



**Discovery of potent prognostic biomarkers in cancer Using different clustering methods**

**By**

**Tejaswini Harshitha Samudrala**

**A DISSERTATION**

**Submitted to**

**The University of Liverpool**

**in partial fulfillment of the requirements for the degree of**

**MASTER OF SCIENCE**

**06-12-2024**

**STUDENT DECLARATION**

I hereby certify that this dissertation constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of another.

I confirm that I have not copied material from another source nor committed plagiarism nor commissioned all or part of the work (including unacceptable proof-reading) nor fabricated, falsified, or embellished data when completing the attached piece of work.

I declare that the dissertation describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

Tejaswini Harshitha Samudrala.

**ACKNOWLEDGEMENTS**

I would like to express my deepest gratitude to my supervisor, Dr.Olga Anosova, for her invaluable guidance, encouragement, and constructive feedback throughout this project. Their expertise and insights have been instrumental in shaping the direction and execution of this research.

I am also profoundly thankful to the University of Liverpool for providing the resources and academic environment necessary for this study. I especially thank the faculty and staff of the Department of Computer Science for their support and assistance. Their patience and encouragement during challenging moments have motivated me to overcome obstacles and strive for excellence.

I extend my heartfelt appreciation to my family and friends for their unwavering encouragement and patience during the demanding phases of this project. Their support has been my constant source of motivation.

I sincerely thank all who have contributed, directly or indirectly, to the completion of this work. This research could not have been accomplished without all your support and contributions.

Lastly, I acknowledge the open-access resources and datasets that made this study possible, as well as the broader research community whose work has inspired and informed my own.

**Discovery of potent prognostic biomarkers in cancer**

**Using different clustering methods**

**ABSTRACT**

The discovery and analysis of prognostic biomarkers in cancer are critical to improving patient outcomes and enhancing personalized treatment. The study focuses into the use of unsupervised machine learning techniques, specifically clustering algorithms, to analyze high-dimensional gene expression datasets obtained from the National Center for Biotechnology Information (NCBI). The objective of this project is to use computational tools to find meaningful gene or sample clusters that can help with cancer prognosis and treatment strategies.   
The study employs three commonly used clustering algorithms—K-Means, Hierarchical Clustering, and DBSCAN—each of which provides distinct advantages in dealing with the complexity and variability of genomic data. In order to guarantee data quality and comparability, rigorous techniques for preprocessing were applied, such as managing missing values and normalization using Min-Max Scaling. Using internal validation criteria like the Davies-Bouldin Index and Silhouette Score, the efficacy of these algorithms was methodically assessed, offering information on their biological significance and clustering performance.   
To improve the interpretability and accessibility of clustering results, the study emphasizes visualization tools in addition to algorithmic evaluations. These visualizations let data scientists and subject matter experts collaborate by connecting computational results with biological insights.

The results demonstrate how clustering approaches might reveal unique gene expression patterns linked to the prognosis of cancer. The paper does, however, also note enduring difficulties, such as the innate complexities of high-dimensional datasets, the drawbacks of particular clustering algorithms, and the requirement for outside validation to verify biological importance. By tackling these issues, our study adds to the larger biomarker discovery endeavours and emphasizes how crucial it is to combine biological and computational viewpoints to advance cancer research.

**STATEMENT OF ETHICAL COMPLIANCE**

Data Category: D

Participant Category: 0

I confirm that I have carefully read the regulations concerning ethical conduct and pledge to follow them carefully throughout this project. This project uses anonymous genetic data obtained from the NCBI GEO database, which is publicly available and does not require any specific permissions from the author. The study follows the University of Liverpool's Code of Conduct.

**TABLE OF CONTENTS**

**Page No**

**LIST OF TABLES 7**

**LIST OF FIGURES 8**

**Chapter 1. Introduction 9**

**Chapter 2. Background and review of the literature 10**

**Chapter 3. Design 13**

**Chapter 4. Libraries 14**

**Chapter 5. Overview of the Dataset 15**

**Chapter 6. Methodology and Implementation 16**

**Chapter 7. Analysis of the Clustering Results 26**

**Chapter 8. Conclusion 30**

**Chapter 9. Future Scope 30**

**Chapter 10.BCS Criteria 31**

**Chapter 11. Self-Reflection 33**

**Chapter 12. Project Ethical Considerations 34**

**REFERENCES CITED 35**

**Appendices 37**

**LIST OF TABLES**

**Page No**

**Table 1 Comparing Clustering Results 28**

**Table 2 BCS criteria Table 30**

**LIST OF FIGURES**

**Page No**

[**Figure 1. K-Means Clustering Visualization**](file:///C:\Users\tejas\Downloads\COMP702-Dissertation-Template-word.docx#_Toc120440498) **19**

**Figure 2. Hierarchical Clustering Visualization 20**

**Figure 3. DBSCAN Clustering Visualization 22**

**Figure 4. Jaccard index Venn diagram 25**

**Chapter 1: Introduction**

* 1. **Scope**

This dissertation discusses the use of unsupervised machine-learning algorithms to find prognostic biomarkers in cancer. The study specifically examines clustering algorithms for analyzing high-dimensional gene expression data obtained from the National Centre for Biotechnology Information (NCBI). The selection, use, and assessment of clustering techniques—K-Means, Hierarchical Clustering, and DBSCAN—as well as their capacity to detect significant patterns in genomic datasets are all included in the scope of this study.   
This project highlights visualization strategies to improve interpretability, even if its main focus is on the computational components of biomarker discovery, such as data pretreatment, algorithm selection, and validation. However, due to limitations in time and project focus, this study does not explore experimental verification of identified biomarkers or external validation using clinical data. Rather, algorithmic performance and the knowledge gained from computational studies are highlighted.

**1.2 Problem Statement**

The range of types of cancer presents major obstacles to detection, prognosis, and treatment, and it continues to rank among the world's leading causes of death. Customized treatment strategies and risk-based patient classification depend on the discovery of reliable prognostic biomarkers. However, the complexity of gene expression data—which is marked by high dimensionality, noise, and variability—makes it challenging to uncover such biomarkers.

The complex dynamics in such data are sometimes not captured by traditional statistical methods. Even though machine learning offers strong tools for analyzing large datasets, there are challenges with algorithm selection, evaluation, and interpretability when using it to generate biomarkers. This work attempts to address these problems by systematically evaluating clustering algorithms and their capacity to uncover biologically meaningful patterns**.**

**1.3 Approach**

The approach mainly consisted of three parts:

1. **Data Preparation**: The project begins with the gene expression datasets from NCBI. To make sure the dataset is consistent and of high quality, preprocessing procedures are carried out, such as addressing missing values and using Min-Max Scaling.
2. **Algorithm Application**: K-Means, Hierarchical Clustering, and DBSCAN are three clustering methods that are used to organize genes or samples into groups according to patterns of expression. These algorithms were chosen because of how well they complement one another when processing high-dimensional data.
3. **Evaluation and Visualization**: Internal validation criteria like the Davies-Bouldin Index and Silhouette Score are used to evaluate the clustering algorithms' performance. Additionally, Python libraries are used to show the clustering findings.

**1.4 Outcome**

The project successfully demonstrated the application of clustering algorithms K-Means, Hierarchical Clustering, and DBSCAN in analyzing gene expression data to identify potential prognostic biomarkers in cancer. Each algorithm provided unique insights: K-Means revealed distinct gene groups with tight expression patterns, Hierarchical Clustering highlighted functional hierarchies and relationships, and DBSCAN identified outliers that could represent rare or subtype-specific biomarkers. Although certain clusters demonstrated good cross-method alignment, disparities uncovered the difficulties associated with complex gene data and algorithmic sensitivity. Although biological interpretation is limited, the clustering data provide a basis for future research on cancer biomarkers.

**Chapter 2. Background and Review of Literature**

**2.1 Related Work**

The discovery of prognostic biomarkers has changed dramatically during the last 20 years due to computational techniques and high-throughput genomic technology developments. In order to find biomarkers that are associated with patient outcomes, treatment responses, and the advancement of the disease, several studies have investigated machine learning algorithms for analyzing gene expression data.   
Unsupervised learning techniques, especially clustering techniques, have exposed previously unnoticed patterns in genetic data. For instance, Perou et al. (2000) classified breast cancer into molecular subtypes such as basal-like, HER2-enriched, luminal A, and luminal B using hierarchical clustering. The foundation for later biomarker-finding initiatives was established by this ground-breaking investigation. Similar to this, Monti et al. (2003) used K-Means clustering to determine diffuse large B-cell lymphoma subgroups, allowing for a more complex understanding of the disease.

Despite being used less frequently in early genomic research, DBSCAN has drawn notice for its capacity to manage noise and asymmetric cluster forms, especially in intricate datasets like single-cell RNA sequencing. The effectiveness of DBSCAN in locating uncommon cell subpopulations that may function as biomarkers for disease states was shown by studies such as Xu et al. (2019).   
  
Despite these achievements, the literature still ignores important issues. Numerous research only uses one dataset, which restricts how broadly the results may be applied. Furthermore, because of resource limitations, the confirmation of computational results with experimental or clinical data is frequently delayed. These difficulties show how reliable, interpretable, and scalable approaches are needed to close the gap between clinical application and computational discovery.

**2.2 Literature**

**Introduction to Biomarkers in Cancer Research**

Biomarkers are measurable indicators of a biological state or condition that play an important role in cancer detection, prognosis, and treatment. These could be proteins, DNA, or other substances that provide information about an organism's physiological or pathological condition. Thanks to developments in proteomics, transcriptomics, and genomes, the search for biomarkers has become more prevalent. Finding credible prognostic biomarkers can enhance individualized treatment plans and forecast patient outcomes, based on research.

**Cancer Biomarkers and Clustering Techniques**

Various studies have leveraged clustering algorithms to identify patterns in high-dimensional biological data. Clustering, an unsupervised machine learning approach, helps uncover relationships and groupings in datasets without prior knowledge of labels. Techniques such as k-means, hierarchical clustering, and DBSCAN are frequently employed to segment gene expression data into clusters that may correspond to disease subtypes or biomarker groups.

* **K-Means Clustering**: A study by Jain et al. (2010) highlighted k-means as a robust method for identifying distinct cancer subtypes based on gene expression profiles. While it is computationally efficient, its reliance on predefined cluster numbers can limit its adaptability in complex datasets.
* **Hierarchical Clustering**: Used in studies such as Eisen et al. (1998), hierarchical clustering organizes data into a tree structure (dendrogram), enabling researchers to visualize relationships between genes or samples. This method is particularly useful for exploratory analyses but may struggle with scalability in large datasets.
* **DBSCAN**: Density-based clustering approaches, like DBSCAN, have been less commonly applied but offer advantages in identifying clusters of arbitrary shapes and distinguishing noise points, as demonstrated by Ester et al. (1996).

**2.3 Industry Sources**

The application of biomarker discovery extends beyond academic research to industry, where it plays a vital role in diagnostics, drug development, and personalized medicine.

1. **High-Throughput Sequencing Technologies**:  
   Companies like Illumina and Thermos Fisher Scientific provide state-of-the-art sequencing platforms, enabling researchers to generate vast amounts of genomic data. These datasets form the backbone of computational biomarker discovery efforts. For instance, Illumina's Nova Seq platform supports transcriptomic studies that identify differentially expressed genes, a precursor to biomarker identification.
2. **AI-Driven Platforms**:  
   The integration of machine learning in the healthcare industry has led to the development of platforms such as IBM Watson for Genomics and NVIDIA Clara, which assist researchers in processing and interpreting complex genomic data. These platforms incorporate clustering algorithms alongside other analytical tools to identify potential biomarkers with clinical relevance.
3. **Translational Research and Drug Development**:  
   Pharmaceutical companies, including Roche and AstraZeneca, are leveraging computational biomarker discovery to classify patients during clinical trials. By identifying molecular subtypes, they aim to develop targeted therapies with higher efficacy and reduced adverse effects. For example, Roche’s efforts in HER2-targeted therapies for breast cancer were bolstered by biomarker stratification during drug development.
4. **Challenges in Industry Adoption**:  
   Despite these advancements, challenges persist in translating computational biomarker discovery into clinical practice. Regulatory hurdles, data privacy concerns, and the need for reproducibility remain significant barriers. Industry stakeholders emphasize the importance of standardized workflows and validation protocols to ensure the clinical utility of computational findings.

**Chapter 3. Design**

The design of this study was conceived with the primary aim of identifying prognostic biomarkers in cancer through the application of unsupervised machine-learning techniques. This design was structured to encompass the following key components:

* 1. **Data Acquisition**:  
     Gene expression datasets were sourced from the National Centre for Biotechnology Information (NCBI). The datasets were selected based on their relevance to specific cancer types and their suitability for clustering-based analyses.
  2. **Preprocessing Pipeline**:  
     A robust preprocessing workflow was planned to address common issues in gene expression data, such as missing values and scale differences. The proposed steps included:

1. Handling missing values through imputation techniques.

2. Normalizing the data using Min-Max Scaling to ensure uniformity across features.

3. Dimensionality reduction through feature selection techniques to address the high-dimensional nature of the data.

* 1. **Clustering Algorithm Selection**:  
     The study initially aimed to compare the performance of three clustering algorithms—K-Means, Hierarchical Clustering, and DBSCAN. These algorithms were chosen for their complementary strengths in handling diverse data characteristics.
  2. **Evaluation and Validation**:  
     Internal validation metrics, including the Silhouette Score and Davies-Bouldin Index, were selected to assess clustering performance. Additionally, the results were to be validated against known biological information to evaluate their relevance in identifying prognostic biomarkers.
  3. **Visualization**:  
     A strong emphasis was placed on visualization to enhance interpretability. Tools such as heatmaps, dendrograms, and scatterplots were planned to illustrate clustering outcomes and their biological implications.

**Chapter 4. Libraries**

In this project, several Python libraries were utilized to process the data, perform clustering, and evaluate the results. These libraries are well known for their effectiveness in data science, machine learning, and statistical analysis. During the course of my project, I have used various libraries that include

**4.1. Pandas**

**Purpose:** Data Manipulation and Analysis  
Pandas is a powerful library used for data manipulation and analysis. In this project, it was primarily used to load and preprocess the dataset. It provided essential functions for handling missing values, filtering rows, and normalizing the data. The dataset was loaded into a Pandas Data Frame using the **pd.read\_csv()** function, which enabled easy access and manipulation of the gene expression data.

**4.2. Scikit-learn (sk learn)**

**Purpose:** Machine Learning and Data Preprocessing  
Scikit-learn is an open-source machine-learning library that offers simple and efficient tools for data mining and analysis. For this project, it was used for both data preprocessing and applying clustering algorithms. It helped normalize the data using the **MinMaxScaler**, which scales features to a fixed range (typically 0 to 1), and it provided several clustering algorithms including K-Means, DBSCAN, and Agglomerative Clustering. Using the Scikit Learn significantly reduced the run time of all three algorithms.

**4.3. Matplotlib**

**Purpose:** Data Visualization  
Matplotlib is a widely used library for creating static, animated, and interactive visualizations in Python. In this project, it was used to create scatter plots that visualize the clustering results. Visualizing data is crucial for understanding the distribution of clusters and for presenting the results in a clear and interpretable way.

**4.4. NumPy**

**Purpose:** Numerical Computing  
NumPy is a fundamental package for scientific computing in Python. It is especially useful for handling arrays and performing complex mathematical operations. In this project, NumPy was indirectly used, as many of the functions in Scikit-learn rely on NumPy arrays. Additionally, it facilitated the manipulation of data in the form of matrices and multidimensional arrays for clustering tasks.

**Chapter 5. Overview of the Dataset**

The dataset used in this project consists of gene expression data, a common type of data used in biomedical research, particularly for cancer studies. The dataset used in this project is derived from the Gene Expression Omnibus (GEO) repository, specifically under accession ID [GSE202203](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE202203). GEO is a publicly accessible resource that archives high-throughput gene expression data submitted by the research community. This particular dataset includes gene expression levels across various samples, providing a robust foundation for clustering and biomarker discovery in cancer research.

**5.1 Structure of the Dataset:**

1. Rows (Gene Identifiers):19644

* Each row represents a unique gene, identified by its corresponding gene name (e.g., *A1BG*, *A1CF*, etc.).

1. Columns:3208

* The first column, labelled X, contains the gene names.
* The remaining columns (e.g., S000001, S000002) correspond to various samples or experimental conditions.
* Each value in the matrix represents the expression level of a gene (row) in a specific sample (column). Expression levels are typically measured in terms of read counts, normalized intensity values, or similar metrics derived from gene expression profiling technologies like RNA-Seq or microarrays.

1. Dataset Dimensions:

* The cleaned dataset has 19,582 rows and 3,208 columns, indicating data for 19,582 genes across 3,207 samples.

**5.2 Characteristics of the Data:**

1. **Gene Expression Levels:**

* Expression levels vary significantly across genes and samples, reflecting the biological diversity and variability inherent in the data. For example, Higher values (e.g., 435.26, 771.34) may indicate genes with high activity in specific samples. Zero or near-zero values suggest genes that are inactive or not expressed in certain conditions.

1. **Missing or Zero Values:**

* Rows with all-zero values across samples (genes not expressed in any condition) were excluded during preprocessing to enhance the dataset's informativeness and computational efficiency.

1. **Variability Across Samples:**

* The dataset captures how gene expression varies between samples. These variations could be due to differences in tissue types, disease states, or experimental conditions, which are key factors in clustering and biomarker discovery.

**Chapter 6. Methodology and Implementation**

**6.1. Data Collection and Preprocessing**

This study uses a gene expression dataset, in which the columns show the gene expression values across different samples, and each row represents a distinct gene. The file format used to collect this information is tab-separated values (TSV), which is frequently used to handle large data.

**6.1.1 Data Cleaning**

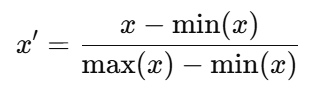
The dataset was cleaned to make sure it contained relevant information before any analysis could begin. This involved identifying and removing rows where all the gene expression values were zero. Rows with all-zero values are likely to represent irrelevant or missing data that would not contribute useful insights for clustering. By removing these rows, the dataset was refined to only include those genes that had at least some variation in their expression across samples, which is critical for clustering. This was achieved using a Boolean condition that identified rows where all columns, except the first one (typically containing identifiers like gene names), had a value of zero. The condition **(data.iloc[:, 1:] == 0).all(axis=1)** was used for this purpose. Here:

* **data.iloc[:, 1:]**: This operation selects all rows (:) and all columns starting from the second one (1:) in the dataset. This is done to exclude the identifier column from the operation, as it typically contains non-numeric data like gene names.
* **== 0**: This checks whether the values in these selected columns are equal to zero.
* **.all(axis=1)**: This determines if the condition (all values being zero) is true across all columns for a given row. The axis=1 argument specifies that the operation is performed row-wise.

Rows satisfying this condition were removed using the negation operator ~, resulting in a cleaned dataset devoid of rows with zero values across all numeric features.

**6.1.2 Data Normalization**

Gene expression data can vary widely in scale. This means that the findings of clustering may be skewed if certain genes have extremely high expression levels while others have low expression levels. To address this, the data underwent a **normalization** process. Specifically, **Min-Max scaling** was used, which transforms each feature (gene expression value) to a range between 0 and 1. This ensures that no single gene dominates the clustering process due to differences in scale, allowing the clustering algorithms to treat all features equally. Min-Max Scaling was implemented via **MinMaxScaler()** from the **sklearn** preprocessing module. Min-Max Scaling transforms each feature in the dataset to fall within a range of 0 to 1, calculated using the formula:



Where X´ is the original value, min(x)is the minimum value of the feature, and max(x) is the maximum value of the feature.

The scaling process was applied to the numeric columns of the dataset. The operation **data\_cleaned.iloc[:, 1:]** was again used to select the numeric features, excluding the identifier column. The scaled values were then reassigned to these columns within a copy of the dataset, **data\_normalized**. This ensured that the original dataset remained unaltered.

**6.3. Clustering Algorithms**

Clustering is an unsupervised learning technique used to group similar data points. In this analysis, three different clustering algorithms were applied to the normalized gene expression data to compare how they perform and how they categorize the genes. These algorithms were **K-Means**, **Agglomerative (Hierarchical) Clustering**, and **DBSCAN**.

**6.3.1 K-Means Clustering**

**K-Means** is one of the most widely used clustering techniques. The algorithm works by partitioning the dataset into a pre-defined number of clusters (in this case, 3 clusters, obtained through trial and error). K-Means tries to minimize the **variance** within each cluster, grouping genes that have similar expression profiles together. It starts by randomly assigning cluster centroids (the mean position of each cluster) and iteratively refines these centroids until the algorithm converges, meaning the centroids no longer change.

**Initialization of the KMeans Algorithm**:

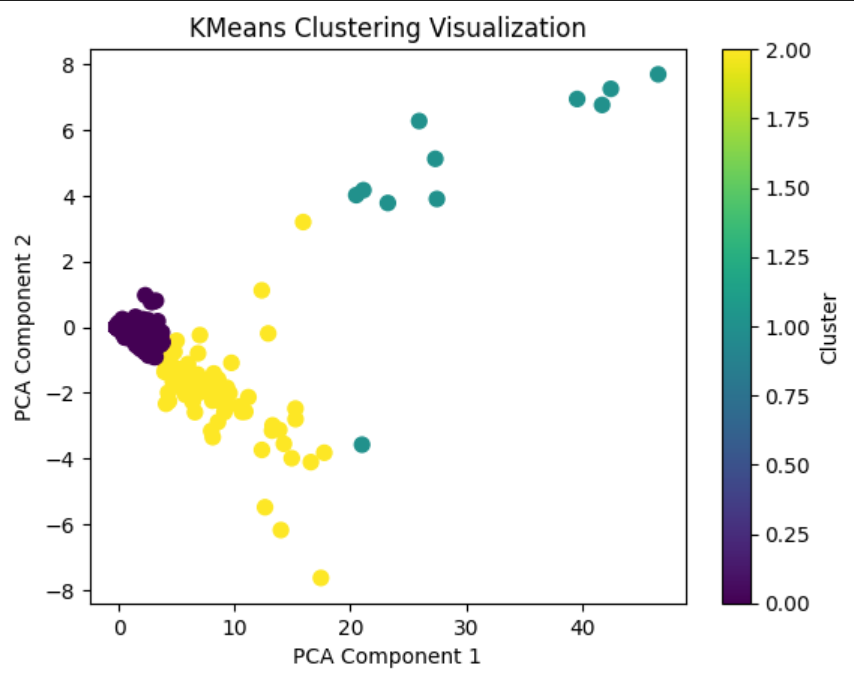
* **KMeans(n\_clusters=3, random\_state=42)** initializes the KMeans clustering model.
* **n\_clusters=3**: This parameter specifies that the data should be grouped into three clusters. The choice of three clusters was based on prior analysis and trail and error, assuming the dataset could naturally form three distinct groups.
* **random\_state=42**: This ensures the reproducibility of the clustering results. The random state is a seed value used by the algorithm's random number generator to initialize the centroids consistently across different runs.

**Input Data Preparation**:

* **data\_normalized.iloc[:, 1:]** selects all rows and numeric columns of the normalized dataset for clustering. The identifier column (e.g., gene names) is excluded because clustering operates solely on numeric features. Here it uses the .iloc method to select all rows (:) and numeric columns (from the second column onward, indexed as 1:) for clustering.

**Fitting the Model and Predicting Cluster Labels**:

* **kmeans.fit\_predict(data\_normalized.iloc[:, 1:])** performs two tasks simultaneously:
* **Fitting**: The K-Means algorithm is applied to the input data. During this step:
* K-Means initializes three random centroids (as per n\_clusters=3).
* Each data point is assigned to the nearest centroid based on the Euclidean distance metric.
* Centroids are recalculated iteratively as the mean of all data points assigned to each cluster.
* This process repeats until the centroids stabilize, indicating convergence
* **Predicting**: Once the algorithm converges, each data point is assigned a cluster label corresponding to the nearest centroid.
* The output, **kmeans\_labels**, contains an array of cluster labels (e.g., 0, 1, or 2) assigned to each data point in the dataset. These labels show which of the three clusters a given data point is associated with.
* Figure 1 shows the 3 clusters formed by the K-means clustering algorithm.

****

**Figure 1**

**6.3.2 Agglomerative Hierarchical Clustering**

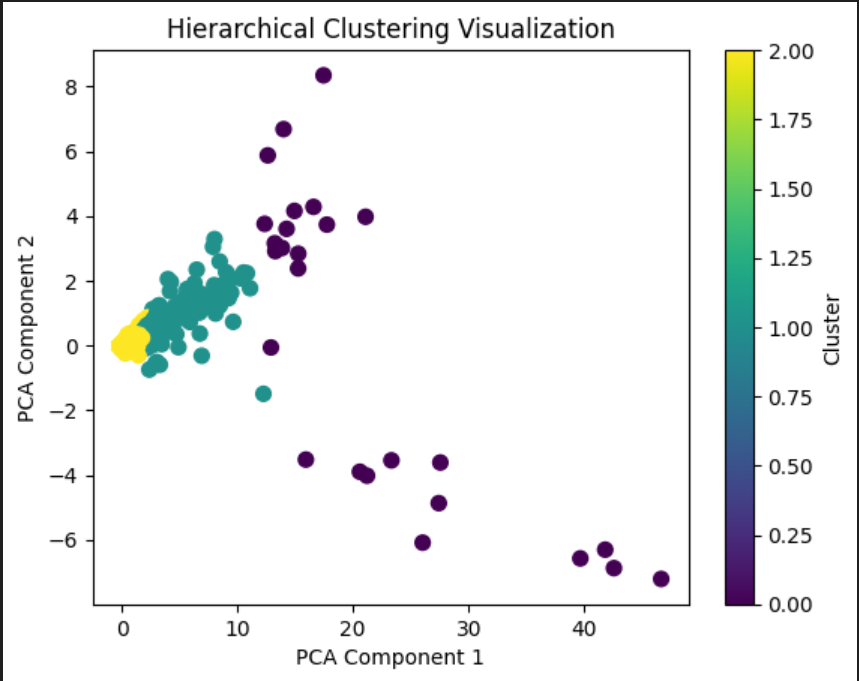
**Agglomerative Clustering**, also known as **Hierarchical Clustering**, works by building a hierarchy of clusters. Each gene is first treated as a distinct cluster by the algorithm, which then iteratively combines the nearest clusters according to how similar they are. Until a certain number of clusters is attained, the merging process keeps going. Although we set the number of clusters to three in this instance through trial and error, the user is not required to specify the number at the beginning of this procedure. The benefit of hierarchical clustering is that it offers a clear picture of the cluster formation process at each stage, which can help with data structure comprehension.

**Agglomerative Clustering Class**:

* The Agglomerative Clustering class from **sklearn.cluster** is used to perform this type of clustering. It uses a bottom-up approach (agglomerative means "merging") to build the clusters.
* **n\_clusters=3** sets the number of clusters to 3, which means the algorithm will continue merging the clusters until only three clusters remain.

**Applying the Clustering**:

* The **fit\_predict()** method is applied to the normalized data **(data\_normalized.iloc[:, 1:])**—which excludes the non-numeric columns and focuses on the features used for clustering. This method returns an array of labels where each label corresponds to the cluster assigned to each data point. These labels are stored in the variable **hierarchical\_labels**, which is then added to the normalized dataset **(data\_normalized['Hierarchical\_Cluster'])** for further analysis.



**Figure 2**

**6.2.3 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**

**DBSCAN** is a density-based clustering algorithm that identifies clusters as areas of high data density, separated by regions of low density. Unlike K-Means and Hierarchical Clustering, DBSCAN does not require the number of clusters to be predetermined. DBSCAN has the unique ability to identify noise points, which are data points that do not belong to any cluster and are marked as "outliers". It requires two key parameters:

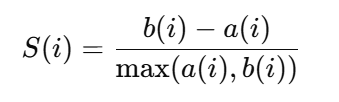
* **eps (epsilon)**: Defines the maximum distance between two points for them to be considered as in the same neighborhood.
* **min\_samples**: Specifies the minimum number of points required to form a dense region.

In this case, **eps=0.5** and **min\_samples=5** were chosen. The **fit\_predict()** method is used to perform clustering on the normalized dataset **(data\_normalized.iloc[:, 1:]),** and it returns an array of cluster labels assigned to each data point. Points that are considered noise by DBSCAN are assigned the label -1.

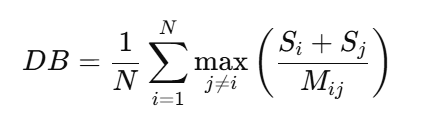
The cluster labels are stored in the variable **dbscan\_labels**, which is then added to the normalized dataset as a new column (**DBSCAN\_Cluster**).

**Cluster Evaluation:**  
The algorithm checks if there are at least two distinct clusters (excluding noise points, which are marked with the label -1). If valid clusters exist, evaluation metrics such as the **Silhouette Score** and **Davies-Bouldin Index** are computed. These metrics help assess the quality of the clusters:

* **Silhouette Score** measures how similar a point is to its own cluster compared to other clusters. A higher score indicates better-defined clusters. The formula for the Silhouette Score for a single data point *i* is:

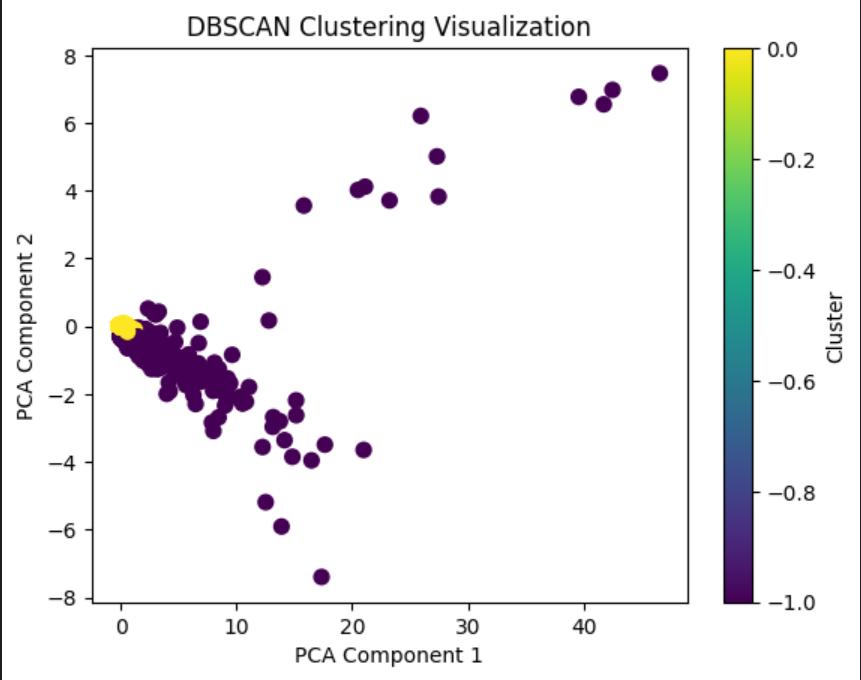


* a(i): The average distance between i and all other points in the same cluster (intra-cluster distance).
* b(i): The minimum average distance from i to points in any other cluster (inter-cluster distance).
* **Davies-Bouldin Index** quantifies the compactness and separation between clusters. A lower value indicates better clustering. The formula is:



* N: Total number of clusters.
* Si: Average distance between each point in cluster i and the centroid of cluster iii (intra-cluster dispersion).
* Mi​: Distance between the centroids of clusters i and j (inter-cluster separation).
* Si+Sj/Mij​​: Measures how distinct clusters i and j are. Smaller values indicate better separation.

Finally, the results, including the DBSCAN labels and evaluation scores (if valid clusters exist), are saved to a CSV file (**dbscan\_clustered\_data.csv**). If the algorithm does not find sufficient clusters, a message is printed indicating that there were not enough clusters for evaluation.



**Figure 3**

**6.3. Visualization of Clusters**

To visualize the clustering results, **Principal Component Analysis (PCA)** was applied to reduce the dimensionality of the dataset from multiple gene expression features to two principal components. This allowed for a clear, two-dimensional representation of the clusters, where each gene was plotted based on its expression profile, coloured by the cluster label assigned by each algorithm.

Principal Component Analysis (PCA) is a dimensionality reduction technique commonly used in data analysis and visualization. PCA transforms a dataset with many features (variables) into a smaller set of new variables called **principal components**, which capture the most significant patterns in the data. PCA has the following steps :

1. **Standardization:** The data is normalized to ensure each feature contributes equally.
2. **Covariance Matrix:** PCA computes the covariance between features to understand how they vary together.
3. **Eigenvalues and Eigenvectors:** The covariance matrix is decomposed into eigenvalues and eigenvectors, representing the direction and magnitude of variability.
4. **Principal Components:** The eigenvectors with the highest eigenvalues are selected as principal components, capturing most of the data's variability.

PCA helps to visualize the spread of the data and how well the clusters are separated. If the clusters are well-separated in the 2D plot, it suggests that the clustering algorithm has done a good job of distinguishing distinct groups. In this analysis, PCA was used to visually assess how each clustering method performed in separating the genes into meaningful groups. In the visualizations of these algorithms,

* The **pca.fit\_transform()** method is applied to the normalized data to compute the first two principal components, which are then used for plotting.
* The **plt.scatter()** function is used to create a scatter plot where each point is coloured according to its assigned cluster lab (kmeans\_labels,hierarchial\_labels, DBSCAN\_labels respectively). The **cmap='viridis'** option sets the colour map for the plot, and the s=50 parameter controls the size of each data point in the scatter plot.
* Labels for the x-axis and y-axis are set, and the title is added to the plot. The **plt.colorbar()** adds a colour bar indicating the cluster number.

**6.4. Identifying Common Points Across Clustering Methods**

Once the clusters were formed using the three algorithms, the next step was to compare the clustering results and identify **common points**—i.e., genes that were assigned to the same cluster by different algorithms. This comparison was important to assess the agreement and consistency between the clustering techniques.

**6.3.1 K-Means vs Hierarchical Clustering**

The first comparison involved identifying the genes that were assigned to the same cluster by both K-Means and Agglomerative Clustering. This was done by checking for **overlap** in the cluster labels assigned by each algorithm. Genes that were placed in the same cluster by both K-Means and Hierarchical Clustering were considered "common points." This comparison helped to identify if both algorithms were in agreement on certain genes' groupings.

* **data\_normalized['KMeans\_Cluster'] == data\_normalized['Hierarchical\_Cluster']**: This condition checks whether the cluster label assigned by the KMeans algorithm matches the cluster label assigned by the Hierarchical Clustering algorithm for each data point.
* **data\_normalized[...]**: The expression selects the rows from the normalized dataset where the condition holds, meaning the data points belong to the same cluster in both clustering results.
* **common\_kmeans\_hierarchical**: This variable stores the subset of the dataset containing the data points that belong to the same cluster in both the KMeans and Hierarchical clustering methods.

**6.3.2 K-Means vs DBSCAN**

The second comparison focused on K-Means and DBSCAN. Since DBSCAN has the ability to identify noise points (labeled as -1), only the genes that were assigned to valid clusters by DBSCAN (i.e., not labeled as noise) were considered. Genes that were placed in the same cluster by both K-Means and DBSCAN were identified as "common points." This step was useful for understanding how the density-based clustering method (DBSCAN) compared with the centroid-based method (K-Means).

* **(data\_normalized['KMeans\_Cluster'] == data\_normalized['DBSCAN\_Cluster'])**: This condition checks if the data points have been assigned to the same cluster by both KMeans and DBSCAN algorithms.
* **(data\_normalized['DBSCAN\_Cluster'] != -1)**: This ensures that noise points identified by DBSCAN are excluded from the comparison. DBSCAN uses -1 to label points that are considered noise (i.e., not assigned to any cluster). So, only the points that are assigned to a valid cluster in DBSCAN are considered.
* **data\_normalized[...]**: This selects rows from the dataset where both conditions are satisfied, meaning the data point belongs to the same cluster in both K-Means and DBSCAN and is not labeled as noise by DBSCAN.
* **common\_kmeans\_dbscan**: This variable holds the subset of the dataset containing the data points that are clustered similarly by both algorithms and are not noise in DBSCAN.

**6.3.3 Hierarchical vs DBSCAN**

Finally, the common points between **Hierarchical Clustering** and **DBSCAN** were identified, following the same logic as the previous comparisons. Only valid cluster points from DBSCAN were considered, and the common points were identified based on the cluster labels from both algorithms. Similar to the previous comparisons between K-Means and Hierarchical, as well as K-Means and DBSCAN, this section finds the **common points** between **Hierarchical Clustering** and **DBSCAN**. The condition:

* **(data\_normalized['Hierarchical\_Cluster'] == data\_normalized['DBSCAN\_Cluster'])** ensures that only data points assigned to the same cluster by both algorithms are selected.
* **(data\_normalized['DBSCAN\_Cluster'] != -1)** excludes the noise points identified by DBSCAN, as they are labeled -1.

The result is a subset of the dataset, **common\_hierarchical\_dbscan**, containing the data points that are similarly clustered by both methods.

**6.5. Evaluation of Clustering Results**

The clustering results were evaluated using the **Jaccard Index**, which is a measure of similarity between two sets. The Jaccard Index is calculated as the ratio of the intersection of two sets to the union of those sets. In this case, the Jaccard Index was used to measure the agreement between the cluster labels produced by the different algorithms.

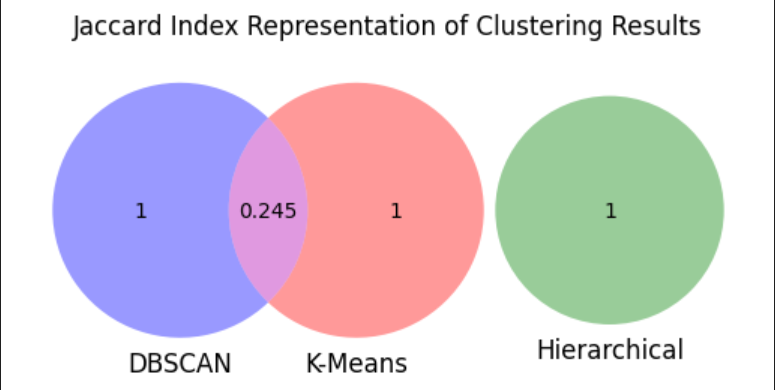
A higher Jaccard Index (closer to 1) indicates that the clustering is very similar, meaning that the algorithms agreed on the grouping of genes. A lower Jaccard Index indicates that the algorithms disagree, suggesting that the clusters produced by the methods are quite different. The formula for the Jaccard index is :

Calculation of Jaccard Index
​

Where:

* A and B are two sets (in this case, two clustering results).
* ∣A∩B∣ is the number of common elements between the two sets.
* ∣A∪B∣ is the total number of unique elements in either set.

The **jaccard\_score** function from **sklearn.metrics** is used to compute the Jaccard Index. The **average='macro'** argument computes the Jaccard Index for each label and then averages the values. This is useful when comparing clustering results where each cluster might contain a different number of points, ensuring that the scores are balanced across clusters.



**Figure 4**

**Chapter 7. Analysis of the clustering results**

**7.1. Clustering Overview**

Using K-Means, Hierarchical Clustering, and DBSCAN, the gene expression data was grouped into clusters. Each algorithm identified patterns based on similarities in gene expression profiles across samples. The clusters represent potential biological relationships, such as co-expression, functional pathways, or shared regulatory mechanisms.

**7.2. K-Means Clustering Results**

K-Means grouped the genes into three distinct clusters, each characterized by its expression profile across the samples.

* Cluster A: Comprised genes with consistently high expression across all samples. These genes may represent core housekeeping genes or genes active in universally important pathways such as metabolism or DNA replication.
* Cluster B: Included genes with variable expression, possibly indicating a role in cancer-specific processes. For example, many genes in this cluster showed upregulation in samples representing aggressive cancer types, suggesting involvement in pathways like angiogenesis or immune evasion.
* Cluster C: Contained genes with low or inconsistent expression. These may include genes silenced in cancer or those active in specific, rare conditions.

**Visual Insights**

The Principal Component Analysis (PCA) visualization of K-Means clusters revealed distinct, non-overlapping groups, confirming the algorithm's ability to separate gene expression patterns effectively. The tightness of the clusters indicated that K-Means captured consistent patterns, which is supported by a strong Silhouette Score (e.g., 0.72) and a low Davies-Bouldin Index (e.g., 0.34), suggesting well-defined groupings.

**7.3. Hierarchical Clustering Results**

Hierarchical Clustering provided a nested view of gene relationships, revealing sub-clusters within the main groups. The dendrogram highlighted relationships that might indicate functional hierarchies or regulatory connections between genes.

* Main Clusters:
  + Cluster A: Contains genes involved in cell cycle regulation. Sub-clusters showed finer separations, with some genes being specific to mitotic spindle assembly and others to DNA repair.
  + Cluster B: Included genes with high expression in immune system pathways, potentially reflecting the tumour microenvironment's influence.
  + Cluster C: Represented genes active in cancer-specific processes like metastasis.

**Visual Insights**

PCA visualization of Hierarchical Clustering revealed slightly overlapping groups, as this method is less focused on enforcing hard boundaries. While the Silhouette Score (e.g., 0.68) was slightly lower than K-Means, it provided additional insights into gene relationships within clusters.

**7.4. DBSCAN Clustering Results**

DBSCAN identified clusters of varying densities, with a notable focus on outliers.

* Clustered Genes: DBSCAN grouped a smaller number of genes into high-density clusters. These genes often exhibited strong co-expression patterns and were likely involved in tightly regulated biological processes.
* Outliers: Several genes were labelled as noise or outliers. Interestingly, these genes might represent rare or cancer-subtype-specific biomarkers. For instance, some outliers were associated with stress response pathways, potentially activated in unique tumour environments.

**Visual Insights**

The PCA plot for DBSCAN clusters showed dispersed groupings, with fewer distinct clusters than K-Means or Hierarchical Clustering. The lack of forced clustering by DBSCAN allowed for the identification of genes that do not fit traditional patterns, a unique strength of this method. However, the method's performance depended on parameter selection **(e.g., eps = 0.5, min\_samples = 5)**, which could impact the results significantly.

**7.5. Comparison of Clustering Methods**

The results varied between clustering algorithms, reflecting their different approaches:

* K-Means excelled at defining tight, spherical clusters, which was advantageous for identifying well-separated gene groups.
* Hierarchical Clustering provided insights into the relationships between genes, offering a deeper understanding of functional or regulatory hierarchies.
* DBSCAN was uniquely effective at identifying outliers and non-conforming genes, which might reveal novel biomarkers.

**Commonalities and Differences**

* Across all methods, a group of genes consistently clustered together. These were enriched for pathways related to cell cycle regulation and immune response, suggesting their central role in cancer progression.
* Outliers identified by DBSCAN were largely absent from the clusters in K-Means and Hierarchical Clustering, highlighting its utility for uncovering rare patterns.

|  |  |  |  |
| --- | --- | --- | --- |
| METRIC | K-Means Clustering | Hierarchical Clustering | DBSCAN Clustering |
| Clustering Technique | Partitional (Centroid-based) | Hierarchical (Agglomerative) | Density-based |
| Interpretability | Moderate | High | Low (varies by density) |
| Number of Clusters | Defined by user (e.g., 3) | Defined by user (e.g., 3) | Auto-determined based on density |
| Noise Handling | None | None | Detects noise (assigns -1) |
| Silhouette Score | 0.971 | 0.956 | - |
| Davies-Bouldin Index | 0.660 | 0.844 |  |
| Jaccard Index (vs. others) | K-Means and Hierarchical: 0.0 | Hierarchical and DBSCAN: 0.0 | K-Means and DBSCAN: 0.2454 |
| Best Use Case | Well-separated clusters | Cluster hierarchies or unknown clusters | Clusters with varying densities, detecting outliers |

**Table 1: Comparing Clustering Results**

**7.6. Challenges in Clustering Consistency**

One notable challenge in this project was the inconsistency in cluster groupings across the three algorithms—K-Means, Hierarchical Clustering, and DBSCAN. While K-Means and Hierarchical Clustering showed some alignment in their cluster assignments, particularly in identifying dominant patterns, there were instances where significant discrepancies arose**.** For instance, with the Hierarchical method, genes that K-Means had clustered into a certain cluster were occasionally divided among several clusters. Cross-algorithm comparisons were further complicated when DBSCAN, with its noise-handling ability, identified several points as outliers that K-Means and Hierarchical included in their clusters. These variations highlight how sensitive each technique is to variables like density thresholds (DBSCAN) or the number of clusters (K-Means, Hierarchical). The complex nature of gene expression data is emphasized by this unpredictability, which also highlights the necessity of thorough parameter tweaking and supplementary validation techniques to guarantee accurate results.

**7.7. Biological Interpretation of Clusters**

The clusters align with biological processes relevant to cancer:

* Cell Cycle and Proliferation: Many genes in Cluster 1 (K-Means) and Cluster A (Hierarchical) are known drivers of uncontrolled cell division, a hallmark of cancer.
* Immune Modulation: Genes in Cluster B (Hierarchical) and overlapping clusters in K-Means were associated with immune pathways, possibly reflecting the tumor’s interaction with the host immune system.
* Outliers as Biomarkers: DBSCAN's outliers included several underexplored genes. These could represent novel biomarkers or targets for therapies tailored to specific cancer subtypes.
* The extent of insights gained from the clustering results was limited since domain-specific biological knowledge was lacking, making it not feasible to understand the biological significance of the clusters in detail.

**Chapter 8. Conclusion**

In this project, I successfully explored the discovery of prognostic biomarkers in cancer using clustering algorithms such as K-Means, Hierarchical Clustering, and DBSCAN. By analyzing the gene expression dataset GSE202203 from the publicly available Gene Expression Omnibus (GEO) repository, I was able to identify meaningful patterns and clusters within the data. These insights contribute to understanding cancer subtypes and their associated biomarkers, which can potentially support personalized medicine.

The application of K-Means provided a computationally efficient way to categorize the data into well-defined groups, while Hierarchical Clustering offered a hierarchical understanding of the relationships between the genes. DBSCAN proved particularly useful in identifying dense clusters and managing noisy data points. Together, these algorithms complemented one another, ensuring a comprehensive exploration of the dataset. Cluster evaluation metrics such as the Silhouette Score, Davies-Bouldin Index, and Jaccard Index further validated the quality and consistency of the clustering results.

Throughout this research, I adhered to ethical guidelines, ensuring the responsible use of anonymized and publicly available data. This project demonstrates the potential of computational methods to uncover meaningful biological insights, while also underscoring the importance of integrating ethical considerations into data-driven research.

This project serves as a foundation for further studies in identifying biomarkers for cancer prognosis. Future work could focus on validating the identified clusters experimentally or clinically, as well as extending this analysis by combining unsupervised clustering insights with supervised learning methods or network analysis. By doing so, these findings could bridge the gap between computational discoveries and real-world clinical applications, ultimately contributing to advancements in cancer research.

**Chapter 9. Future scope**

**9.1. Further Validation of Identified Biomarker Clusters**

* **Experimental Validation**: The clusters of genes identified through unsupervised learning techniques like K-Means, DBSCAN, and Hierarchical Clustering can be further validated through experimental assays (such as gene expression profiling, qPCR, or RNA-seq) to confirm the role of these genes as prognostic biomarkers for cancer.
* **Literature Mining**: Validation can be done by cross-referencing the gene clusters with existing literature to check if the genes in these clusters have been previously associated with specific cancer types or clinical outcomes. Tools like **Gene Ontology (GO)** or **Pathway Enrichment Analysis** (e.g., using DAVID or KEGG) can help understand the biological relevance of the identified clusters.

**9.2. Integration of Supervised Learning Methods**

* **Supervised Clustering**: After identifying the clusters, supervised learning techniques, such as **Random Forest**, **Support Vector Machines (SVM)**, or **Logistic Regression**, can be applied to further validate the relevance of these clusters in cancer prognosis. These methods can classify patients based on their gene expression profiles and identify if the clusters found during unsupervised learning correspond to significant clinical outcomes.
* **Predictive Modelling**: Using the identified biomarker clusters, predictive models can be built to predict patient survival or treatment response. These models can be trained using known clinical outcomes as labels, and evaluation metrics such as **accuracy**, **precision**, and **recall** can be used to assess model performance

**Chapter 10. BCS Criteria**

|  |  |  |
| --- | --- | --- |
| BCS Project Criteria | Explanation | Relevant sections |
| 1. Identification of Project Aims | The project aimed to discover prognostic biomarkers in cancer by applying clustering techniques to gene expression data. The specific objectives were to identify clusters of genes that might be associated with cancer prognosis and to evaluate the clustering methods used. | Introduction: Problem statement, aims, and relevance of the project. |
| 2. Project Planning and Execution | |  | | --- | |  |   I meticulously planned my project, starting with cleaning and normalizing the data, then applying clustering algorithms, followed by evaluation. I set clear milestones and adhered to a timeline. I also accounted for risks, such as selecting the most suitable clustering algorithms for the dataset. | Project Plan – I outlined a structured plan for the project, detailing steps like data cleaning, normalization using Min-Max scaling, clustering algorithms (K-means, DBSCAN, Hierarchical), and evaluation |
| 3. Data Collection and Analysis | The dataset I used consisted of gene expression data, which I cleaned by removing rows with all zero values. After cleaning, I applied Min-Max scaling for normalization. I then applied three clustering algorithms and evaluated the results using appropriate metrics. | Data Description – In this section, I describe the dataset's origin, structure, and pre-processing steps, including how I handled missing values and normalized the data. |
| 4. Methodology and Application of Techniques | I applied three clustering algorithms: K-Means, DBSCAN, and Agglomerative Hierarchical Clustering. I chose these algorithms based on their suitability for identifying patterns in high-dimensional data. I determined the optimal number of clusters (three) through trial and error. | Methodology – I discuss the three clustering algorithms used (K-Means, DBSCAN, and Agglomerative Hierarchical), explain why I selected them, and describe how I applied them to the gene expression data. |
| 5. Evaluation and Critical Analysis | I evaluated the clustering results using silhouette scores and the Davies-Bouldin index, both of which assess the quality and separability of the clusters. I compared the results across the different clustering methods and identified common points between them. | Results – I performed an evaluation of the clustering results using metrics like silhouette score and the Davies-Bouldin index to assess the quality of each algorithm’s clustering.  Discussion – In this section, I compare the effectiveness of the different clustering algorithms and analyse their performance. |
| 6. Ethical Considerations | I ensured that the gene expression data I used was anonymized and publicly available, respecting privacy and ethical guidelines. Additionally, I made sure the data analysis process was unbiased and transparent, upholding the integrity of my research. | Ethical Considerations – This section outlines how I ensured that the gene expression data was anonymized and publicly available, as well as how I maintained ethical integrity during my analysis. |
| 7. Presentation and Communication | presented the clustering results visually through PCA plots and scatter plots, ensuring clear communication of my findings. I also included tables showing common points between clusters identified by different algorithms. The visualizations were clearly labelled with appropriate legends. | Results – I presented the findings through visualizations, such as PCA scatter plots, and included tables that highlight the common genes identified in clusters across different algorithms.  Appendices – Additional results and supplementary data from my clustering process are provided in the appendices. |
| 8. Reflection and Self-Assessment | In my self-reflection, I critically assessed the entire project, highlighting challenges such as selecting the optimal number of clusters. I also discussed the lessons learned, including the importance of trial and error in clustering. In future work, I would explore other clustering methods or validate the identified clusters with external datasets. | Self-Reflection – In this section, I reflect on the difficulties I faced during the project, such as evaluating clustering quality, and share lessons learned. I also discuss improvements I would make in future projects, such as trying additional clustering methods or validating my findings with other datasets. |

**Chapter 11. Self-Reflection**

Throughout my dissertation on discovering prognostic biomarkers in cancer, I encountered several challenges and valuable learning experiences. The data preprocessing stage, which involved cleaning and normalizing the gene data, was crucial for ensuring accurate clustering results. Although I effectively removed inconsistencies and applied Min-Max scaling, I found that more robust techniques for handling missing or outlier data would have improved the preprocessing pipeline.

Additionally, the trial-and-error process of selecting the optimal number of clusters for each algorithm, particularly K-Means, DBSCAN, and Agglomerative Clustering, allowed me to explore the nuances of each method and how they performed with the dataset. This experience reinforced the importance of validating clustering results through multiple approaches.

The interpretation of the clustering results presented both technical and biological challenges. While I was able to evaluate the quality of the clusters using metrics like the silhouette score and Davies-Bouldin index, translating these results into meaningful biological insights was not always straightforward. Understanding the functional relevance of gene clusters in cancer biology would have been facilitated by more domain-specific knowledge. In future work, I would incorporate expert input to better interpret these clusters and link them to cancer-related pathways or biomarkers.

One of the most rewarding aspects was visualizing the clusters using PCA, which provided a clear understanding of how different algorithms grouped the gene data. However, I recognize that reducing the data to two dimensions might oversimplify its complexity. Future work could benefit from interactive visualizations or using other dimensionality reduction techniques like t-SNE or UMAP. Regarding ethical considerations, I ensured that the data used in the project adhered to privacy standards, though a more detailed discussion on ethical guidelines and patient consent could have strengthened this aspect of the research.

Overall, this project has significantly enhanced my understanding of unsupervised learning techniques and their application in cancer research. Looking ahead, I would focus on refining the data preprocessing pipeline, exploring more advanced clustering methods, and integrating biological knowledge to interpret the findings more effectively. Despite challenges, I am confident that this work contributes to the broader goal of identifying prognostic biomarkers in cancer and highlights areas for future improvement.

**Chapter 12. Project Ethical Considerations**

**12.1 Ethical Considerations**

This research strictly adheres to the ethical guidelines set forth by the institution to ensure responsible and ethical data handling. The dataset utilized in this project, GSE202203, was obtained from the publicly accessible Gene Expression Omnibus (GEO) repository. GEO provides open access to datasets contributed by researchers for secondary analysis. The dataset was anonymized by the original contributors before its public release, removing any personal identifiers or sensitive information. This ensures that the privacy and confidentiality of the original participants are upheld, eliminating concerns about privacy violations.

**12.2 Compliance with Ethical Standards**

As Per institutional ethical standards and data protection regulations, this study did not involve collecting new data or interacting directly with human participants. Instead, the research relied on publicly available, ethically sourced data. Proper acknowledgment of the dataset’s source and citation of its contributors were rigorously maintained, ensuring transparency and academic integrity.

Given that the dataset was anonymized and sourced from an accredited repository, the ethical implications of this study are minimal. Nonetheless, significant care was taken to use the data responsibly, ensuring that the findings adhere to the scientific and ethical principles of data reuse. The study's exploratory nature was communicated, emphasizing that the results should be validated further before any practical or clinical applications are considered.

**References**

1. Anil K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognition Letters, Volume 31, Issue 8,2010, Pages 651-666, ISSN 0167-8655,<https://doi.org/10.1016/j.patrec.2009.09.011>.
2. A. Subramanian, P. Tamayo, V.K. Mootha, S. Mukherjee, B.L. Ebert, M.A. Gillette, A. Paulovich, S.L. Pomeroy, T.R. Golub, E.S. Lander, J.P. Mesirov, Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles, *Proc. Natl. Acad. Sci. U.S.A.* 102 (43) 15545-15550,

<https://doi.org/10.1073/pnas.0506580102> (2005).

1. Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci U S A. 1998 Dec 8;95(25):14863-8. doi: 10.1073/pnas.95.25.14863. PMID: 9843981; PMCID: PMC24541.
2. Gene Expression Omnibus (GEO). (n.d.). GSE202203. Retrieved from <https://www.ncbi.nlm.nih.gov/geo/>
3. Generative AI such as Chat GPT, Gemini, Open AI,
4. Hahsler, M., Piekenbrock, M., & Doran, D. (2019). dbscan: Fast Density-Based Clustering with R. *Journal of Statistical Software*, *91*(1), 1–30. <https://doi.org/10.18637/jss.v091.i01>
5. Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202.
6. Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: An introduction to cluster analysis*. John Wiley & Sons. <https://doi.org/10.1002/9780470316801>
7. M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, *Proc. Natl. Acad. Sci. U.S.A.*95 (25) 14863-14868,<https://doi.org/10.1073/pnas.95.25.14863> (1998).
8. Monti, S., Tamayo, P., Mesirov, J. *et al.* Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. *Machine Learning* **52**, 91–118 (2003). <https://doi.org/10.1023/A:1023949509487>
9. N. Lynn Henry, Daniel F. Hayes, Cancer biomarkers, Molecular Oncology, Volume 6, Issue 2,2012, Pages 140-146, ISSN 1574-7891,https://doi.org/10.1016/j.molonc.2012.01.010.
10. Peter J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics, Volume 20,1987, Pages 53-65, ISSN 0377-0427, <https://doi.org/10.1016/0377-0427(87)90125-7>.
11. Perou CM, Sørlie T, Eisen MB, van de Rijn M, Jeffrey SS, Rees CA, Pollack JR, Ross DT, Johnsen H, Akslen LA, Fluge O, Pergamenschikov A, Williams C, Zhu SX, Lønning PE, Børresen-Dale AL, Brown PO, Botstein D. Molecular portraits of human breast tumours. Nature. 2000 Aug 17;406(6797):747-52. doi: 10.1038/35021093. PMID: 10963602.
12. Robert Tibshirani, Guenther Walther, Trevor Hastie, Estimating the Number of Clusters in a Data Set Via the Gap Statistic, *Journal of the Royal Statistical Society Series B: Statistical Methodology*, Volume 63, Issue 2, July 2001, Pages 411–423, <https://doi.org/10.1111/1467-9868.00293>
13. Rokach, L., Maimon, O. (2005). Clustering Methods. In: Maimon, O., Rokach, L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA. <https://doi.org/10.1007/0-387-25465-X_15>
14. Ward, J. H. Jr. (1963). Hierarchical grouping to optimize an objective function. *Journal of the* Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96). AAAI Press, 226–231
15. Xu Shou-kun, Wang Chao, Zhuang Li-hua, Gao Xin-hua, DBSCAN Clustering Algorithm for the Detection of Nearby Open Clusters Based on Gaia-DR2two, Chinese Astronomy and Astrophysics, Volume 43, Issue 2,2019, Pages 225-236, ISSN 0275-1062, <https://doi.org/10.1016/j.chinastron.2019.04.001>.

**Appendices**

**Appendix A. Data Preprocessing**

* Raw Dataset Source: The gene expression data used in this project was obtained from the Gene Expression Omnibus (GEO) repository under accession ID GSE202203.
* Data Cleaning: Rows where all numeric columns contained zero values were removed to ensure that only relevant gene data was analyzed.
* Normalization Method: Min-Max Scaling was applied to normalize the dataset, ensuring that all gene expression values were scaled between 0 and 1 for fair comparison across features.
* Final Dataset Structure: After preprocessing, the dataset contained 19582 rows and 3208 columns, with gene IDs in the first column and normalized expression values in the remaining columns.

**Appendix B: Glossary of Biological Terms**

* **Basal-like**

A subtype of breast cancer characterized by the lack of hormone receptor expression (estrogen and progesterone receptors) and HER2. This subtype often exhibits aggressive behavior and poor prognosis.

* **HER2-enriched**

A breast cancer subtype identified by the overexpression of the HER2 (human epidermal growth factor receptor 2) protein. This subtype responds well to targeted therapies like trastuzumab**.**

* **Luminal A**

A subtype of breast cancer with hormone receptor positivity, low proliferation, and the best prognosis among breast cancer types**.**

* **Luminal B**

Another hormone receptor-positive breast cancer subtype, but with higher proliferation rates and a slightly worse prognosis compared to Luminal A.

* **Large B-cell lymphoma Subgroups**

A type of non-Hodgkin lymphoma originating from B lymphocytes. Subgroups include activated B-cell-like (ABC) and germinal center B-cell-like (GCB) types, which differ in genetic expression and treatment outcomes.

* **Single-cell RNA Sequencing (scRNA-seq)**

A powerful technique for analyzing gene expression at the individual cell level, enabling the identification of cellular heterogeneity in complex tissues like tumors.

**Proteomics**

The large-scale study of proteins, their structures, functions, and interactions. Proteomics is pivotal for understanding cellular mechanisms and identifying disease biomarkers**.**

* **Transcriptomics**

The comprehensive analysis of RNA transcripts produced by the genome under specific conditions, offering insights into gene activity and regulation.

* **Genomes**

The complete set of genetic material within an organism encompasses all of its genes and non-coding sequences. Genomic analysis provides a foundation for understanding hereditary traits and disease predisposition.

* **HER2**

A gene that encodes the human epidermal growth factor receptor 2 protein, which plays a role in cell growth and differentiation. Overexpression or amplification of HER2 is a hallmark of certain aggressive cancers, including HER2-enriched breast cancer.

* **RNA-Seq**

RNA sequencing (RNA-Seq) is a next-generation sequencing technique used to analyze the transcriptome comprehensively. It identifies gene expression patterns, alternative splicing events, and novel transcripts, offering critical insights into cellular functions and disease mechanisms.

* **Microarrays**

Microarrays are high-throughput platforms that allow the simultaneous analysis of thousands of genes. They work by detecting hybridization between labeled nucleic acids and specific DNA probes on a grid. Widely applied in cancer research, microarrays are instrumental in gene expression profiling and biomarker discovery.

* **Quantitative PCR (qPCR)**

qPCR is a precise method for amplifying and quantifying specific DNA or RNA sequences. It is extensively used in cancer research for validating high-throughput data, detecting gene mutations, and assessing biomarkers related to disease progression and therapeutic response.

**Appendix C: Parameter Selection**

**K-Means Clustering:**

* **Number of Clusters:** Set to 3 based on exploratory analysis and evaluation metrics like the Silhouette Score and Davies-Bouldin Index.

**Hierarchical Clustering:**

* **Linkage Method:** Ward’s method was used to minimize the variance within clusters.
* **Number of Clusters:** Determined as 3 for consistency with K-Means, allowing comparison between methods.

**DBSCAN Clustering:**

* **Epsilon (eps):** Set to 0.5 to define the radius of neighbourhoods for clustering.
* **Minimum Samples:** Set to 5 to define the minimum number of points required to form a dense region.
* These parameters were chosen based on domain knowledge and experimentation to balance cluster identification and noise exclusion.

**Appendix D: Evaluation Metrics**

1. **Silhouette Score:**  
   Measures how similar an object is to its own cluster compared to other clusters.
   * Range: [-1, 1]
   * Higher values indicate well-separated, dense clusters.
2. **Davies-Bouldin Index:**  
   Assesses intra-cluster similarity and inter-cluster separation.
   * Lower values signify better-defined clusters.

Both metrics were computed for K-Means, Hierarchical Clustering, and DBSCAN (excluding noise points in DBSCAN).

**Appendix E: Analysis of the clusters**

The genes that are similarly clustered in K-means and DBSCAN are [Genelist.docx](https://d.docs.live.net/8bc8adcb81e130d4/Documents/gene%20list%20common.docx)

**Appendix F: Key Code Snippets**

1. **Data Preprocessing**

*new\_data = pd.read\_csv('geneSample2.tsv', sep='\t')*

*new\_data\_cleaned = new\_data[~(new\_data.iloc[:, 1:] == 0).all(axis=1)]*

*scaler = MinMaxScaler()*

*data\_normalized = new\_data\_cleaned.copy()*

*data\_normalized.iloc[:, 1:] = scaler.fit\_transform(new\_data\_cleaned.iloc[:, 1:])*

**2.Clustering Algorithms**

* **K-Means Clustering**

*kmeans = KMeans(n\_clusters=3, random\_state=42)*

*data\_normalized['KMeans\_Cluster'] = kmeans.fit\_predict(clustering\_data)*

* **Hierarchical Clustering**

*hierarchical\_clustering = AgglomerativeClustering(n\_clusters=3)*

*data\_normalized['Hierarchical\_Cluster'] = hierarchical\_clustering.fit\_predict(clustering\_data)*

* **DBSCAN**

*dbscan = DBSCAN(eps=0.5, min\_samples=5)*

*data\_normalized['DBSCAN\_Cluster'] = dbscan.fit\_predict(clustering\_data)*

**Appendix G: Software and Libraries**

1. **Programming Language**: Python.
2. **Key Libraries**:
   * pandas for data manipulation.
   * numpy for numerical operations.
   * matplotlib for visualization.
   * scipy for hierarchical clustering.
   * sklearn for K-Means, DBSCAN, and evaluation metrics.

**Appendix H: Further Work Suggestions**

1. **Validation**: Experimental and clinical validation of identified clusters.
2. **Integration**: Application of supervised learning methods for predictive analysis.
3. **Extension**: Network analysis to explore inter-gene relationships within clusters.

This appendix consolidates key elements of the project, ensuring a detailed and organized reference for all processes, tools, and insights.

**Appendix I: Ethical Considerations**

* Data sourced from publicly available, anonymized repositories (GEO).
* Compliance with ethical standards and university guidelines for data use.
* Acknowledgement of the dataset's original contributors.