True
Number of False:  0
```

Check Output 3

In [8]:

```

test1 = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_3/sample_500.gct",'r')
expression_file = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/TCGA.BRCA.sampleMap_HiSeqV2",'r')
def merge(list1,list2):
    merge_list = list(zip(list1,list2))
    return merge_list
number_of_random_samples = 200
number_of_random_genes = 200
index = 0
sample_value = [] #has the value of all the (random_sample,random_genes)
for pos,value in enumerate(test1):
    if pos == 1:
        rows = value.split()[0]
        column = value.split()[1]
    if pos == 2:
        random_gene = sorted(random.sample(range(int(rows)),number_of_random_genes))
        value = value.split()[2:]
        random_sample = random.sample(value, number_of_random_samples)
        for a,valuex in enumerate(value):
            if valuex in random_sample:
                random_sample[random_sample.index(valuex)] = (valuex,a)
        combine_list_index = merge(random_sample,random_gene)
    if pos > 2:
        value = value.split()[2:]
        for x in combine_list_index:
            if (pos - 2) == x[1]:
                sample_value.append(value[x[0][1]])

final_val = merge(combine_list_index,sample_value)

#reading the value from the large main expression data file.
indx = 0
expression_file_common_list = []
sample_value_2 = []
for pos,value in enumerate(expression_file):
    if pos == 0:
        value = value.split()[1:]
        for y in final_val:
            for x in range(len(value)):
                if value[x] == y[0][0][0]:
                    expression_file_common_list.append(x)
        combine_expression_index = merge(expression_file_common_list, random_gene)
    if pos > 0:
        value = value.split()[1:]
        for x in combine_expression_index:
            if pos == x[1]:
                a = x[0]
                sample_value_2.append(value[a])
        indx += 1

counter = True
x1 = 0
for x in range(len(sample_value)):
    if sample_value[x] != sample_value_2[x]:
        x1 += 1
        counter = False
print(merge(sample_value, sample_value_2))

print("Comparing all the random values from expression data and .gct file : ",counter)
print(x1)

```

```
[('10.1075', '10.1075'), ('0.0000', '0.0000'), ('9.1174', '9.1174'), ('7.1908', '7.1908'), ('7.1600', '7.1600'), ('6.7161', '6.7161'), ('9.1911', '9.1911'), ('9.6292', '9.6292'), ('2.4604', '2.4604'), ('8.2979', '8.2979'), ('9.2896', '9.2896'), ('2.1342', '2.1342'), ('4.5350', '4.5350'), ('3.4622', '3.4622'), ('7.9525', '7.9525'), ('8.7362', '8.7362'), ('9.5733', '9.5733'), ('6.6751', '6.6751'), ('10.5358', '10.5358'), ('0.0000', '0.0000'), ('10.1006', '10.1006'), ('0.0000', '0.0000'), ('3.5250', '3.5250'), ('0.3565', '0.3565'), ('4.2530', '4.2530'), ('3.1859', '3.1859'), ('3.9742', '3.9742'), ('9.9317', '9.9317'), ('9.9816', '9.9816'), ('10.9870', '10.9870'), ('10.2558', '10.2558'), ('10.5020', '10.5020'), ('8.9690', '8.9690'), ('0.0000', '0.0000'), ('2.5751', '2.5751'), ('0.0000', '0.0000'), ('7.4451', '7.4451'), ('5.8606', '5.8606'), ('7.3479', '7.3479'), ('0.0000', '0.0000'), ('12.4630', '12.4630'), ('0.0000', '0.0000'), ('6.6107', '6.6107'), ('5.8492', '5.8492'), ('3.4009', '3.4009'), ('9.9019', '9.9019'), ('2.4556', '2.4556'), ('0.0000', '0.0000'), ('11.1734', '11.1734'), ('1.4666', '1.4666'), ('0.0000', '0.0000'), ('7.0344', '7.0344'), ('10.3036', '10.3036'), ('0.0000', '0.0000'), ('14.8623', '14.8623'), ('0.0000', '0.0000'), ('2.8806', '2.8806'), ('9.5956', '9.5956'), ('0.0000', '0.0000'), ('0.4846', '0.4846'), ('8.0840', '8.0840'), ('11.2615', '11.2615'), ('8.0162', '8.0162'), ('5.6771', '5.6771'), ('10.2146', '10.2146'), ('0.0000', '0.0000'), ('2.5936', '2.5936'), ('5.7118', '5.7118'), ('12.5705', '12.5705'), ('11.6335', '11.6335'), ('8.0884', '8.0884'), ('0.0000', '0.0000'), ('9.6422', '9.6422'), ('8.0810', '8.0810'), ('8.0914', '8.0914'), ('7.6469', '7.6469'), ('10.4457', '10.4457'), ('3.5320', '3.5320'), ('5.8228', '5.8228'), ('0.0000', '0.0000'), ('4.0276', '4.0276'), ('9.4696', '9.4696'), ('11.1596', '11.1596'), ('3.3523', '3.3523'), ('8.6153', '8.6153'), ('10.5170', '10.5170'), ('0.0000', '0.0000'), ('7.1792', '7.1792'), ('10.1591', '10.1591'), ('6.2155', '6.2155'), ('8.4843', '8.4843'), ('0.0000', '0.0000'), ('9.7954', '9.7954'), ('6.8947', '6.8947'), ('9.7054', '9.7054'), ('3.2101', '3.2101'), ('10.2834', '10.2834'), ('0.6130', '0.6130'), ('6.1523', '6.1523'), ('8.4158', '8.4158'), ('7.4194', '7.4194'), ('10.7978', '10.7978'), ('0.9750', '0.9750'), ('11.8252', '11.8252'), ('5.3363', '5.3363'), ('5.6977', '5.6977'), ('2.9819', '2.9819'), ('12.1269', '12.1269'), ('1.9209', '1.9209'), ('11.0478', '11.0478'), ('11.5954', '11.5954'), ('7.9684', '7.9684'), ('11.1587', '11.1587'), ('8.5773', '8.5773'), ('9.2351', '9.2351'), ('4.9490', '4.9490'), ('5.8604', '5.8604'), ('7.0417', '7.0417'), ('5.4235', '5.4235'), ('8.6977', '8.6977'), ('9.2799', '9.2799'), ('6.4572', '6.4572'), ('2.1795', '2.1795'), ('12.0463', '12.0463'), ('0.0000', '0.0000'), ('8.6338', '8.6338'), ('0.0000', '0.0000'), ('9.2318', '9.2318'), ('0.0000', '0.0000'), ('12.4687', '12.4687'), ('10.4325', '10.4325'), ('6.9534', '6.9534'), ('5.1502', '5.1502'), ('8.5519', '8.5519'), ('8.0899', '8.0899'), ('5.7705', '5.7705'), ('6.2898', '6.2898'), ('9.1564', '9.1564'), ('4.1547', '4.1547'), ('11.8403', '11.8403'), ('9.4487', '9.4487'), ('11.8929', '11.8929'), ('0.0000', '0.0000'), ('5.0606', '5.0606'), ('0.0000', '0.0000'), ('7.3525', '7.3525'), ('9.7699', '9.7699'), ('2.5738', '2.5738'), ('9.8428', '9.8428'), ('8.9353', '8.9353'), ('0.6554', '0.6554'), ('9.6641', '9.6641'), ('0.0000', '0.0000'), ('8.5442', '8.5442'), ('9.1822', '9.1822'), ('9.4484', '9.4484'), ('8.3920', '8.3920'), ('10.8192', '10.8192'), ('0.0000', '0.0000'), ('5.9810', '5.9810'), ('1.1375', '1.1375'), ('0.0000', '0.0000'), ('5.7253', '5.7253'), ('12.6445', '12.6445'), ('0.0000', '0.0000'), ('9.3744', '9.3744'), ('0.0000', '0.0000'), ('5.2169', '5.2169'), ('9.2943', '9.2943'), ('1.7518', '1.7518'), ('8.5789', '8.5789'), ('6.6612', '6.6612'), ('0.0000', '0.0000'), ('5.7185', '5.7185'), ('0.0000', '0.0000'), ('8.2171', '8.2171'), ('6.2035', '6.2035'), ('5.9052', '5.9052'), ('9.5080', '9.5080'), ('8.4455', '8.4455'), ('10.1017', '10.1017'), ('8.5304', '8.5304'), ('3.1457', '3.1457'), ('5.3851', '5.3851'), ('1.1615', '1.1615'), ('3.5745', '3.5745'), ('8.8529', '8.8529'), ('10.3764', '10.3764'), ('0.0000', '0.0000'), ('8.8042', '8.8042'), ('10.0928', '10.0928'), ('9.0169', '9.0169'), ('5.2258', '5.2258'), ('1.1046', '1.1046'), ('0.0000', '0.0000'), ('8.7386', '8.7386'), ('10.0530', '10.0530'), ('5.8902', '5.8902'), ('1.2337', '1.2337'), ('8.1317', '8.1317')]
Comparing all the random values from expression data and .gct file :  True
0
```

Check 2: Phenotype (Script 2 and 3 only)

Check Script 2 Output

- * Generate random index from .cls file
- * Find the Sample ID in Same index location
- * Check if the Sample ID from .gct file and corresponding phenotype from .cls file match in the .tsv file.

In [4]:

```
import random
file1_content = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_2/sample_500.gct', 'r')
file2_content = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/BasalLum.tsv", 'r')
cls_content = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_2/sample_500.cls', 'r')
Number_of_test_samples = 200 #200 #300 #400 #500
#splitting the gct file in a list and getting the position
for pos, value in enumerate(file1_content):
    if pos == 2:
        gct_file = value.split()[2:]
    if pos == 3: break

#pheno_type_tsv = [x.split() for pos,x in enumerate(file2_content) if pos > 0]
for val in cls_content:
    cls_data = val.split()
    #print(cls_data)
    for pos,x in enumerate(cls_data):
        cls_data[pos] = [pos,x] #(index position in cls, phenotype value in the index postion)
random_cls_data = random.sample(cls_data, Number_of_test_samples) # get random index and class data
#print(random_cls_data)

#checking the Sample ID in same index in gct file
data = []
for pos,value in enumerate(gct_file):
    for random_val in random_cls_data:
        if pos == random_val[0]:
            if random_val[1] == '0':class_sample= 'Lum'
            else: class_sample = 'Basal'
            data.append([value, class_sample])

#print(data)
#check the phenotype in BasalLum.tsv file
index = 0
lum_basal = [x.split() for pos,x in enumerate(file2_content) if pos > 0]
for test_sample_phenotype in data:
    counter = False
    for sample_phenotype in lum_basal:
        if test_sample_phenotype == sample_phenotype:
            index += 1
            counter = True
    print(index,sample_phenotype,test_sample_phenotype, counter)

#[Here we have the Sample ID from .gct, and phenptype from cls] comparing against BasalLum.tsv [Sample ID, Phenotype] if same True
```

```
1 ['TCGA-A8-A096-01', 'Lum'] ['TCGA-A8-A096-01', 'Lum'] True
2 ['TCGA-BH-A0C7-01', 'Lum'] ['TCGA-BH-A0C7-01', 'Lum'] True
3 ['TCGA-AR-A0TV-01', 'Lum'] ['TCGA-AR-A0TV-01', 'Lum'] True
4 ['TCGA-A7-A0CE-01', 'Basal'] ['TCGA-A7-A0CE-01', 'Basal'] True
5 ['TCGA-EW-A1OZ-01', 'Lum'] ['TCGA-EW-A1OZ-01', 'Lum'] True
6 ['TCGA-A8-A07P-01', 'Lum'] ['TCGA-A8-A07P-01', 'Lum'] True
7 ['TCGA-E9-A22B-01', 'Lum'] ['TCGA-E9-A22B-01', 'Lum'] True
8 ['TCGA-D8-A1Y2-01', 'Lum'] ['TCGA-D8-A1Y2-01', 'Lum'] True
9 ['TCGA-BH-A0HK-01', 'Lum'] ['TCGA-BH-A0HK-01', 'Lum'] True
10 ['TCGA-D8-A1Y1-01', 'Lum'] ['TCGA-D8-A1Y1-01', 'Lum'] True
11 ['TCGA-A2-A04P-01', 'Basal'] ['TCGA-A2-A04P-01', 'Basal'] True
12 ['TCGA-A8-A0A1-01', 'Lum'] ['TCGA-A8-A0A1-01', 'Lum'] True
13 ['TCGA-E2-A14T-01', 'Lum'] ['TCGA-E2-A14T-01', 'Lum'] True
14 ['TCGA-A2-A0D4-01', 'Lum'] ['TCGA-A2-A0D4-01', 'Lum'] True
15 ['TCGA-E2-A106-01', 'Lum'] ['TCGA-E2-A106-01', 'Lum'] True
16 ['TCGA-E2-A15R-01', 'Lum'] ['TCGA-E2-A15R-01', 'Lum'] True
17 ['TCGA-A8-A06R-01', 'Lum'] ['TCGA-A8-A06R-01', 'Lum'] True
18 ['TCGA-E9-A1R5-01', 'Lum'] ['TCGA-E9-A1R5-01', 'Lum'] True
19 ['TCGA-D8-A1JS-01', 'Lum'] ['TCGA-D8-A1JS-01', 'Lum'] True
20 ['TCGA-AR-A1AJ-01', 'Basal'] ['TCGA-AR-A1AJ-01', 'Basal'] True
21 ['TCGA-A2-A0EU-01', 'Lum'] ['TCGA-A2-A0EU-01', 'Lum'] True
22 ['TCGA-AR-A24P-01', 'Lum'] ['TCGA-AR-A24P-01', 'Lum'] True
23 ['TCGA-E2-A14W-01', 'Lum'] ['TCGA-E2-A14W-01', 'Lum'] True
24 ['TCGA-A8-A075-01', 'Lum'] ['TCGA-A8-A075-01', 'Lum'] True
25 ['TCGA-A8-A093-01', 'Lum'] ['TCGA-A8-A093-01', 'Lum'] True
26 ['TCGA-A8-A099-01', 'Lum'] ['TCGA-A8-A099-01', 'Lum'] True
27 ['TCGA-AR-A24H-01', 'Lum'] ['TCGA-AR-A24H-01', 'Lum'] True
28 ['TCGA-A7-A26E-01', 'Lum'] ['TCGA-A7-A26E-01', 'Lum'] True
29 ['TCGA-A2-A0T2-01', 'Basal'] ['TCGA-A2-A0T2-01', 'Basal'] True
30 ['TCGA-AR-A2LL-01', 'Lum'] ['TCGA-AR-A2LL-01', 'Lum'] True
31 ['TCGA-EW-A1P0-01', 'Lum'] ['TCGA-EW-A1P0-01', 'Lum'] True
32 ['TCGA-E9-A244-01', 'Basal'] ['TCGA-E9-A244-01', 'Basal'] True
33 ['TCGA-B6-A0RM-01', 'Lum'] ['TCGA-B6-A0RM-01', 'Lum'] True
34 ['TCGA-A8-A08G-01', 'Lum'] ['TCGA-A8-A08G-01', 'Lum'] True
35 ['TCGA-C8-A1HL-01', 'Lum'] ['TCGA-C8-A1HL-01', 'Lum'] True
36 ['TCGA-BH-A1FG-01', 'Lum'] ['TCGA-BH-A1FG-01', 'Lum'] True
37 ['TCGA-D8-A1JJ-01', 'Lum'] ['TCGA-D8-A1JJ-01', 'Lum'] True
38 ['TCGA-E9-A1RB-01', 'Lum'] ['TCGA-E9-A1RB-01', 'Lum'] True
39 ['TCGA-E9-A1R6-01', 'Lum'] ['TCGA-E9-A1R6-01', 'Lum'] True
40 ['TCGA-E2-A10C-01', 'Lum'] ['TCGA-E2-A10C-01', 'Lum'] True
41 ['TCGA-B6-A0WW-01', 'Lum'] ['TCGA-B6-A0WW-01', 'Lum'] True
42 ['TCGA-E2-A1LI-01', 'Basal'] ['TCGA-E2-A1LI-01', 'Basal'] True
43 ['TCGA-D8-A1XV-01', 'Lum'] ['TCGA-D8-A1XV-01', 'Lum'] True
44 ['TCGA-C8-A1HG-01', 'Lum'] ['TCGA-C8-A1HG-01', 'Lum'] True
45 ['TCGA-BH-A18L-01', 'Lum'] ['TCGA-BH-A18L-01', 'Lum'] True
46 ['TCGA-A8-A083-01', 'Lum'] ['TCGA-A8-A083-01', 'Lum'] True
47 ['TCGA-AO-A0J6-01', 'Basal'] ['TCGA-AO-A0J6-01', 'Basal'] True
48 ['TCGA-AO-A0J7-01', 'Lum'] ['TCGA-AO-A0J7-01', 'Lum'] True
49 ['TCGA-AQ-A04H-01', 'Lum'] ['TCGA-AQ-A04H-01', 'Lum'] True
50 ['TCGA-BH-A1F6-01', 'Basal'] ['TCGA-BH-A1F6-01', 'Basal'] True
51 ['TCGA-A8-A07U-01', 'Basal'] ['TCGA-A8-A07U-01', 'Basal'] True
52 ['TCGA-BH-A18H-01', 'Lum'] ['TCGA-BH-A18H-01', 'Lum'] True
53 ['TCGA-AN-A0FS-01', 'Lum'] ['TCGA-AN-A0FS-01', 'Lum'] True
54 ['TCGA-AO-A1KS-01', 'Lum'] ['TCGA-AO-A1KS-01', 'Lum'] True
```

```
55 ['TCGA-EW-A1J6-01', 'Lum'] ['TCGA-EW-A1J6-01', 'Lum'] True
56 ['TCGA-E2-A159-01', 'Basal'] ['TCGA-E2-A159-01', 'Basal'] True
57 ['TCGA-A8-A08C-01', 'Lum'] ['TCGA-A8-A08C-01', 'Lum'] True
58 ['TCGA-BH-A0BP-01', 'Lum'] ['TCGA-BH-A0BP-01', 'Lum'] True
59 ['TCGA-B6-A0RV-01', 'Lum'] ['TCGA-B6-A0RV-01', 'Lum'] True
60 ['TCGA-A2-A04Y-01', 'Lum'] ['TCGA-A2-A04Y-01', 'Lum'] True
61 ['TCGA-A2-A0CQ-01', 'Lum'] ['TCGA-A2-A0CQ-01', 'Lum'] True
62 ['TCGA-E9-A1NH-01', 'Lum'] ['TCGA-E9-A1NH-01', 'Lum'] True
63 ['TCGA-D8-A1JL-01', 'Basal'] ['TCGA-D8-A1JL-01', 'Basal'] True
64 ['TCGA-C8-A132-01', 'Lum'] ['TCGA-C8-A132-01', 'Lum'] True
65 ['TCGA-D8-A1XS-01', 'Lum'] ['TCGA-D8-A1XS-01', 'Lum'] True
66 ['TCGA-AN-A0XU-01', 'Basal'] ['TCGA-AN-A0XU-01', 'Basal'] True
67 ['TCGA-BH-A0BV-01', 'Lum'] ['TCGA-BH-A0BV-01', 'Lum'] True
68 ['TCGA-BH-A1FJ-01', 'Lum'] ['TCGA-BH-A1FJ-01', 'Lum'] True
69 ['TCGA-BH-A0HF-01', 'Lum'] ['TCGA-BH-A0HF-01', 'Lum'] True
70 ['TCGA-D8-A1JM-01', 'Basal'] ['TCGA-D8-A1JM-01', 'Basal'] True
71 ['TCGA-A2-A0SX-01', 'Basal'] ['TCGA-A2-A0SX-01', 'Basal'] True
72 ['TCGA-D8-A1Y0-01', 'Lum'] ['TCGA-D8-A1Y0-01', 'Lum'] True
73 ['TCGA-A8-A085-01', 'Lum'] ['TCGA-A8-A085-01', 'Lum'] True
74 ['TCGA-AN-A0XL-01', 'Lum'] ['TCGA-AN-A0XL-01', 'Lum'] True
75 ['TCGA-AR-A1AW-01', 'Lum'] ['TCGA-AR-A1AW-01', 'Lum'] True
76 ['TCGA-A8-A09M-01', 'Lum'] ['TCGA-A8-A09M-01', 'Lum'] True
77 ['TCGA-E2-A1IP-01', 'Basal'] ['TCGA-E2-A1IP-01', 'Basal'] True
78 ['TCGA-EW-A1J3-01', 'Lum'] ['TCGA-EW-A1J3-01', 'Lum'] True
79 ['TCGA-AC-A2BM-01', 'Lum'] ['TCGA-AC-A2BM-01', 'Lum'] True
80 ['TCGA-A7-A3J0-01', 'Lum'] ['TCGA-A7-A3J0-01', 'Lum'] True
81 ['TCGA-A2-A0ET-01', 'Lum'] ['TCGA-A2-A0ET-01', 'Lum'] True
82 ['TCGA-E2-A1LL-01', 'Basal'] ['TCGA-E2-A1LL-01', 'Basal'] True
83 ['TCGA-E2-A156-01', 'Lum'] ['TCGA-E2-A156-01', 'Lum'] True
84 ['TCGA-E2-A14Y-01', 'Basal'] ['TCGA-E2-A14Y-01', 'Basal'] True
85 ['TCGA-D8-A1JN-01', 'Lum'] ['TCGA-D8-A1JN-01', 'Lum'] True
86 ['TCGA-AN-A0AJ-01', 'Lum'] ['TCGA-AN-A0AJ-01', 'Lum'] True
87 ['TCGA-A8-A076-01', 'Lum'] ['TCGA-A8-A076-01', 'Lum'] True
88 ['TCGA-A1-A0SI-01', 'Lum'] ['TCGA-A1-A0SI-01', 'Lum'] True
89 ['TCGA-BH-A0E0-01', 'Basal'] ['TCGA-BH-A0E0-01', 'Basal'] True
90 ['TCGA-A7-A3IY-01', 'Lum'] ['TCGA-A7-A3IY-01', 'Lum'] True
91 ['TCGA-EW-A1P3-01', 'Lum'] ['TCGA-EW-A1P3-01', 'Lum'] True
92 ['TCGA-A2-A0YF-01', 'Lum'] ['TCGA-A2-A0YF-01', 'Lum'] True
93 ['TCGA-A7-A26G-01', 'Basal'] ['TCGA-A7-A26G-01', 'Basal'] True
94 ['TCGA-AO-A12H-01', 'Lum'] ['TCGA-AO-A12H-01', 'Lum'] True
95 ['TCGA-A8-A0A9-01', 'Lum'] ['TCGA-A8-A0A9-01', 'Lum'] True
96 ['TCGA-B6-A0IP-01', 'Lum'] ['TCGA-B6-A0IP-01', 'Lum'] True
97 ['TCGA-E2-A154-01', 'Lum'] ['TCGA-E2-A154-01', 'Lum'] True
98 ['TCGA-AR-A24N-01', 'Lum'] ['TCGA-AR-A24N-01', 'Lum'] True
99 ['TCGA-E9-A22D-01', 'Lum'] ['TCGA-E9-A22D-01', 'Lum'] True
100 ['TCGA-AN-A0XO-01', 'Lum'] ['TCGA-AN-A0XO-01', 'Lum'] True
101 ['TCGA-AO-A129-01', 'Basal'] ['TCGA-AO-A129-01', 'Basal'] True
102 ['TCGA-GM-A2DF-01', 'Basal'] ['TCGA-GM-A2DF-01', 'Basal'] True
103 ['TCGA-AN-A049-01', 'Lum'] ['TCGA-AN-A049-01', 'Lum'] True
104 ['TCGA-A8-A06Z-01', 'Lum'] ['TCGA-A8-A06Z-01', 'Lum'] True
105 ['TCGA-AR-A2LR-01', 'Basal'] ['TCGA-AR-A2LR-01', 'Basal'] True
106 ['TCGA-B6-A0RT-01', 'Basal'] ['TCGA-B6-A0RT-01', 'Basal'] True
107 ['TCGA-AR-A251-01', 'Basal'] ['TCGA-AR-A251-01', 'Basal'] True
108 ['TCGA-B6-A0X4-01', 'Lum'] ['TCGA-B6-A0X4-01', 'Lum'] True
109 ['TCGA-A8-A06T-01', 'Lum'] ['TCGA-A8-A06T-01', 'Lum'] True
110 ['TCGA-A8-A090-01', 'Lum'] ['TCGA-A8-A090-01', 'Lum'] True
111 ['TCGA-AO-A030-01', 'Lum'] ['TCGA-AO-A030-01', 'Lum'] True
112 ['TCGA-A7-A26H-01', 'Lum'] ['TCGA-A7-A26H-01', 'Lum'] True
113 ['TCGA-BH-A209-01', 'Lum'] ['TCGA-BH-A209-01', 'Lum'] True
114 ['TCGA-A8-A092-01', 'Lum'] ['TCGA-A8-A092-01', 'Lum'] True
115 ['TCGA-AC-A30D-01', 'Lum'] ['TCGA-AC-A30D-01', 'Lum'] True
116 ['TCGA-E9-A1RD-01', 'Lum'] ['TCGA-E9-A1RD-01', 'Lum'] True
117 ['TCGA-A7-A13D-01', 'Basal'] ['TCGA-A7-A13D-01', 'Basal'] True
118 ['TCGA-C8-A26V-01', 'Lum'] ['TCGA-C8-A26V-01', 'Lum'] True
119 ['TCGA-A2-A0CW-01', 'Lum'] ['TCGA-A2-A0CW-01', 'Lum'] True
120 ['TCGA-AR-A24K-01', 'Lum'] ['TCGA-AR-A24K-01', 'Lum'] True
121 ['TCGA-AO-A12B-01', 'Lum'] ['TCGA-AO-A12B-01', 'Lum'] True
122 ['TCGA-A8-A07O-01', 'Basal'] ['TCGA-A8-A07O-01', 'Basal'] True
123 ['TCGA-BH-A0HB-01', 'Lum'] ['TCGA-BH-A0HB-01', 'Lum'] True
124 ['TCGA-E2-A15S-01', 'Lum'] ['TCGA-E2-A15S-01', 'Lum'] True
125 ['TCGA-A2-A0YD-01', 'Lum'] ['TCGA-A2-A0YD-01', 'Lum'] True
126 ['TCGA-C8-A27A-01', 'Lum'] ['TCGA-C8-A27A-01', 'Lum'] True
127 ['TCGA-A8-A082-01', 'Lum'] ['TCGA-A8-A082-01', 'Lum'] True
128 ['TCGA-D8-A1JI-01', 'Lum'] ['TCGA-D8-A1JI-01', 'Lum'] True
129 ['TCGA-AN-A0FK-01', 'Lum'] ['TCGA-AN-A0FK-01', 'Lum'] True
130 ['TCGA-A1-A0SP-01', 'Basal'] ['TCGA-A1-A0SP-01', 'Basal'] True
131 ['TCGA-E2-A1B4-01', 'Lum'] ['TCGA-E2-A1B4-01', 'Lum'] True
132 ['TCGA-B6-A0IM-01', 'Lum'] ['TCGA-B6-A0IM-01', 'Lum'] True
133 ['TCGA-A8-A07W-01', 'Lum'] ['TCGA-A8-A07W-01', 'Lum'] True
134 ['TCGA-A2-A04V-01', 'Lum'] ['TCGA-A2-A04V-01', 'Lum'] True
135 ['TCGA-E2-A1IL-01', 'Lum'] ['TCGA-E2-A1IL-01', 'Lum'] True
136 ['TCGA-AO-A0J8-01', 'Lum'] ['TCGA-AO-A0J8-01', 'Lum'] True
137 ['TCGA-BH-A1F0-01', 'Basal'] ['TCGA-BH-A1F0-01', 'Basal'] True
138 ['TCGA-HN-A2NL-01', 'Basal'] ['TCGA-HN-A2NL-01', 'Basal'] True
139 ['TCGA-BH-A1FM-01', 'Lum'] ['TCGA-BH-A1FM-01', 'Lum'] True
140 ['TCGA-D8-A1J8-01', 'Lum'] ['TCGA-D8-A1J8-01', 'Lum'] True
141 ['TCGA-BH-A1F8-01', 'Lum'] ['TCGA-BH-A1F8-01', 'Lum'] True
142 ['TCGA-AR-A0TU-01', 'Basal'] ['TCGA-AR-A0TU-01', 'Basal'] True
143 ['TCGA-AQ-A1H2-01', 'Lum'] ['TCGA-AQ-A1H2-01', 'Lum'] True
144 ['TCGA-B6-A0WZ-01', 'Lum'] ['TCGA-B6-A0WZ-01', 'Lum'] True
145 ['TCGA-GI-A2C9-01', 'Basal'] ['TCGA-GI-A2C9-01', 'Basal'] True
146 ['TCGA-E9-A226-01', 'Lum'] ['TCGA-E9-A226-01', 'Lum'] True
147 ['TCGA-A8-A095-01', 'Lum'] ['TCGA-A8-A095-01', 'Lum'] True
148 ['TCGA-A2-A259-01', 'Lum'] ['TCGA-A2-A259-01', 'Lum'] True
149 ['TCGA-EW-A1PA-01', 'Lum'] ['TCGA-EW-A1PA-01', 'Lum'] True
150 ['TCGA-E9-A24A-01', 'Lum'] ['TCGA-E9-A24A-01', 'Lum'] True
151 ['TCGA-BH-A204-01', 'Lum'] ['TCGA-BH-A204-01', 'Lum'] True
152 ['TCGA-A7-A0CD-01', 'Lum'] ['TCGA-A7-A0CD-01', 'Lum'] True
```



```
153 ['TCGA-BH-A0BD-01', 'Lum'] ['TCGA-BH-A0BD-01', 'Lum'] True
154 ['TCGA-BH-A18U-01', 'Lum'] ['TCGA-BH-A18U-01', 'Lum'] True
155 ['TCGA-AO-A0JI-01', 'Lum'] ['TCGA-AO-A0JI-01', 'Lum'] True
156 ['TCGA-AR-A250-01', 'Lum'] ['TCGA-AR-A250-01', 'Lum'] True
157 ['TCGA-BH-A0HQ-01', 'Lum'] ['TCGA-BH-A0HQ-01', 'Lum'] True
158 ['TCGA-C8-A12U-01', 'Lum'] ['TCGA-C8-A12U-01', 'Lum'] True
159 ['TCGA-D8-A1JD-01', 'Lum'] ['TCGA-D8-A1JD-01', 'Lum'] True
160 ['TCGA-EW-A1IY-01', 'Lum'] ['TCGA-EW-A1IY-01', 'Lum'] True
161 ['TCGA-EW-A1PB-01', 'Basal'] ['TCGA-EW-A1PB-01', 'Basal'] True
162 ['TCGA-AN-A03Y-01', 'Lum'] ['TCGA-AN-A03Y-01', 'Lum'] True
163 ['TCGA-A8-A08I-01', 'Lum'] ['TCGA-A8-A08I-01', 'Lum'] True
164 ['TCGA-AQ-A04J-01', 'Basal'] ['TCGA-AQ-A04J-01', 'Basal'] True
165 ['TCGA-A2-A0YM-01', 'Basal'] ['TCGA-A2-A0YM-01', 'Basal'] True
166 ['TCGA-AR-A0U3-01', 'Lum'] ['TCGA-AR-A0U3-01', 'Lum'] True
167 ['TCGA-E9-A1RC-01', 'Lum'] ['TCGA-E9-A1RC-01', 'Lum'] True
168 ['TCGA-GM-A3NY-01', 'Lum'] ['TCGA-GM-A3NY-01', 'Lum'] True
169 ['TCGA-E9-A2JS-01', 'Lum'] ['TCGA-E9-A2JS-01', 'Lum'] True
170 ['TCGA-BH-A1FN-01', 'Lum'] ['TCGA-BH-A1FN-01', 'Lum'] True
171 ['TCGA-A2-A25E-01', 'Lum'] ['TCGA-A2-A25E-01', 'Lum'] True
172 ['TCGA-A2-A0CT-01', 'Lum'] ['TCGA-A2-A0CT-01', 'Lum'] True
173 ['TCGA-E9-A22E-01', 'Lum'] ['TCGA-E9-A22E-01', 'Lum'] True
174 ['TCGA-A2-A1FV-01', 'Lum'] ['TCGA-A2-A1FV-01', 'Lum'] True
175 ['TCGA-B6-A0WV-01', 'Lum'] ['TCGA-B6-A0WV-01', 'Lum'] True
176 ['TCGA-EW-A1P8-01', 'Basal'] ['TCGA-EW-A1P8-01', 'Basal'] True
177 ['TCGA-B6-A0I2-01', 'Basal'] ['TCGA-B6-A0I2-01', 'Basal'] True
178 ['TCGA-C8-A26W-01', 'Lum'] ['TCGA-C8-A26W-01', 'Lum'] True
179 ['TCGA-BH-A0DX-01', 'Lum'] ['TCGA-BH-A0DX-01', 'Lum'] True
180 ['TCGA-C8-A12W-01', 'Lum'] ['TCGA-C8-A12W-01', 'Lum'] True
181 ['TCGA-AO-A1KR-01', 'Basal'] ['TCGA-AO-A1KR-01', 'Basal'] True
182 ['TCGA-EW-A1PD-01', 'Lum'] ['TCGA-EW-A1PD-01', 'Lum'] True
183 ['TCGA-EW-A1PC-01', 'Lum'] ['TCGA-EW-A1PC-01', 'Lum'] True
184 ['TCGA-AR-A0TQ-01', 'Lum'] ['TCGA-AR-A0TQ-01', 'Lum'] True
185 ['TCGA-D8-A1X5-01', 'Lum'] ['TCGA-D8-A1X5-01', 'Lum'] True
186 ['TCGA-A8-A08R-01', 'Basal'] ['TCGA-A8-A08R-01', 'Basal'] True
187 ['TCGA-A2-A0T0-01', 'Basal'] ['TCGA-A2-A0T0-01', 'Basal'] True
188 ['TCGA-E2-A15A-01', 'Lum'] ['TCGA-E2-A15A-01', 'Lum'] True
189 ['TCGA-C8-A130-01', 'Lum'] ['TCGA-C8-A130-01', 'Lum'] True
190 ['TCGA-A8-A07S-01', 'Lum'] ['TCGA-A8-A07S-01', 'Lum'] True
191 ['TCGA-AO-A1KP-01', 'Lum'] ['TCGA-AO-A1KP-01', 'Lum'] True
192 ['TCGA-BH-A0C3-01', 'Lum'] ['TCGA-BH-A0C3-01', 'Lum'] True
193 ['TCGA-BH-A0H9-01', 'Lum'] ['TCGA-BH-A0H9-01', 'Lum'] True
194 ['TCGA-AR-A1AI-01', 'Basal'] ['TCGA-AR-A1AI-01', 'Basal'] True
195 ['TCGA-A7-A0DB-01', 'Lum'] ['TCGA-A7-A0DB-01', 'Lum'] True
196 ['TCGA-BH-A202-01', 'Lum'] ['TCGA-BH-A202-01', 'Lum'] True
197 ['TCGA-BH-A0BL-01', 'Basal'] ['TCGA-BH-A0BL-01', 'Basal'] True
198 ['TCGA-BH-A0DO-01', 'Lum'] ['TCGA-BH-A0DO-01', 'Lum'] True
199 ['TCGA-BH-A0HY-01', 'Lum'] ['TCGA-BH-A0HY-01', 'Lum'] True
200 ['TCGA-B6-A0X1-01', 'Basal'] ['TCGA-B6-A0X1-01', 'Basal'] True
```

In [16]:

```
import random
file1_content = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_3/sample_500.gct', 'r')
file2_content = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/ESR1.tsv", 'r')
cls_content = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_3/sample_500.cls', 'r')
Number_of_test_samples = 200 #200 #300 #400 #500
for pos, value in enumerate(file1_content):
    if pos == 2:
        gct_file = value.split()[2:]
        break
#print(pheno_type_gct)
#pheno_type_tsv = [x.split() for pos,x in enumerate(file2_content) if pos > 0]
#Generating random index and correspinding phenotype data from .cls file
for val in cls_content:
    cls_data = val.split()
    for pos,x in enumerate(cls_data):
        cls_data[pos] = [pos,x] #(index position in cls, phenotype value in the index postion)
random_cls_data = random.sample(cls_data, Number_of_test_samples)
#print(random_cls_data)

# get the same index from cls file and go to gct file and get the Sample ID
data = []
for pos,value in enumerate(gct_file):
    for random_val in random_cls_data:
        if pos == random_val[0]:
            if random_val[1] == '0' :class_sample= 'ESR1_low'
            else: class_sample = 'ESR1_high'
            data.append([value, class_sample])
# checking it corresponding to the ESR1.tsv file
phenotype_tsv = [x.split() for pos,x in enumerate(file2_content) if pos > 0]
length = len(phenotype_tsv)
for pos, value in enumerate(phenotype_tsv):
    if pos <= length/2 - 1:
        value[1] = "ESR1_low"
    else: value[1] = "ESR1_high"
counter = 0
for val_x in data:
    index = False
    for val_y in phenotype_tsv:
        if val_x == val_y:
            index = True
            counter += 1
            print(val_x, val_y, index)

print(str(counter) + " randomly generated data found in the file.")
print(str( Number_of_test_samples - counter) + " randomly generated data not found in the file.")
```



```
[ 'TCGA-BH-A0W7-01', 'ESR1_low' ] [ 'TCGA-BH-A0W7-01', 'ESR1_low' ] True
[ 'TCGA-BH-A0HA-01', 'ESR1_low' ] [ 'TCGA-BH-A0HA-01', 'ESR1_low' ] True
[ 'TCGA-AR-A0U1-01', 'ESR1_low' ] [ 'TCGA-AR-A0U1-01', 'ESR1_low' ] True
[ 'TCGA-B6-A0IH-01', 'ESR1_high' ] [ 'TCGA-B6-A0IH-01', 'ESR1_high' ] True
[ 'TCGA-A1-A0SH-01', 'ESR1_low' ] [ 'TCGA-A1-A0SH-01', 'ESR1_low' ] True
[ 'TCGA-A1-A0SN-01', 'ESR1_high' ] [ 'TCGA-A1-A0SN-01', 'ESR1_high' ] True
[ 'TCGA-BH-A18T-01', 'ESR1_low' ] [ 'TCGA-BH-A18T-01', 'ESR1_low' ] True
[ 'TCGA-AO-A0J9-01', 'ESR1_high' ] [ 'TCGA-AO-A0J9-01', 'ESR1_high' ] True
[ 'TCGA-A2-A04U-01', 'ESR1_low' ] [ 'TCGA-A2-A04U-01', 'ESR1_low' ] True
[ 'TCGA-E9-A229-01', 'ESR1_low' ] [ 'TCGA-E9-A229-01', 'ESR1_low' ] True
[ 'TCGA-BH-A0HK-01', 'ESR1_high' ] [ 'TCGA-BH-A0HK-01', 'ESR1_high' ] True
[ 'TCGA-E2-A14Z-01', 'ESR1_high' ] [ 'TCGA-E2-A14Z-01', 'ESR1_high' ] True
[ 'TCGA-D8-A1Y1-01', 'ESR1_high' ] [ 'TCGA-D8-A1Y1-01', 'ESR1_high' ] True
[ 'TCGA-A2-A04P-01', 'ESR1_low' ] [ 'TCGA-A2-A04P-01', 'ESR1_low' ] True
[ 'TCGA-A7-A13E-01', 'ESR1_low' ] [ 'TCGA-A7-A13E-01', 'ESR1_low' ] True
[ 'TCGA-E2-A1L7-01', 'ESR1_high' ] [ 'TCGA-E2-A1L7-01', 'ESR1_high' ] True
[ 'TCGA-A2-A0D4-01', 'ESR1_high' ] [ 'TCGA-A2-A0D4-01', 'ESR1_high' ] True
[ 'TCGA-AO-A12F-01', 'ESR1_low' ] [ 'TCGA-AO-A12F-01', 'ESR1_low' ] True
[ 'TCGA-AC-A3HN-01', 'ESR1_high' ] [ 'TCGA-AC-A3HN-01', 'ESR1_high' ] True
[ 'TCGA-A2-A25C-01', 'ESR1_low' ] [ 'TCGA-A2-A25C-01', 'ESR1_low' ] True
[ 'TCGA-E9-A1N4-01', 'ESR1_low' ] [ 'TCGA-E9-A1N4-01', 'ESR1_low' ] True
[ 'TCGA-AQ-A1H3-01', 'ESR1_high' ] [ 'TCGA-AQ-A1H3-01', 'ESR1_high' ] True
[ 'TCGA-A8-A0A4-01', 'ESR1_high' ] [ 'TCGA-A8-A0A4-01', 'ESR1_high' ] True
[ 'TCGA-AR-A1AJ-01', 'ESR1_low' ] [ 'TCGA-AR-A1AJ-01', 'ESR1_low' ] True
[ 'TCGA-BH-A0DI-01', 'ESR1_low' ] [ 'TCGA-BH-A0DI-01', 'ESR1_low' ] True
[ 'TCGA-A2-A0T6-01', 'ESR1_low' ] [ 'TCGA-A2-A0T6-01', 'ESR1_low' ] True
[ 'TCGA-A2-A25B-01', 'ESR1_low' ] [ 'TCGA-A2-A25B-01', 'ESR1_low' ] True
[ 'TCGA-AQ-A0Y5-01', 'ESR1_low' ] [ 'TCGA-AQ-A0Y5-01', 'ESR1_low' ] True
[ 'TCGA-BH-A201-01', 'ESR1_high' ] [ 'TCGA-BH-A201-01', 'ESR1_high' ] True
[ 'TCGA-BH-A2L8-01', 'ESR1_low' ] [ 'TCGA-BH-A2L8-01', 'ESR1_low' ] True
[ 'TCGA-E9-A1N8-01', 'ESR1_low' ] [ 'TCGA-E9-A1N8-01', 'ESR1_low' ] True
[ 'TCGA-EW-A1P0-01', 'ESR1_high' ] [ 'TCGA-EW-A1P0-01', 'ESR1_high' ] True
[ 'TCGA-AO-A0JC-01', 'ESR1_low' ] [ 'TCGA-AO-A0JC-01', 'ESR1_low' ] True
[ 'TCGA-E9-A3Q9-01', 'ESR1_high' ] [ 'TCGA-E9-A3Q9-01', 'ESR1_high' ] True
[ 'TCGA-C8-A1HL-01', 'ESR1_low' ] [ 'TCGA-C8-A1HL-01', 'ESR1_low' ] True
[ 'TCGA-AR-A24W-01', 'ESR1_high' ] [ 'TCGA-AR-A24W-01', 'ESR1_high' ] True
[ 'TCGA-D8-A13Y-01', 'ESR1_low' ] [ 'TCGA-D8-A13Y-01', 'ESR1_low' ] True
[ 'TCGA-C8-A1HJ-01', 'ESR1_low' ] [ 'TCGA-C8-A1HJ-01', 'ESR1_low' ] True
[ 'TCGA-E9-A1R3-01', 'ESR1_low' ] [ 'TCGA-E9-A1R3-01', 'ESR1_low' ] True
[ 'TCGA-C8-A133-01', 'ESR1_high' ] [ 'TCGA-C8-A133-01', 'ESR1_high' ] True
[ 'TCGA-A2-A0EO-01', 'ESR1_high' ] [ 'TCGA-A2-A0EO-01', 'ESR1_high' ] True
[ 'TCGA-BH-A1EW-01', 'ESR1_high' ] [ 'TCGA-BH-A1EW-01', 'ESR1_high' ] True
[ 'TCGA-BH-A0DZ-01', 'ESR1_low' ] [ 'TCGA-BH-A0DZ-01', 'ESR1_low' ] True
[ 'TCGA-AR-A255-01', 'ESR1_high' ] [ 'TCGA-AR-A255-01', 'ESR1_high' ] True
[ 'TCGA-B6-A0IE-01', 'ESR1_low' ] [ 'TCGA-B6-A0IE-01', 'ESR1_low' ] True
[ 'TCGA-AR-A1AS-01', 'ESR1_high' ] [ 'TCGA-AR-A1AS-01', 'ESR1_high' ] True
[ 'TCGA-EW-A1PH-01', 'ESR1_low' ] [ 'TCGA-EW-A1PH-01', 'ESR1_low' ] True
[ 'TCGA-E2-A1L9-01', 'ESR1_high' ] [ 'TCGA-E2-A1L9-01', 'ESR1_high' ] True
[ 'TCGA-BH-A0BA-01', 'ESR1_high' ] [ 'TCGA-BH-A0BA-01', 'ESR1_high' ] True
[ 'TCGA-A2-A0D2-01', 'ESR1_low' ] [ 'TCGA-A2-A0D2-01', 'ESR1_low' ] True
[ 'TCGA-BH-A0B5-01', 'ESR1_low' ] [ 'TCGA-BH-A0B5-01', 'ESR1_low' ] True
[ 'TCGA-BH-A1F6-01', 'ESR1_low' ] [ 'TCGA-BH-A1F6-01', 'ESR1_low' ] True
[ 'TCGA-BH-A0HO-01', 'ESR1_high' ] [ 'TCGA-BH-A0HO-01', 'ESR1_high' ] True
[ 'TCGA-A7-A26J-01', 'ESR1_high' ] [ 'TCGA-A7-A26J-01', 'ESR1_high' ] True
[ 'TCGA-A2-A0SW-01', 'ESR1_high' ] [ 'TCGA-A2-A0SW-01', 'ESR1_high' ] True
[ 'TCGA-BH-A0W4-01', 'ESR1_low' ] [ 'TCGA-BH-A0W4-01', 'ESR1_low' ] True
[ 'TCGA-AN-A0XS-01', 'ESR1_high' ] [ 'TCGA-AN-A0XS-01', 'ESR1_high' ] True
[ 'TCGA-D8-A1JB-01', 'ESR1_high' ] [ 'TCGA-D8-A1JB-01', 'ESR1_high' ] True
[ 'TCGA-E9-A1NE-01', 'ESR1_low' ] [ 'TCGA-E9-A1NE-01', 'ESR1_low' ] True
[ 'TCGA-AR-A2LM-01', 'ESR1_high' ] [ 'TCGA-AR-A2LM-01', 'ESR1_high' ] True
[ 'TCGA-B6-A0I1-01', 'ESR1_low' ] [ 'TCGA-B6-A0I1-01', 'ESR1_low' ] True
[ 'TCGA-AO-A126-01', 'ESR1_low' ] [ 'TCGA-AO-A126-01', 'ESR1_low' ] True
[ 'TCGA-EW-A2FS-01', 'ESR1_high' ] [ 'TCGA-EW-A2FS-01', 'ESR1_high' ] True
[ 'TCGA-E2-A1LH-01', 'ESR1_low' ] [ 'TCGA-E2-A1LH-01', 'ESR1_low' ] True
[ 'TCGA-A2-A25D-01', 'ESR1_low' ] [ 'TCGA-A2-A25D-01', 'ESR1_low' ] True
[ 'TCGA-E9-A1N6-01', 'ESR1_high' ] [ 'TCGA-E9-A1N6-01', 'ESR1_high' ] True
[ 'TCGA-BH-A0W5-01', 'ESR1_high' ] [ 'TCGA-BH-A0W5-01', 'ESR1_high' ] True
[ 'TCGA-BH-A0H5-01', 'ESR1_high' ] [ 'TCGA-BH-A0H5-01', 'ESR1_high' ] True
[ 'TCGA-E9-A227-01', 'ESR1_high' ] [ 'TCGA-E9-A227-01', 'ESR1_high' ] True
[ 'TCGA-C8-A12K-01', 'ESR1_low' ] [ 'TCGA-C8-A12K-01', 'ESR1_low' ] True
[ 'TCGA-A8-A07E-01', 'ESR1_high' ] [ 'TCGA-A8-A07E-01', 'ESR1_high' ] True
[ 'TCGA-A1-A0SJ-01', 'ESR1_low' ] [ 'TCGA-A1-A0SJ-01', 'ESR1_low' ] True
[ 'TCGA-D8-A1JL-01', 'ESR1_low' ] [ 'TCGA-D8-A1JL-01', 'ESR1_low' ] True
[ 'TCGA-BH-A0H6-01', 'ESR1_high' ] [ 'TCGA-BH-A0H6-01', 'ESR1_high' ] True
[ 'TCGA-D8-A142-01', 'ESR1_low' ] [ 'TCGA-D8-A142-01', 'ESR1_low' ] True
[ 'TCGA-AO-A0JG-01', 'ESR1_high' ] [ 'TCGA-AO-A0JG-01', 'ESR1_high' ] True
[ 'TCGA-E2-A1AZ-01', 'ESR1_low' ] [ 'TCGA-E2-A1AZ-01', 'ESR1_low' ] True
[ 'TCGA-D8-A1JM-01', 'ESR1_low' ] [ 'TCGA-D8-A1JM-01', 'ESR1_low' ] True
[ 'TCGA-B6-A0X0-01', 'ESR1_high' ] [ 'TCGA-B6-A0X0-01', 'ESR1_high' ] True
[ 'TCGA-AO-A0J4-01', 'ESR1_low' ] [ 'TCGA-AO-A0J4-01', 'ESR1_low' ] True
[ 'TCGA-D8-A27H-01', 'ESR1_low' ] [ 'TCGA-D8-A27H-01', 'ESR1_low' ] True
[ 'TCGA-E9-A22G-01', 'ESR1_low' ] [ 'TCGA-E9-A22G-01', 'ESR1_low' ] True
[ 'TCGA-A7-A2KD-01', 'ESR1_high' ] [ 'TCGA-A7-A2KD-01', 'ESR1_high' ] True
[ 'TCGA-E2-A1IP-01', 'ESR1_low' ] [ 'TCGA-E2-A1IP-01', 'ESR1_low' ] True
[ 'TCGA-AO-A128-01', 'ESR1_low' ] [ 'TCGA-AO-A128-01', 'ESR1_low' ] True
[ 'TCGA-AC-A2BM-01', 'ESR1_low' ] [ 'TCGA-AC-A2BM-01', 'ESR1_low' ] True
[ 'TCGA-A8-A09A-01', 'ESR1_high' ] [ 'TCGA-A8-A09A-01', 'ESR1_high' ] True
[ 'TCGA-B6-A0RN-01', 'ESR1_high' ] [ 'TCGA-B6-A0RN-01', 'ESR1_high' ] True
[ 'TCGA-GM-A2DC-01', 'ESR1_high' ] [ 'TCGA-GM-A2DC-01', 'ESR1_high' ] True
[ 'TCGA-AN-A04A-01', 'ESR1_low' ] [ 'TCGA-AN-A04A-01', 'ESR1_low' ] True
[ 'TCGA-A8-A06Q-01', 'ESR1_high' ] [ 'TCGA-A8-A06Q-01', 'ESR1_high' ] True
[ 'TCGA-BH-A0B0-01', 'ESR1_high' ] [ 'TCGA-BH-A0B0-01', 'ESR1_high' ] True
[ 'TCGA-A8-A09D-01', 'ESR1_low' ] [ 'TCGA-A8-A09D-01', 'ESR1_low' ] True
[ 'TCGA-A7-A3IY-01', 'ESR1_high' ] [ 'TCGA-A7-A3IY-01', 'ESR1_high' ] True
[ 'TCGA-EW-A1P3-01', 'ESR1_high' ] [ 'TCGA-EW-A1P3-01', 'ESR1_high' ] True
[ 'TCGA-AR-A24V-01', 'ESR1_high' ] [ 'TCGA-AR-A24V-01', 'ESR1_high' ] True
[ 'TCGA-BH-A0BM-01', 'ESR1_high' ] [ 'TCGA-BH-A0BM-01', 'ESR1_high' ] True
[ 'TCGA-A2-A0EW-01', 'ESR1_high' ] [ 'TCGA-A2-A0EW-01', 'ESR1_high' ] True
```

```
[ 'TCGA-AO-A03V-01', 'ESR1_low' ] [ 'TCGA-AO-A03V-01', 'ESR1_low' ] True
[ 'TCGA-D8-A1JH-01', 'ESR1_high' ] [ 'TCGA-D8-A1JH-01', 'ESR1_high' ] True
[ 'TCGA-BH-A0C0-01', 'ESR1_high' ] [ 'TCGA-BH-A0C0-01', 'ESR1_high' ] True
[ 'TCGA-AR-A24X-01', 'ESR1_low' ] [ 'TCGA-AR-A24X-01', 'ESR1_low' ] True
[ 'TCGA-EW-A10W-01', 'ESR1_low' ] [ 'TCGA-EW-A10W-01', 'ESR1_low' ] True
[ 'TCGA-E2-A15D-01', 'ESR1_high' ] [ 'TCGA-E2-A15D-01', 'ESR1_high' ] True
[ 'TCGA-B6-A0IQ-01', 'ESR1_low' ] [ 'TCGA-B6-A0IQ-01', 'ESR1_low' ] True
[ 'TCGA-GM-A2DF-01', 'ESR1_low' ] [ 'TCGA-GM-A2DF-01', 'ESR1_low' ] True
[ 'TCGA-D8-A27M-01', 'ESR1_low' ] [ 'TCGA-D8-A27M-01', 'ESR1_low' ] True
[ 'TCGA-AR-A2LR-01', 'ESR1_low' ] [ 'TCGA-AR-A2LR-01', 'ESR1_low' ] True
[ 'TCGA-B6-A0RT-01', 'ESR1_low' ] [ 'TCGA-B6-A0RT-01', 'ESR1_low' ] True
[ 'TCGA-AR-A251-01', 'ESR1_low' ] [ 'TCGA-AR-A251-01', 'ESR1_low' ] True
[ 'TCGA-A2-A0YH-01', 'ESR1_high' ] [ 'TCGA-A2-A0YH-01', 'ESR1_high' ] True
[ 'TCGA-BH-A0DE-01', 'ESR1_high' ] [ 'TCGA-BH-A0DE-01', 'ESR1_high' ] True
[ 'TCGA-AC-A2FF-01', 'ESR1_low' ] [ 'TCGA-AC-A2FF-01', 'ESR1_low' ] True
[ 'TCGA-A2-A25F-01', 'ESR1_low' ] [ 'TCGA-A2-A25F-01', 'ESR1_low' ] True
[ 'TCGA-B6-A0WT-01', 'ESR1_high' ] [ 'TCGA-B6-A0WT-01', 'ESR1_high' ] True
[ 'TCGA-BH-A209-01', 'ESR1_high' ] [ 'TCGA-BH-A209-01', 'ESR1_high' ] True
[ 'TCGA-GI-A2C8-01', 'ESR1_high' ] [ 'TCGA-GI-A2C8-01', 'ESR1_high' ] True
[ 'TCGA-BH-A0EA-01', 'ESR1_high' ] [ 'TCGA-BH-A0EA-01', 'ESR1_high' ] True
[ 'TCGA-BH-A18F-01', 'ESR1_low' ] [ 'TCGA-BH-A18F-01', 'ESR1_low' ] True
[ 'TCGA-C8-A12V-01', 'ESR1_low' ] [ 'TCGA-C8-A12V-01', 'ESR1_low' ] True
[ 'TCGA-AR-A0TW-01', 'ESR1_low' ] [ 'TCGA-AR-A0TW-01', 'ESR1_low' ] True
[ 'TCGA-BH-A0H7-01', 'ESR1_high' ] [ 'TCGA-BH-A0H7-01', 'ESR1_high' ] True
[ 'TCGA-C8-A26V-01', 'ESR1_high' ] [ 'TCGA-C8-A26V-01', 'ESR1_high' ] True
[ 'TCGA-AR-A24K-01', 'ESR1_high' ] [ 'TCGA-AR-A24K-01', 'ESR1_high' ] True
[ 'TCGA-A2-A3KC-01', 'ESR1_high' ] [ 'TCGA-A2-A3KC-01', 'ESR1_high' ] True
[ 'TCGA-GM-A2DI-01', 'ESR1_high' ] [ 'TCGA-GM-A2DI-01', 'ESR1_high' ] True
[ 'TCGA-E2-A10A-01', 'ESR1_high' ] [ 'TCGA-E2-A10A-01', 'ESR1_high' ] True
[ 'TCGA-B6-A2IU-01', 'ESR1_high' ] [ 'TCGA-B6-A2IU-01', 'ESR1_high' ] True
[ 'TCGA-A7-A0DA-01', 'ESR1_low' ] [ 'TCGA-A7-A0DA-01', 'ESR1_low' ] True
[ 'TCGA-A8-A0A6-01', 'ESR1_low' ] [ 'TCGA-A8-A0A6-01', 'ESR1_low' ] True
[ 'TCGA-E2-A15S-01', 'ESR1_high' ] [ 'TCGA-E2-A15S-01', 'ESR1_high' ] True
[ 'TCGA-A2-A0YD-01', 'ESR1_high' ] [ 'TCGA-A2-A0YD-01', 'ESR1_high' ] True
[ 'TCGA-C8-A27A-01', 'ESR1_high' ] [ 'TCGA-C8-A27A-01', 'ESR1_high' ] True
[ 'TCGA-A8-A082-01', 'ESR1_high' ] [ 'TCGA-A8-A082-01', 'ESR1_high' ] True
[ 'TCGA-D8-A1JI-01', 'ESR1_high' ] [ 'TCGA-D8-A1JI-01', 'ESR1_high' ] True
[ 'TCGA-A8-A06P-01', 'ESR1_high' ] [ 'TCGA-A8-A06P-01', 'ESR1_high' ] True
[ 'TCGA-A8-A07J-01', 'ESR1_low' ] [ 'TCGA-A8-A07J-01', 'ESR1_low' ] True
[ 'TCGA-A2-A0EV-01', 'ESR1_high' ] [ 'TCGA-A2-A0EV-01', 'ESR1_high' ] True
[ 'TCGA-AO-A12C-01', 'ESR1_low' ] [ 'TCGA-AO-A12C-01', 'ESR1_low' ] True
[ 'TCGA-E2-A150-01', 'ESR1_low' ] [ 'TCGA-E2-A150-01', 'ESR1_low' ] True
[ 'TCGA-BH-A0BO-01', 'ESR1_high' ] [ 'TCGA-BH-A0BO-01', 'ESR1_high' ] True
[ 'TCGA-A2-A1FX-01', 'ESR1_high' ] [ 'TCGA-A2-A1FX-01', 'ESR1_high' ] True
[ 'TCGA-BH-A1EX-01', 'ESR1_low' ] [ 'TCGA-BH-A1EX-01', 'ESR1_low' ] True
[ 'TCGA-HN-A2NL-01', 'ESR1_low' ] [ 'TCGA-HN-A2NL-01', 'ESR1_low' ] True
[ 'TCGA-A2-A0SY-01', 'ESR1_high' ] [ 'TCGA-A2-A0SY-01', 'ESR1_high' ] True
[ 'TCGA-E2-A1IO-01', 'ESR1_low' ] [ 'TCGA-E2-A1IO-01', 'ESR1_low' ] True
[ 'TCGA-AR-A0TU-01', 'ESR1_low' ] [ 'TCGA-AR-A0TU-01', 'ESR1_low' ] True
[ 'TCGA-E2-A15K-01', 'ESR1_high' ] [ 'TCGA-E2-A15K-01', 'ESR1_high' ] True
[ 'TCGA-C8-A12X-01', 'ESR1_high' ] [ 'TCGA-C8-A12X-01', 'ESR1_high' ] True
[ 'TCGA-E9-A226-01', 'ESR1_high' ] [ 'TCGA-E9-A226-01', 'ESR1_high' ] True
[ 'TCGA-A1-A0SD-01', 'ESR1_high' ] [ 'TCGA-A1-A0SD-01', 'ESR1_high' ] True
[ 'TCGA-BH-A18U-01', 'ESR1_high' ] [ 'TCGA-BH-A18U-01', 'ESR1_high' ] True
[ 'TCGA-AO-A03N-01', 'ESR1_low' ] [ 'TCGA-AO-A03N-01', 'ESR1_low' ] True
[ 'TCGA-AO-A0JL-01', 'ESR1_low' ] [ 'TCGA-AO-A0JL-01', 'ESR1_low' ] True
[ 'TCGA-D8-A1JD-01', 'ESR1_high' ] [ 'TCGA-D8-A1JD-01', 'ESR1_high' ] True
[ 'TCGA-AR-A24R-01', 'ESR1_high' ] [ 'TCGA-AR-A24R-01', 'ESR1_high' ] True
[ 'TCGA-EW-A1IY-01', 'ESR1_high' ] [ 'TCGA-EW-A1IY-01', 'ESR1_high' ] True
[ 'TCGA-EW-A1PB-01', 'ESR1_low' ] [ 'TCGA-EW-A1PB-01', 'ESR1_low' ] True
[ 'TCGA-AN-A0FJ-01', 'ESR1_low' ] [ 'TCGA-AN-A0FJ-01', 'ESR1_low' ] True
[ 'TCGA-AN-A04D-01', 'ESR1_low' ] [ 'TCGA-AN-A04D-01', 'ESR1_low' ] True
[ 'TCGA-A2-A0YM-01', 'ESR1_low' ] [ 'TCGA-A2-A0YM-01', 'ESR1_low' ] True
[ 'TCGA-A8-A08Z-01', 'ESR1_low' ] [ 'TCGA-A8-A08Z-01', 'ESR1_low' ] True
[ 'TCGA-AN-A041-01', 'ESR1_high' ] [ 'TCGA-AN-A041-01', 'ESR1_high' ] True
[ 'TCGA-A2-A0D3-01', 'ESR1_high' ] [ 'TCGA-A2-A0D3-01', 'ESR1_high' ] True
[ 'TCGA-GM-A3NY-01', 'ESR1_high' ] [ 'TCGA-GM-A3NY-01', 'ESR1_high' ] True
[ 'TCGA-AR-A1AK-01', 'ESR1_high' ] [ 'TCGA-AR-A1AK-01', 'ESR1_high' ] True
[ 'TCGA-A2-A25E-01', 'ESR1_high' ] [ 'TCGA-A2-A25E-01', 'ESR1_high' ] True
[ 'TCGA-A8-A09V-01', 'ESR1_high' ] [ 'TCGA-A8-A09V-01', 'ESR1_high' ] True
[ 'TCGA-A2-A0CT-01', 'ESR1_high' ] [ 'TCGA-A2-A0CT-01', 'ESR1_high' ] True
[ 'TCGA-E9-A22E-01', 'ESR1_high' ] [ 'TCGA-E9-A22E-01', 'ESR1_high' ] True
[ 'TCGA-BH-A18Q-01', 'ESR1_low' ] [ 'TCGA-BH-A18Q-01', 'ESR1_low' ] True
[ 'TCGA-AN-A0XW-01', 'ESR1_low' ] [ 'TCGA-AN-A0XW-01', 'ESR1_low' ] True
[ 'TCGA-B6-A0WX-01', 'ESR1_low' ] [ 'TCGA-B6-A0WX-01', 'ESR1_low' ] True
[ 'TCGA-AO-A1KR-01', 'ESR1_low' ] [ 'TCGA-AO-A1KR-01', 'ESR1_low' ] True
[ 'TCGA-EW-A1PD-01', 'ESR1_high' ] [ 'TCGA-EW-A1PD-01', 'ESR1_high' ] True
[ 'TCGA-EW-A1PC-01', 'ESR1_low' ] [ 'TCGA-EW-A1PC-01', 'ESR1_low' ] True
[ 'TCGA-E2-A14N-01', 'ESR1_low' ] [ 'TCGA-E2-A14N-01', 'ESR1_low' ] True
[ 'TCGA-E9-A249-01', 'ESR1_high' ] [ 'TCGA-E9-A249-01', 'ESR1_high' ] True
[ 'TCGA-D8-A27R-01', 'ESR1_low' ] [ 'TCGA-D8-A27R-01', 'ESR1_low' ] True
[ 'TCGA-A2-A0T0-01', 'ESR1_low' ] [ 'TCGA-A2-A0T0-01', 'ESR1_low' ] True
[ 'TCGA-AN-A0FT-01', 'ESR1_low' ] [ 'TCGA-AN-A0FT-01', 'ESR1_low' ] True
[ 'TCGA-D8-A1XL-01', 'ESR1_low' ] [ 'TCGA-D8-A1XL-01', 'ESR1_low' ] True
[ 'TCGA-C8-A12N-01', 'ESR1_high' ] [ 'TCGA-C8-A12N-01', 'ESR1_high' ] True
[ 'TCGA-B6-A1KN-01', 'ESR1_high' ] [ 'TCGA-B6-A1KN-01', 'ESR1_high' ] True
[ 'TCGA-EW-A1OX-01', 'ESR1_high' ] [ 'TCGA-EW-A1OX-01', 'ESR1_high' ] True
[ 'TCGA-E9-A22A-01', 'ESR1_low' ] [ 'TCGA-E9-A22A-01', 'ESR1_low' ] True
[ 'TCGA-C8-A1HO-01', 'ESR1_high' ] [ 'TCGA-C8-A1HO-01', 'ESR1_high' ] True
[ 'TCGA-E2-A15J-01', 'ESR1_low' ] [ 'TCGA-E2-A15J-01', 'ESR1_low' ] True
[ 'TCGA-D8-A27G-01', 'ESR1_low' ] [ 'TCGA-D8-A27G-01', 'ESR1_low' ] True
[ 'TCGA-AN-A0FY-01', 'ESR1_high' ] [ 'TCGA-AN-A0FY-01', 'ESR1_high' ] True
[ 'TCGA-D8-A27P-01', 'ESR1_low' ] [ 'TCGA-D8-A27P-01', 'ESR1_low' ] True
[ 'TCGA-BH-A0DG-01', 'ESR1_low' ] [ 'TCGA-BH-A0DG-01', 'ESR1_low' ] True
[ 'TCGA-A8-A09W-01', 'ESR1_high' ] [ 'TCGA-A8-A09W-01', 'ESR1_high' ] True
[ 'TCGA-AR-A252-01', 'ESR1_high' ] [ 'TCGA-AR-A252-01', 'ESR1_high' ] True
[ 'TCGA-AO-A1KT-01', 'ESR1_high' ] [ 'TCGA-AO-A1KT-01', 'ESR1_high' ] True
[ 'TCGA-E2-A153-01', 'ESR1_low' ] [ 'TCGA-E2-A153-01', 'ESR1_low' ] True
```

```
['TCGA-A2-A0YG-01', 'ESR1_high'] ['TCGA-A2-A0YG-01', 'ESR1_high'] True
['TCGA-BH-A0BL-01', 'ESR1_low'] ['TCGA-BH-A0BL-01', 'ESR1_low'] True
['TCGA-BH-A0HY-01', 'ESR1_high'] ['TCGA-BH-A0HY-01', 'ESR1_high'] True
['TCGA-AO-A0JA-01', 'ESR1_high'] ['TCGA-AO-A0JA-01', 'ESR1_high'] True
200 randomly generated data found in the file.
0 randomly generated data not found in the file.
```

Check 3 : Is the ordering of the cls file consistent with gct

Check 3 Steps

* Get Random Sample ID and Phenotype from .tsv file.

*Find the Sample ID in .gct file

*Get the location/position of the file in .gct file

*Check the phenotype in same position in .cls file

*Compare the value from .tsv and .cls file

Check 3

Check output from script 2 (BasalLum)

```
In [9]: import random
file1_content = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_2/sample_500.gct', 'r')
file2_content = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/BasalLum.tsv", 'r')
cls_content = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_2/sample_500.cls', 'r')
Number_of_test_samples = 500 #200 #300 #400 #500
for pos, value in enumerate(file1_content):
    if pos == 2:
        pheno_type_gct = value.split()[2:]
phenotype_tsv = [x.split() for pos,x in enumerate(file2_content) if pos > 0]
random_tsv = random.sample(phenotype_tsv,Number_of_test_samples)
gct_random_location = []
for pos,value in enumerate(pheno_type_gct):
    for random_val in random_tsv:
        if value == random_val[0]:
            if random_val[1] == 'Lum':classX = 0
            else: classX = 1
            gct_random_location.append((pos,value,classX))

index = 0
for valx in cls_content:
    Lum_Basal = valx.split()
    for location in gct_random_location:
        counter = False
        if int(Lum_Basal[location[0]]) == location[2]:
            counter = True
            print("CLS_data : ",Lum_Basal[location[0]],"GCT_random_phenotype : ", location[2] ,"Check : ", counter)
        else:
            print(counter)
            index += 1
print(index)
#print(gct_random_location)
#print(Lum_Basal)
```

```
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 1 GCT_random_phenotype : 1 Check : True
CLS_data : 1 GCT_random_phenotype : 1 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 1 GCT_random_phenotype : 1 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 1 GCT_random_phenotype : 1 Check : True
CLS_data : 1 GCT_random_phenotype : 1 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
CLS_data : 0 GCT_random_phenotype : 0 Check : True
```

Page 19 of 30

Page 20 of 30

Page 21 of 30

Page 22 of 30

Check output from Script 3 (ESR1)

In [15]:

```
import random
file1_content = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_3/sample_500.gct', 'r')
file2_content = open("/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/ESR1.tsv", 'r')
cls_content = open('/Users/rausharm/Desktop/UCSC_Xena/GSEA_FILES/Output_3/sample_500.cls', 'r')
Number_of_test_samples = 500 #200 #300 #400 #500

for pos, value in enumerate(file1_content):
    if pos == 2:
        gct_file = value.split()[2:]
phenotype_tsv = [x.split() for pos,x in enumerate(file2_content) if pos > 0]
index = len(phenotype_tsv) / 2

#configuring the ESR1 file with ESR1_low and ESR1_high
for pos, value in enumerate(phenotype_tsv):
    if pos <= index - 1:
        phenotype_tsv[pos] = [value[0], 'ESR1_low']
    else:
        phenotype_tsv[pos] = [value[0], 'ESR1_high']

#generating random in tsv file

random_tsv = random.sample(phenotype_tsv,Number_of_test_samples)
#print(random_tsv)

gct_random_location = [] # index position of Phenotype sample and expected data
#Finding the location of the random samples in gct file
for pos,value in enumerate(gct_file):
    for random_val in random_tsv:
        if value == random_val[0]:
            if random_val[1] == 'ESR1_low':classX = 0
            else: classX = 1
            gct_random_location.append((pos,value,classX))#(position in gct file, sampleID, expected Phenotype)
print(gct_random_location)

#comparing the phenotype value in cls file getting the location from corresponding gct file
for valx in cls_content:
    counter = False
    ESR1 = valx.split()
    index_counter = 0
    for location in gct_random_location:
        if int(ESR1[location[0]]) == location[2]: #comparing the value from ESR1 and gct_random_location
            counter = True # if the values are different then the counter is False
            print("CLS_data : ",ESR1[location[0]],"GCT_random_phenotype : ", location[2] ,"Check : ", counter)
        else:
            counter = False
            index_counter += 1
            print(location, counter)

print(index_counter)
#print(gct_random_location)
```

[(0, 'TCGA-BH-A0BQ-01', 0), (1, 'TCGA-BH-A0BT-01', 1), (2, 'TCGA-BH-A18J-01', 0), (3, 'TCGA-BH-A0W7-01', 0), (4, 'TCGA-BH-A0HA-01', 0), (5, 'TCGA-A7-A0CE-01', 0), (6, 'TCGA-AR-A0U1-01', 0), (7, 'TCGA-EW-A1OZ-01', 0), (8, 'TCGA-A2-A0EY-01', 1), (9, 'TCGA-A8-A09R-01', 1), (10, 'TCGA-E2-A1II-01', 0), (11, 'TCGA-C8-A273-01', 1), (12, 'TCGA-BH-A1EO-01', 1), (13, 'TCGA-AN-A0AS-01', 0), (14, 'TCGA-A2-A1G0-01', 0), (15, 'TCGA-B6-A0I8-01', 0), (16, 'TCGA-AR-A24S-01', 1), (17, 'TCGA-D8-A1XR-01', 1), (18, 'TCGA-E2-A1IJ-01', 1), (19, 'TCGA-BH-A0AY-01', 0), (20, 'TCGA-B6-A0IH-01', 1), (21, 'TCGA-A1-A0SH-01', 0), (22, 'TCGA-AR-A1AQ-01', 0), (23, 'TCGA-A1-A0SN-01', 1), (24, 'TCGA-BH-A18T-01', 0), (25, 'TCGA-E9-A22B-01', 1), (26, 'TCGA-AO-A0J9-01', 1), (27, 'TCGA-A1-A0SE-01', 0), (28, 'TCGA-A2-A04U-01', 0), (29, 'TCGA-D8-A1Y2-01', 1), (30, 'TCGA-E9-A229-01', 0), (31, 'TCGA-BH-A0HK-01', 1), (32, 'TCGA-E2-A14Z-01', 1), (33, 'TCGA-D8-A1Y1-01', 1), (34, 'TCGA-A2-A04P-01', 0), (35, 'TCGA-A7-A13E-01', 0), (36, 'TCGA-A7-A0D9-01', 1), (37, 'TCGA-E2-A14R-01', 0), (38, 'TCGA-BH-A0E2-01', 1), (39, 'TCGA-E2-A1L7-01', 1), (40, 'TCGA-GM-A2DM-01', 1), (41, 'TCGA-A2-A0D4-01', 1), (42, 'TCGA-AO-A12F-01', 0), (43, 'TCGA-AC-A3HN-01', 1), (44, 'TCGA-A2-A25C-01', 0), (45, 'TCGA-E9-A1N4-01', 0), (46, 'TCGA-AQ-A1H3-01', 1), (47, 'TCGA-E9-A295-01', 0), (48, 'TCGA-D8-A1XD-01', 1), (49, 'TCGA-AC-A2FB-01', 0), (50, 'TCGA-BH-A0DQ-01', 1), (51, 'TCGA-BH-A0BW-01', 0), (52, 'TCGA-A8-A0A4-01', 1), (53, 'TCGA-D8-A1JS-01', 1), (54, 'TCGA-AR-A1AJ-01', 0), (55, 'TCGA-AN-A0AK-01', 1), (56, 'TCGA-BH-A0BZ-01', 1), (57, 'TCGA-AR-A24P-01', 1), (58, 'TCGA-BH-A0DI-01', 0), (59, 'TCGA-A2-A0T6-01', 0), (60, 'TCGA-A2-A25B-01', 0), (61, 'TCGA-A2-A0SV-01', 0), (62, 'TCGA-BH-A18G-01', 0), (63, 'TCGA-BH-A1FC-01', 0), (64, 'TCGA-E2-A14W-01', 1), (65, 'TCGA-AO-A124-01', 0), (66, 'TCGA-A8-A075-01', 0), (67, 'TCGA-AQ-A0Y5-01', 0), (68, 'TCGA-A8-A07C-01', 0), (69, 'TCGA-BH-A201-01', 1), (70, 'TCGA-AR-A24H-01', 1), (71, 'TCGA-BH-A0WA-01', 0), (72, 'TCGA-A8-A08P-01', 1), (73, 'TCGA-BH-A2L8-01', 0), (74, 'TCGA-BH-A0DK-01', 0), (75, 'TCGA-A2-A0T2-01', 0), (76, 'TCGA-E2-A1IF-01', 1), (77, 'TCGA-E9-A1N8-01', 0), (78, 'TCGA-EW-A1P0-01', 1), (79, 'TCGA-E9-A244-01', 0), (80, 'TCGA-AO-A0JD-01', 1), (81, 'TCGA-A1-A0SO-01', 0), (82, 'TCGA-A8-A09T-01', 1), (83, 'TCGA-BH-A18I-01', 1), (84, 'TCGA-B6-A0RU-01', 0), (85, 'TCGA-AO-A0JC-01', 0), (86, 'TCGA-E9-A3Q9-01', 1), (87, 'TCGA-C8-A1HL-01', 0), (88, 'TCGA-AR-A24W-01', 1), (89, 'TCGA-D8-A13Y-01', 0), (90, 'TCGA-AR-A2LO-01', 0), (91, 'TCGA-E2-A1LS-01', 0), (92, 'TCGA-AR-A0TS-01', 0), (93, 'TCGA-AR-A0TT-01', 0), (94, 'TCGA-D8-A1JJ-01', 1), (95, 'TCGA-BH-A1ES-01', 1), (96, 'TCGA-C8-A1HJ-01', 0), (97, 'TCGA-BH-A0BR-01', 0), (98, 'TCGA-AO-A12E-01', 0), (99, 'TCGA-A2-A0CM-01', 0), (100, 'TCGA-E9-A1RB-01', 1), (101, 'TCGA-E9-A1R3-01', 0), (102, 'TCGA-D8-A1JK-01', 0), (103, 'TCGA-D8-A27L-01', 1), (104, 'TCGA-C8-A133-01', 1), (105, 'TCGA-A2-A0EO-01', 1), (106, 'TCGA-A2-A0YL-01', 0), (107, 'TCGA-AR-A1AY-01', 0), (108, 'TCGA-AR-A24L-01', 1), (109, 'TCGA-AR-A0U0-01', 0), (110, 'TCGA-BH-A1EW-01', 1), (111, 'TCGA-AN-A0FF-01', 1), (112, 'TCGA-AN-A0FX-01', 0), (113, 'TCGA-BH-A0DZ-01', 0), (114, 'TCGA-E2-A1LI-01', 0), (115, 'TCGA-AR-A255-01', 1), (116, 'TCGA-B6-A0IE-01', 0), (117, 'TCGA-A8-A07F-01', 1), (118, 'TCGA-AR-A1AS-01', 1), (119, 'TCGA-EW-A1PH-01', 0), (120, 'TCGA-BH-A0HP-01', 1), (121, 'TCGA-C8-A1HG-01', 1), (122, 'TCGA-E2-A1L9-01', 1), (123, 'TCGA-BH-A0BA-01', 1), (124, 'TCGA-D8-A1XY-01', 1), (125, 'TCGA-A2-A0D2-01', 0), (126, 'TCGA-AR-A1AL-01', 1), (127, 'TCGA-AO-A0J6-01', 0), (128, 'TCGA-D8-A1J9-01', 0), (129, 'TCGA-BH-A0B5-01', 0), (130, 'TCGA-BH-A0B4-01', 1), (131, 'TCGA-AR-A1AX-01', 0), (132, 'TCGA-B6-A0IN-01', 1), (133, 'TCGA-BH-A1F6-01', 0), (134, 'TCGA-A8-A07U-01', 0), (135, 'TCGA-BH-A18V-01', 0), (136, 'TCGA-AQ-A04L-01', 0), (137, 'TCGA-AR-A1AP-01', 1), (138, 'TCGA-AC-A2QH-01', 0), (139, 'TCGA-BH-A0HO-01', 1), (140, 'TCGA-E2-A1L8-01', 1), (141, 'TCGA-BH-A18H-01', 1), (142, 'TCGA-BH-A0HU-01', 1), (143, 'TCGA-A7-A26J-01', 1), (144, 'TCGA-A2-A0ST-01', 0), (145, 'TCGA-D8-A27W-01', 1), (146, 'TCGA-A2-A0SW-01', 1), (147, 'TCGA-GM-A2DL-01', 1), (148, 'TCGA-E2-A159-01', 0), (149, 'TCGA-BH-A0W4-01', 0), (150, 'TCGA-D8-A1Y3-01', 0), (151, 'TCGA-A2-A0T5-01', 1), (152, 'TCGA-AN-A0XS-01', 1), (153, 'TCGA-D8-A1JB-01', 1), (154, 'TCGA-B6-A0RE-01', 0), (155, 'TCGA-C8-A134-01', 0), (156, 'TCGA-AN-A0AT-01', 0), (157, 'TCGA-GM-A2DN-01', 1), (158, 'TCGA-BH-A0HX-01', 1), (159, 'TCGA-E9-A1NE-01', 0), (160, 'TCGA-BH-A0BP-01', 1), (161, 'TCGA-AR-A2LM-01', 1), (162, 'TCGA-B6-A0I1-01', 0), (163, 'TCGA-A2-A04R-01', 1), (164, 'TCGA-AO-A126-01', 0), (165, 'TCGA-AO-A0JM-01', 0), (166, 'TCGA-EW-A2FS-01', 1), (167, 'TCGA-E2-A1LH-01', 0), (168, 'TCGA-A2-A25D-01', 0), (169, 'TCGA-E9-A243-01', 0), (170, 'TCGA-E9-A1N6-01', 1), (171, 'TCGA-BH-A0W5-01', 1), (172, 'TCGA-E2-A15I-01', 1), (173, 'TCGA-BH-A0H5-01', 1), (174, 'TCGA-E9-A227-01', 1), (175, 'TCGA-EW-A2FV-01', 0), (176, 'TCGA-BH-A0BG-01', 0), (177, 'TCGA-E2-A1LA-01', 1), (178, 'TCGA-A2-A04Y-01', 1), (179, 'TCGA-AR-A240-01', 1), (180, 'TCGA-GM-A2DA-01', 0), (181, 'TCGA-AO-A12A-01', 1), (182, 'TCGA-EW-A1IW-01', 1), (183, 'TCGA-C8-A12K-01', 0), (184, 'TCGA-E9-A1NA-01', 1), (185, 'TCGA-A8-A07E-01', 1), (186, 'TCGA-BH-A18K-01', 1), (187, 'TCGA-A1-A0SJ-01', 0), (188, 'TCGA-D8-A1JL-01', 0), (189, 'TCGA-A8-A097-01', 1), (190, 'TCGA-AN-A0FZ-01', 1), (191, 'TCGA-BH-A0E6-01', 0), (192, 'TCGA-C8-A132-01', 1), (193, 'TCGA-BH-A0H6-01', 1), (194, 'TCGA-D8-A1XS-01', 1), (195, 'TCGA-C8-A12Y-01', 1), (196, 'TCGA-E9-A1QZ-01', 1), (197, 'TCGA-GM-A2D9-01', 1), (198, 'TCGA-AN-A0XU-01', 0), (199, 'TCGA-D8-A142-01', 0), (200, 'TCGA-EW-A1J2-01', 1), (201, 'TCGA-BH-A0HF-01', 1), (202, 'TCGA-AO-A0JG-01', 1), (203, 'TCGA-C8-A131-01', 0), (204, 'TCGA-E2-A1AZ-01', 0), (205, 'TCGA-E9-A3HO-01', 1), (206, 'TCGA-D8-A1JM-01', 0), (207, 'TCGA-BH-A0BF-01', 0), (208, 'TCGA-BH-A

Page 25 of 30

Page 26 of 30

Page 27 of 30

Page 28 of 30

Page 29 of 30

