LMP1210H Winter 2025 Assignment 3

Yuliang Xiao – yl.xiao@mail.utoronto.ca

February 25, 2025

Instructors: Rahul G. Krishnan & Bo Wang

By turning in this assignment, I agree by the UofT honor code and declare that all of this is my own work.

Problem 1 - Revisiting Single-Cell RNA-Seq

(a) The ARI values for all of the 8 models are listed

Adjusted Rand Index				
K	2	4	6	8
K-means	1.000	0.845	0.918	0.955
GMM	0.160	0.0104	0.000	0.743

Table 1: Adjusted Rand Index Report for K-means and GMM Models with K=[2, 4, 6, 8]

(b) Given the results printed on terminal, K-means model runs much faster than the GMM at each K values and the performance of K-means is much better than that of GMM. K-means model runs stably for each K=[2, 4, 6, 8] while the GMM only works well when K=8, showing that K-means is model stronger and more robust than GMM.

Problem 2 - Hierarchical Clustering

single link

$$\begin{array}{ccccc} & A & B & C & D \\ A & 0 & 1 & 5 & 6 \\ B & 1 & 0 & 3 & 8 \\ C & 5 & 3 & 0 & 4 \\ D & 6 & 8 & 4 & 0 \end{array} \rightarrow \begin{array}{cccc} & (A,B) & C & D \\ (A,B) & 0 & 3 & 6 \\ C & 3 & 0 & 4 \\ D & 6 & 4 & 0 \end{array} \rightarrow \begin{array}{ccc} & (A,B,C) & D \\ (A,B,C) & 0 & 4 \\ D & 4 & 0 \end{array}$$

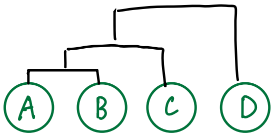
(A,B) has minimum distance

((A,B),C) has the minimum distance

$$d_{(A,B),C} = \min\{d_{A,C}, d_{B,C}\} = \min\{5,3\} = 3$$

$$d_{(A,B),D} = \min\{d_{A,D}, d_{B,D}\} = \min\{6,8\} = 6$$

$$d_{(A,B,C),D} = \min\{d_{(A,B),D}, d_{C,D}\} = \min\{6,4\} = 4$$



complete link

$$\begin{array}{ccccc} & A & B & C & D \\ A & 0 & 1 & 5 & 6 \\ B & 1 & 0 & 3 & 8 \\ C & 5 & 3 & 0 & 4 \\ D & 6 & 8 & 4 & 0 \end{array} \rightarrow \begin{array}{cccc} & (A,B) & C & D \\ (A,B) & 0 & 5 & 8 \\ C & 5 & 0 & 4 \\ D & 8 & 4 & 0 \end{array} \rightarrow \begin{array}{ccc} & (A,B) & (C,D) \\ (A,B) & 0 & 8 \\ (C,D) & 8 & 0 \end{array}$$

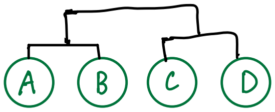
(A,B) has minimum distance

(C,D) has minimum distance

$$d_{(A,B),C} = \max\{d_{A,C}, d_{B,C}\} = \max\{5,3\} = 5$$

$$d_{(A,B),D} = \max\{d_{A,D}, d_{B,D}\} = \max\{6,8\} = 8$$

$$d_{(A,B),(C,D)} = \max\{d_{(A,B),C}, d_{(A,B),D}\} = \max\{5,8\} = 8$$



Problem 3 - Adaboost Implementation

Answers: please click and check this Google Colab Notebook.

Problem 4 - PCA

(a) The 15 explained variances are reported

Explained Variance															
PC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Variance	63.1	7.87	6.78	6.55	6.38	6.19	5.94	5.83	5.74	5.70	5.62	5.59	5.56	5.52	5.50

Table 2: Explained Variance of 15 Principal Components

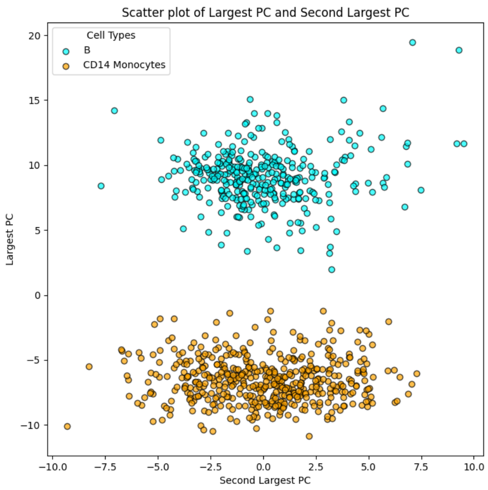
Given the table, the first principal component is the most important for representing the data.

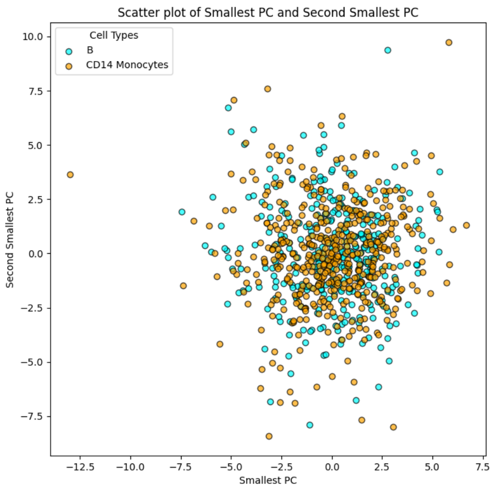
(b) To obtain at least 80 percent of data variance, the minimum number of principal components that we need is 398. Here is the calculation process:

Algorithm 1 Obtain Minimum Number of PCs for 80% Data Variance

```
procedure OBTAIN_MIN_N(data)
    pca ← PCA(random_state=17)                       ▷ Initialize PCA with random seed=17
    pca.fit(data)                                    ▷ Fit PCA to data
    exp_var_ratio ← pca.explained_variance_ratio_    ▷ Get explained variance ratio
    cum_var ← np.cumsum(exp_var_ratio)               ▷ Get cumulative explained variance ratio
    Min_N ← np.argmax(cum_var >= 0.8) + 1            ▷ Get minimum number of PCs
    return Min_N
```

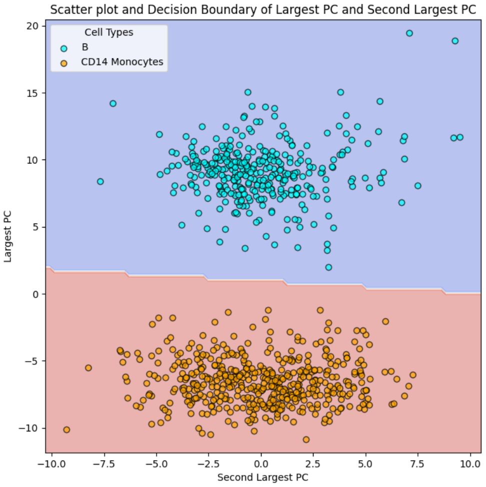
(c) Here are two scatter plots

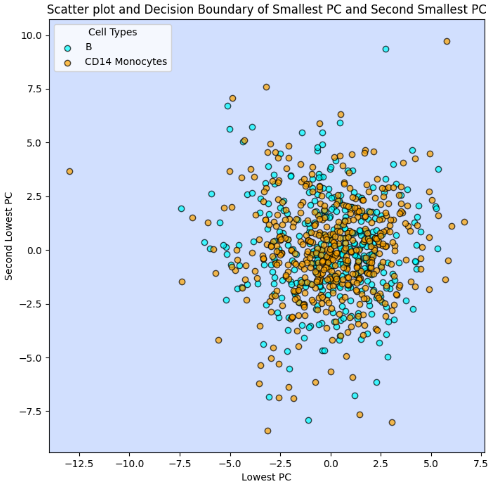




Given the plots, I can conclude that largest PC and second largest PC have clear boundary to separate themselves while the smallest and second smallest PC cannot be distinguished easily.

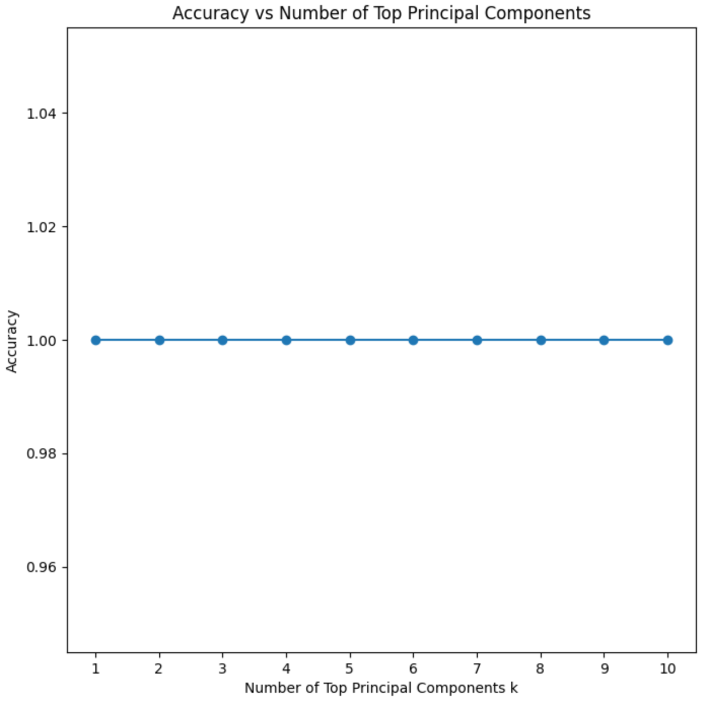
(d) Here are plots of logistic regression





The accuracy of logistic regression model on largest and second largest PCs is 100% while the accuracy for smallest and second smallest PCs is 58.29%. Therefore, we can conclude that the performance of classification using 2 largest PCs is much better than that using 2 smallest PCs.

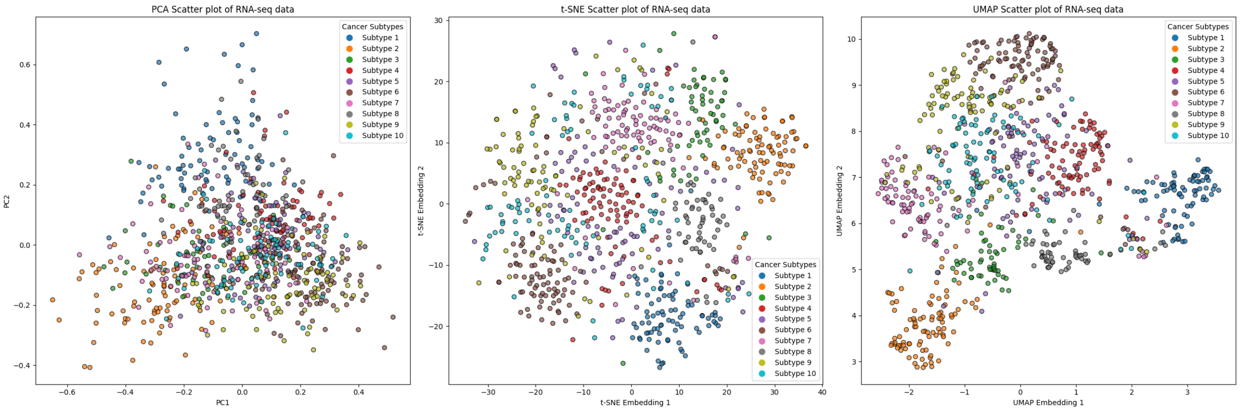
(e) Here is the accuracy plot of logistic regression model



Given the plot, I found that the accuracy is same across different number of principal components using for the classification, it means that some of the added components might not contribute additional useful information. Since the first principal component (PC1) (dominant variance ratio) already captures most of the class separability, adding more PCs might not significantly improve the model.

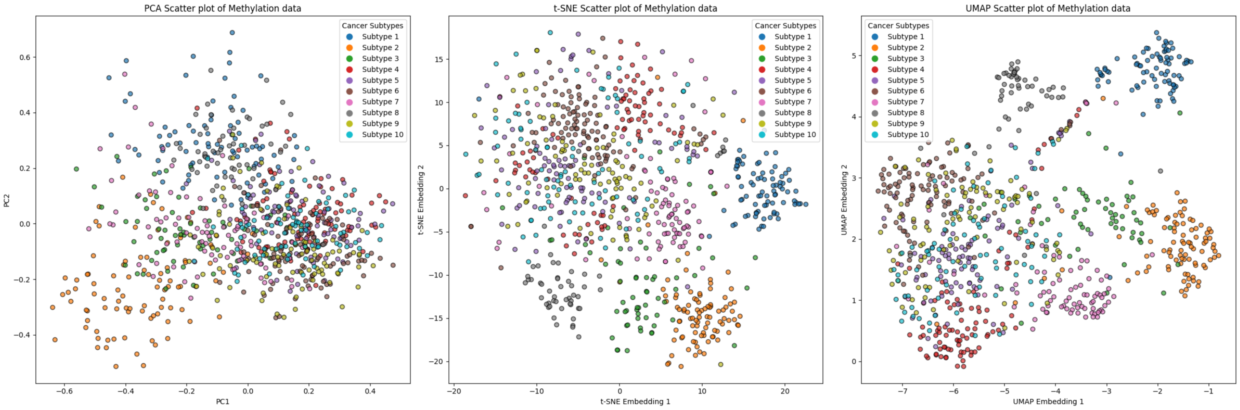
Problem 5 - Multi-Omics Analysis for Cancer Subtyping

(a) Here is the plot of all three visualization methods for RNA-seq data



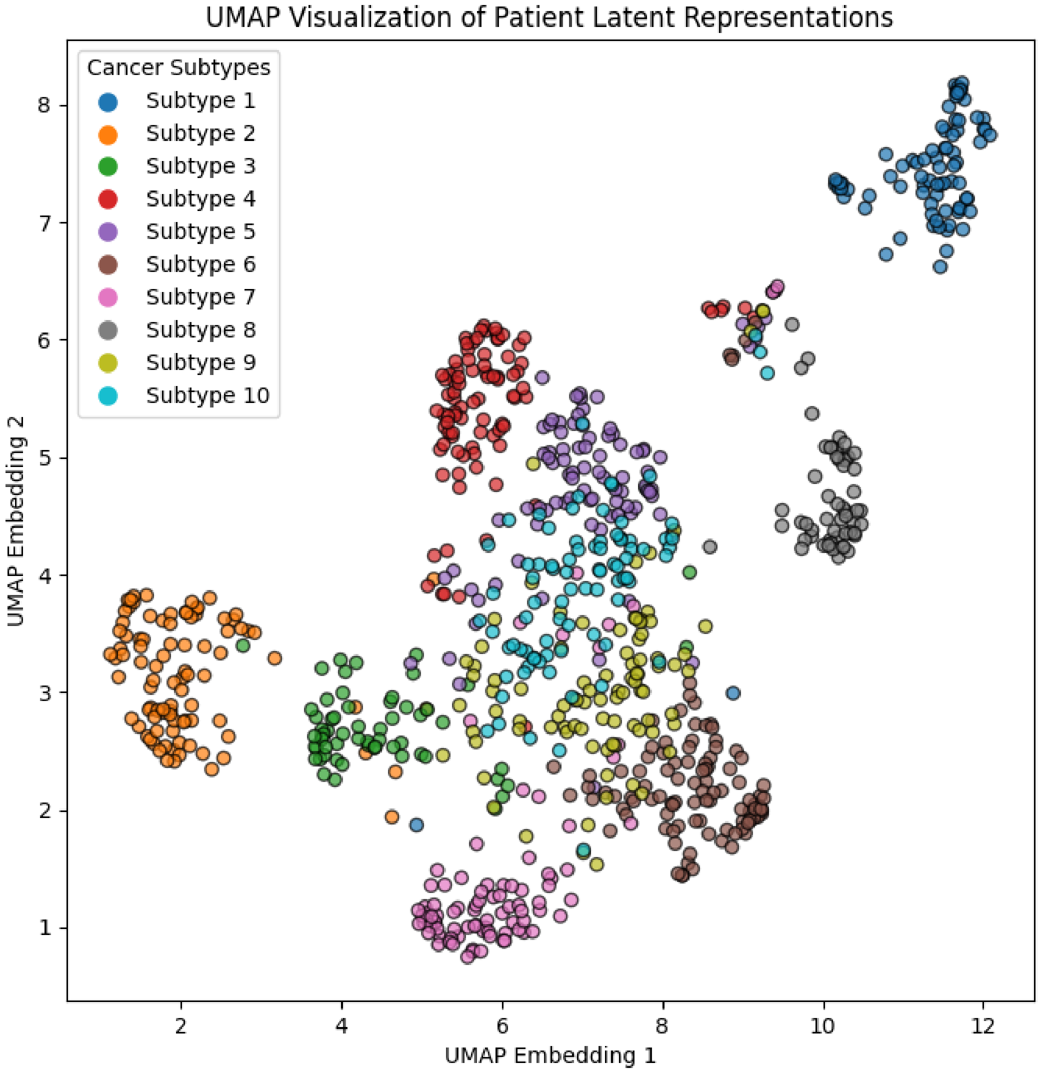
The ARI value for K-means model with $k = 10$ is 0.331.

(b) Here is the plot of all three visualization methods for Methylation data



The ARI value for K-means model with $k = 10$ is 0.278.

(c) To obtain a good latent representation of patients, I built a training process using **Autoencoder** where the encoder projects the raw data into low dimensional space and decoder do the inverse projection from latent space to original dimension. The network architecture of **Autoencoder** is implemented in the file `autoencoder.py` which utilize multiple fully connected layers combined with the activation function `Mish()`. The `Mean-Square-Error` loss function is used to evaluate the performance of the model and the `Adam` optimizer with learning rate $lr = 5e^{-5}$ is applied to iteratively update the model parameters. Once the training is finished, we only used the optimized encoder to extract latent features and follow the same operations in **Part (a) & (b)** to project data and visualize them in 2-D space using `UMAP`. Lastly, the AMI value of K-means model with $k = 10$ is 0.721.



Problem 6 - Convolutional Neural Networks (CNN)

(a) We can use the formula to compute the output size after convolution operation

$$\begin{aligned} H_{out} &= \left\lfloor \frac{H_{in} + 2 \times padding - dilation \times (kernel - 1) - 1}{stride} + 1 \right\rfloor \\ W_{out} &= \left\lfloor \frac{W_{in} + 2 \times padding - dilation \times (kernel - 1) - 1}{stride} + 1 \right\rfloor \\ D_{out} &= Num\_filter \end{aligned} \tag{1}$$

Then we can place value of input spatial dimension into the formula

$$\begin{aligned} H_{out} &= \left\lfloor \frac{32 + 2 \times 1 - 1 \times (6 - 1) - 1}{2} + 1 \right\rfloor = 15 \\ W_{out} &= \left\lfloor \frac{64 + 2 \times 1 - 1 \times (6 - 1) - 1}{2} + 1 \right\rfloor = 31 \\ D_{out} &= 12 \end{aligned} \tag{2}$$

Finally we get the output size of a convolution layer which is 31×15×12 (Width*Height*Depth).

(b) We can use the formula to compute the number of parameters in a convolution operation

$$\texttt{Total Parameters} = (C_{in} \times C_{out} \times Kernel \times Kernel) + bias \tag{3}$$

The total trainable parameters in a 2D Convolutional later with input channel 5 and output channel 6 and kernel size 3 is $5 \times 6 \times 3 \times 3 + 6 = 276$.

(c) We can use the formula to compute the output size after convolution operation

$$\begin{aligned} H_{out} &= \left\lfloor \frac{H_{in} + 2 \times padding - dilation \times (kernel - 1) - 1}{stride} + 1 \right\rfloor \\ W_{out} &= \left\lfloor \frac{W_{in} + 2 \times padding - dilation \times (kernel - 1) - 1}{stride} + 1 \right\rfloor \\ D_{out} &= D_{in} \end{aligned} \tag{4}$$

Then we can place value of input spatial dimension into the formula

$$\begin{aligned} H_{out} &= \left\lfloor \frac{128 + 2 \times 0 - 1 \times (8 - 1) - 1}{4} + 1 \right\rfloor = 31 \\ W_{out} &= \left\lfloor \frac{128 + 2 \times 0 - 1 \times (8 - 1) - 1}{4} + 1 \right\rfloor = 31 \\ D_{out} &= 3 \end{aligned} \tag{5}$$

Finally we get the output size of a convolution layer which is 31×31×3 (Width*Height*Depth).

Problem 7 - Cash-Swift Video

As far as I am thinking, people will first use source separation tools to extract the vocal track from a song and preprocess the sound waveform into the spectrogram for next step training. They will also extract key acoustic features like pitch information from both the source and target singers. Then, they will use the deep learning model to train the voice conversion model use the spectrogram of source singers as input and compute the loss function with the spectrogram of target singer. Once the model is converged, the postprocessing step is applied to make the sound natural.