How different are the PanIN, normal, acinar and tumor cell types to each other? Are there any batch effects or patient specific effects?

Principal Component Analysis

The data given is stored as a matrix, the rows correspond to the different genes and the columns correspond to patient id-tumor site.

Principal component analysis is a method of dimension reduction used to observe differing gene expression between cells. Since the dataset has a lot of different genes being analyzed for a lot of patients, a PCA plot is used to convert the correlations between cells into a 2D plot. upon identifying clusters, it can be concluded that the number of clusters correspond to the number of cells doing those many types of things with their genes. PC1 represents the genes with most variation and PC2 represents genes with second most variation.

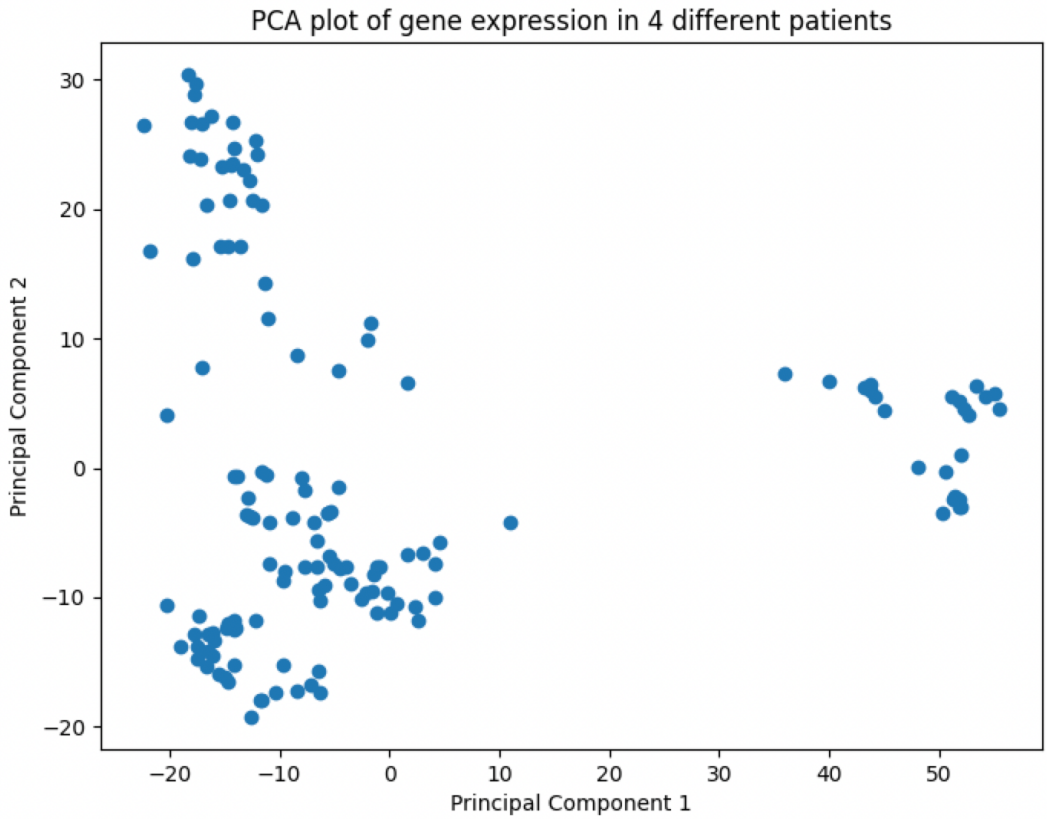
My hypothesis of the graph: since there are 4 regions, there will be 4 colors/ clusters of points at patient's representative of: PanIN, normal, acinar, tumor. Where PanIN will be the most varied followed by tumor (glandular/ poorly differentiated) then acinar then normal ductal.

Pycharm 3.10 has been used alongside the libraries above for data processing.

```python
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans

file_path = 'processed-tumour-DSP.tsv'
df = pd.read_csv(file_path, sep='\t', index_col=0).T  # transpose matrix

matplotlib.use('TkAgg')  # this is needed to run matplotlib on my laptop, otherwise cuases error: partial initialisation

pca = PCA()
pca_space = pca.fit_transform(df)  # normalise the data and its variance
pca_space_df = pd.DataFrame(pca_space, index=df.index)

clustering = KMeans(4, random_state=42)  # random state used for reproducibility
clusters = clustering.fit_predict(pca_space_df)

scatter = plt.scatter(pca_space_df[0], pca_space_df[1], c=clusters,
                      cmap='viridis', marker='o')

legend_labels = ['Tumor (glandular/poor)', 'PanIN', 'Acinar', 'Normal Ductal']  # legend names
plt.legend(handles=scatter.legend_elements()[0], labels=legend_labels, title='Clusters', loc='upper right')

plt.title('PCA plot of gene expression in 4 different patients')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.show()
```

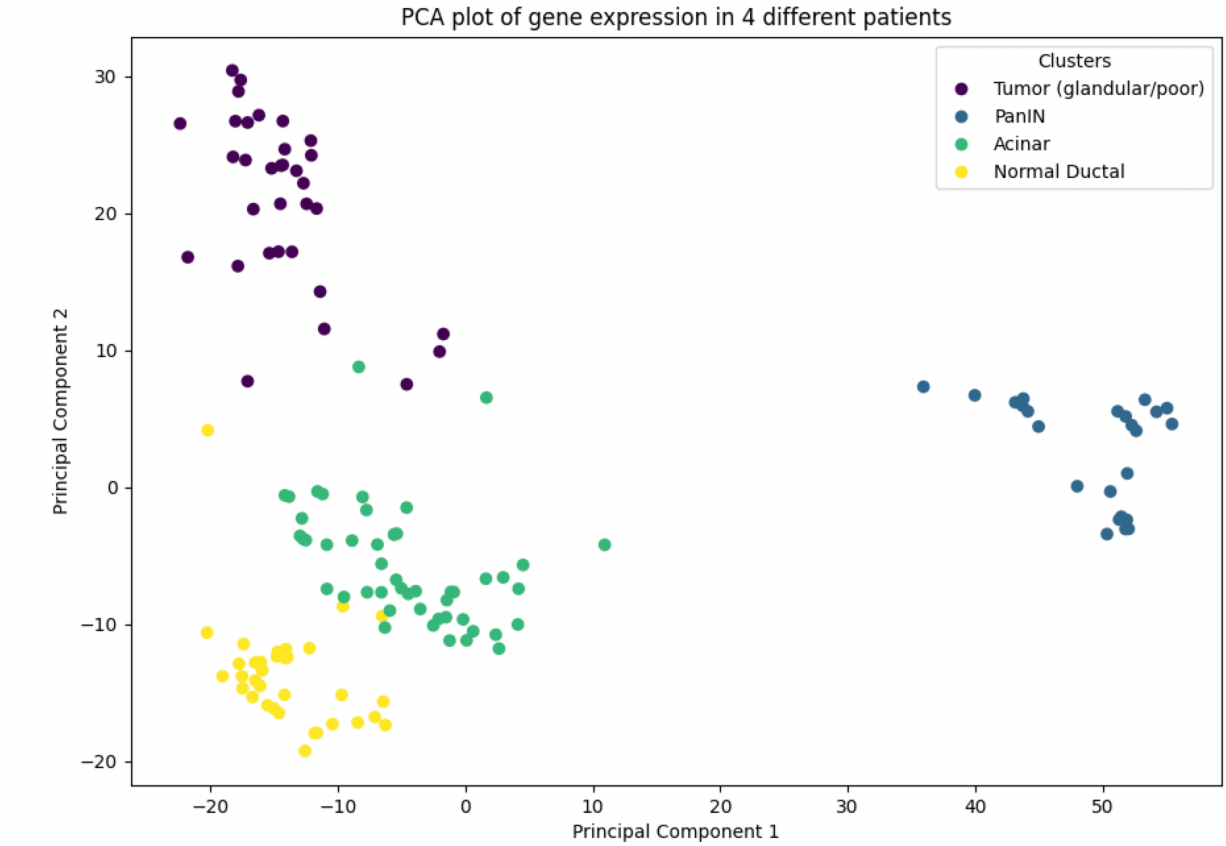
For matplotlib.use('TkAgg) source error:
https://stackoverflow.com/questions/73745245/error-using-matplotlib-in-pycharm-has-no-attribute-figurecanvas

Initial plot that has not been colored based on cell type:



Coloring the plot based on cell type

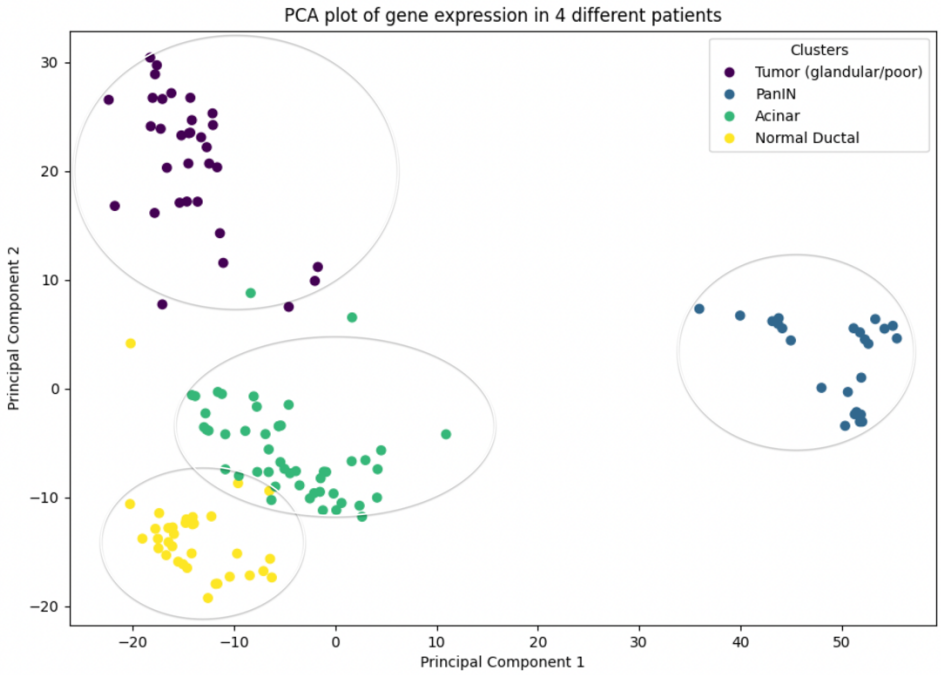
After coloring the plot based on cell type:



There are 4 different clusters of cells obtained which represent 4 regions of interest.

Analysis

Based on my hypothesis, there are 4 clusters that were visible to me, they have been colored coded accordingly. The PanIN gene is the most varied followed by tumor (glandular/ poorly differentiated) then acinar then normal ductal.



Normal ductal and acinar are quite similar as they both have a similar role to do with digestive enzyme storage and production. This is further supported in the research paper “Following quality control and data normalization, we performed PCA and detected a prominent batch effect. Upon correction, we obtained acinar and PanIN ROIs that clustered separately, whereas ADM and normal duct ROIs were interspersed with each other, indicating transcriptional similarity, which is expected” (Carpenter Et al).

Based on the hypotheses given, I think that Acinar to ductal metaplasia is the hypothesis that is a more appropriate fit to this data. Supported by clustering patterns post batch correction, transcriptional similarity between acinar to ductal metaplasia and other region of interest, as they were both interspersed with each other in the clustering, this aligns with the expected transcriptional similarity between acinar cells undergoing metaplasia and normal ductal cells.

Further experimentation

To further understand this data set, a SCREE plot can potentially be used to further minimize complexity like the genes that have the same or very similar expression. Only the genes that are highly varied can be focused upon. This can be conducted before the PCA and the subsequent data in the PCA can be focused on these highly varied genes. The varied genes can also specifically be labelled.

Works cited

Carpenter, E. S., Elhossiny, A. M., Kadiyala, P., Li, J., McGue, J., Griffith, B. D., Zhang, Y., Edwards, J., Nelson, S., Lima, F., Donahue, K. L., Du, W., Bischoff, A. C., Alomari, D., Watkoske, H. R., Mattea, M., The, S., Espinoza, C. E., Barrett, M., ... Pasca di Magliano, M. (2023). Analysis of Donor Pancreata Defines the Transcriptomic Signature and Microenvironment of Early Neoplastic Lesions. *Cancer Discovery*, *13*(6), 1324–1345.

Appendix

Code pasted:

```
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans

file_path = 'processed-tumour-DSP.tsv'
df = pd.read_csv(file_path, sep='\t', index_col=0).T  # transpose matrix

matplotlib.use('TkAgg')  # this is needed to run matplotlib on my laptop, otherwise cuases error: partial initialisation

pca = PCA()
pca_space = pca.fit_transform(df)  # normalise the data and its variance
pca_space_df = pd.DataFrame(pca_space, index=df.index)

clustering = KMeans(4, random_state=42)  # random state used for reproducibility
clusters = clustering.fit_predict(pca_space_df)

scatter = plt.scatter(pca_space_df[0], pca_space_df[1], c=clusters,
                      cmap='viridis', marker='o')

legend_labels = ['Tumor (glandular/poor)', 'PanIN', 'Acinar', 'Normal Ductal']  # legend names
plt.legend(handles=scatter.legend_elements()[0], labels=legend_labels, title='Clusters', loc='upper right')

plt.title('PCA plot of gene expression in 4 different patients')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.show()
```