Observing Similarities in Tumor formation from Cap Analysis Gene Expression

Problem:

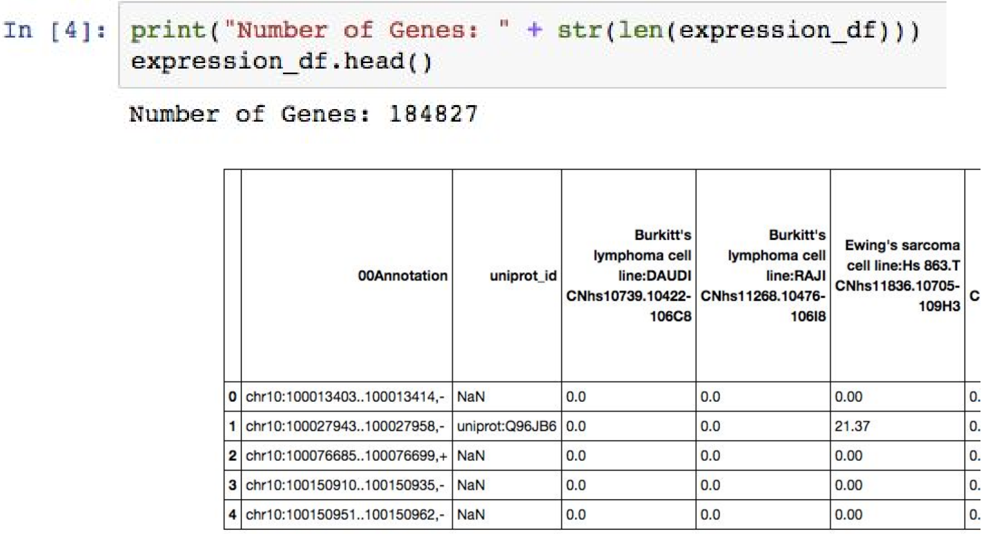
Cancer is a disease that is affecting millions of people worldwide. As the scientific community continues to progress towards further understanding of how cancer forms, many questions still remain especially in regards to the genetic basis of cancer. Further developing our understanding of gene expression profiles in various forms of cancer could greatly shape the future of how cancer is attacked at a genetic level moving forward.

Purpose:

The purpose of this project is to find out whether or not there are overlapping gene expression profiles between various types of tumors, both benign and malignant. Finding similarities in gene expression between different types of cancers could allow scientists to dig deeper beyond those similarities into why these genes are being expressed in multiple types of cancer. Observing this could also allow scientists to spend more time on genes that are showing more importance in the gene expression landscape of cancer and provide a more targeted approach that could possibly lead us to faster cures.

Data:

The dataset was acquired from Cap analysis gene expression (CAGE) from human samples by the FANTOM5 consortium. The initial dataset contains 247 different experiments/cell types as the columns and 184,827 different genes and as the rows. For this project I will be looking at both the matrix and the transpose where the genes are the samples and the cell types are the variables in the former, and the cell types are the samples and genes are the variables in the latter.



```
In [4]: print("Number of Genes: " + str(len(expression_df)))
        expression_df.head()
```

```
Number of Genes: 184827
```

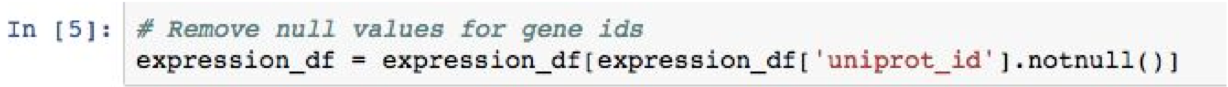
	00Annotation	uniprot_id	Burkitt's lymphoma cell line:DAUDI CNhs10739.10422-106C8	Burkitt's lymphoma cell line:RAJI CNhs11268.10476-106I8	Ewing's sarcoma cell line:Hs 863.T CNhs11836.10705-109H3	C
0	chr10:100013403..100013414,-	NaN	0.0	0.0	0.00	0.
1	chr10:100027943..100027958,-	uniprot:Q96JB6	0.0	0.0	21.37	0.
2	chr10:100076685..100076699,+	NaN	0.0	0.0	0.00	0.
3	chr10:100150910..100150935,-	NaN	0.0	0.0	0.00	0.
4	chr10:100150951..100150962,-	NaN	0.0	0.0	0.00	0.

Wrangling:

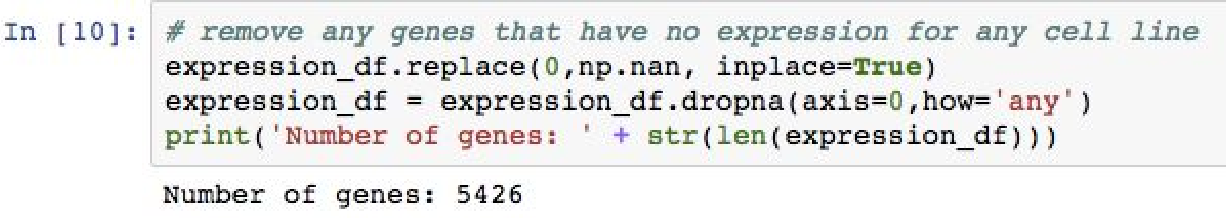
Since the data was accumulated from a number of sources and contained high variability, I applied different wrangling and cleaning techniques using pandas to remove noise and produce a dataset from which strong biological signals could be recovered. No normalization was required since the tags-per-million count is already normalized by the total number of reads in a given experiment. I experimented with various thresholds for dropping genes by computing basic statistics at each stage, and settled on the following procedure.

Missing Values

Many values in the matrix were zeros, indicating that for a given experiment, the gene transcript was not observed at all. This is unexpected and most likely due to anomalies in the experiments where either the tags are inaccurate or the transcript was not tested. In order to avoid imputing values or determining a safe cut-off for number of zeros tolerated per gene, I dropped all genes with any zero values.



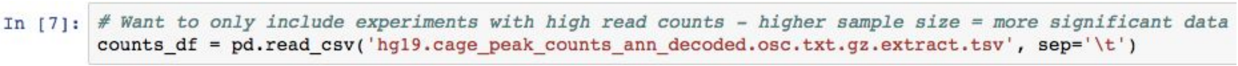
```
In [5]: # Remove null values for gene ids
        expression_df = expression_df[expression_df['uniprot_id'].notnull()]
```



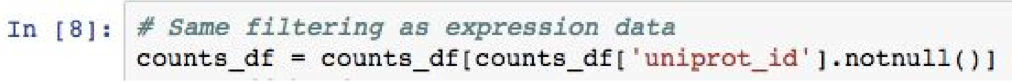
```
In [10]: # remove any genes that have no expression for any cell line
         expression_df.replace(0,np.nan, inplace=True)
         expression_df = expression_df.dropna(axis=0,how='any')
         print('Number of genes: ' + str(len(expression_df)))

         Number of genes: 5426
```

Total Reads

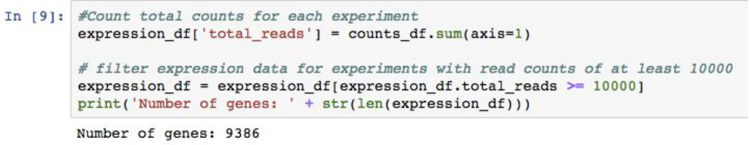
The number of total reads in the experiments varied significantly. I first uploaded the identical dataset from the FANTOM5 consortium, but instead of choosing tags per million I chose read counts as the variable counted. I removed all experiments with fewer than 10,000 reads, as low read count introduces uncertainty.



```
In [7]: # Want to only include experiments with high read counts - higher sample size = more significant data
        counts_df = pd.read_csv('hg19.cage_peak_counts_ann_decoded.osc.txt.gz.extract.tsv', sep='\t')
```



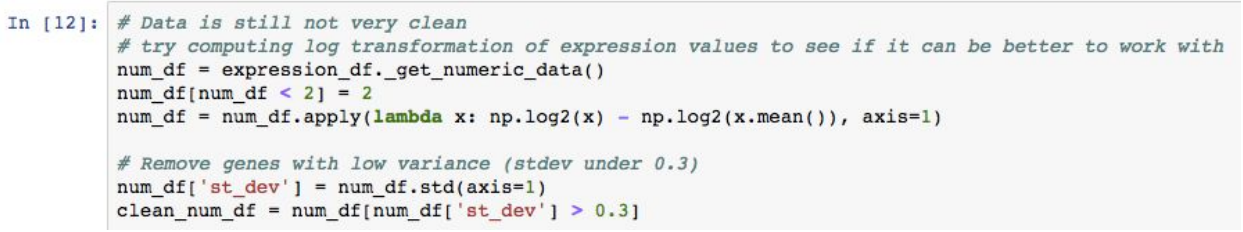
```
In [8]: # Same filtering as expression data
        counts_df = counts_df[counts_df['uniprot_id'].notnull()]
```



```
In [9]: #Count total counts for each experiment
        expression_df['total_reads'] = counts_df.sum(axis=1)

        # filter expression data for experiments with read counts of at least 10000
        expression_df = expression_df[expression_df.total_reads >= 10000]
        print('Number of genes: ' + str(len(expression_df)))

        Number of genes: 9386
```

Transformation

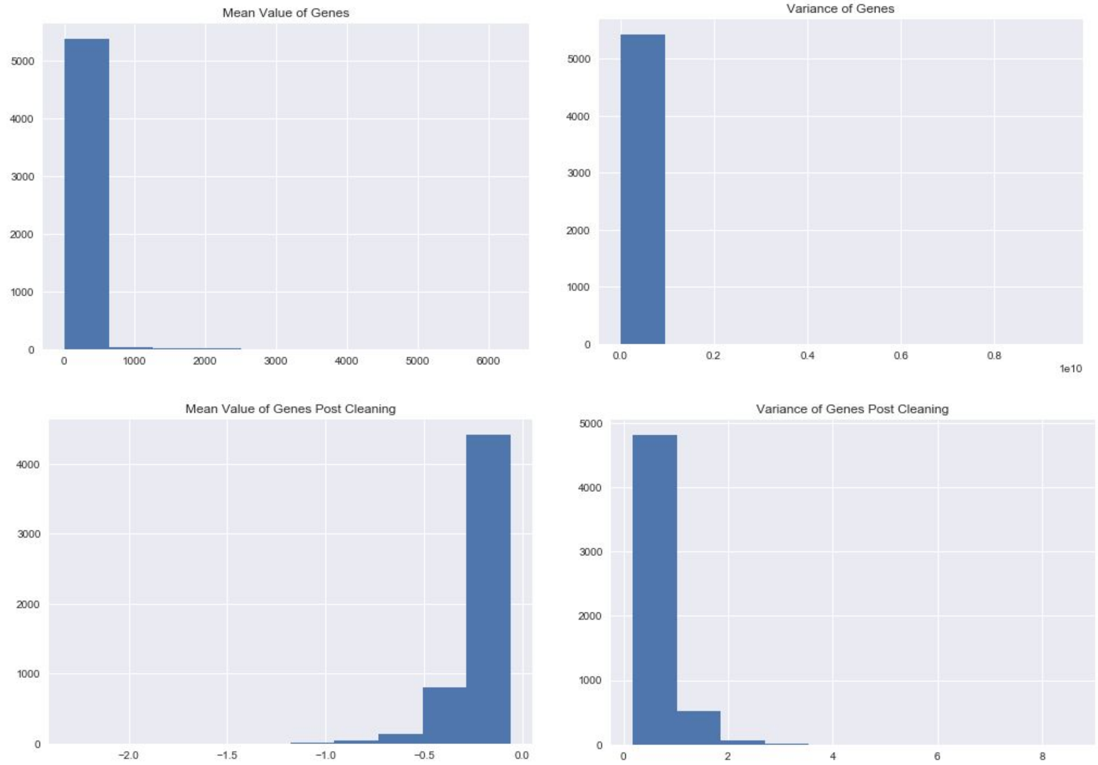
Although no normalization was required, log2 transformation is common in most gene expression datasets to bring the data closer to a normal distribution. I computed the log2 transform of the data, first changing values less than two to two. I then computed the log fold change log(xgene,exp) − log(meangene(experiments)). I removed genes with low variance (standard deviation < 0.3).

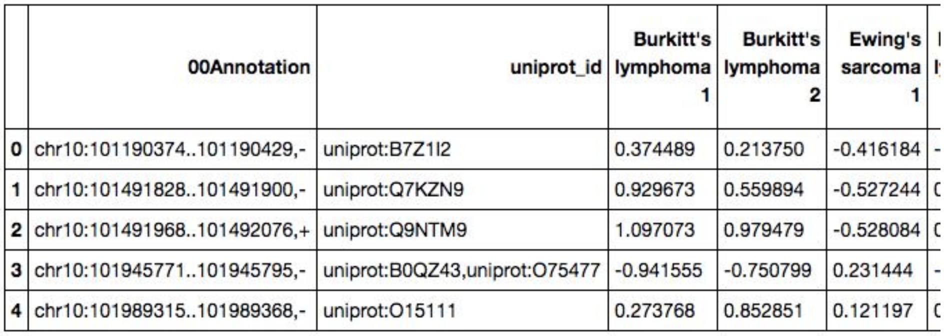


```
In [12]: # Data is still not very clean
         # try computing log transformation of expression values to see if it can be better to work with
         num_df = expression_df._get_numeric_data()
         num_df[num_df < 2] = 2
         num_df = num_df.apply(lambda x: np.log2(x) - np.log2(x.mean()), axis=1)

         # Remove genes with low variance (stdev under 0.3)
         num_df['st_dev'] = num_df.std(axis=1)
         clean_num_df = num_df[num_df['st_dev'] > 0.3]
```

Final Data

After all stages of filtering, the remaining data consisted of 5426 genes and 247 experiments or cell types. To determine the quality of this filtered dataset, I examined the basic statistics. These statistics included the mean, median, minimum values, maximum values, standard deviation, and the variance. The filtered data has a cleaner distribution than the original data





	00Annotation	uniprot_id	Burkitt's lymphoma 1	Burkitt's lymphoma 2	Ewing's sarcoma 1
0	chr10:101190374..101190429,-	uniprot:B7Z1I2	0.374489	0.213750	-0.416184
1	chr10:101491828..101491900,-	uniprot:Q7KZN9	0.929673	0.559894	-0.527244
2	chr10:101491968..101492076,+	uniprot:Q9NTM9	1.097073	0.979479	-0.528084
3	chr10:101945771..101945795,-	uniprot:B0QZ43,uniprot:O75477	-0.941555	-0.750799	0.231444
4	chr10:101989315..101989368,-	uniprot:O15111	0.273768	0.852851	0.121197

Inferential Statistics and Exploratory Data Analysis

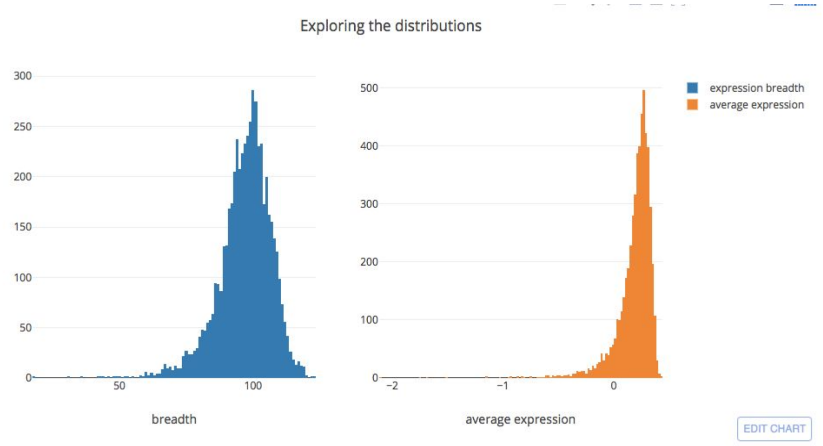
Inferential Statistics

The data is much cleaner after the in depth wrangling process, but still not ideal. I first used a 1-sample t-test to find any statistically significant genes. The null hypothesis for each gene is that the different expression profile for each cell type is due to random chance. For each gene's t-statistic I received a p-value of 1, showing that the null hypothesis could not be rejected and each gene's expression profile across all cell types could have been to random chance.

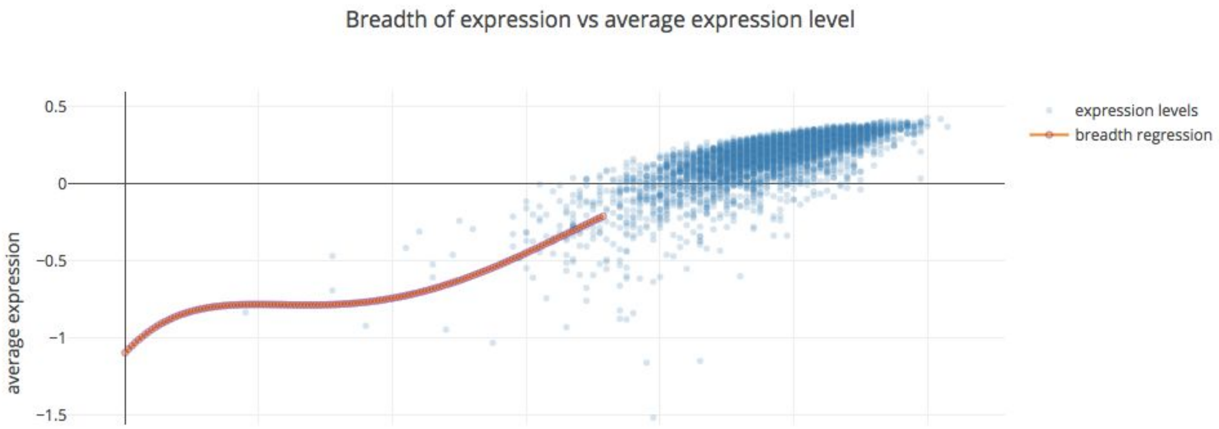
Exploratory Data Analysis

At this point it was becoming clear that the data was going to be very tough to analyze and make inferences due to it's lack of symmetrical distribution. To counter this I thought of using interactive visualizations that would allow one to observe interesting points in data by simply hovering over with the mouse. I chose plotly due to having prior experience with it in a much smaller gene expression dataset.

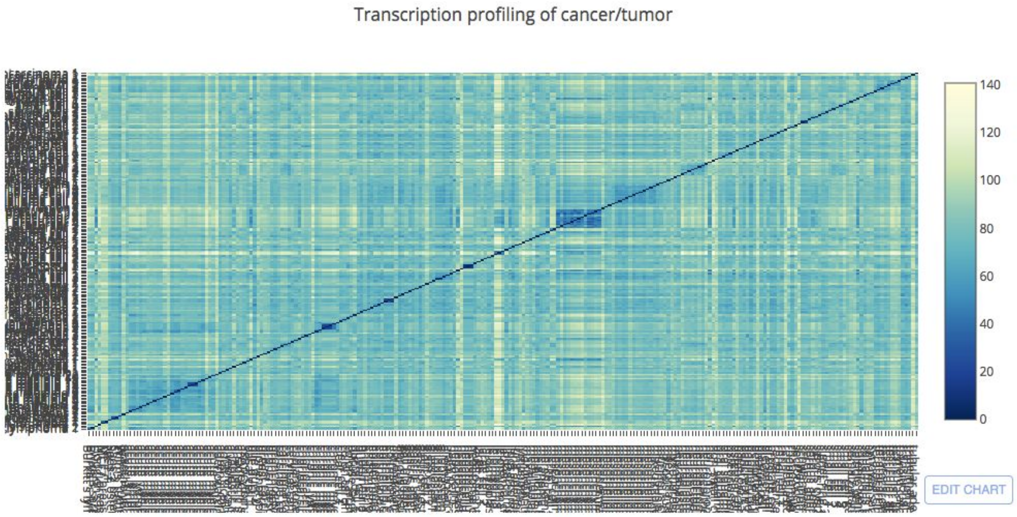
The first visualization I created was a side-by-side histogram of the genes' expression breadth – in how many cell types are the genes' expressed as well as the average gene expression across all samples. Both charts showed abnormal distribution. The average gene expression appeared to be very low for most genes.



The second visualization I created was a scatter plot with a trend line. I continued to zero in on breadth and expression and plotted the relationship between the two. The average expression levels were scattered with the breadth regression line. This chart showed that the average expression was fairly random and the clear correlation between breadth and average expression that one would expect.

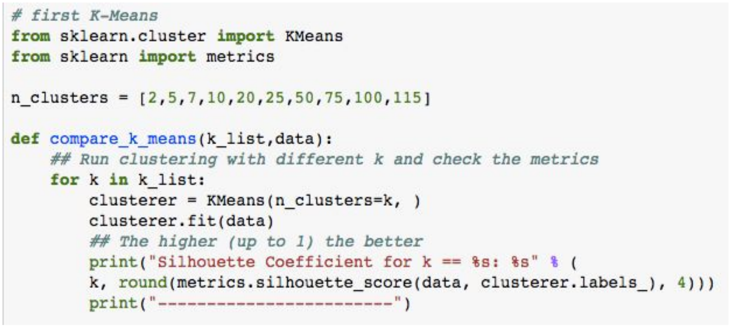


The last visualization I created was the most important one, a heatmap of the gene expression that would be able to show clear outliers and give a preview on whether or not the clustering that I was planning to do had any potential to show something interesting. The metric of the heatmap was euclidean distance, cell types that were closer in gene expression would be more blue, and cell types that were further in gene expression would be more yellow. The vast majority of the heatmap was yellow, with very few and tiny blue spots scattered in random areas that appear to be anomalies.

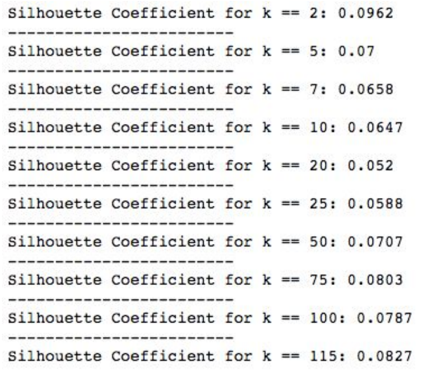


Clustering Models and Results

Although the heatmap of the gene expression showed less than promising results, I was looking to validate the non-existent patterns within the data with a few forms of clustering. I chose to cluster with the cell types as the samples first. The first clustering model I observed was K-means clustering. Again, my expectations were not high utilizing this model after observing the heatmap. The scoring metric that I used was the silhouette coefficient. I used this metric because it accurately depicts the separation distance between clusters. Silhouette coefficients near 1 indicate that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster. For my values of k I used 2, 5, 7, 10, 20, 25, 50, 75, 100, 115 with 115 being the number of different cell types. All of the silhouette scores for each value of k were close to zero, with the highest ones being around 0.1.

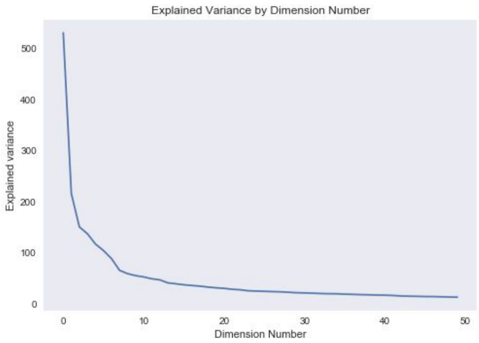


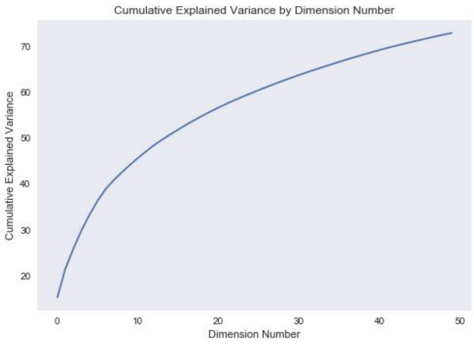
```
# first K-Means
from sklearn.cluster import KMeans
from sklearn import metrics

n_clusters = [2,5,7,10,20,25,50,75,100,115]

def compare_k_means(k_list,data):
    ## Run clustering with different k and check the metrics
    for k in k_list:
        clusterer = KMeans(n_clusters=k, )
        clusterer.fit(data)
        ## The higher (up to 1) the better
        print("Silhouette Coefficient for k == %s: %s" % (
        k, round(metrics.silhouette_score(data, clusterer.labels_), 4)))
        print("-------------------------")
```

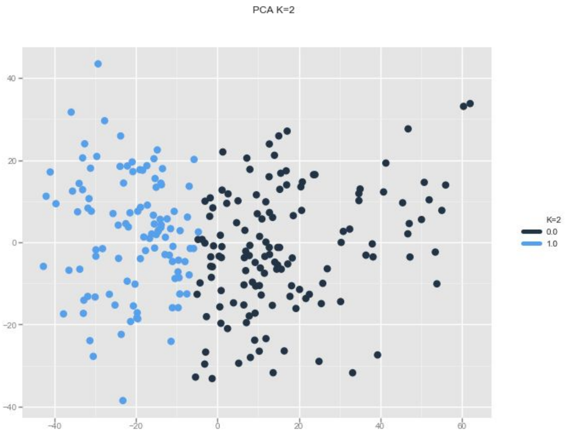


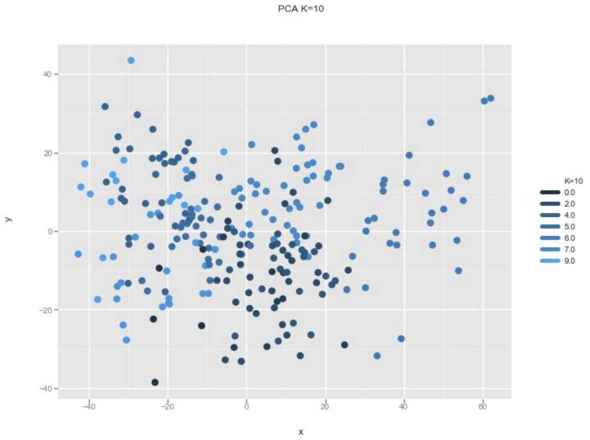
```
Silhouette Coefficient for k == 2: 0.0962
------------------------
Silhouette Coefficient for k == 5: 0.07
------------------------
Silhouette Coefficient for k == 7: 0.0658
------------------------
Silhouette Coefficient for k == 10: 0.0647
------------------------
Silhouette Coefficient for k == 20: 0.052
------------------------
Silhouette Coefficient for k == 25: 0.0588
------------------------
Silhouette Coefficient for k == 50: 0.0707
------------------------
Silhouette Coefficient for k == 75: 0.0803
------------------------
Silhouette Coefficient for k == 100: 0.0787
------------------------
Silhouette Coefficient for k == 115: 0.0827
```

After observing the poor scores I considered using Principal Component Analysis (PCA) to reduce the dimensionality of the dataset and see if clusters would more smoothly form after the reduction. After further analysis of PCA on the dataset, it appeared that reducing the dataset to 50 dimensions would still only explain about 70% of the data's variance, showing that there was no shape of the data with cell type as the sample. For the sake of observing the effect, I reduced the dataset to 2 dimensions and then plotted a few values of k on the reduced data. The reduced dataset interestingly separated into 2 clusters, but more than that and the clustering appeared to be completely random.









At this point it was clear that the data was not going to show much overlap between cell type gene expression. I tried a couple different other clustering methods such as Agglomerative clustering and Affinity Propagation and received similar silhouette scores.

The same above procedure was done using genes as the samples. Although for both K-means and Agglomerative clustering I have higher silhouette coefficients of around 0.25, they were still not quite high enough to show the data clustering well.

Conclusion and Future Recommendations

My further recommendation for this project would be to acquire more data for more genes across similar cell types to validate whether or not most cancer cells seem to have distinct expression profiles in general.

I would also like to include some personal insight as I continued to learn more about this dataset. Immediately after observing the heat-map that I created using plotly, I was able to interactively view each cluster that had a dark blue color. In the context of this heatmap, dark blue represents a smaller Euclidean distance of gene expression. The majority of the dark blue portions of the heatmap happen to reside along the middle diagonal of the visual. The diagonal of the heatmap is where the identical cell line expression value intersects. With the majority of the dark blue areas being around the diagonal I immediately felt the fear that these blue areas are simply the different cells of the same cell line. When I hovered over these areas, that fear was confirmed. It was at this moment that I began to contemplate whether or not I should continue to dive deeper into learning more about this gene expression dataset. To observe a gene expression dataset with 247 different types of specific tumor cells, containing about 40-50 different cell lines of both malignant and benign tumors, and determine that the gene expression profiles of each tumor cell line does not overlap over one other seems to be a rather disappointing result to say the least.

However, going back to my roots as a pure scientist I realized a severe flaw in that thinking. It is very easy to get caught up solely on scientific research that will report data that shocks the world and gets more attention. Reporting data that is less shocking seems to be overlooked simply due to the fact that it gets less attention as we look to market our work. The integrity of science, however, remains a constant regardless of the reaction to one's findings. In the context of this project, it may not be the most "impressive" finding to conclude that specific tumor cell lines have distinct gene expression profiles that have little overlap with other tumor cell lines. Yet this finding could be interpreted as an extremely important inference to how we approach tackling cancer moving forward. With such distinct gene expression profiles with each individual tumor cell line, the ideal approach to cancer would appear to be a more individualized and targeted approach as opposed to a blanket thought process for how to defeat cancer. These findings also lead us to further understand how complex of a disease cancer is and how many different ways that it can develop.

In conclusion, when one aims to perform empirical research in any field we must remember why we first asked the question of interest. Whatever that question may be any findings that we conclude in our study get us one step closer to answering that question, which is ultimately the goal of science.