Contents

1 Introduction **4**

2 Methodology **4**

2.1 Data Preprocessing 4

2.1.1 Data Cleaning 5

2.1.2 Feature Encoding 5

2.1.3 Feature Scaling 5

2.2 Feature Selection 5

2.2.1 Feature Selection via ANOVA F-test 6

2.3 Dimensionality Reduction 6

2.3.1 Removal of low variance genes 6

2.3.2 Principal Component Analysis (PCA) 6

2.3.3 Nonlinear Dimensionality Reduction with UMAP 7

3 Model Training & Evaluation **9**

3.1 Clustering Methodology 9

3.1.1 K-Means with Euclidean Distance 9

3.1.2 K-Means with Mahalanobis Distance 11

3.1.3 Agglomerative Clustering 13

3.1.4 DBSCAN 16

3.1.5 Observation 18

3.1.6 Gaussian Mixture Model 18

3.1.7 Fuzzy C-Means 20

3.2 Performance Evaluation 22

3.2.1 Evaluation Metrics for $k = 6$ Clusters 22

3.2.2 Evaluation Metrics for Optimal Number of Clusters 23

3.2.3 Evaluation Metric Definitions 23

3.3 Determining Potential Biomarkers within each cluster 24

3.4 Discussion 25

4 Additional Observations and Insights **26**

5 Conclusion **27**

List of Figures

1 Eigenvalues in descending order showing the variance explained by each principal component . . . 7
2 Scree plot showing cumulative explained variance versus number of principal components . . . 8
3 Visualization of Original Clusters on PCA space . . . 8
4 K-Means Clustering with $k = 6$ . . . 10
5 K-Means Clustering with Optimal $k = 3$ . . . 11
6 K-Means Clustering with $k = 6$ using Mahalanobis Distance . . . 12
7 K-Means Clustering with Optimal $k = 4$ using Mahalanobis Distance . . . 13
8 Dendrogram from Agglomerative Clustering with Cosine Distance . . . 14
9 Agglomerative Clustering with $k = 6$ . . . 15
10 Agglomerative Clustering with Optimal $k = 4$ . . . 16
11 Elbow Method for Determining Optimal ε . . . 17
12 DBSCAN Clustering Results . . . 17
13 GMM Clustering with $k = 6$ . . . 19
14 GMM Clustering with Optimal $k = 3$ . . . 20
15 Fuzzy C-Means Clustering with $k = 6$ . . . 21
16 Fuzzy C-Means Clustering with Optimal $k = 3$ . . . 22
17 Heatmap of Top 25 Variable Genes Ordered by Agglomerative Clustering 26

List of Tables

1 Top 10 most discriminative genes selected using ANOVA F-test 6
2 Silhouette Scores for Different Values of k 11
3 Silhouette Scores for Different Values of k using Mahalanobis Distance . 13
4 Silhouette Scores for Different Values of k using Agglomerative Clustering with Cosine Distance 15
5 Silhouette Scores for Different Values of k using GMM 19
6 Silhouette Scores for Different Values of k using FCM 21
7 Clustering Performance Metrics for $k = 6$ Clusters 22
8 Clustering Performance Metrics for Optimal Number of Clusters 23
9 Top 5 Biomarker Genes for Each Cluster 25

1 Introduction

Breast cancer's molecular heterogeneity necessitates advanced computational approaches for subtype identification. This study analyzes the **GSE45827** dataset (151 samples, 54,676 genes) from CuMiDa, [1]. which includes six established subtypes: Luminal A/B, HER2-enriched, Basal-like, Claudin-low, and Normal-like. Data was preprocessed with RMA background correction and quantile normalization.

- **Objectives**:
  - Identify novel subgroups using K-Means,Hierarchical Clustering and other clustering methods
  - Validate clusters against known classifications
  - Discover potential biomarkers
- **Significance**:
  - Enables precise diagnostic classification
  - Guides personalized treatment strategies
  - Demonstrates ML applications in oncology

Recent advances in clustering high-dimensional genomic data have shown promise for refining breast cancer taxonomy. Our work builds on these approaches while addressing computational challenges specific to gene expression data.

- Class distribution:
  - Class 0: 30 samples (HER2-enriched)
  - Class 1: 41 samples (Basal-like)
  - Class 2: 14 samples (Cell line)
  - Class 3: 29 samples (Luminal A)
  - Class 4: 30 samples (Luminal B)
  - Class 5: 7 samples (Normal-like)

2 Methodology

2.1 Data Preprocessing

The dataset consists of 151 samples with gene expression profiles of 54,676 genes, categorized into six distinct classes of breast cancer. The preprocessing pipeline included the following steps:

2.1.1 Data Cleaning

- **Missing Values Handling**: The dataset was checked for missing values using `df.isnull().sum()`, which revealed no missing values in any of the attributes.
- **Duplicate Removal**: Duplicate rows were identified and removed using `df.duplicated()`, with no duplicates found in the dataset.
- **Negative Values Check**: All numerical columns were verified for negative values using `(df[col] < 0).any()`, with no negative values detected.

2.1.2 Feature Encoding

- The categorical target variable `type` was encoded using ordinal encoding with `OrdinalEncoder()` from scikit-learn, converting the six breast cancer classes to numerical values.

2.1.3 Feature Scaling

- **Standard Scaling**: Implemented using `StandardScaler()` to standardize features to zero mean and unit variance

2.2 Feature Selection

The problem statement stated that we needed to identify the most relevant genes contributing to breast cancer classification. To do this we used ANOVA F-TEST. The top 10 genes with the highest F-scores were selected, aiming to capture the most significant variance related to the target variable. In high-dimensional datasets, especially those encountered in fields like genomics , feature selection is crucial for improving model performance and interpretability. One effective method for feature selection in classification tasks is the Analysis of Variance (ANOVA) F-test, implemented in scikit-learn as `f_classif`.
The ANOVA F-test assesses the significance of each feature by evaluating the ratio of variance between different class means to the variance within the classes. Features with higher F-scores are considered more discriminative.

- **Efficiency**: Evaluates each feature independently, making it computationally efficient for high-dimensional data.
- **Effectiveness**: Identifies features that contribute significantly to class separation..

Mathematical Foundation

For each feature, the F-statistic is calculated as:

$$F = \frac{\text{Variance Between Groups}}{\text{Variance Within Groups}}$$

A higher F-value indicates that the feature has a greater ability to distinguish between classes.

2.2.1 Feature Selection via ANOVA F-test

Rank	Gene ID
1	200795_at
2	202878_s_at
3	205225_at
4	210052_s_at
5	210809_s_at
6	216836_s_at
7	222358_x_at
8	222608_s_at
9	232814_x_at
10	240205_x_at

Table 1: Top 10 most discriminative genes selected using ANOVA F-test

Note: The results obtained in this section were not used for clustering analysis, as the class labels were utilized to identify the most relevant genes contributing to breast cancer classification. Since clustering is an unsupervised method, incorporating label-based feature selection would introduce bias and is therefore avoided.

2.3 Dimensionality Reduction

2.3.1 Removal of low variance genes

Genes with low variance across samples typically carry minimal discriminative information and can introduce noise into the analysis. Therefore, we removed the bottom 50% of genes with the least variance to retain only those features most likely to contribute meaningfully to clustering and downstream interpretation.

2.3.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) was performed to reduce the high-dimensional gene expression data:

– Eigenvalue analysis showed the first few principal components capture most of the variance (Figure 1)

PCA was employed on the reduced dataset to understand the variance distribution and determine an appropriate number of components for further analysis.

– **Explained Variance Thresholds:**
  * **95% Variance:** Achieved with 115 principal components.
  * **85% Variance:** Achieved with 74 principal components.

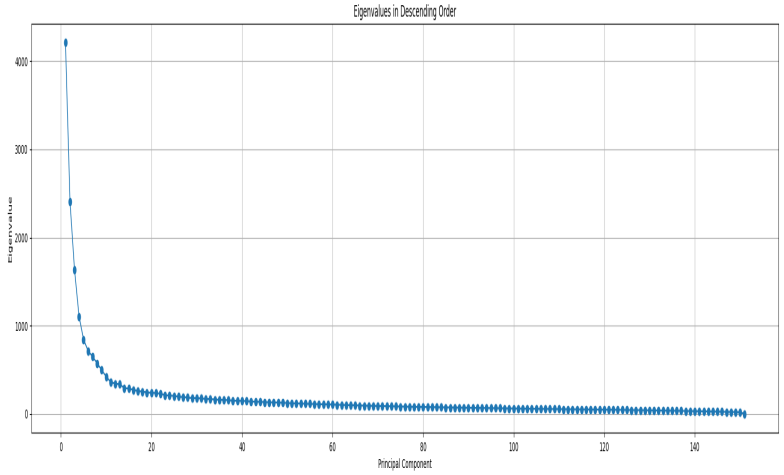

Figure 1: Eigenvalues in descending order showing the variance explained by each principal component

Given the sample size of 151, retaining 115 components was deemed excessive, potentially leading to overfitting. Therefore, a threshold of 85% explained variance was selected, resulting in 74 principal components for subsequent analysis.

2.3.3 Nonlinear Dimensionality Reduction with UMAP

To further reduce dimensionality and capture nonlinear relationships within the data, UMAP was applied to the 74 principal components, reducing them to 5 dimensions.

Why UMAP?

UMAP was chosen for its ability to preserve both local and global data structures, making it effective for visualizing and analyzing complex, high-dimensional datasets. Unlike PCA, which is limited to capturing linear relationships, UMAP can uncover

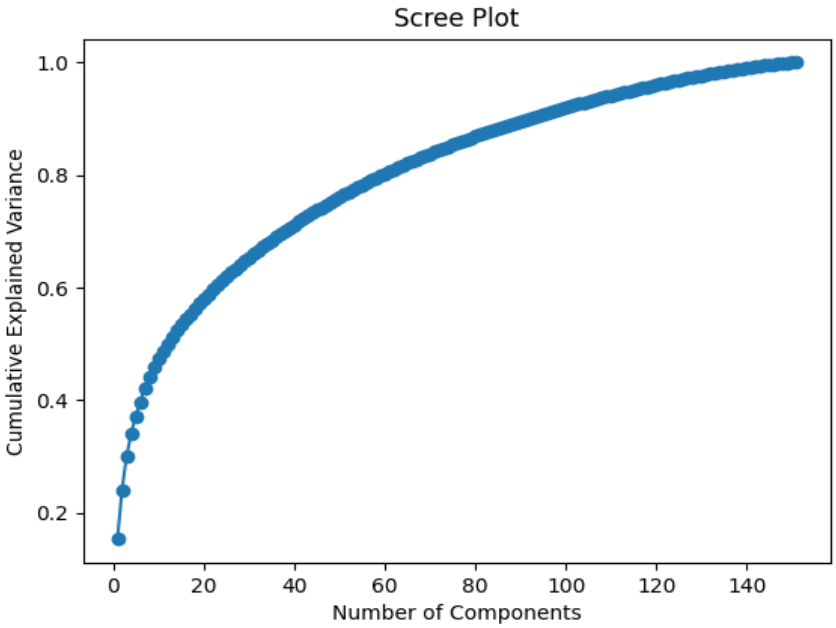

Figure 2: Scree plot showing cumulative explained variance versus number of principal components

nonlinear patterns, providing a more nuanced representation of the data's intrinsic structure.

Data Visualization

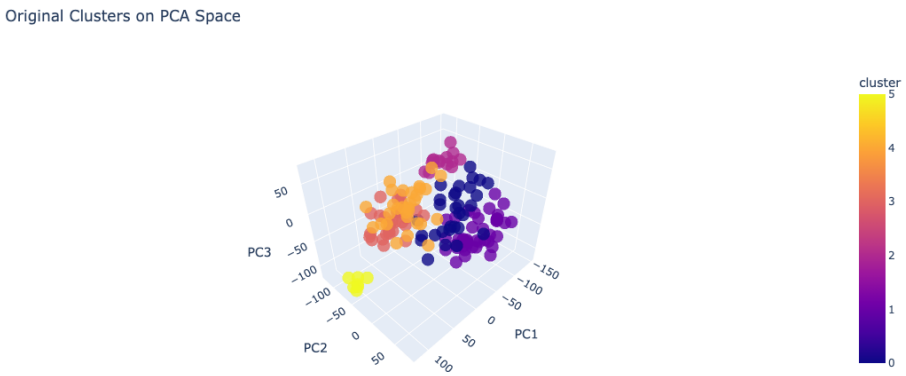

Figure 3: Visualization of Original Clusters on PCA space

3 Model Training & Evaluation

3.1 Clustering Methodology

We evaluated 6 clustering approaches on the reduced gene expression data:

- **K-Means**: Partition-based clustering
- **K-Means with Mahalanobis Distance**: Partition-based clustering
- **Hierarchical**: Agglomerative clustering with Cosine distance
- **DBSCAN**: Density-based clustering
- **Gaussian Mixture Models**: Soft clustering
- **Fuzzy C Means**: Soft Clustering

3.1.1 K-Means with Euclidean Distance

K-Means clustering is an unsupervised machine learning algorithm designed to partition a dataset into k distinct, non-overlapping clusters. Each cluster is characterized by its centroid, which is the mean position of all the points within the cluster.

Mathematical Formulation

Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ in R^d, the objective of K-Means is to minimize the within-cluster sum of squares (WCSS):

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$$

where:

- $S = \{S_1, S_2, \ldots, S_k\}$ represents the set of clusters,
- μ_i is the centroid of cluster S_i, calculated as $\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x$.

The algorithm operates iteratively through the following steps:

1. **Initialization:** Select k initial centroids, often chosen randomly.
2. **Assignment:** Assign each data point to the nearest centroid based on Euclidean distance.
3. **Update:** Recalculate the centroids as the mean of all points assigned to each cluster.

4. **Convergence:** Repeat the assignment and update steps until the centroids no longer change significantly or a maximum number of iterations is reached.

K-Means is particularly effective for datasets where clusters are spherical and of similar size. However, it assumes that the number of clusters k is known a priori and can be sensitive to the initial placement of centroids. Techniques such as the Elbow Method and Silhouette Analysis are commonly used to determine an appropriate value for k.

Clustering with $k = 6$

A K-Means clustering algorithm was applied with $k = 6$ clusters using Euclidean distance. The resulting clusters are visualized in Figure 4.

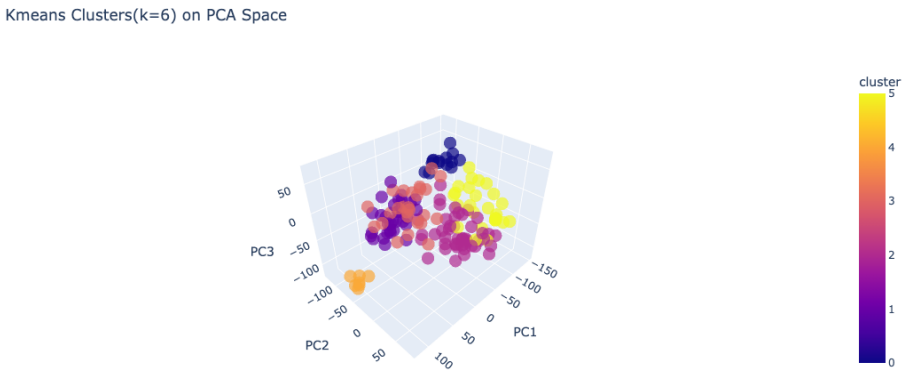

Figure 4: K-Means Clustering with $k = 6$

Silhouette Score Analysis

To determine the optimal number of clusters, silhouette scores were computed for different values of k. The silhouette score measures how similar an object is to its own cluster compared to other clusters. The scores range from -1 to 1, with higher values indicating better clustering.

Table 2: Silhouette Scores for Different Values of k

Number of Clusters (k)	Silhouette Score
3	0.6288
4	0.5021
5	0.3855
6	0.5565
7	0.4910
8	0.4677

The highest silhouette score is observed at $k = 3$, suggesting that three clusters may be optimal for this dataset.

Final Clustering with Optimal k

Based on the silhouette analysis, K-Means clustering was performed with $k = 3$. The resulting clusters are visualized in Figure 5.

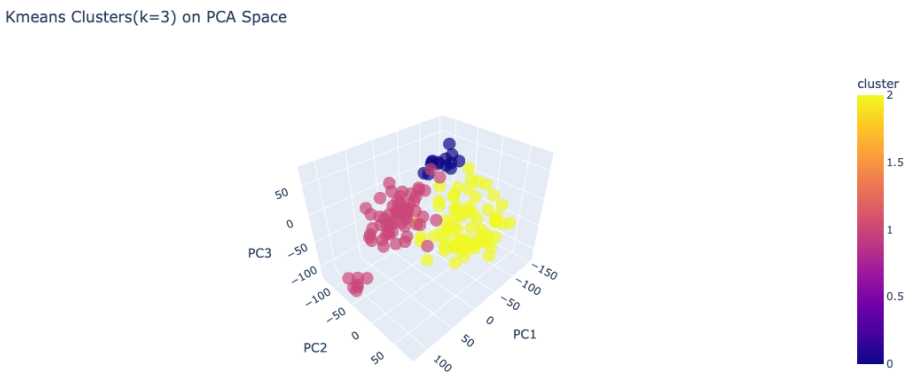

Figure 5: K-Means Clustering with Optimal $k = 3$

3.1.2 K-Means with Mahalanobis Distance

While traditional K-Means clustering uses Euclidean distance to assign data points to clusters, this approach assumes that clusters are spherical and of similar size. To accommodate clusters with different shapes and orientations, Mahalanobis distance can be employed. Mahalanobis distance accounts for the covariance among variables, making it suitable for identifying elliptical clusters.

Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ in R^d, the Mahalanobis distance between a point x and a cluster centroid μ is defined as:

$$D_M(x, \mu) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)}$$

where S is the covariance matrix of the cluster.

The objective of K-Means with Mahalanobis distance is to minimize the within-cluster sum of squared Mahalanobis distances:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} D_M(x, \mu_i)^2$$

Clustering with $k = 6$

A K-Means clustering algorithm was applied with $k = 6$ clusters using the Mahalanobis distance. The resulting clusters are visualized in Figure 6.

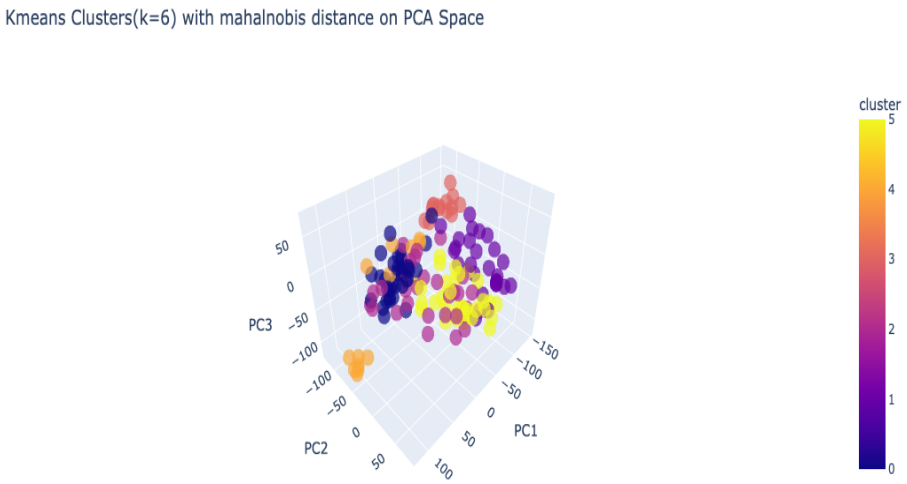

Figure 6: K-Means Clustering with $k = 6$ using Mahalanobis Distance

Silhouette Score Analysis

To determine the optimal number of clusters, silhouette scores were computed for different values of k. The silhouette score measures how similar an object is to its own cluster compared to other clusters. The scores range from -1 to 1, with higher values indicating better clustering.

Table 3 presents the silhouette scores for various values of k.

Table 3: Silhouette Scores for Different Values of k using Mahalanobis Distance

Number of Clusters (k)	Silhouette Score
3	0.0376
4	0.4702
5	0.0877
6	0.2488
7	0.2024
8	0.1292

The highest silhouette score is observed at $k = 4$, suggesting that four clusters may be optimal for this dataset.

Final Clustering with Optimal k

Based on the silhouette analysis, K-Means clustering was performed with $k = 4$ using Mahalanobis distance. The resulting clusters are visualized in Figure 7.

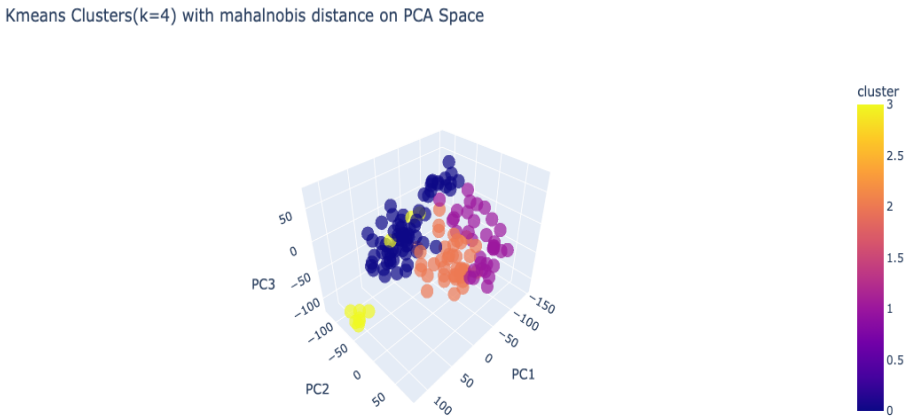

Figure 7: K-Means Clustering with Optimal $k = 4$ using Mahalanobis Distance

3.1.3 Agglomerative Clustering

Agglomerative clustering is a hierarchical clustering method that builds nested clusters by successively merging or splitting them based on a distance metric.This model works well on small datasets primarily because of its computational complexity and space requirements When applied to gene expression data, the choice of distance metric is crucial. Cosine distance is particularly effective in this context because it measures the orientation between vectors rather than their magnitude. This is

beneficial for gene expression data, where the pattern of expression (i.e., the relative expression levels across genes) is often more informative than the absolute expression levels.

Mathematical Formulation

Given two vectors $\mathbf{x}$ and $\mathbf{y}$, the cosine similarity is defined as:

$$\text{cosine\_similarity}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

The cosine distance is then:

$$\text{cosine\_distance}(\mathbf{x}, \mathbf{y}) = 1 - \text{cosine\_similarity}(\mathbf{x}, \mathbf{y})$$

This metric emphasizes the directionality of the data vectors, making it suitable for high-dimensional data like gene expression profiles.

Dendrogram Visualization

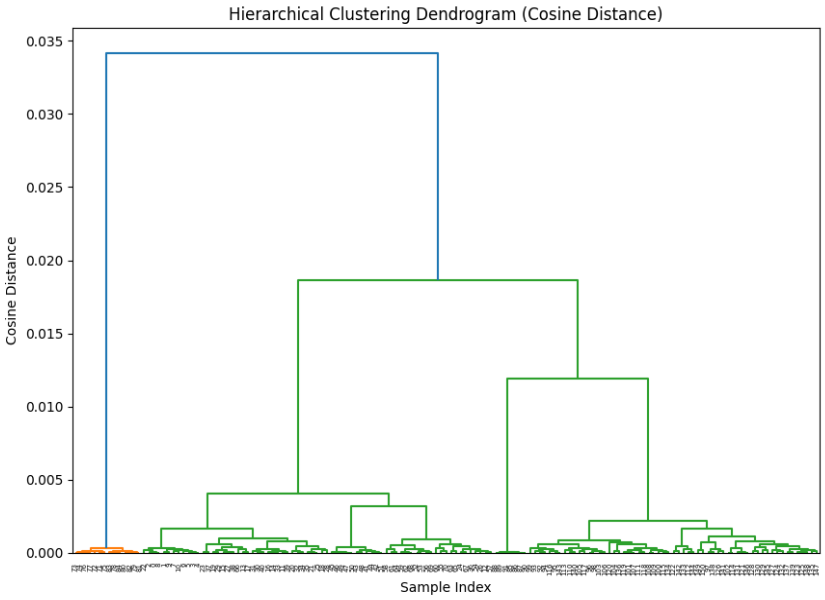

Figure 8: Dendrogram from Agglomerative Clustering with Cosine Distance

Clustering with $k = 6$

Agglomerative clustering was applied with $k = 6$ components. The resulting clusters are visualized in Figure 9.

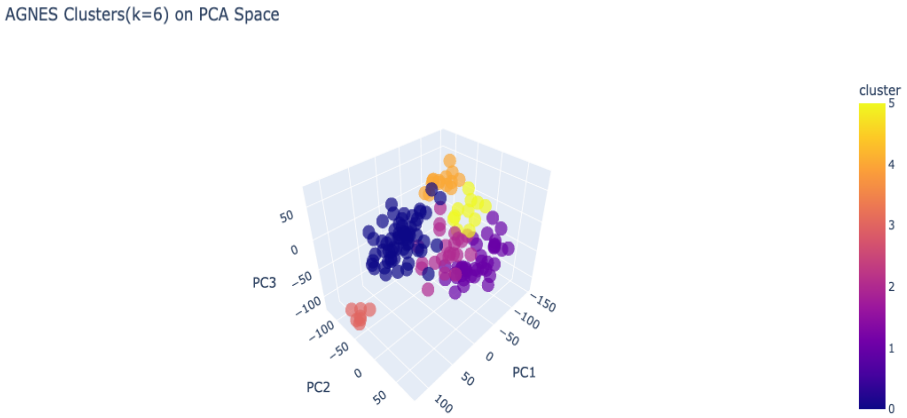

Figure 9: Agglomerative Clustering with $k = 6$

Silhouette Score Analysis

To evaluate the quality of clustering and determine the optimal number of clusters, silhouette scores were computed for different values of k. The silhouette score assesses how similar an object is to its own cluster compared to other clusters.

Table 4: Silhouette Scores for Different Values of k using Agglomerative Clustering with Cosine Distance

Number of Clusters (k)	Silhouette Score
3	0.8210
4	0.8330
5	0.7360
6	0.7534
7	0.6539
8	0.6326

The highest silhouette score is observed at $k = 4$, indicating that three clusters provide the most appropriate grouping for this dataset.

Final Clustering with Optimal k

Based on the silhouette analysis, agglomerative clustering was performed with $k = 3$. The resulting clusters are visualized in Figure 10.

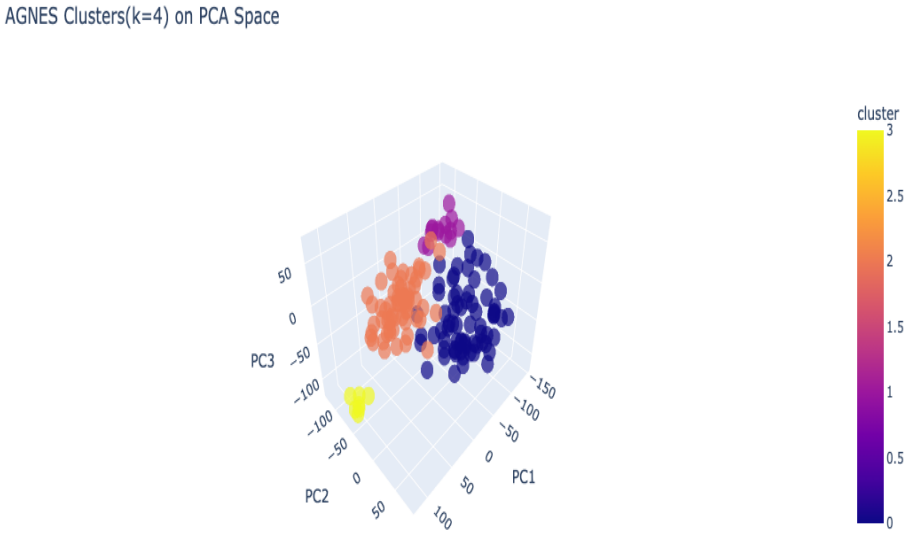

Figure 10: Agglomerative Clustering with Optimal $k = 4$

3.1.4 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering algorithm that groups together points that are closely packed together, marking as outliers points that lie alone in low-density regions. Unlike partitioning methods, DBSCAN does not require specifying the number of clusters beforehand and can find clusters of arbitrary shapes.

Mathematical Formulation

DBSCAN requires two parameters:

- **Epsilon (ε)**: The maximum distance between two samples for them to be considered as in the same neighborhood.
- **MinPts**: The number of samples in a neighborhood for a point to be considered as a core point.

A point is classified as:

- **Core Point**: If it has at least MinPts points within ε.
- **Border Point**: If it has fewer than MinPts within ε, but is in the neighborhood of a core point.
- **Noise Point**: If it is neither a core point nor a border point.

Elbow Method for Optimal ε

To determine the optimal value of ε, the elbow method was employed. This involves plotting the distances to the k-th nearest neighbor (with $k = \text{MinPts}$) for each point and identifying the "elbow" point in the plot.

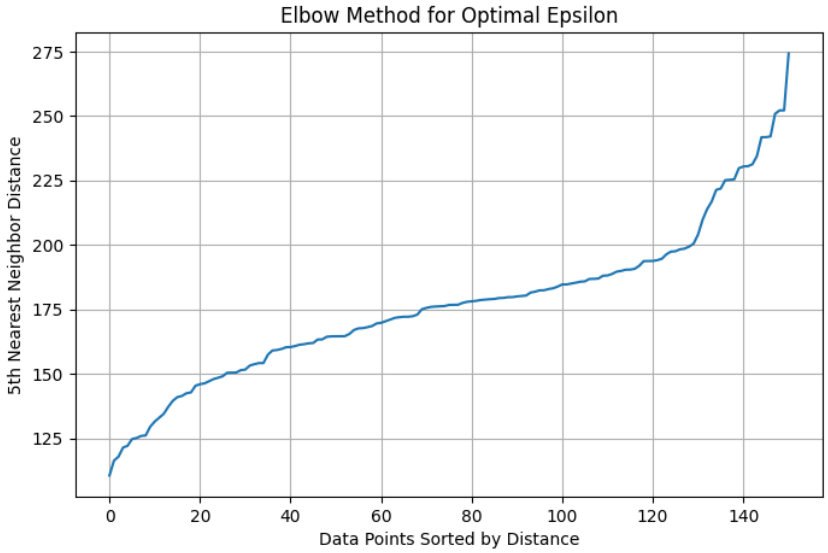

Figure 11: Elbow Method for Determining Optimal ε

DBSCAN Clustering Results

Using the optimal ε determined from the elbow method, DBSCAN was applied to the dataset. The resulting clusters are visualized in Figure 12.

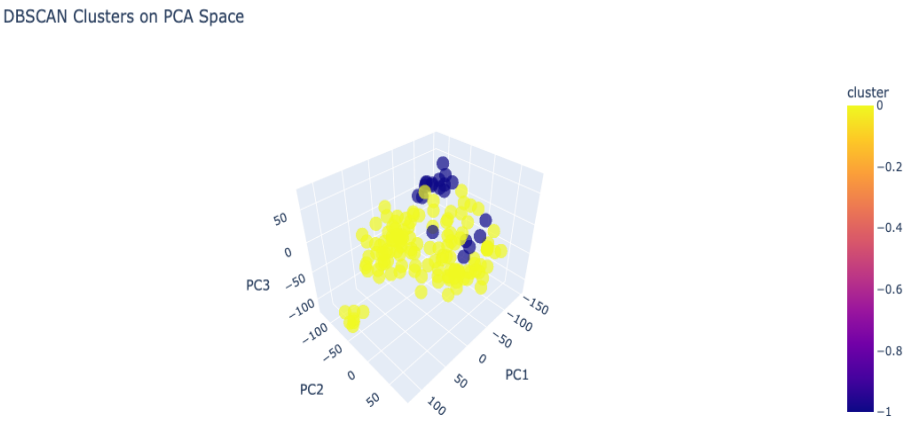

Figure 12: DBSCAN Clustering Results

3.1.5 Observation

The DBSCAN algorithm was unable to identify meaningful clusters in the dataset. Most data points were either assigned to a single cluster or labeled as noise. This outcome suggests that the dataset does not have well-defined density-based clusters, or the chosen parameters (ε and MinPts) were not suitable. Factors such as varying densities, high dimensionality, or the presence of noise can affect DBSCAN's performance.

3.1.6 Gaussian Mixture Model

Gaussian Mixture Models (GMMs) are probabilistic models that represent a mixture of multiple Gaussian distributions. Unlike K-Means clustering, which assigns each data point to a single cluster, GMMs provide a soft clustering approach by assigning probabilities to each data point belonging to each cluster.

Mathematical Formulation

Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$, GMM assumes that the data is generated from a mixture of k Gaussian distributions. The probability density function for a data point x is:

$$p(x) = \sum_{i=1}^{k} \pi_i \cdot \mathcal{N}(x \mid \mu_i, \Sigma_i)$$

where:

- π_i is the mixing coefficient for the i^{th} Gaussian component, with $\sum_{i=1}^{k} \pi_i = 1$ and $\pi_i \geq 0$.
- $\mathcal{N}(x \mid \mu_i, \Sigma_i)$ is the Gaussian distribution with mean μ_i and covariance matrix Σ_i.

The parameters $\{\pi_i, \mu_i, \Sigma_i\}$ are typically estimated using the Expectation-Maximization (EM) algorithm.

Clustering with $k = 6$

A Gaussian Mixture Model was applied with $k = 6$ components. The resulting clusters are visualized in Figure 13.

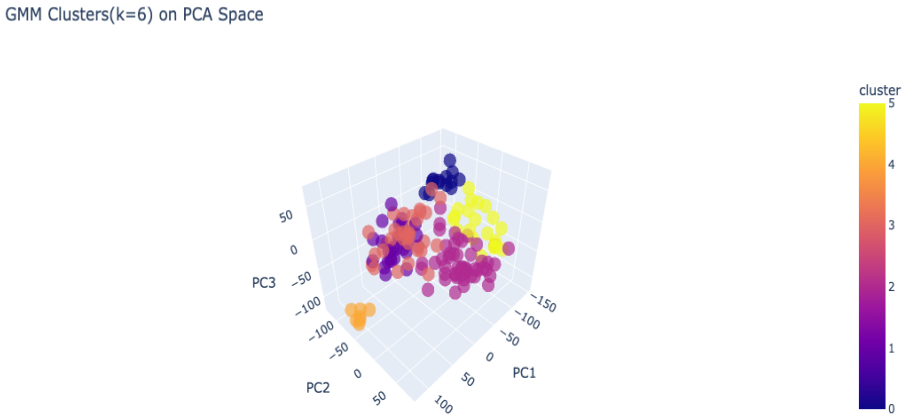

Figure 13: GMM Clustering with $k = 6$

Silhouette Score Analysis

To determine the optimal number of clusters, silhouette scores were computed for different values of k. The silhouette score measures how similar an object is to its own cluster compared to other clusters. The scores range from -1 to 1, with higher values indicating better clustering.

Table 5 presents the silhouette scores for various values of k.

Table 5: Silhouette Scores for Different Values of k using GMM

Number of Clusters (k)	Silhouette Score
3	0.6288
4	0.4910
5	0.4163
6	0.5291
7	0.4456
8	0.4298

The highest silhouette score is observed at $k = 3$, suggesting that three clusters may be optimal for this dataset.

Final Clustering with Optimal k

Based on the silhouette analysis, GMM clustering was performed with $k = 3$. The resulting clusters are visualized in Figure 14.

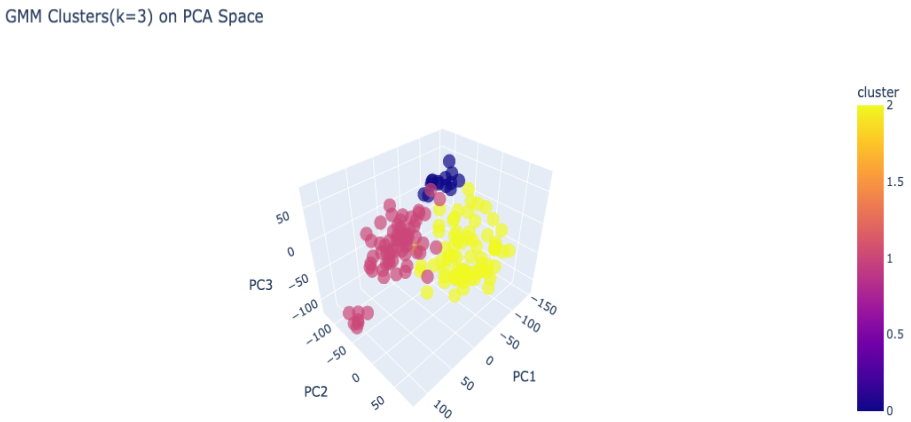

Figure 14: GMM Clustering with Optimal $k = 3$

3.1.7 Fuzzy C-Means

Fuzzy C-Means (FCM) [2] is a clustering algorithm that allows data points to belong to multiple clusters with varying degrees of membership. Unlike traditional hard clustering methods, such as K-Means, where each data point is assigned to a single cluster, FCM provides a soft clustering approach by assigning membership levels to each data point for every cluster. This approach is particularly useful for datasets where clusters may overlap or have ambiguous boundaries.

Mathematical Formulation

Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$, FCM aims to partition the data into c clusters by minimizing the following objective function:

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m \|x_i - c_j\|^2$$

where:

- u_{ij} is the degree of membership of data point x_i in cluster j,
- $m > 1$ is the fuzziness parameter that determines the level of cluster fuzziness,
- c_j is the center of cluster j,
- $\|x_i - c_j\|$ is the Euclidean distance between data point x_i and cluster center c_j.

The algorithm iteratively updates the membership degrees and cluster centers until convergence.

Clustering with $k = 6$

The FCM algorithm was applied with $k = 6$ clusters. The resulting clusters are visualized in Figure 15.

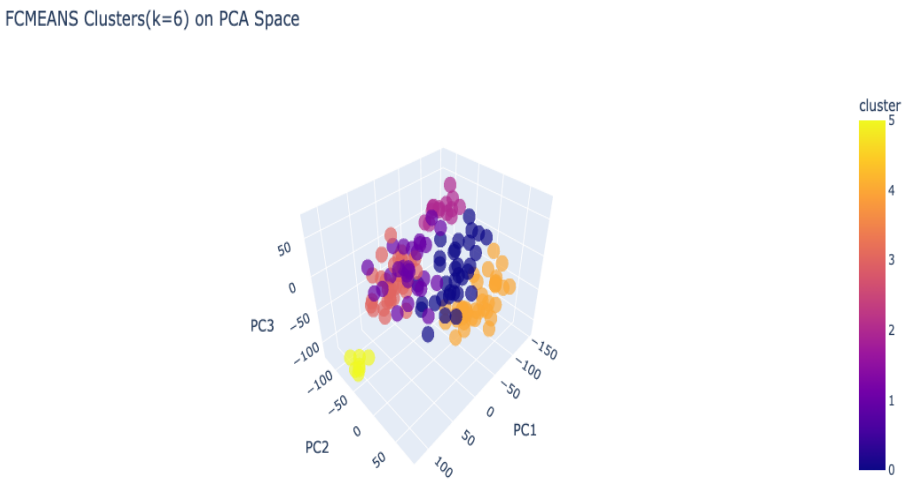

Figure 15: Fuzzy C-Means Clustering with $k = 6$

Silhouette Score Analysis

To determine the optimal number of clusters, silhouette scores were computed for different values of k. The silhouette score measures how similar an object is to its own cluster compared to other clusters. The scores range from -1 to 1, with higher values indicating better clustering.

Silhouette scores are shown in Table 6

Table 6: Silhouette Scores for Different Values of k using FCM

Number of Clusters (k)	Silhouette Score
3	0.6288
4	0.2468
5	0.3892
6	0.4699
7	0.4901
8	0.4677

The highest silhouette score is observed at $k = 3$, suggesting that three clusters may be optimal for this data set.

Final Clustering with Optimal k

Based on the silhouette analysis, the FCM clustering was performed with $k = 3$. The resulting clusters are visualized in Figure 16.

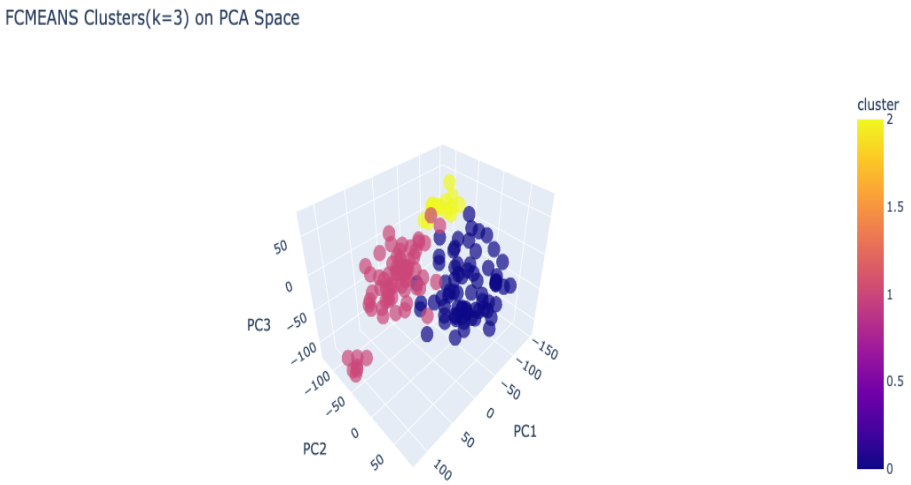

Figure 16: Fuzzy C-Means Clustering with Optimal $k = 3$

3.2 Performance Evaluation

Evaluating the performance of clustering algorithms is crucial to understand the quality of the clusters formed. Common evaluation metrics include Adjusted Rand Index (ARI), Silhouette Score, and Purity. This report presents these metrics for various clustering algorithms applied to the dataset.

3.2.1 Evaluation Metrics for $k = 6$ Clusters

The following table summarizes the performance of different clustering algorithms when the number of clusters is fixed at $k = 6$.

Table 7: Clustering Performance Metrics for $k = 6$ Clusters

Method	ARI	Silhouette Score	Purity
KMeans	0.6341	0.5565	0.7616
Mahalanobis	0.3348	0.2488	0.6093
GMM	0.5607	0.5291	0.7218
DBSCAN	0.0803	0.3803	0.3311
FCM	0.8173	0.4699	0.9272
Agglomerative	0.6549	0.7534	0.7748

3.2.2 Evaluation Metrics for Optimal Number of Clusters

The optimal number of clusters for each method was determined using appropriate techniques (e.g., silhouette analysis. The following table presents the performance metrics for each method using its optimal number of clusters.

Table 8: Clustering Performance Metrics for Optimal Number of Clusters

Method	**ARI**	**Silhouette Score**	**Purity**
KMeans	0.5061	0.6288	0.5629
Mahalanobis	0.3251	0.4702	0.5099
GMM	0.5061	0.6288	0.5629
DBSCAN	0.0803	0.3803	0.3311
FCM	0.5061	0.6288	0.5629
Agglomerative	0.5658	0.8330	0.6093

3.2.3 Evaluation Metric Definitions

Adjusted Rand Index (ARI)

The Adjusted Rand Index measures the similarity between two clustering assignments, adjusted for chance grouping. It is defined as:

$$\text{ARI} = \frac{\text{RI} - E[\text{RI}]}{\max(\text{RI}) - E[\text{RI}]}$$

where:

- RI is the Rand Index.
- $E[\text{RI}]$ is the expected Rand Index for random clusterings.

The ARI ranges from -1 (complete disagreement) to 1 (perfect agreement), with 0 indicating random labeling.

Silhouette Score

The Silhouette Score assesses how similar an object is to its own cluster compared to other clusters. For a data point i, it is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where:

- $a(i)$ is the average distance from i to all other points in the same cluster.
- $b(i)$ is the minimum average distance from i to points in a different cluster.

The score ranges from -1 to 1, with higher values indicating better clustering structure.

Purity

Purity measures the extent to which clusters contain a single class. It is calculated as:

$$\text{Purity} = \frac{1}{N} \sum_{k} \max_{j} |C_k \cap L_j|$$

where:

- N is the total number of data points.
- C_k is the set of data points in cluster k.
- L_j is the set of data points in class j.

Purity ranges from 0 to 1, with higher values indicating better clustering performance.In general Purity score is not a good evaluation metric if number of clusters is high but in this case the number of clusters is low(less than 6) so we can use purity score as a metric.

3.3 Determining Potential Biomarkers within each cluster

In the analysis of gene expression data, identifying biomarker genes that are significantly differentially expressed across clusters is crucial. For each cluster, a two-sample t-test was performed comparing the expression levels of genes within the cluster against those outside the cluster. Genes with a p-value less than 0.01 were considered significant, and among these, the top 5 genes with the highest absolute t-statistics were selected as potential biomarkers for each cluster.

Table 9: Top 5 Biomarker Genes for Each Cluster

Class	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
0	216836_s_at	210930_s_at	210761_s_at	224447_s_at	234354_x_at
1	218211_s_at	226961_at	235046_at	229150_at	228302_x_at
2	202878_s_at	225353_s_at	223343_at	200795_at	213975_s_at
3	204603_at	212195_at	236641_at	205046_at	219010_at
4	228241_at	205225_at	232322_x_at	209173_at	223103_at
5	222981_s_at	219935_at	222262_s_at	240277_at	215438_x_at

3.4 Discussion

The clustering results generally aligned well with known breast cancer subtypes when the number of clusters was fixed at six. Among the methods, Agglomerative Clustering with Cosine distance and Gaussian Mixture Models showed the strongest alignment based on evaluation metrics like ARI, Silhouette Score, and Purity. However, silhouette score trends across all methods consistently indicated that the optimal number of clusters may be closer to three, suggesting biological overlaps among certain subtypes.

Additionally, class imbalance—particularly the small sizes of Class 2 (Cell line) and Class 5 (Normal-like)—likely impacted the clustering quality and model robustness. Despite this, the biomarker analysis revealed several differentially expressed genes with strong statistical significance, providing insights into underlying biological mechanisms that distinguish clusters.

This supports the potential of unsupervised learning not only in subtype separation but also in the discovery of candidate genes for further clinical validation.

The alignment between our clustering results and known subtypes suggests that these methods could help improve breast cancer classification, especially in cases where the standard subtype labels don't fully capture the tumor's biology. This is particularly valuable for personalized medicine, where accurate classification can guide more effective treatments.

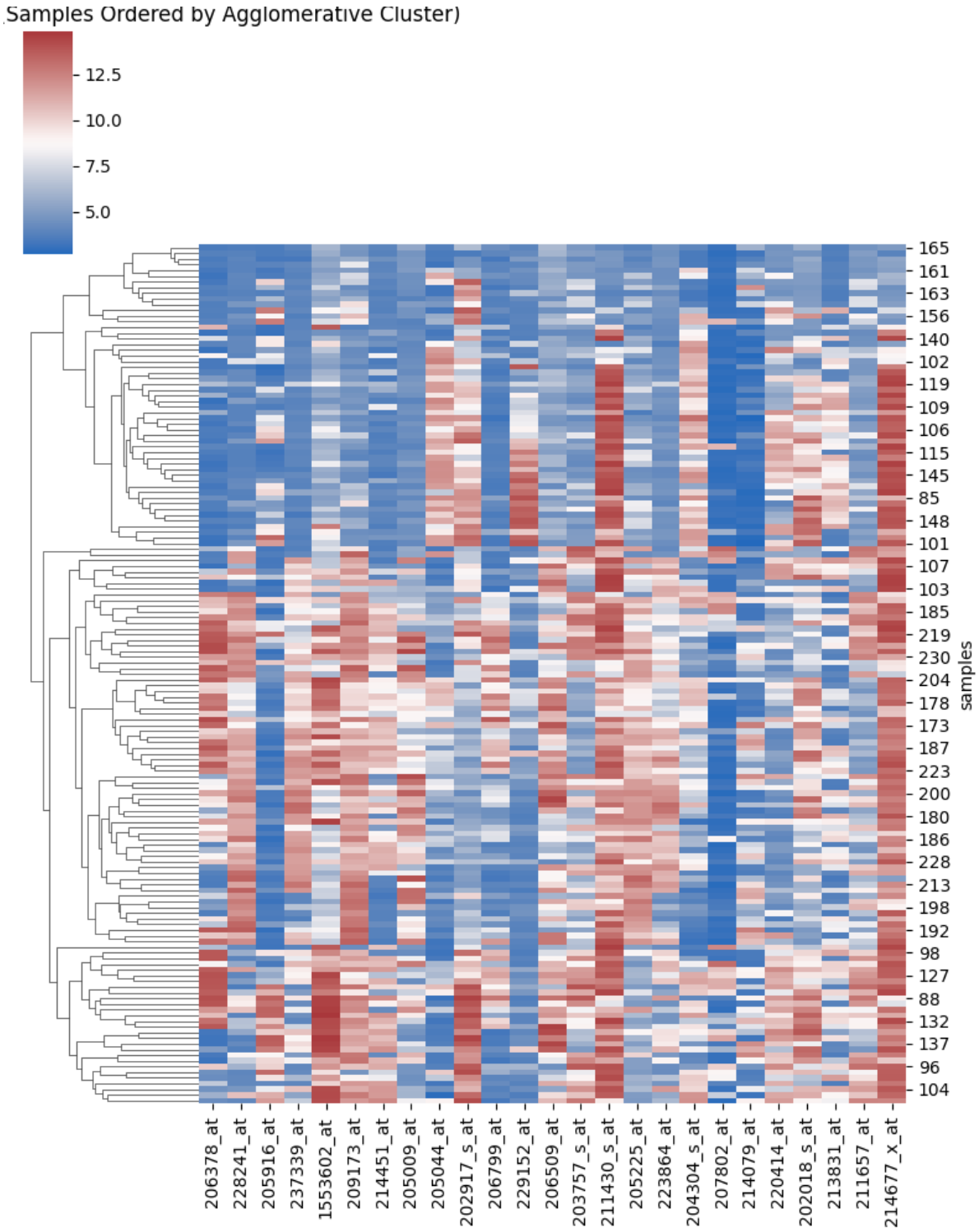

Figure 17: Heatmap of Top 25 Variable Genes Ordered by Agglomerative Clustering

4 Additional Observations and Insights

– The silhouette analysis consistently indicated $k = 3$ as the optimal number of clusters in multiple algorithms, despite the known six-class structure. This could suggest a higher-level latent structure or similarities between subtypes

such as luminal A / B or HER2 / Basel.

- The observed overlap in expression signatures between Luminal A and B, as well as between HER2-enriched and basal-like subtypes, may point toward a latent biological axis that governs tumor behavior beyond the classical subtype classification.Instead of being clearly separate groups, Luminal A and B might lie on a spectrum, with gradual differences in how fast the cancer cells grow or how aggressive they are.
- UMAP provided significantly better visual separation of clusters compared to PCA, highlighting the importance of non-linear dimensionality reduction techniques in gene expression analysis.
- DBSCAN underperformed due to the curse of dimensionality and lack of clear density-separated clusters. This demonstrates the need for careful parameter tuning or the use of adaptive density-based methods for high-dimensional data.
- Algorithms like GMM and Fuzzy C-Means showed greater tolerance to class imbalance, as evidenced by their relatively stable performance metrics even when small clusters were present.
- The Mahalanobis distance-based K-Means variant underperformed due to unreliable covariance estimates in high-dimensional, low-sample contexts, reaffirming the risk of overfitting with small sample sizes.
- Some biomarker genes appeared consistently across different clustering methods, suggesting their robustness and potential biological relevance across various analytical pipelines.

5 Conclusion

This project explored unsupervised learning methods to uncover the latent structure in breast cancer gene expression data from the CuMiDa dataset [1]. After applying preprocessing and dimensionality reduction via PCA and UMAP, we evaluated several clustering algorithms to uncover potential subtypes and biomarkers.

Despite the six known classes, clustering evaluations consistently favored three clusters, suggesting possible biological overlap or higher-level subtype relationships. Among the methods used, Agglomerative Clustering with Cosine distance and Gaussian Mixture Models achieved the best performance, aligning closely with known subtype distributions.

Biomarker analysis revealed differentially expressed genes specific to each cluster, demonstrating the potential of clustering methods in real-world genomic insight.

These results underscore the value of unsupervised learning in oncology and set a foundation for future integrative analyzes involving clinical and molecular data.

Future work could explore integrating clinical metadata or using ensemble clustering to improve robustness

References

[1] Bruno Grisci. *Breast Cancer Gene Expression - CuMiDa*, 2023.

[2] Maria Amélia Nascimento, Hugo C. Medeiros, Raul C. Lima, and João Paulo Papa. *Breast cancer diagnosis based on mammary thermography and extreme learning machines*. Computational and Mathematical Methods in Medicine, 2016:1–11, 2016.