Contents

1 Introduction

In the rapidly advancing field of genomics, understanding how genes respond to various doses of treatments is crucial for the progression of personalized medicine and targeted therapies. High-dimensional dose-response data, which capture changes in gene expression levels across a spectrum of treatment dosages, hold immense potential for significant biomedical discoveries. However, the complexity and vastness of such data present substantial challenges for effective analysis and interpretation. One promising method to address these challenges is through clustering algorithms, which can group genes exhibiting similar expression patterns, thereby uncovering underlying biological processes and treatment effects (Brittain et al., 2017, p. 40-41).

Clustering, a core technique in unsupervised machine learning, aims to partition data into groups, or clusters, where data points within the same group exhibit high similarity, while those in different groups are distinct. In the context of high-dimensional gene expression data, clustering can identify genes that respond similarly to treatments, suggesting shared regulatory mechanisms or functional relationships. This is particularly valuable in medical research for several key reasons:

1. **Identification of Biomarkers**: Clustering can reveal genes that consistently respond to specific drug doses, pinpointing potential biomarkers for drug efficacy or toxicity. These biomarkers are essential for developing diagnostic tools and personalized treatment plans.
2. **Inference of Drug Mechanisms**: By grouping genes with similar dose-response profiles, researchers can infer potential pathways and mechanisms of action for drugs. This can enhance the understanding of how drugs influence biological systems and help identify possible side effects.
3. **Reduction of Data Complexity**: High-dimensional data often contain redundant information. Clustering reduces this complexity by summarizing the data into a smaller number of representative groups, making the analysis more manageable and interpretable.
4. **Guidance for Experimental Design**: Insights from clustering analysis can inform the design of future experiments, such as selecting specific doses for detailed investigation or identifying key genes for further study.

The dataset analyzed in this project consists of gene expression data from several genes, measured at eight different concentrations of valproic acid (VPA), ranging from 0 mM to 1000 mM. Multiple biological replicates were included for each of the other concentrations. The data, i.e. the gene expression value, presented on a logarithmic scale (log2), enables the exploration of dose-response relationships and the identification of differentially expressed genes.

This thesis is organized into several sections. The next section problem statement introduces the dataset used and clearly outlines the project's objectives. The statistical sethods section summarizes the algorithms—such as K-Means, Hierarchical Clustering, DBSCAN, OPTICS, and ORICC—as well as visualization techniques like Matplots, PCA plots, and others utilized throughout the study. The statistical analysis section presents the results, beginning with a brief analysis of the VPA dataset, followed by the application of clustering algorithms. Simulated datasets with varied characteristics, but similar behavior to the original dataset, are then analyzed using the same clustering methods. Finally, the consistency of these algorithms across different simulated datasets are evaluated. The thesis concludes with a concise summary of the entire research process.

This thesis is significant for several reasons. First, it addresses a critical need in bioinformatics for robust methods to analyze complex gene expression data. Second, by simulating realistic datasets, it offers a controlled environment to systematically compare different clustering algorithms, generating insights that can be generalized to real-world applications. Third, the findings will aid researchers in medical genomics in understanding drug responses, identifying potential biomarkers, and designing more effective experiments. Lastly, it offers a more dependable and stable clustering algorithm suitable for real-world high-dimensional gene expression datasets.

2 Problem statement

2.1 Dataset and data quality

The dataset analyzed in this study originates from the work of Krug et al. (2013), who focused on developmental neurotoxicity (DNT) and reproductive toxicity (RT). RT encompasses various influences, including chemicals, drugs, and other substances, which can disrupt biological processes linked to reproduction. DNT, a subset of RT, specifically

addresses disturbances in the development of embryos or fetuses. Both fields traditionally require extensive animal testing, which is both costly and ethically challenging when evaluating the effects of a single chemical on DNT or RT.

To address these challenges, Krug et al. (2013) developed in vitro approaches that reduce the need for animal testing and enhance the cost-effectiveness of chemical safety assessments. They utilized human embryonic stem cells (hESC) alongside substances like valproic acid (VPA) and methylmercury. VPA, commonly used in treating epilepsy, has been associated with an increased risk of congenital abnormalities. Krug et al. (2013) highlighted that analyzing transcriptome changes induced by toxic substances can provide significant mechanistic insights, motivating their research.

This dataset includes expression values for 54,675 genes across multiple treatment conditions. The gene expression levels were measured at eight different concentrations of VPA: 0 mM (control), 25 mM, 150 mM, 350 mM, 450 mM, 550 mM, 800 mM, and 1000 mM. For robustness, the dataset includes multiple biological replicates: six replicates for the control group (0 mM) and three replicates for each of the other concentration groups.

The gene expression data were obtained using the GeneChip® Human Genome U133 Plus 2.0 and pre-processed using the Robust Multi-Chip Average (RMA) algorithm as described by Irizarry et al. (2003). The expression measurements are presented on a logarithmic scale with base 2, which facilitates interpretation of the data in terms of fold changes. Specifically, an increase of 1 on the log2 scale represents a doubling of gene expression, while a decrease of 1 indicates a halving of expression. Lastly, the absence of missing values in the dataset indicates that the quality of this dataset is high.

2.2 Project objectives

The primary objective of this thesis is to evaluate the performance of various unsupervised clustering algorithms on high-dimensional data. In this thesis, five clustering algorithms—K-Means, Hierarchical Clustering, DBSCAN, OPTICS, and ORICC—are used. Initially, the key features of the original dataset, the VPA dataset, are examined by analyzing boxplots of gene expression values across different valproic acid concentrations. Subsequently, the above mentioned clustering algorithms are applied to the dataset, and their comparative performance is evaluated. Following this, the same algorithms are applied to several simulated datasets, which differ in characteristics such as standard

deviations or gene structure, yet maintain similarities with the original dataset. The performance of these algorithms are evaluated using the Adjusted Rand Index (ARI) as a metric. By examining the ARI values across different scenarios, a comparative analysis of their performance is conducted. The algorithm that consistently demonstrates better performance, regardless of dataset characteristics, is considered to be the most stable method among the clustering algorithm studied. Insights gained from this study will advance the field of computational biology and provide practical guidance for researchers analyzing real-world gene expression data.

Ultimately, this thesis aims to bridge the gap between computational methods and biological insights, contributing to the advancement of personalized medicine and targeted therapies. By identifying the most effective clustering algorithms for high-dimensional dose-response data, this research will enable more accurate and meaningful interpretations of gene expression studies, paving the way for new discoveries in medical research.

3 Statistical methods

This section discusses the various statistical and graphical tools used to analyze the data. The software R (R Development Core Team, 2021) with packages ggplot2 (Wickham, 2016), ggpubr (Kassambara and Kassambara, 2020), dplyr (Wickham et al., 2022), rstatix (Kassambara, 2021), cluster (Maechler et al., 2013), dbscan (Michael Hahsler, 2024), opticskxi (Charlon, 2019), NbClust (Charrad et al., 2015), ORIClust (Tianqing Liu and Zhang, 2009), tidyr (Wickham, 2021), stats (Team and contributors worldwide, 2020b) are utilized for analysis of the data and factoextra (Alboukadel Kassambara, 2020), graphics (Team and contributors worldwide, 2020a), reshape2 (Wickham, 2020), gridExtra (Baptiste Auguie, 2017) are used for visualization of the data.

3.1 Clustering Algorithms

In this subsection, the clustering algorithms utilized in this thesis—namely K-Means, hierarchical clustering, DBSCAN, OPTICS, and ORICC—are explained in detail. Clustering algorithms are essential tools in data analysis that group a set of objects into clusters based on their similarities. The primary goal of clustering is to organize a dataset into meaningful structures, where objects within the same cluster are more similar to each other than to those in other clusters.

3.1.1 K-Means clustering algorithm

K-Means is a widely used unsupervised machine learning algorithm employed for clustering data into groups. The objective is to partition the dataset into k clusters, where each data point is assigned to the cluster with the nearest mean (Bishop, 2006, p. 375-376).

Algorithm Steps:

1. **Initialization**: The value of k, representing the final number of clusters, is determined using the elbow plot of the entire dataset, which is discussed in detail in the later subsection. The algorithm then randomly selects k data points from the dataset as the initial centroids.

2. **Assignment of points to the nearest cluster centroid**: The distance between each data point and each centroid is calculated using the Euclidean distance, formula of which is provided below:

$$\|x_i - \mu_j\| = \sqrt{\sum_{l=1}^{d} (x_{il} - \mu_{jl})^2}$$

   where:

   - x_i is the i-th data point in the dataset, represented as a vector in a d-dimensional space.
   - μ_j is the j-th centroid, also a vector in the same d-dimensional space.
   - x_{il} is the l-th component (or feature) of the i-th data point.
   - μ_{jl} is the l-th component of the j-th centroid.
   - d is the total number of dimensions (or features) in the data.

   Each data point is then assigned to the cluster with the nearest centroid, i.e., the one with the minimum Euclidean distance.

3. **Update**: The new centroid for each cluster is computed calculating the mean of all data points assigned to that cluster.

4. **Repeat**: The assignment and update steps are repeated until the centroids stabilize or the maximum number of iterations is reached.

Drawbacks of the K-Means Algorithm: One significant drawback of the K-Means algorithm is that it is sensitive to the initial placement of centroids. Convergence to different final clusters can occur based on the initial random selection of centroids, leading to suboptimal clustering results as the algorithm may get stuck in local optima.

To mitigate this issue, one of the the techniques that can be implemented is :

Multiple Runs: The K-Means algorithm can be executed multiple times with different initial centroid seeds, and the result with the lowest within-cluster sum of squares (WCSS) can be selected.

The K-Means algorithm provides a simple and efficient method for clustering data. Data points are iteratively assigned to the nearest centroid, and the centroids are updated until stabilization occurs. Through this process, the dataset is partitioned into k clusters, with each cluster represented by a centroid, which is the mean of the data points within that cluster (Bishop, 2006, p. 377).

3.1.2 Hierarchical Clustering

Hierarchical clustering is widely used as an unsupervised machine learning algorithm to cluster data into a hierarchy of clusters. A tree-like structure (dendrogram) is built, representing nested groups of data points, which can be used to analyze the data at varying levels of granularity.

Types of Hierarchical Clustering: Two main types of hierarchical clustering are identified (Reddy, 2018, p. 85-87):

- **Agglomerative (Bottom-Up)**: This method starts with each data point as a single cluster, with pairs of clusters being merged iteratively until all data points are in a single cluster.
- **Divisive (Top-Down)**: This method begins with all data points in a single cluster, which is recursively split until each data point forms its own cluster.

Algorithm steps of Agglomerative Hierarchical Clustering: Let $X = \{x_1, x_2, \ldots, x_n\}$ represent the set of n data points, where each $x_i \in \mathbb{R}^d$.

1. **Initialization**: n clusters $\{C_1, C_2, \ldots, C_n\}$ are initialized, where $C_i = \{x_i\}$.

2. **Distance Calculation**: The distance between all pairs of clusters C_i and C_j is computed. For Euclidean distance, this is defined as:

$$d(C_i, C_j) = \|\mu_i - \mu_j\| = \sqrt{\sum_{l=1}^{d} (\mu_{il} - \mu_{jl})^2}$$

where:

- μ_i is the centroid of cluster C_i, represented as a vector in d-dimensional space.
- μ_{il} is the l-th component of the centroid μ_i.
- d is the number of dimensions or features of the data.

3. **Linkage Methods**: The distance between clusters is determined based on the selected linkage method:

- **Single Linkage**: $d(C_i, C_j) = \min\{d(x_a, x_b) : x_a \in C_i, x_b \in C_j\}$
- **Complete Linkage**: $d(C_i, C_j) = \max\{d(x_a, x_b) : x_a \in C_i, x_b \in C_j\}$
- **Average Linkage**: $d(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{x_a \in C_i} \sum_{x_b \in C_j} d(x_a, x_b)$

Each linkage method provides a different way of measuring the distance between clusters, which can significantly impact the shape and composition of the resulting clusters. Choosing the appropriate linkage method depends on the data structure and the clustering objectives—single linkage tends to form elongated clusters, complete linkage creates more compact clusters, while average linkage offers a balance between the two.

4. **Cluster Merging**: The two clusters with the smallest distance are merged.

5. **Update**: The distances between the new cluster and the remaining clusters are updated.

6. **Repetition**: The process is repeated until all data points are merged into a single cluster.

7. **Determining Final Clusters**: To determine the final clusters, the dendrogram is cut at a certain height, which represents the desired number of clusters at the end. This cut height can be estimated by analyzing the elbow plot, which helps identify the point where adding more clusters yields diminishing returns. Details of elbow method is discussed in the later subsection.

(Ian and Eibe, 2005, p. 271-277).

Drawbacks of Hierarchical Clustering: One major drawback of hierarchical clustering is its computational complexity. The calculation of distances between all pairs of data points, as well as the updating of distances during each merging step, is computationally intensive, making it unsuitable for large datasets.

Hierarchical clustering offers a powerful and intuitive method for clustering data, producing a detailed hierarchy of clusters. However, the computational demands limit its applicability for large-scale datasets. Various linkage methods provide flexibility in defining similarity between data points, offering different ways to interpret the structure of the data.

3.1.3 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is widely recognized as a density-based clustering algorithm used in unsupervised machine learning. Unlike K-Means, in which spherical clusters are assumed and the number of clusters k must be predefined, DBSCAN allows clusters of arbitrary shapes to be discovered, and outliers (noise) in the data are effectively identified.

Key Concepts of DBSCAN

- **Core Points**: A point is classified as a core point if at least a minimum number of points (MinPts) are located within a specified radius (ε).
- **Border Points**: A point is classified as a border point if it lies within ε distance of a core point but does not have enough neighbors to qualify as a core point.
- **Noise Points (Outliers)**: Points that do not meet the criteria for being either core or border points are considered noise (Ian and Eibe, 2011, p. 298-299).

Algorithm steps: Let $X = \{x_1, x_2, \ldots, x_n\}$ represent the set of n data points, where each $x_i \in \mathbb{R}^d$.

1. **Parameter Selection**: The values for ε (neighborhood radius) and MinPts (minimum number of points in a neighborhood) are chosen. To determine the optimal values for ε and MinPts, different combinations of these parameters are tested. The quality of the resulting clusters is evaluated using the silhouette coefficient, which provides a measure of how well each point is clustered. Details of silhouette

coefficient is discussed in upcoming subsections. The combination of ε and MinPts that results in the highest average silhouette coefficient is selected as the optimal parameter set. This combination provides the best balance between cluster cohesion and separation.

2. **Core Point Identification**: A point p is identified as a core point if at least $MinPts$ points are found within its ε-neighborhood:

$$N_\varepsilon(p) = \{q \in X \mid \|p - q\| \leq \varepsilon\}$$

where $\|p-q\|$ denotes the Euclidean distance between points p and q, and $|N_\varepsilon(p)| \geq MinPts$.

3. **Cluster Formation**: Clusters are formed by expanding around each core point. A point q is directly reachable from a core point p if it lies within the ε-neighborhood of p, i.e., $\|p - q\| \leq \varepsilon$. A point is considered reachable from p if there is a sequence of directly reachable points connecting p to q. All reachable points are assigned to the same cluster as the core point.

4. **Noise Detection**: Points that remain unvisited during the clustering process are identified as noise (outliers).

(Ian and Eibe, 2011, p. 300).

Advantages and Drawbacks of DBSCAN

Advantages: DBSCAN offers several significant advantages. One of its key strengths is its ability to identify clusters of arbitrary shapes, which allows it to discover complex cluster structures that other algorithms might miss. The algorithm is also robust to noise and outliers, ensuring that such irregularities do not unduly impact the clustering results. Additionally, DBSCAN does not require the number of clusters k to be predefined, providing a flexible approach to clustering where the number of clusters can be determined from the data itself.

Drawbacks: However, DBSCAN does have some drawbacks. The performance of the algorithm depends a lot on the choice of parameters, particularly ε (the radius) and $MinPts$ (the minimum number of points), which requires careful tuning to achieve optimal results. Additionally, DBSCAN may struggle with high-dimensional data due

to the curse of dimensionality, where distance metrics become less meaningful. The algorithm can also face challenges when dealing with clusters of varying densities, as it may not perform well in scenarios where the density of clusters differs significantly.

DBSCAN has been recognized as a versatile clustering algorithm that identifies clusters of arbitrary shapes and sizes, while handling noise effectively. The concepts of core points, border points, and noise points have been leveraged to provide a robust clustering solution, suitable for a variety of datasets.

3.1.4 OPTICS

OPTICS (Ordering Points To Identify the Clustering Structure) is a density-based clustering algorithm that extends DBSCAN by providing a more flexible clustering solution. It produces a hierarchical clustering called the reachability plot, which allows for the identification of clusters of varying density and shapes (Ankerst et al., 1999, p. 155-160).

Key Concepts of OPTICS

- **Core Distance**: For each point, the core distance is the Euclidean distance to its MinPts-th nearest neighbor. The parameter MinPts represents the minimum number of points needed to define a cluster.
- **Reachability Distance**: The reachability distance of a point p with respect to another point q is the maximum of the core distance of q and the distance between p and q. It measures how strongly point p is connected to point q.
- **Reachability Plot**: A plot that represents the reachability distances of all points in the dataset, ordered by their reachability distance.

Steps of the OPTICS Algorithm :

Let $X = \{x_1, x_2, \ldots, x_n\}$ be the set of n data points, where each $x_i \in \mathbb{R}^d$.

1. **Core Distance Calculation**: Compute the core distance $CD(p)$ for each point p as:

$$CD(p) = \text{distance to the } MinPts\text{-th nearest neighbor}$$

2. **Reachability Distance Calculation**: Compute the reachability distance $RD(p, q)$ between points p and q as:

$$RD(p, q) = \max\{CD(q), d(p, q)\}$$

where $d(p, q)$ is the distance between points p and q.

3. **Reachability Plot**: Construct the reachability plot, ordering points by their reachability distances, which visualizes the density-based clustering structure.

4. **Cluster Formation**: To extract clusters from the reachability plot, a threshold ε is used to determine cluster boundaries. This process involves:

   - **Local Minima Detection**: Identifying local minima in the reachability plot, which represent potential cluster cores or boundaries.
   - **Cluster Expansion**: Expanding clusters from each local minimum by including all points within the specified reachability distance ε. Expansion continues until no additional points can be included within this distance.
   - **Separation of Clusters**: Points that are not reachable within the threshold ε from any cluster are classified as outliers or noise. These points are excluded from cluster formation.
   - **Cluster Validation**: Validating the clusters to ensure that points within the same cluster have similar reachability distances, while points in different clusters exhibit significantly different distances.

   (Ian and Eibe, 2011, p. 301-305).

Advantages and Drawbacks of OPTICS

Advantages: OPTICS excels in several areas, particularly in its ability to identify clusters of varying densities and shapes. Unlike algorithms that assume clusters to be of similar size or shape, OPTICS can detect clusters with diverse structures, which makes it highly versatile for different types of data. Additionally, the algorithm provides a hierarchical view of the clustering structure through its reachability plot, offering a deeper insight into the clustering process and allowing for a more nuanced understanding of the data. The robustness of OPTICS to noise and outliers is another significant

advantage, as it ensures that the presence of irregular data points does not adversely affect the clustering results.

Drawbacks: Despite its strengths, OPTICS has some limitations. The algorithm requires careful parameter tuning for $MinPts$ and ε, which can be a complex and time-consuming process. Selecting inappropriate parameters can lead to suboptimal clustering results. Additionally, the performance of OPTICS may degrade when dealing with high-dimensional data due to the curse of dimensionality, where the distance metrics become less meaningful. Furthermore, the computational cost of OPTICS can be high, especially for large datasets, which may impact its scalability and efficiency in practical applications.

OPTICS is a powerful density-based clustering algorithm that extends DBSCAN by providing a hierarchical clustering structure and flexibility in identifying clusters of varying densities and shapes. By leveraging core distances and reachability distances, OPTICS offers robust clustering solutions suitable for complex datasets.

3.1.5 ORICC

Orthogonal Recursive Cluster Identification with Constraints (ORICC) is a clustering method developed for short time-course microarray experiments. The challenge of identifying gene clusters with similar expression patterns while accounting for the inherent constraints of time-course data is addressed by this method. Order-restricted candidate profiles are leveraged by ORICC to improve clustering performance and interpretability.

Objective: The primary objective of ORICC is to identify clusters of genes whose expression profiles are consistent with specific order constraints over time. By incorporating these constraints, which often reflect biological relevance such as temporal expression patterns in gene regulation, the clustering results are enhanced.

Methodology: A two-stage approach is employed by ORICC:

1. **Profile Identification:**
    - **Order-Restricted Candidate Profiles:** In the first stage, candidate profiles that conform to the order constraints imposed by the time-course data are identified. These constraints may include monotonic increases or decreases in

gene expression over time, as well as more complex patterns. Each candidate profile is considered a potential gene cluster. For a candidate profile vector $\mathbf{c}$ in a time-course dataset with T time points, the profile is represented as:

$$\mathbf{c} = (c_1, c_2, \ldots, c_T)$$

where:

- c_i is the gene expression value at time point i.
- i ranges from 1 to T, where T represents the total number of time points.

The following types of candidate profiles are considered:

- **Monotone Increasing Candidate Profiles:** Gene expression values exhibiting a non-decreasing trend over time are characterized by this profile.
$$C^{\uparrow} = \{\mathbf{c} \in \mathbb{R}^T \mid c_1 \leq c_2 \leq \cdots \leq c_T\}$$
- **Monotone Decreasing Candidate Profiles:** Gene expression values showing a non-increasing trend over time are defined by this profile.
$$C^{\downarrow} = \{\mathbf{c} \in \mathbb{R}^T \mid c_1 \geq c_2 \geq \cdots \geq c_T\}$$
- **Up-Down Candidate Profiles (Umbrella Order):** Gene expression values that increase up to a maximum at time point i and then decrease thereafter are represented by this profile.
$$C^{\wedge} = \{\mathbf{c} \in \mathbb{R}^T \mid c_1 \leq c_2 \leq \cdots \leq c_i \text{ and } c_i \geq c_{i+1} \geq \cdots \geq c_T\}$$
- **Down-Up Candidate Profiles:** This profile is characterized by gene expression values that first decrease to a minimum and then increase.
$$C^{\vee} = \{\mathbf{c} \in \mathbb{R}^T \mid c_1 \geq c_2 \geq \cdots \geq c_i \text{ and } c_i \leq c_{i+1} \leq \cdots \leq c_T\}$$
- **Cyclical Candidate Profiles:** Oscillations with minima and maxima at specified time points are shown by this profile.

$$C^{\wedge}\wedge = \{\mathbf{c} \in \mathbb{R}^T \mid c_1 \leq c_2 \leq \cdots \leq c_i \geq c_{i+1} \geq \cdots \geq c_j \leq c_{j+1} \leq \cdots \leq c_k \geq c_{k+1} \geq \cdots \geq$$

– **Incomplete Inequality Candidate Profiles:** These profiles are utilized when specific ordering constraints cannot be fully defined.

$$C^I = \{\mathbf{c} \in \mathbb{R}^T \mid c_1 \leq c_2 \leq \cdots \leq c_i \text{ and } c_i \geq c_{i+1} \geq \cdots \geq c_T \text{ with partial ordering}\}$$

- **Orthogonal Transformations:** Orthogonal transformations are applied by ORICC to ensure that the identified profiles are mutually orthogonal, thereby simplifying the clustering process and reducing redundancy in the candidate profiles.

2. **Recursive Clustering:**
   - **Recursive Algorithm:** In the second stage, recursive clustering is performed by ORICC using the identified order-restricted profiles. Data are partitioned into clusters based on the similarity of gene expression profiles to these candidate profiles.
   - **Information Criterion-Based Model Selection:** An information criterion (such as BIC or AIC) is used to select the optimal number of clusters and refine the clustering results. This step ensures that the chosen model best fits the data while accounting for the complexity of the clustering solution.

Implementation: The ORICC method has been implemented in a computational algorithm that performs profile identification and recursive clustering steps. This algorithm was specifically developed for short time-course microarray experiments, which involve gene expression data collected over a series of time points. The goal is to identify clusters of genes that exhibit similar temporal expression patterns (Liu et al., 2009).

Advantage and Drawbacks: A notable advantage of the ORICC method is its capacity to generate biologically relevant clusters through the integration of order restrictions. This feature makes it a valuable tool for analyzing gene expression data that adheres to specific temporal or experimental sequences. Additionally, the inclusion of an information criterion serves to mitigate the risk of overfitting, thus enhancing the generalizability of the results.

However, certain limitations are inherent to ORICC. The effectiveness of the algorithm is dependent on the accurate specification of order restrictions; if the assumed order is incorrect, the resultant clusters may not be meaningful.

In R programming, two variants of ORICC exists: ORICC1 and ORICC2. The primary distinction between these variants are that ORICC2 incorporates a pre-selection phase to remove uninformative data elements, such as flat genes, before applying the ORICC algorithm. This pre-selection phase can reduce the dataset size compared to the original dataset.

3.2 Elbow Method for Determining Optimal Parameters in K-Means and Hierarchical Clustering Algorithms

Clustering algorithms, such as K-Means and Hierarchical Clustering, require tuning certain parameters to achieve optimal performance. One crucial decision is the number of clusters. The elbow method is a heuristic technique that helps determine the ideal number of clusters (in K-Means) or the optimal cut-tree height in hierarchical clustering, balancing model complexity and fit (Han et al., 2022, p. 491-492).

Application in K-Means Clustering: The objective of K-Means clustering is to minimize the within-cluster sum of squares (WCSS). WCSS decreases as the number of clusters K increases. However, beyond a certain point, adding more clusters yields diminishing returns. The Elbow Method helps identify this point of diminishing returns, suggesting the optimal number of clusters.

Steps to find elbow point for K-Means Algorithm:

- **Clustering:** Run K-Means for a range of K values and calculate WCSS.
- **Plotting:** Plot WCSS against K. The curve typically shows a sharp decline followed by a flattening.
- **Elbow Point:** The optimal K is found at the "elbow," where adding more clusters does not significantly reduce WCSS.

Application in Hierarchical Clustering: In hierarchical clustering, a dendrogram is used to visualize the arrangement of clusters. The cut-tree height determines the number of clusters by cutting the dendrogram at a certain level. The elbow method can be used to find the optimal cut-tree height by analyzing the within-cluster dissimilarity (similar to WCSS in K-Means).

Steps to find elbow point for Hierarchical Clustering Algorithm:

- **Dendrogram Analysis:** Compute the hierarchical clustering dendrogram and calculate the dissimilarity for a range of cut heights.
- **Plotting:** Plot the dissimilarity or change in dissimilarity against the cut height.
- **Elbow Point:** The optimal cut-tree height is where the reduction in dissimilarity starts to slow, indicating a good balance between cluster compactness and separation.

Conclusion: The elbow method is a versatile technique for parameter selection in clustering algorithms. It can be applied to determine the optimal number of clusters in K-Means and the appropriate cut tree height in hierarchical clustering. While the method is intuitive, it may require visual interpretation or supplemental techniques when the "elbow" is not distinct.

3.3 Visualizations

In this subsection, the visualization techniques applied in this thesis, including Matplot for general data's structure, PCA plot for dimensionality reduction for statistical analysis, are thoroughly discussed.

3.3.1 Matplots

Visualization is a crucial aspect of gene clustering analysis as it allows researchers to understand complex biological data more effectively. This plot provides the tools needed to visualize the structure of gene clusters, offering insights into the patterns and relationships within the data (Kassambara, 2017, p. 25-26). Visualization helps bridge the gap between raw data and biological interpretation, making it a fundamental component of clustering analysis.

Matplot is widely used to visualize the structure of each gene in a dataset. Gene expression values are plotted across different time points, concentrations, or treatments. Through Matplot, the overall structure of gene profiles in the dataset can be observed, and it can be determined how many types of gene expression patterns are present.

Gene expression values are plotted across different time points, concentrations, or treatments to provide a visual representation of gene profiles. Matplot is used to create

these plots, which allow researchers to observe the overall structure of gene profiles and identify various expression patterns within the dataset.

Mathematically, each gene profile can be represented as a function $f_i(t)$, where t denotes time points or concentrations, and $f_i(t)$ denotes the expression level of gene i at time t. The plot is typically constructed as follows:

$$E_i = \{(t_j, f_i(t_j)) \mid j = 1, 2, \ldots, m\}$$

where E_i denotes the expression profile of gene i with m observations at different time points or concentrations. In a typical plot, the y-axis represents the expression values, and the x-axis represents the time points, concentrations, or other relevant variables.

This visualization helps in understanding how gene expression changes over time or across different conditions, facilitating the identification of patterns and trends in the data.

3.3.2 PCA Plot

Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms a dataset into a set of linearly uncorrelated variables known as principal components. PCA is widely applied in exploratory data analysis and in the development of predictive models. The results of PCA are visualized to understand the structure and relationships in high-dimensional data (Kassambara, 2017, p. 30-31).

When PCA Plots are Used:

Dimensionality Reduction: PCA reduces the number of variables in a dataset while retaining as much information as possible. Mathematically, PCA involves the following steps:

1. **Centering the Data**: - Given a dataset $X \in \mathbb{R}^{n \times d}$, where n is the number of samples and d is the number of features, center the data by subtracting the mean of each feature:

   $$X_{\text{centered}} = X - \bar{X}$$

   Here, $\bar{X} \in \mathbb{R}^{1 \times d}$ is the mean vector of the features, with each element $\bar{X}_j$ being the mean of the j-th feature.

2. **Covariance Matrix Calculation**: - Compute the covariance matrix Σ of the centered data:

$$\Sigma = \frac{1}{n-1} X_{\text{centered}}^T X_{\text{centered}}$$

The covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$ captures the variance and covariance between the features. Each element Σ_{ij} represents the covariance between the i-th and j-th features.

3. **Eigenvalue Decomposition**: - Perform eigenvalue decomposition on the covariance matrix Σ:

$$\Sigma V = V \Lambda$$

where $\Lambda \in \mathbb{R}^{d \times d}$ is the diagonal matrix of eigenvalues λ_i, and $V \in \mathbb{R}^{d \times d}$ is the matrix of eigenvectors. Each eigenvector v_i corresponds to an eigenvalue λ_i, representing the variance captured by that principal component.

4. **Selecting Principal Components**: - Choose the top k eigenvectors corresponding to the largest eigenvalues to form the matrix V_k. Project the data onto these principal components:

$$X_{\text{PCA}} = X_{\text{centered}} V_k$$

where $V_k \in \mathbb{R}^{d \times k}$ is the matrix containing the top k eigenvectors, and $X_{\text{PCA}} \in \mathbb{R}^{n \times k}$ is the reduced-dimensional representation of the data.

PCA helps in identifying patterns and clusters in the data. By projecting the data onto the top principal components, researchers can visualize and analyze the structure of the data in a lower-dimensional space. PCA enables the visualization of high-dimensional data in 2D or 3D plots by projecting it onto the first two or three principal components. This visualization aids in understanding the relationships between data points and the overall data structure.

3.4 Clusters performance measurement

In this subsection, the performance metrics used to evaluate clustering results, such as the silhouette coefficient and adjusted rand index, are detailed and discussed.

3.4.1 Silhouette Coefficient

The silhouette coefficient is used to measure cluster cohesion and separation, assessing the quality of clusters formed by clustering algorithms. For a single data point i, the silhouette coefficient is computed as follows (Bishop, 2006, p. 405-411):

1. **Cohesion (a_i)**: The average distance between data point i and all other points in the same cluster C_i is calculated as:

$$a_i = \frac{1}{|C_i| - 1} \sum_{j \in C_i, j \neq i} d(i, j)$$

   where $d(i, j)$ represents the distance between data points i and j, and $|C_i|$ denotes the number of points in cluster C_i.

2. **Separation (b_i)**: The average distance between data point i and all points in the nearest neighboring cluster C_{nearest} is determined as:

$$b_i = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

3. **Silhouette Coefficient (s_i)**: The silhouette coefficient for data point i is computed as:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

   The silhouette coefficient s_i ranges between -1 and 1:

   - $s_i \approx 1$: Data point i is well-clustered.
   - $s_i \approx 0$: Data point i is on the boundary of two clusters.
   - $s_i \approx -1$: Data point i may be assigned to the wrong cluster.

4. **Overall Silhouette Coefficient (S)**: The average silhouette coefficient across all data points is calculated as:

$$S = \frac{1}{n} \sum_{i=1}^{n} s_i$$

   where n denotes the total number of data points.

The silhouette coefficient is employed to provide a quantitative measure for evaluating the effectiveness of clustering algorithms in generating well-separated and compact clusters.

3.4.2 Adjusted Rand Index

The similarity between two clusterings is measured by the Adjusted Rand Index (ARI). This measure considers all pairs of samples and counts those pairs that are assigned to the same or different clusters in both the predicted and true clusterings. The ARI adjusts for chance agreement and is corrected for the number of clusters and the size of the dataset. For two clusterings X and Y with n samples, the ARI is calculated as (Bishop, 2006, p. 135-137):

$$ARI(X,Y) = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}$$

where:

- n_{ij} represents the number of pairs of samples that are assigned to the same cluster in both X and Y.
- a_i denotes the number of samples in cluster i in X.
- b_j denotes the number of samples in cluster j in Y.
- $\binom{x}{2} = \frac{x(x-1)}{2}$ represents the binomial coefficient.

The ARI ranges from -1 to 1:

- ARI = 1: Indicates a perfect match between the clusterings.
- ARI = 0: Represents the expected value for random clusterings.
- ARI $<$ 0: Indicates less agreement than would be expected by chance.

A robust measure of similarity between two clusterings is provided by the Adjusted Rand Index, accounting for the effect of chance.

4 Statistical analysis

The clustering algorithms and methods outlined in the previous section are employed here to analyze and interpret the given VPA dataset. Additionally, those methods are applied to various simulated datasets as well in order to evaluate the consistency of the algorithm's result.

4.1 VPA Dataset Characteristics

Before simulating any dataset, characteristics of the VPA dataset is analysed first. This helps to incorporate some of the properties of the original data to the simulated dataset later. The statistical characteristics of the original dataset are therefore thoroughly covered in this paragraph. As previously mentioned, there are 54,675 genes in the Valporic Acid dataset. To reduce the runtime when working with this large datasets, 1,000 genes with the highest variability in gene expression across the concentration groups were selected from the entire dataset for further analysis.

	Control	25mM	150mM	350mM	450mM	550mM	800mM	1000mM
Min	2.73	2.74	2.78	2.73	2.96	3.02	2.82	2.85
1st Qu.	4.63	4.42	4.68	5.15	5.31	5.41	5.69	5.74
Median	5.46	5.49	5.84	6.25	6.34	6.46	6.86	7.04
Mean	5.80	5.80	6.06	6.40	6.54	6.63	6.90	7.00
3rd Qu.	6.81	6.78	7.12	7.43	7.57	7.65	8.02	8.20
Max	13.15	13.09	12.83	12.40	12.15	12.17	12.58	12.81

Table 1: Summary statistics of top 1000 most variabled genes from VPA dataset.

Some statistical measures of the selected gene's gene expression value across various concentration groups such as Min, Median, Mean, Max etc. are included in the Table 1.

The preceding table is displayed in compact form by the boxplot Figure 1.

Both the table and boxplot indicate a pattern of increasing gene expression values with higher concentrations. However, the distribution of gene expression values, including similar median values and nearly identical first and third quartiles, appears consistent within each concentration group across replicates. The interquartile range (IQR) of gene expression values across all concentrations and replicates remains relatively consistent. While this pattern is observed in the top 1,000 most variable genes, indicating an increasing gene expression profile, it is not representative of all genes in the VPA dataset.

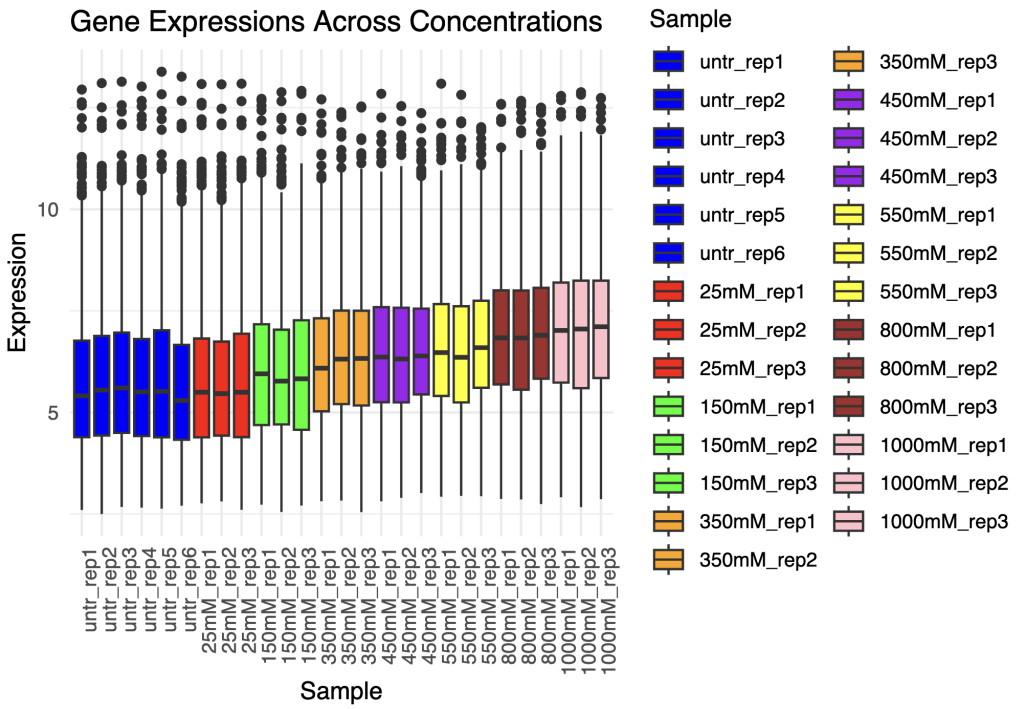


Figure 1: Boxplot of top 1000 most variabled genes of VPA dataset.

Figure 30 in the Appendix presents a boxplot for the gene expression values of all 55,000 genes in the VPA dataset, where the average gene expression values appear relatively flat. This suggests that not all genes in the VPA dataset exhibit the same expression profile as the top 1,000 most variabled genes. However, to simplify the analysis by reducing the runtime and complexity of using bigger dataset, only the top 1,000 genes with the highest variability in expression were selected for further investigation.

The 1000 chosen genes' individual structures are shown by matplot Figure 31 at Appendix section. Multiple profiles/ gene structure can be detected from the plot for example increasing, decreasing, flat structure etc.

Next, 5 clustering algorithms namely K-Means, Hierarchical clustering, DBSCAN, OPTICS and ORICC is applied on the selected VPA dataset. After applying the algorithms on the dataset, the result of the same is visualized. However, as the dataset is high dimensional, before plotting the result, principle component analysis or PCA method is applied to the the clustered dataset and the high dimensional dataset is projected into 2 dimensional dataset.

In 2 dimensional PCA plot, x-axis shows the principle component 1 which represents the maximum varianced points (genes) of the dataset. And the y-axis shows the 2nd most varianced data points of the dataset known as principle component 2.

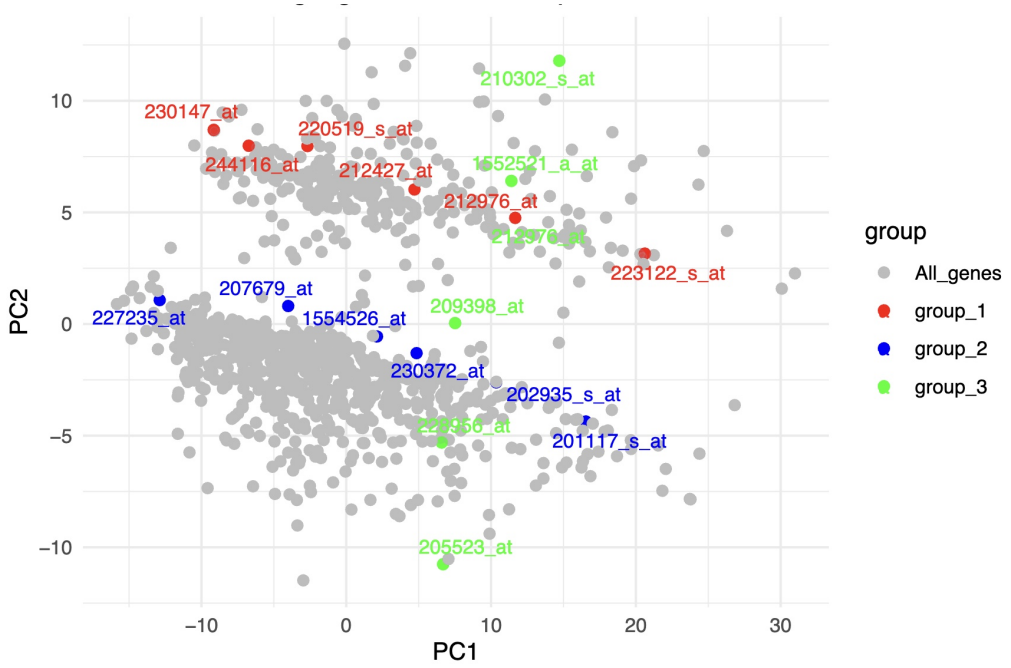


Figure 2: 2 dimensional PCA plot of the top 1,000 most variabled genes from the VPA dataset, highlighting some of the genes.

Figure 2 presents the two-dimensional PCA plot of the selected VPA dataset, with certain genes highlighted by name and position. Three gene groups were selected based on their positions in the PCA plot. Gene group_1, highlighted in red, consists of six genes located in the upper half of the plot, aligned nearly in a straight line. Gene group_2, marked in blue, is composed of genes from the lower half of the plot, positioned along a similar horizontal axis. Finally, gene group_3, shown in green, consists of genes located along the same vertical axis.

Figure 3, 4 and 5 shows the individual structure of the genes selected from the gene group 1,2 and 3 respectively where x-axis denotes the concentration value and y-axis shows the gene expression values of that gene. The dotted line in each of gene plots represents the connection between the average means of the replicates at each concentration.

In Figure 3, the structure of the genes from gene group_1 is shown who are selected from left to right of the PCA plot. It is noticeable that genes positioned on the right side exhibit higher average gene expression values compared to those on the left. For

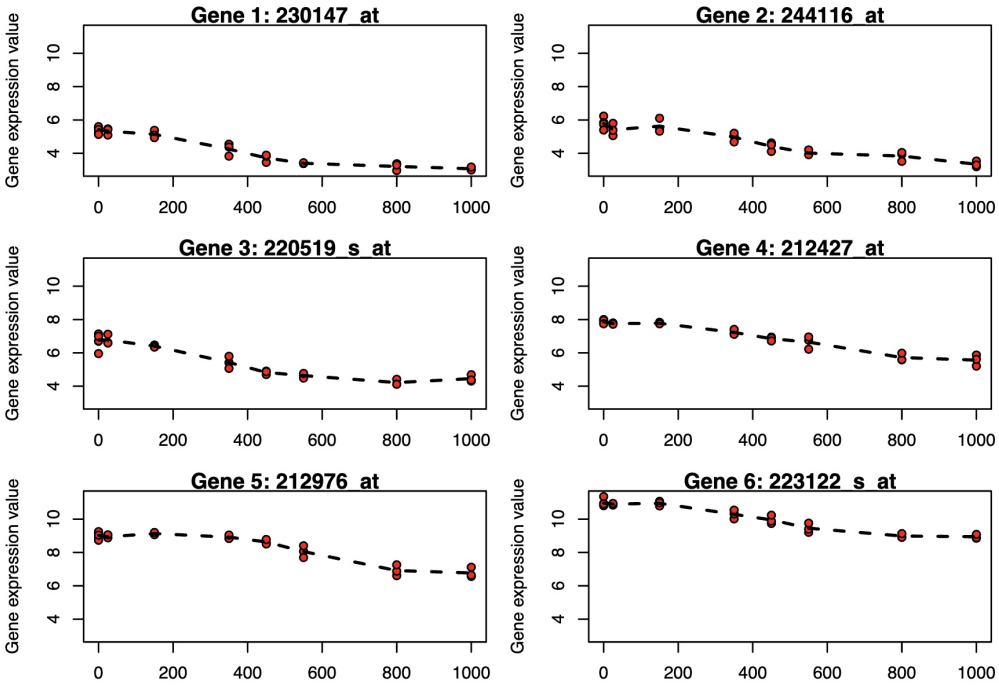


Figure 3: Structure of each genes from group_1.

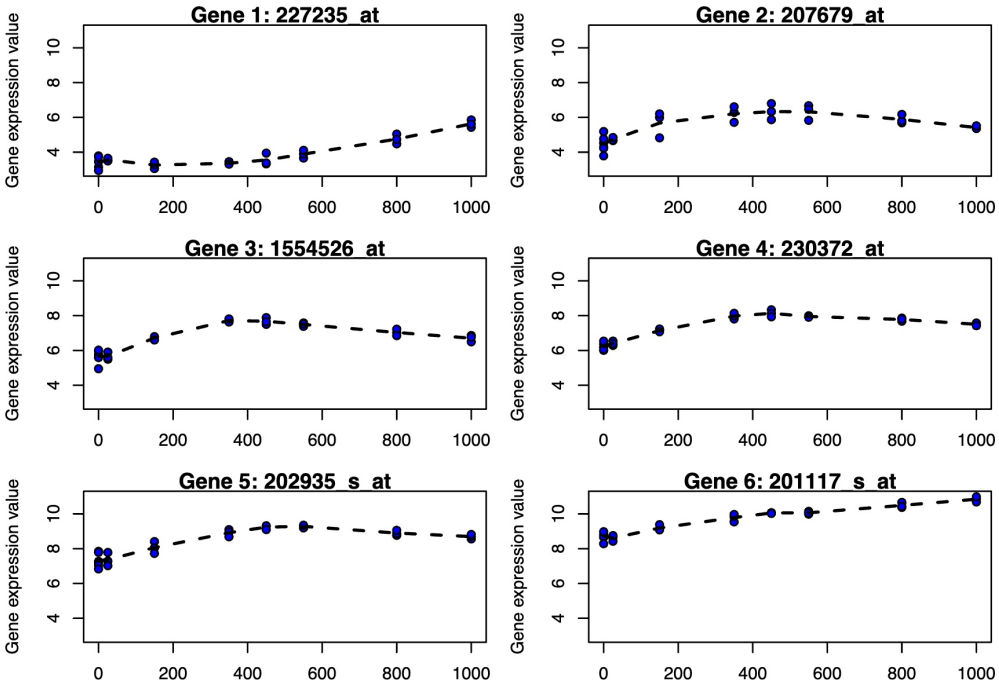


Figure 4: Structure of each genes from group_2.

example, gene **230147_at**, located at the far left of the PCA plot Figure 2, has a mean

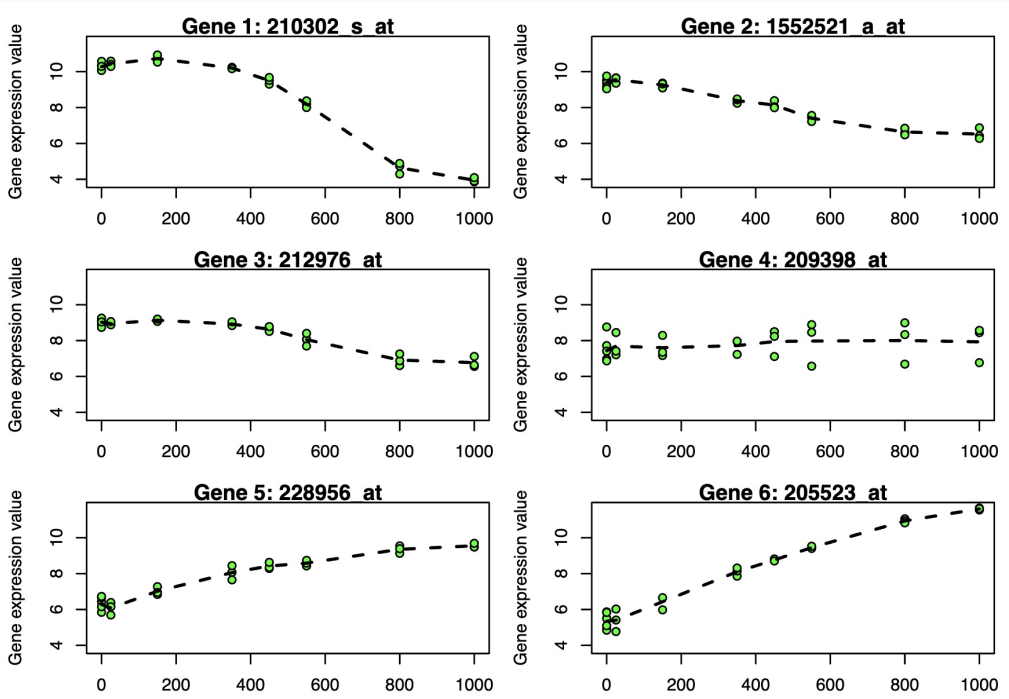


Figure 5: Structure of each genes from group_3.

gene expression value of approximately 4, whereas gene **223122_s_at**, positioned at the far right, shows a mean gene expression value of around 10. This indicates that as one moves from left to right along the x-axis, or the first principal component axis, there is a general increase in gene expression values. Additionally, it is observed that all the selected genes exhibit a decreasing expression profile, meaning that gene expression decreases as concentration increases, regardless of their position on the plot.

Individual gene structure are shown by Figure 4 from left to right for gene group_2. Similar as before, as genes are picked from left to right of the PCA plot, the average gene expression value increases in general. For extreme left gene **227235_at** the gene expression value is approximately within the range of 4 to 6 where as for the extreme right gene **201117_s_at** it is between 8 to 11. However, here, all genes seems to have increasing gene profiles i.e. with increase in concentration of VPA, the gene expression value increases.

In Figure 5, the individual gene structure of gene group_3 is plotted selecting them from top to bottom. Now the average gene expression value of the genes depends on the position where the genes resides in the PCA plot along the y axis i.e principle component 2 axis. That means, gene **210302_s_at** placed at extreme top of PCA plot and gene

205523_at placed at extreme bottom of the plot, are almost equidistant from x axis. So, their upper and lower limit of gene expression value is pretty much same which is between 10 and 4. This is same for all other genes as well. If genes are selected near from x axis for example gene **209398_at**, it have flat gene profile . Also, as mentioned earlier, genes which are from the upper half of the PCA plot (**210302_s_at**, **1552521_a_at** and **212976_at**), seems to have decreasing profile and genes which are from lower half of the plot (**228956_at**, **205523_at**), seems to have increasing gene profile. Also, genes residing near the x axis or the PC1 axis i.e at PC2 = 0, have flat gene profile.

In summary, upper half of the PCA plot contains genes with decreasing profile, lower half of the plot contains genes with increasing profile and as we go from top to bottom, the gene's structure (profile) changes from decreasing to increasing profile. Last but not the least, gene expression value of the genes increases as we go from left to right of the PCA plot horizontally.

Application of Clustering Algorithms on VPA dataset: In this part, all clustering algorithms, including K-Means, Hierarchical clustering, DBSCAN, OPTICS, and ORICC, are applied to the VPA dataset, focusing on the top 1,000 most variabled genes.

Distance based clustering algorithm:

In this subsection, discussion of K-Means and Hierarchical clustering algorithms applied to the selected VPA dataset are done. The optimal value of K for the K-Means algorithm and the optimal cut-tree height for the Hierarchical Clustering algorithm are determined using the elbow plot and silhouette coefficient.

Figure 6 presents the elbow plot for the VPA dataset, focusing on the top 1,000 most variable genes. The plot suggests that the elbow appears at approximately $k = 2$, 4, or 6, indicating that the optimal k-value for k-means and the optimal cut-tree height for hierarchical clustering could potentially be any of these values.

In addition to the elbow plot, the optimal k-value and cut-tree height for the k-means and hierarchical clustering algorithms are further refined using the silhouette score. A custom function is developed in R to evaluate a range of k-values or cut-tree heights. For each k value, the k-means and hierarchical clustering algorithms are applied to the selected VPA dataset, and the total silhouette coefficient is calculated based on the clusters assigned to each data point. The k-value or cut-tree height that results in the maximum silhouette coefficient is considered optimal, as a higher silhouette coefficient indicates better clustering quality for the dataset.

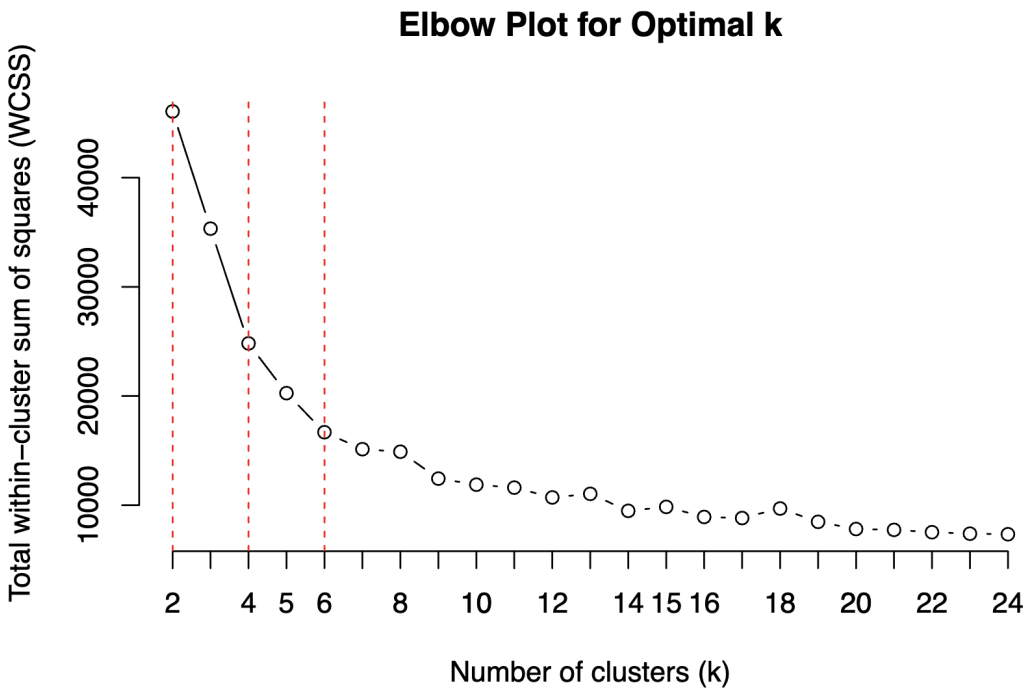


Figure 6: Elbow plot of top 1000 genes from VPA dataset.

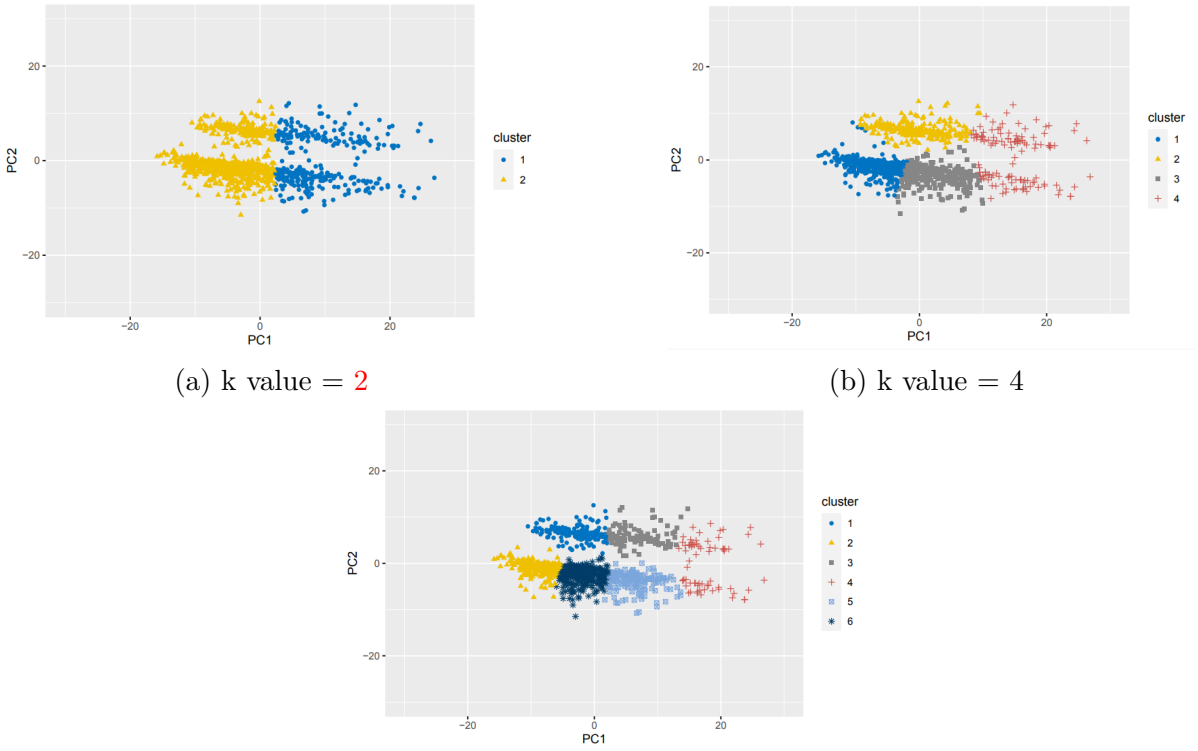
Parameter settings	K-val/cut-tree height	SC K-means	SC HC
1	2	0.46	0.40
2	4	0.44	0.43
3	6	0.45	0.40

Table 2: K-values/sut-tree height and corresponding silhouette coefficients for k-means and Hierarchical clustering algorithm.

Table 2 presents the optimal k-values and cut-tree heights for the k-means and hierarchical clustering algorithms, respectively, as determined by the elbow plot. It also shows the corresponding total silhouette coefficients for the dataset after applying these algorithms based on the user-defined function.

According to the silhouette score, parameter setting 1, where the k-value is 2, is optimal for the k-means clustering algorithm, as it results in the highest silhouette coefficient of 0.46 for the selected VPA dataset. Conversely, parameter setting 2, with a cut-tree height of 4, is optimal for the hierarchical clustering algorithm, yielding a maximum silhouette coefficient of 0.43.

Despite this, both the k-means and hierarchical clustering algorithms were applied to the selected VPA dataset for all k values of 2, 4, and 6, respectively. Figures 7 and 8 illustrate



(a) k value = 2

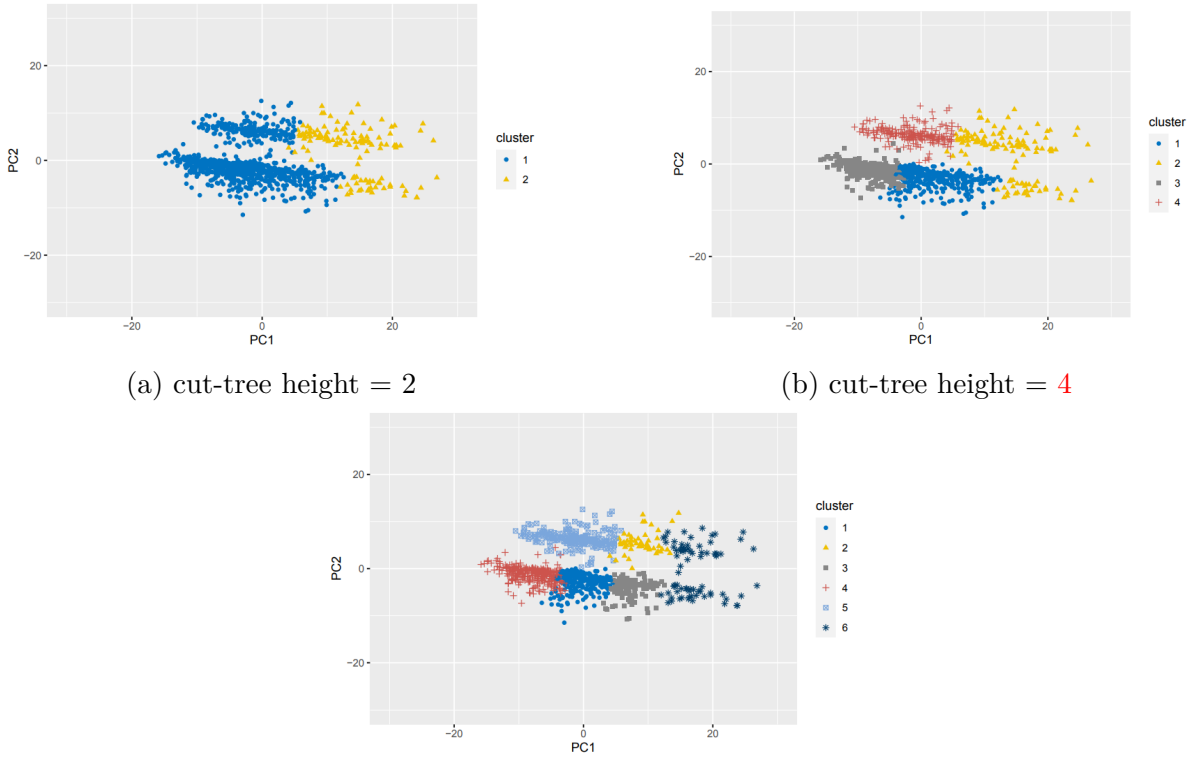
(b) k value = 4

(c) k value = 6

Figure 7: 2 dimensional PCA plot for K-Means algorithm with different K values.

the 2-dimensional PCA plots of the k-means and hierarchical clustering algorithms with different k-values and cut-tree heights.

At first glance, both algorithms cluster the data in a broadly similar way. However, as mentioned in the earlier subsection, the two-dimensional PCA plot of the selected VPA dataset indicates that the upper half of the plot is associated with genes showing decreasing expression profiles, while the lower half corresponds to genes with increasing profiles. Despite these patterns, neither distance-based clustering algorithm effectively distinguished the dataset based on gene expression profile. Instead, clustering was observed along the y-axis, specifically at the value where PC1 = 0. In this case, genes on the left side of the PC1 = 0 axis were grouped together based on relatively lower gene expression values, while genes on the right side were clustered based on higher expression values, regardless of their gene profiles. Since both algorithms consider the distance between gene expression values as the primary clustering criterion, rather than the overall gene expression structure, the resulting clusters grouped genes with similar expression ranges, irrespective of their gene expression profiles.



(a) cut-tree height = 2

(b) cut-tree height = 4

(c) cut-tree height = 6

Figure 8: 2 dimensional PCA plot for Hierarchical clustering algorithm with different cut-tree height.

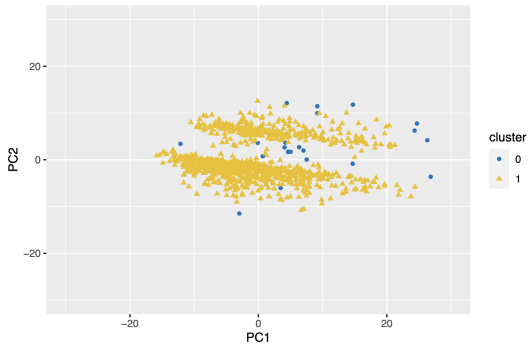
Density based clustering algorithm

In the context of density-based clustering, the DBSCAN and OPTICS algorithms are applied. For the DBSCAN algorithm, two user-defined parameters—ϵ (the neighborhood radius) and MinPts (the minimum number of points)—are required to perform clustering on the VPA dataset. To determine the optimal values for these parameters, a custom function is developed in R. This function explores a range of possible ϵ and MinPts values, applying the DBSCAN algorithm to the dataset for each combination. For the ϵ parameter, a range of values between 1 and 20, incremented by 0.1, was used. For the MinPts parameter, every value between 3 and 30 was tested. These ranges were determined through extensive trial and error to optimize the parameters for the dataset. The silhouette coefficient is then calculated for each scenario, and the combination of ϵ and MinPts that yields the highest silhouette coefficient is considered optimal for the dataset.

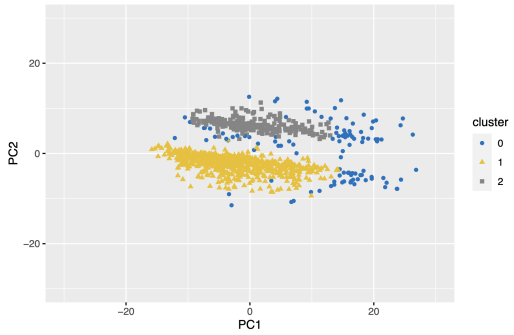
Parameter Setting	eps	minPts	Num. Clusters	Total avg. Silhouette Coefficient
1	1.4	6	1	0.42
2	1.9	11	2	0.37
3	1.1	7	3	0.34

Table 3: Optimal value of user-defined parameters based on silhouette coefficient for DBSCAN on top 1000 genes of VPA dataset.

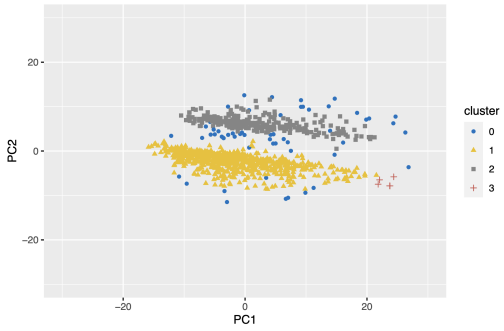
Table 3 shows the top 3 parameters combination of DBSCAN for which highest silhouette coefficient is achieved.



(a) Parameter Settings = 1



(b) Parameter Settings = 2



(c) Parameter Settings = 3

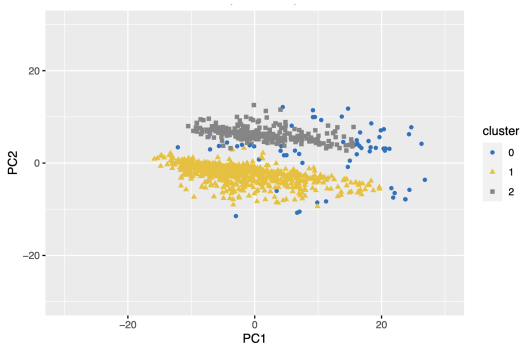
Figure 9: 2 dimensional PCA Plot of the dataset after applying DBSCAN algorithm.

Figure 9 presents the 2 dimensional PCA plot of the VPA dataset after applying the DBSCAN algorithm using the aforementioned parameter settings. Although parameter setting 1, with $\epsilon = 1.4$ and MinPts $= 6$, appears to be optimal for DBSCAN based on the highest silhouette coefficient of 0.42, it is evident that nearly all data points are grouped into a single large cluster, with some genes classified as outliers (cluster level 0). This raises concerns about the effectiveness of this parameter setting. However, the PCA plots for parameter settings 2 and 3 suggest more meaningful clustering, as

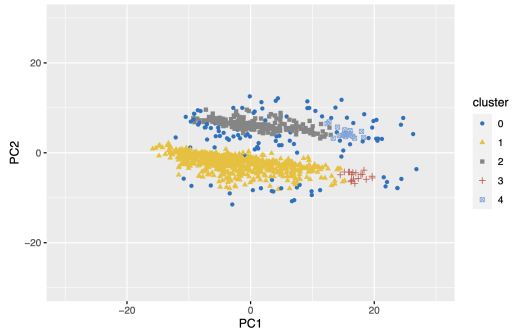
they more effectively differentiate between the upper half of the dataset (which contains decreasing gene profiles) and the lower half (which contains increasing gene profiles).

Parameter Setting	minPts	Num. Clusters	Total avg. Silhouette Coefficient
1	10	2	0.32
2	8	4	0.30
3	17	3	0.28

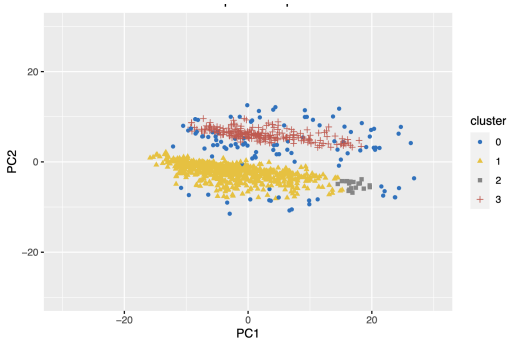
Table 4: Optimal value of user-defined parameters based on silhouette coefficient for OPTICS on top 1000 genes of VPA dataset.



(a) Parameter Settings = 1



(b) Parameter Settings = 2



(c) Parameter Settings = 3

Figure 10: 2 dimensional PCA Plot of the dataset after applying OPTICS algorithm.

To determine the optimal MinPts value for the OPTICS algorithm, a custom function is employed, which tests a range of MinPts values on the VPA dataset. Here also, similar as DBSCAN algorithm, range of MinPts values between 3 and 30 were explored. These parameter ranges were chosen after extensive experimentation to identify the most suitable values for the dataset. For each MinPts value, the OPTICS algorithm is executed, and the silhouette coefficient is calculated. The MinPts value yielding the highest silhouette coefficient is deemed optimal for the dataset. Table 4 presents the top three parameter settings with the highest silhouette coefficients. Figure 10 illustrates the

2 dimensional PCA plot following the application of the OPTICS algorithm with these settings. Based on the silhouette coefficient, parameter setting 1, with MinPts of 10 and a total average silhouette coefficient of 0.32, appears to be optimal. In comparison to DBSCAN, this optimal setting for OPTICS offers a clearer outcome, showing two distinct clusters: one representing the upper half of the dataset with decreasing gene profiles, and the other representing the lower half with increasing gene profiles, along with some outliers.

Model Based Clustering Algorithm:

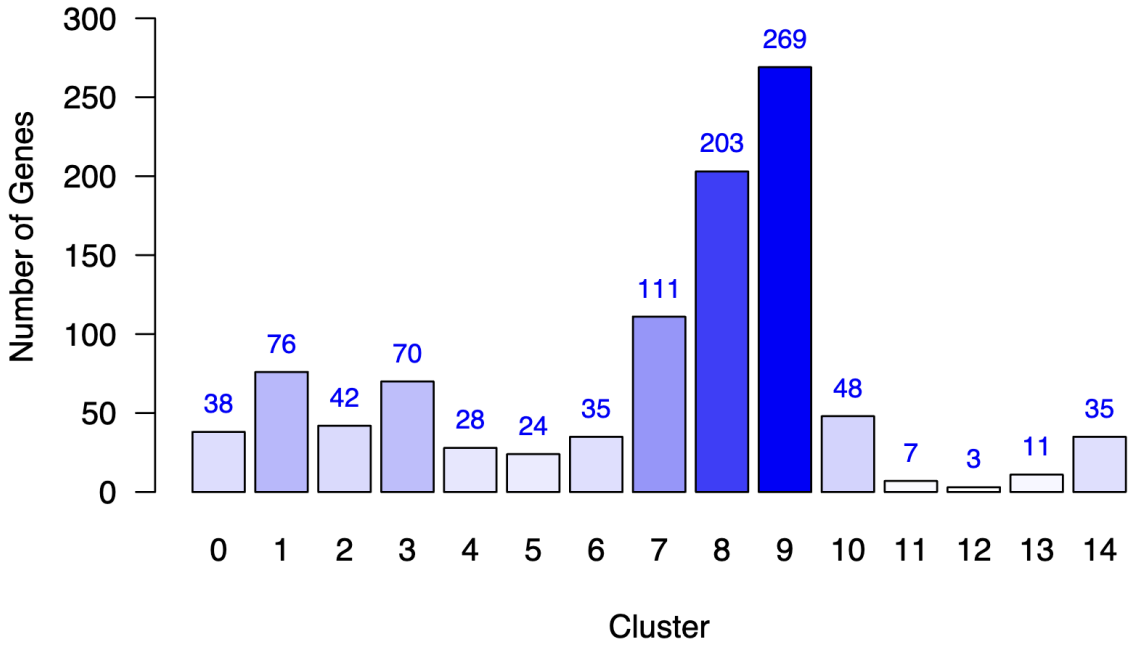


Figure 11: Barplot showing number of genes assigned to the gene profile number by ORICC on top 1000 genes of VPA dataset.

Due to the challenge of determining optimal values for user-defined parameters in clustering algorithms, and the significant impact these parameters have on the algorithm's outcome, the ORICC algorithm was applied to the selected VPA dataset. Since ORICC is specifically designed for order-restricted short time-course microarray experiments (such as gene expression datasets), ORICC is also used here and its performance was evaluated in comparison to k-means, hierarchical clustering, DBSCAN, and OPTICS algorithms. The ORICC algorithm does not require user-defined parameters, making it a more straightforward choice for clustering in this context. Table 5 lists the gene profile

Profile number	Gene profile
1	Decreasing
2	Up down max at 2
3	Up down max at 3
4	Up down max at 4
5	Up down max at 5
6	Up down max at 6
7	Up down max at 7
8	Increasing
9	Down up min at 2
10	Down up min at 3
11	Down up min at 4
12	Down up min at 5
13	Down up min at 6
14	Down up min at 7

Table 5: Gene profiles and corresponding profile number of ORICC on selected VPA dataset.

and their corresponding profile numbers after applying the ORICC algorithm on the VPA dataset. Figure 11 shows the number of genes assigned to the gene profiles. It is observed that the majority of genes fall into the gene profile types - Down Up Minimum at 2 and Increasing and Up Down Maximum at 7. Thus, the top 1000 most variable genes in the VPA dataset are predominantly associated with gene profiles 7, 8, and 9. Very less number of genes are associated with gene profile 11, 12 and 13 with number of genes of 7, 3 and 11 out of 1000 genes respectively. Lastly, the silhouette coefficient for the selected VPA dataset, after applying the ORICC clustering algorithm, is 0.48.

4.2 Data simulation

To assess the consistency and reliability of the clustering algorithms, datasets with varying characteristics are simulated. These datasets are then analyzed by applying the clustering algorithms, and their performance is evaluated by comparing the true cluster assignments with the predicted clusters for each gene, using the Adjusted Rand Index (ARI).

Each simulated dataset mirrors the dimensionality of the selected portion of original dataset, comprising 1000 rows (each representing synthetic genes) and 27 columns (each of which represents the combinations of 8 concentrations and their replicates). Similar

to the VPA dataset, the first 6 columns correspond to 6 replicates of the control group, and the subsequent 3 columns represent replicates for each of the 7 concentrations. Each cell in the dataset holds the synthetic estimate of the gene expression value for a gene at a specific concentration.

Three different methods are used to generate the synthetic data:

- Normally distributed gene expression value with a standard deviation of 1 which adds variability to the replicate values, simulating the natural randomness and noise inherent in the data. The peak value is set as 8 for the gene expressions.
- Normally distributed data with a standard deviation of 2 to add higher variability in the gene expression value of the replicates and a peak value of gene expressions of 8.
- Normally distributed data with a standard deviation of 5 which adds much higher randomness and noise to the gene expression value of the replicates with a peak value of 8.

For each data generation method, 3 gene profile groups are created, each consisting of various gene's profile structure.

In the following subsections, each group of gene profiles are discussed in details, with the corresponding number of genes simulated for each profile. The true cluster level of each gene profile within every group is also indicated in the tables, shown in parentheses next to the gene profile name.

In this thesis, a total of 15 distinct gene profiles were simulated. These profiles are visualized in Figure 32 in the appendix. For example, the gene profile Up_Down_Max_7 represents a profile that shows a steady increase in gene expression value up to the 7th concentration (800mM), followed by a steady decrease. That is the peak value of this gene profile is at concentration 800mM. Likewise, the Down_Up_Min_7 group represents a profile that decreases continuously until the 7th concentration (800mM), then begins to increase, with the lowest gene expression value at 800mM.

4.2.1 Gene Profile Group 1

Table 6 displays the various gene profiles simulated for gene profile group 1 along with their corresponding gene counts. A total of 8 distinct gene profiles are utilized in this group. Number of genes are defined for each gene profile in such a way that they are

Gene Profile	Number of Genes
Flat (0)	200
Decreasing (1)	150
Up Down Max 2 (2)	100
Up Down Max 3 (3)	100
Up Down Max 6 (6)	100
Increasing (8)	150
Down Up Min 3 (10)	100
Down Up Min 6 (13)	100

Table 6: Genes profile group 1 with gene profile, its true cluster level and number of genes simulated for each profile.

more or less equal to maintain the homogeneity of the gene profile group. However, slightly higher number of genes are assigned to Flat, Decreasing and Increasing genes profiles because of 2 reason. One, more emphasize is given to these 3 profiles or in other words, these 3 profiles are considered as the main/dominating gene profiles for these gene profile group. Second, variety in the simulated dataset is expected as a result of which different gene profile groups are constructed with different dominating gene profiles.

4.2.2 Gene Profile Group 2

Table 7 presents the simulated gene profiles and the corresponding number of genes for each gene profiles. Total 10 gene profiles are simulated in this gene group. The profiles labeled Increasing, Up Down Max 2, 3, 4, and 5 exhibit closely aligned peak values of gene expression which is observable from the Figure 32, found in the Appendix section. In other words, the basic structure of these 5 gene profile groups are similar because all of them have either increasing or increasing-decrease tendency in gene expression profile. The gene profiles Down Up Min 2, 3, 4, and 5 display comparable trough values and in general similar gene profile structure as well.

Here, for all the gene profile group, 100 genes are simulated. So there are no dominating gene profile in this simulated gene profile group. Equal number of genes are simulated for every gene profile to maintain the homogeneity of the gene profiles withing the gene profile group.

Gene Profile	Number of Genes
Flat (0)	100
Up Down Max 2 (2)	100
Up Down Max 3 (3)	100
Up Down Max 4 (4)	100
Up Down Max 5 (5)	100
Increasing (8)	100
Down Up Min 2 (9)	100
Down Up Min 3 (10)	100
Down Up Min 4 (11)	100
Down Up Min 5 (12)	100

Table 7: Genes profile group 2 with gene profile, its true cluster level and number of genes simulated for each profile.

4.2.3 Gene Profile Group 3

Table 8 show the details in tabular form, Totaling 14 distinct gene profiles which are simulated for gene profile group 3.

Gene Profile	Number of Genes
Decreasing (1)	100
Up down max 2 (2)	75
Up down max 3 (3)	75
Up down max 4 (4)	75
Up down max 5 (5)	75
Up down max 6 (6)	50
Up down max 7 (7)	50
Increasing (8)	100
Down up min 2 (9)	75
Down up min 3 (10)	75
Down up min 4 (11)	75
Down up min 5 (12)	75
Down up min 6 (13)	50
Down up min 7 (14)	50

Table 8: Genes profile group 3 with gene profile, its true cluster level and number of genes simulated for each profile.

Here also, number of genes are simulated for each gene profile in such a way that homogeneity within each gene profile is maintained. However, more number of genes are simulated for both decreasing and increasing profile, making these profiles dominant.

Gene Profile Group	Dataset Parameter	Num. Simulated Dataset
Gene Profile Group 1	Std. Dev = 1, Peak Val = 8	500
Gene Profile Group 1	Std. Dev = 2, Peak Val = 8	500
Gene Profile Group 1	Std. Dev = 5, Peak Val = 8	500
Gene Profile Group 2	Std. Dev = 1, Peak Val = 8	500
Gene Profile Group 2	Std. Dev = 2, Peak Val = 8	500
Gene Profile Group 2	Std. Dev = 5, Peak Val = 8	500
Gene Profile Group 3	Std. Dev = 1, Peak Val = 8	500
Gene Profile Group 3	Std. Dev = 2, Peak Val = 8	500
Gene Profile Group 3	Std. Dev = 5, Peak Val = 8	500

Table 9: Data simulation setup summary.

Lastly, Table 9 provides an overview of the data simulation setup, detailing various combinations of gene profile groups and dataset standard deviations. For each gene profile group and for each dataset with different data parameters, 500 distinct datasets are simulated. Each datasets are distinct and different from each other because random noise are incorporated at each gene expression value every time. In total, 4500 datasets, each of size 1000 rows and 27 columns are simulated for this thesis.

4.3 Application of different clustering algorithms on simulated dataset

This subsection examines the application of several clustering algorithms—including k-means, hierarchical clustering, DBSCAN, OPTICS, and two variants of the ORICC algorithm (ORICC1 and ORICC2)—across all simulated datasets. The performance of these algorithms are assessed using the Adjusted Rand Index (ARI), and the results are summarized with a boxplot showing the ARI values for 500 datasets.

Since determining the optimal k-value or cut-tree height from the elbow plot is impractical due to the variability across 500 distinct datasets, two distinct functions were developed to identify the optimal k-value and cut-tree height for each dataset. These functions require the user to input a possible range of k-values or cut-tree heights. Given the prior knowledge of the optimal k-value and cut-tree height for the selected VPA dataset (which were 2, 4, and 6), a range of values from 2 to 10 was provided

for these functions, as the simulated datasets exhibited similar characteristics to the original VPA dataset (based on the top 1,000 most variable genes). For each possible k-value or cut-tree height, K-Means or hierarchical clustering algorithms were applied to the dataset, and the corresponding average silhouette coefficient of the resultant clusters was calculated. The value of k or cut-tree height with the highest average silhouette coefficient was considered optimal.

4.3.1 Gene Profile Group 1

Dataset with standard deviation 1

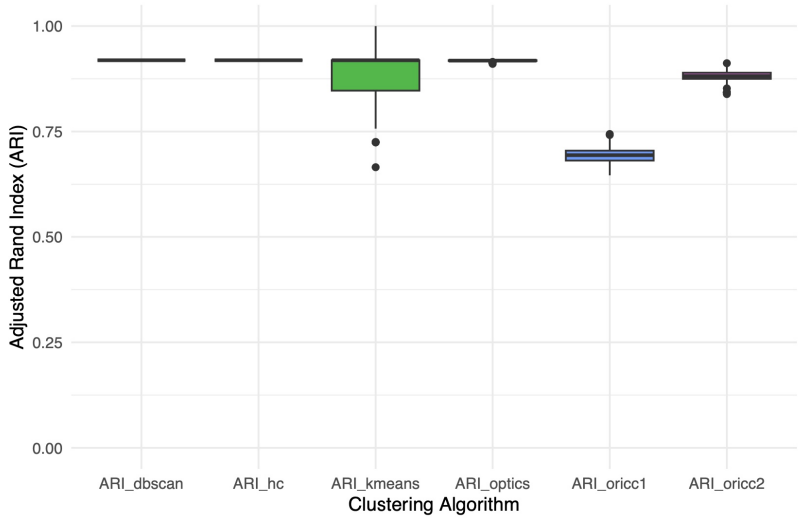


Figure 12: ARI value of 500 simulations for gene profile group 1 standard deviation 1.

Figure 12 presents the Adjusted Rand Index (ARI) results of applying the clustering algorithms on the 500 simulated datasets of gene profile group 1 with standard deviation of 1. Overall, the performance of nearly all algorithms is quite well, as indicated by the ARI values approaching 1 in the boxplots. However, the k-means algorithm shows slightly more variability in its ARI values, resulting in a boxplot with a wider interquartile range (IQR) and a larger lower quartile value. In comparison, ORICC1 exhibits lower overall performance across the datasets but still achieves an ARI close to 0.75, which is generally considered a good result.

To evaluate the accuracy of the k-value or cut-tree height predictions made by the user-defined functions based on the silhouette coefficient, compared to the true number of clusters (which is 8, as indicated in Table 6), barplots are generated to show the

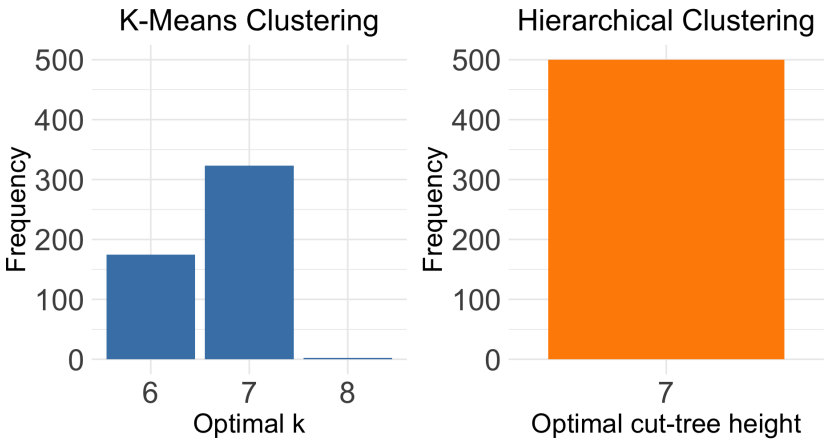


Figure 13: Frequency distribution of optimal parameter values across 500 simulations for two clustering algorithms based on silhouette coefficient.

frequency of selected k-values and cut-tree heights from the 500 simulation. Figure 13 displays the barplots for the same. It is evident that, out of 500 simulations, the k-value of 7 was chosen 325 times, while a k-value of 6 was selected around 173 times, and a k-value of 8 was chosen about 2 times. For hierarchical clustering, the cut-tree height was consistently selected as 7 in every case. These selections are reasonably close to the true number of clusters, especially for hierarchical clustering, which is why it performed well overall, as reflected by the ARI boxplot approaching the perfect value of 1. In contrast, k-means exhibited some fluctuation in performance due to the fact that nearly 37% of the cases involved a k-value of 6, which is slightly less than the actual number of clusters, leading to a relatively larger lower IQR in its performance.

	1	2	3	4	5	6	7
0	0	0	0	0	200	0	0
1	0	0	0	0	0	0	150
2	0	100	0	0	0	0	0
3	0	100	0	0	0	0	0
6	0	0	0	100	0	0	0
8	72	0	0	0	0	78	0
10	0	0	100	0	0	0	0
13	0	0	100	0	0	0	0

(a). K-means algorithm.

	1	2	3	4	5	6	7
0	200	0	0	0	0	0	0
1	0	0	150	0	0	0	0
2	0	0	0	0	0	100	0
3	0	0	0	0	0	100	0
6	0	0	0	100	0	0	0
8	0	150	0	0	0	78	0
10	0	0	0	0	0	0	100
13	0	0	0	0	100	0	0

(b). Hierarchical clustering algorithm.

Table 10: Confusion Matrix of true vs. predicted cluster level for gene profile group 1 with standard deviation 1.

Tables 10(a) and 10(b) shows the confusion matrices for the k-means and hierarchical clustering algorithms, respectively, applied to one of the simulated datasets. These matrices facilitate a comparison between the true and predicted cluster assignments.

In the confusion matrices, the predicted clusters are listed column-wise, while the true clusters are listed row-wise. The true cluster level of each gene profile for gene profile group 1 are detailed already in Table 6 in bracket beside each gene profile name.

From the confusion matrix of the k-means algorithm shown in Table 10(a), it is clear that certain gene profile groups—such as Flat (true cluster: 0, predicted cluster: 5), Decreasing (true cluster: 1, predicted cluster: 7), and Up Down Max 6 (true cluster: 6, predicted cluster: 4)—are accurately predicted. However, the Up Down Max 2 (true cluster: 2) and Up Down Max 3 (true cluster: 3) profiles are combined into a single cluster (predicted cluster level: 2). Similarly, the Down Up Min 3 (true cluster: 10) and Down Up Min 6 (true cluster: 13) profiles are merged into another single cluster (predicted cluster level: 3). Additionally, the Increasing profile group with 150 genes is divided into two clusters with sizes 72 and 78 (predicted levels 1 and 6). A similar pattern is observed in the confusion matrix for the hierarchical clustering algorithm. This merging of profiles is likely due to the proximity of peak values among the groups (e.g., Up Down Max 2, Up Down Max 3 and Down Up Min 3, Down Up Min 6), resulting in their combination into larger clusters.

The boxplot shown in Figure 12 indicates that ORICC2 performs better than ORICC1. Furthermore, Boxplot 33(a) in the appendix reveals that the median number of genes rejected by ORICC2 for the dataset with standard deviation 1 is 200. This is consistent with Table 6, which lists 200 genes simulated as **Flat Genes** for this gene profile group (1). Figure 34 in the Appendix illustrates a comparative analysis of the number of flat genes simulated for the dataset **at a randomly selected simulation run** for different gene profile groups, and the extent to which ORICC2 rejected these genes during the pre-selection stage due to their "uninformative flatness". It also shows the number of genes rejected from other profiles, which were not simulated as flat genes. In Figure 34(a), the bars corresponding to a standard deviation of 1 clearly indicate that ORICC2 exclusively discards genes simulated as flat, as no bars are present for rejected genes (from other profiles).

Dataset with standard deviation 2

Figure 14 displays the ARI values obtained from 500 simulations using gene profile group 1 with a dataset standard deviation of 2. The performance of all clustering algorithms appears to have declined compared to the previous scenario. Nonetheless, k-means and hierarchical clustering still outperform the other algorithms.

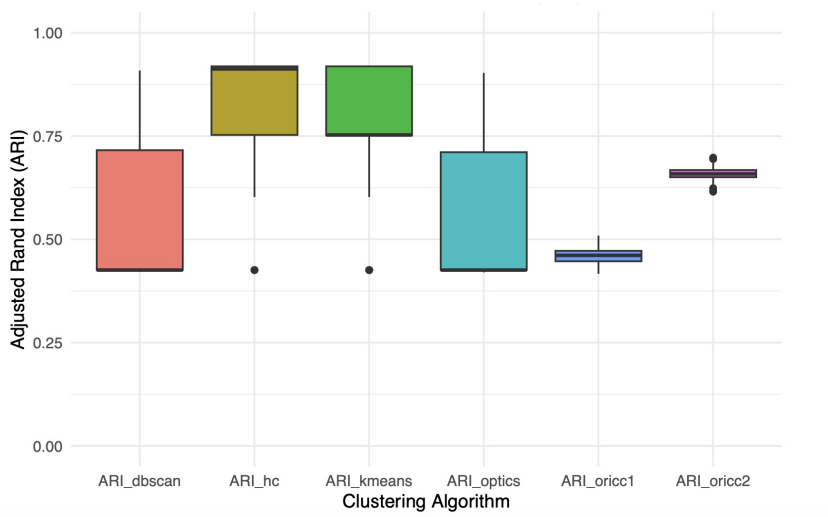


Figure 14: ARI value of 500 simulations for gene profile group 1 standard deviation 2.

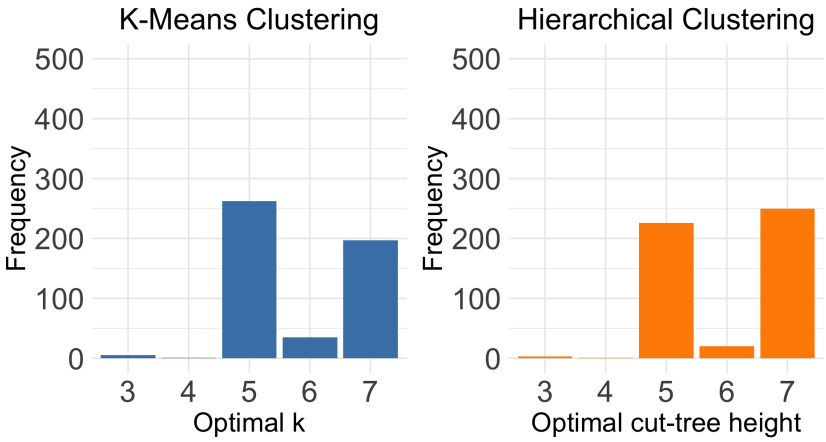


Figure 15: Frequency distribution of optimal parameter values across 500 simulations for two clustering algorithms based on silhouette coefficient.

Figure 15, which illustrates the selected k-values and cut-tree heights for the simulated datasets across 500 simulations, shows that in most cases, the k-values and cut-tree heights for the k-means and hierarchical clustering algorithms are typically chosen as 5 or 7.

From the confusion matrices for true vs. predicted clusters shown in Table 11(a) and 11(b), where the k-value and cut-tree height are set to 7, it is evident that gene profiles from true clusters 2 and 3 have been combined into a larger cluster. This could be due to the same reason as before: since these two clusters have peak values that are very close to each other, they likely do not have a significant distance between these 2 gene

	1	2	3	4	5	6	7
0	0	0	0	0	0	200	0
1	149	0	0	1	0	0	0
2	0	0	0	0	0	0	100
3	0	0	0	0	0	0	100
6	0	0	100	0	0	0	0
8	0	150	0	0	0	0	0
10	0	0	0	0	100	0	0
13	0	0	0	100	0	0	0

(a). K-means algorithm.

	1	2	3	4	5	6	7
0	200	0	0	0	0	0	0
1	0	0	150	0	0	0	0
2	0	0	0	0	0	100	0
3	0	0	0	0	0	100	0
6	0	0	0	100	0	0	0
8	0	150	0	0	0	78	0
10	0	0	0	0	0	0	100
13	0	0	0	0	100	0	0

(b). Hierarchical clustering algorithm.

Table 11: Confusion Matrix of true vs. predicted cluster level for gene profile group 1 with standard deviation 2.

profile. As a result, they were indistinguishable by distance-based clustering algorithms like k-means and hierarchical clustering.

The boxplot in Figure 14 again demonstrates that ORICC2 outperforms ORICC1. Additionally, the boxplot in Figure 33(a) in the appendix shows that the median number of genes rejected for the dataset with standard deviation 2 of gene profile group 1 by ORICC2 is 203, which is almost equal to the number of flat genes simulated for this gene profile group (refer to Table 6). Further bars in Figure 34(a), corresponding to a standard deviation of 2, indicate that out of 200 simulated flat genes, ORICC2 has rejected all of them as well as 3 genes from other gene profile groups during the pre-selection stage.

Dataset with standard deviation 5

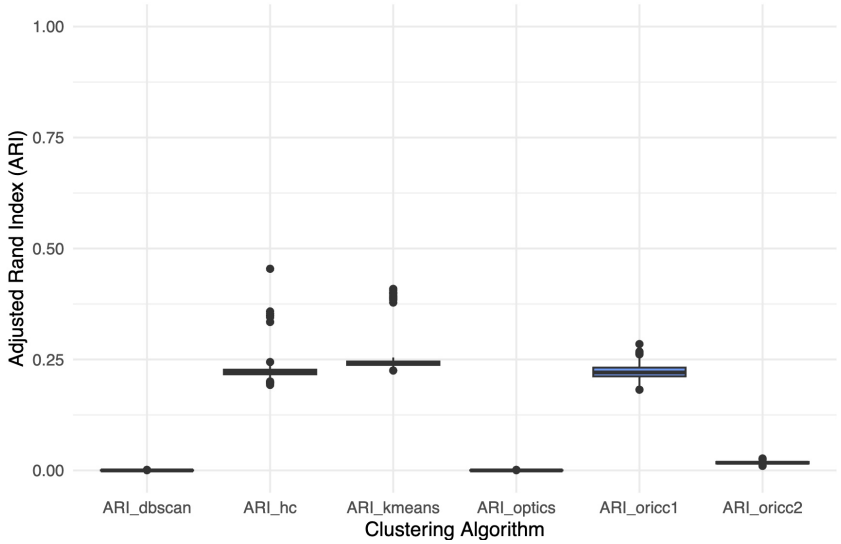


Figure 16: ARI value of 500 simulations for gene profile group 1 standard deviation 5.

Figure 16 presents the ARI results for all clustering algorithms applied to 500 simulated datasets from the same gene profile group but with a larger standard deviation of 5. In this setup, DBSCAN, OPTICS, and ORICC2 appear to have performed worse. Additionally, k-means and hierarchical clustering did not perform as well as in previous cases.

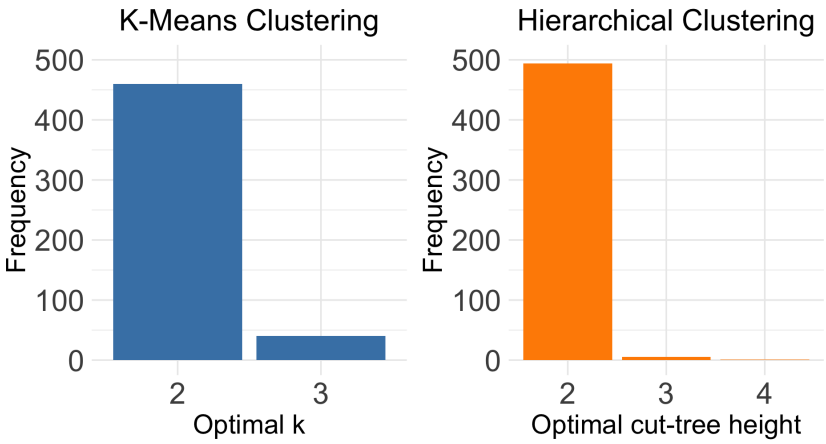


Figure 17: Frequency distribution of optimal parameter values across 500 simulations for two clustering algorithms based on silhouette coefficient.

To further investigate these two algorithms, Figure 17 presents a barplot showing the selected k-values and cut-tree heights for each data simulation. It is noticeable that in most simulations, the optimal k-value or cut-tree height was selected as 2, whereas the actual number of distinct clusters is 8. This discrepancy is the primary reason for the failure of these two algorithms.

	1	2
0	1	199
1	10	140
2	96	4
3	97	3
6	99	1
8	148	2
10	2	98
13	0	100

(a). K-means algorithm.

	1	2
0	200	0
1	127	23
2	6	94
3	8	92
6	4	96
8	10	140
10	97	3
13	100	0

(b). Hierarchical clustering algorithm.

Table 12: Confusion Matrix of true vs. predicted cluster level for gene profile group 1 with Standard deviation 5.

Further investigation was conducted by examining the confusion matrices from one of the 500 simulated datasets. Figures 12(a) and 12(b) show snippets of the true vs. predicted cluster levels when the optimal k-value or cut-tree height was set to 2 for the k-means

and hierarchical clustering algorithms, respectively. From both confusion matrices, it is clear that the gene profiles—Flat (true cluster level: 0), Decreasing (true cluster level: 1), Down Up Min 3 (true cluster level: 10), and Down Up Min 6 (true cluster level: 13)—were all combined into a single large cluster. Similarly, other gene profiles such as Increasing (true cluster level: 8), Up Down Max 2 (true cluster level: 2), and Up Down Max 3 (true cluster level: 3) were merged into another large cluster. This suggests that due to the high standard deviation in these datasets, the distance-based clustering algorithms (k-means and hierarchical clustering) were only able to differentiate whether a gene profile was generally increasing or decreasing. They failed to distinguish clusters based on more nuanced features, such as peak or trough values.

The boxplot in Figure 16 confirms that ORICC2 performed poorly under these conditions. The boxplot for standard deviation 5 in Figure 33(a) in the appendix reveals that approximately 850 out of 1000 genes were rejected by ORICC2 in the pre-selection step during each simulation. As a result, instead of being applied to 1000 genes, ORICC2 was effectively only used on around 150 genes.

Bars in Figure 34(a), corresponding to a standard deviation of 5, proves that out of 850 rejected genes, 650 were from gene profile groups that were not simulated as flat profiles. This substantial gene rejection could be due to the high standard deviation in the dataset, which caused other gene profiles to become nearly flat and thus were rejected by ORICC2 before the clustering algorithm was even applied. Consequently, this led to the overall poor performance of the algorithm on these datasets.

4.3.2 Gene Profile Group 2

Dataset with standard deviation 1

Similar steps were followed for gene profile group 2, whose profile structure and the number of simulated genes for each profile are shown in Table 7 earlier. A total of 10 distinct gene profiles were simulated in this gene group. However, it is evident from the boxplot Figure 18 that DBSCAN, OPTICS, k-means, and hierarchical clustering algorithms performed worse compared to ORICC1 and ORICC2.

Figure 19 shows that the optimal k-value and cut-tree height for the k-means and hierarchical clustering algorithms, as selected by the user-defined function based on the silhouette coefficient, is 4 in most cases, which is really less than the true number of cluster profiles, which is 10.

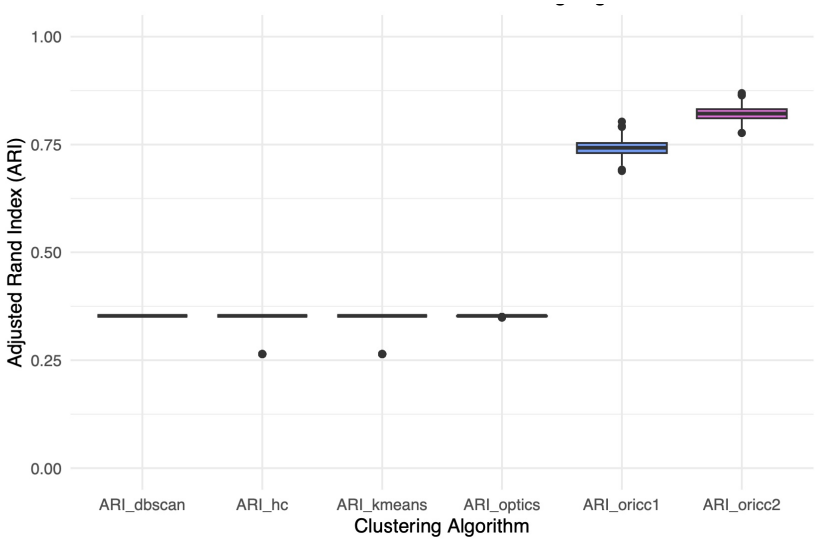


Figure 18: ARI value of 500 simulations for gene profile group 2 standard deviation 1.

	1	2	3	4
0	0	0	0	100
2	0	0	100	0
3	0	0	100	0
4	0	0	100	0
5	0	0	100	0
8	0	100	0	0
9	100	0	0	0
10	100	0	0	0
11	100	0	0	0
12	100	0	0	0

(a). K-means algorithm.

	1	2	3	4
0	100	0	0	0
2	0	0	100	0
3	0	0	100	0
4	0	0	100	0
5	0	0	100	0
8	0	100	0	0
9	0	0	0	100
10	0	0	0	100
11	0	0	0	100
12	0	0	0	100

(b). Hierarchical clustering algorithm.

Table 13: Confusion Matrix of true vs. predicted cluster level for gene profile group 2 with Standard deviation 1.

Tables 13(a) and 13(b) display the confusion matrices comparing true cluster levels with predicted cluster levels for one of the 500 simulated datasets. It appears that both algorithms successfully identified the Flat gene profile (true cluster level: 0) and the Increasing gene profile (true cluster level: 8) as expected. However, all other increasing-decreasing profiles, namely Up Down Max 2 (true cluster level: 2), Up Down Max 3 (true cluster level: 3), Up Down Max 4 (true cluster level: 4), and Up Down Max 5 (true cluster level: 5), were merged into one large cluster. Similarly, the decreasing-increasing gene profiles, including Down Up Min 2 (true cluster level: 9), Down Up Min 3 (true cluster level: 10), Down Up Min 4 (true cluster level: 11), and Down Up Min 5 (true cluster level: 12), were grouped into another large cluster after applying the algorithms, which was not the expected outcome.

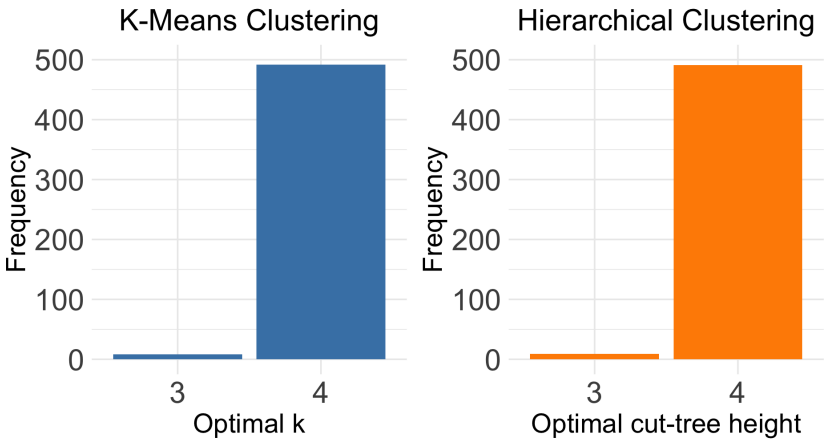


Figure 19: Frequency distribution of optimal parameter values across 500 simulations for two clustering algorithms based on silhouette coefficient.

In this scenario, ORICC2 performed slightly better than ORICC1. Figure 33(b) in the appendix shows that the median number of genes rejected by ORICC2 during the pre-selection step across simulations with standard deviation 1 is approximately 99. Table 7 also indicated earlier that 100 genes were simulated as **Flat Genes**. In Figure 34(b), the bars for a standard deviation of 1 show that, for that dataset, all 100 rejected genes were from mulated "flat" genes.

Dataset with standard deviation 2

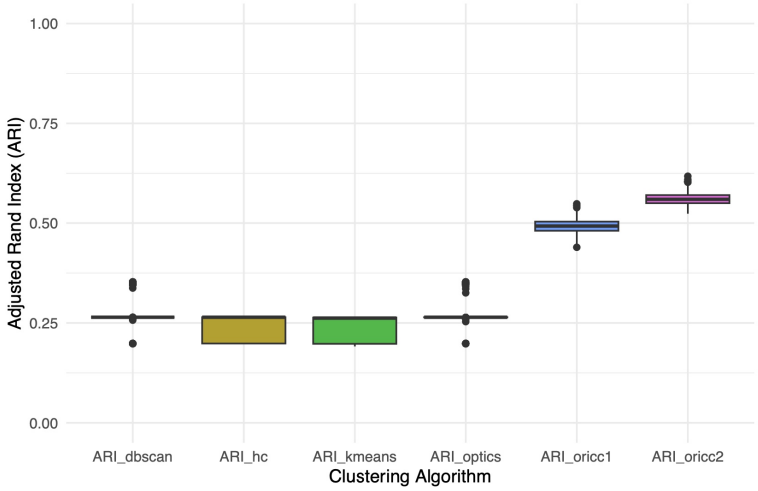


Figure 20: ARI value of 500 simulations for gene profile group 2 standard deviation 2.

Figure 20 presents the ARI scores for all clustering algorithms across 500 simulated datasets of gene profile group 2 with a standard deviation of 2. Overall, a decrease

in performance is observed, as indicated by the lower ARI values for all the clustering algorithms compared to the performance of all the clustering algorithms on the same gene profile group (2) with less standard deviation 1.

Figure 21: Frequency distribution of optimal parameter values across 500 simulations for two clustering algorithms based on silhouette coefficient.

	1	2	3
0	0	0	100
2	100	0	0
3	100	0	0
4	100	0	0
5	100	0	0
8	97	3	0
9	0	100	0
10	0	100	0
11	0	100	0
12	0	100	0

(a). K-means algorithm.

	1	2	3
0	100	0	0
2	0	100	0
3	0	100	0
4	0	100	0
5	0	100	0
8	0	100	0
9	0	0	100
10	0	0	100
11	0	0	100
12	0	0	100

(b). Hierarchical clustering algorithm.

Table 14: Confusion Matrix of true vs. predicted cluster level for gene profile group 2 with Standard deviation 2.

Figure 21 shows the optimal k-values and cut-tree heights selected for the k-means and hierarchical clustering algorithms across the datasets. For 325 datasets, the optimal values were chosen as 3, while for about 175 datasets, the values were set to 2.

The confusion matrices for one of the datasets, shown in Table 14(a) and 14(b), reveal similar behavior to previous observations. However, in this case, genes with Increasing profiles (true cluster level: 8) are merged with all other increasing-decreasing gene profiles for both k-means and hierarchical clustering algorithms.

ORICC2 also outperformed ORICC1 in this scenario. Additionally, the boxplot of standadrd deviation 2 in Figure 33(b) from the appendix shows that approximately 110 out

of 1000 genes are rejected by ORICC2 in the pre-selection step for each simulation, which is close to the number of actual simulated flat profiled genes (100). Bars representing a standard deviation of 2 in Figure 34(b) in the Appendix demonstrate that, for that simulation, ORICC2 rejected total approximately 103 gene where 99 out of 100 were flat genes, and approximately 4 genes that did not exhibit a flat profile were rejected.

Dataset with standard deviation 5

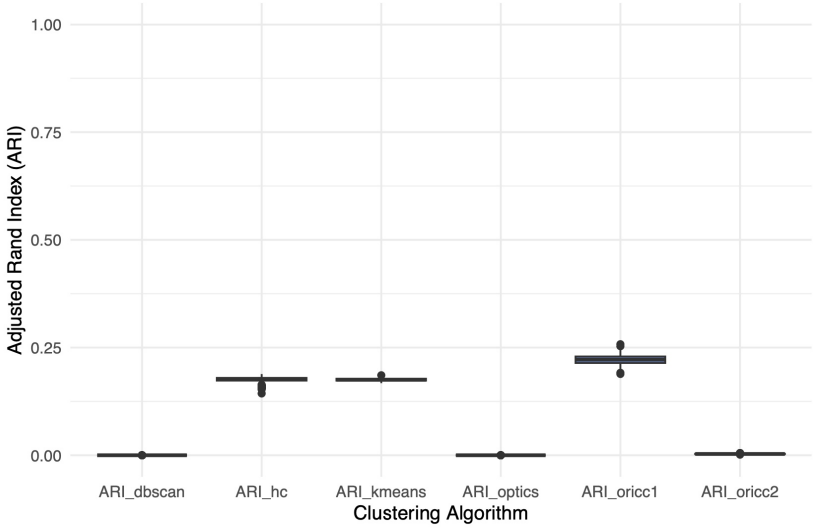


Figure 22: ARI value of 500 simulations for gene profile group 2 standard deviation 5.

The clustering algorithms were again applied to datasets with a relatively large standard deviation of 5. Figure 22 shows the performance of these algorithms across all simulated datasets. It is evident that DBSCAN, OPTICS, and ORICC2 performed poorly, with ARI values close to 0. In contrast, k-means, hierarchical clustering and ORICC1 algorithms demonstrated better performance compared to the other algorithms.

	1	2
0	97	3
2	0	100
3	1	99
4	0	100
5	1	99
8	25	75
9	100	0
10	100	0
11	100	0
12	100	0

(a). K-means algorithm.

	1	2
0	99	1
2	1	99
3	0	100
4	1	99
5	2	98
8	16	84
9	96	4
10	95	5
11	98	2
12	99	1

(b). Hierarchical clustering algorithm.

Table 15: Confusion Matrix of true vs. predicted cluster level for gene profile group 2 with Standard deviation 5.

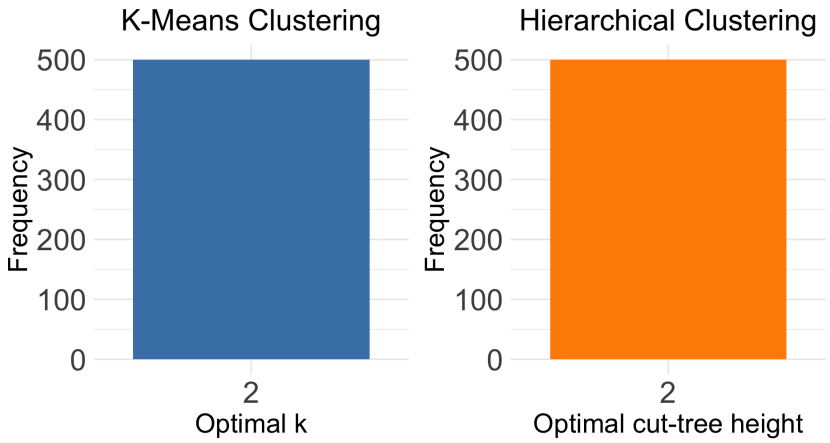


Figure 23: Frequency distribution of optimal parameter values across 500 simulations for two clustering algorithms based on silhouette coefficient.

However, Figure 23 reveals that the optimal k-value and cut-tree height were incorrectly selected for all simulated datasets. Instead of the true number of clusters of 10, the selected value was 2. Consequently, many clusters were combined into a single large cluster, as illustrated in the confusion matrices shown in Tables 15(a) and 15(b). Further analysis of the confusion matrices indicates that these distance-based algorithms can only detect the basic structure of the gene profiles, such as whether they are increasing-decreasing or decreasing-increasing. They are unable to identify clusters based on peak or trough values. Additionally, Flat profiles are merged with increasing-decreasing profile groups due to the presence of random noise.

Similar to gene profile group 1, for datasets with a standard deviation of 5, ORICC2 also performed poorly in gene profile group 2 due to rejection of lots of genes at the pre-selection stage of ORICC2. This is further supported by the boxplot in Figure 33(b) for standard deviation of 5, which shows that approximately 830 out of 1000 genes were rejected by ORICC2 in each simulation. This rejection rate is significantly higher than the number of simulated flat genes, which is 100 (see Table 7). The high standard deviation in the dataset likely resulted in most genes being sparse, regardless of the profiles simulated, causing many gene profiles to appear flat and leading ORICC2 to reject them during the pre-selection step. This is supported by the bars representing a standard deviation of 5 in Figure 34(b) in the Appendix. It is evident that, out of 850 rejected genes, 97 were simulated as flat genes, while the remaining genes were from other gene profiles.

4.3.3 Gene Profile Group 3

Dataset with standard deviation 1

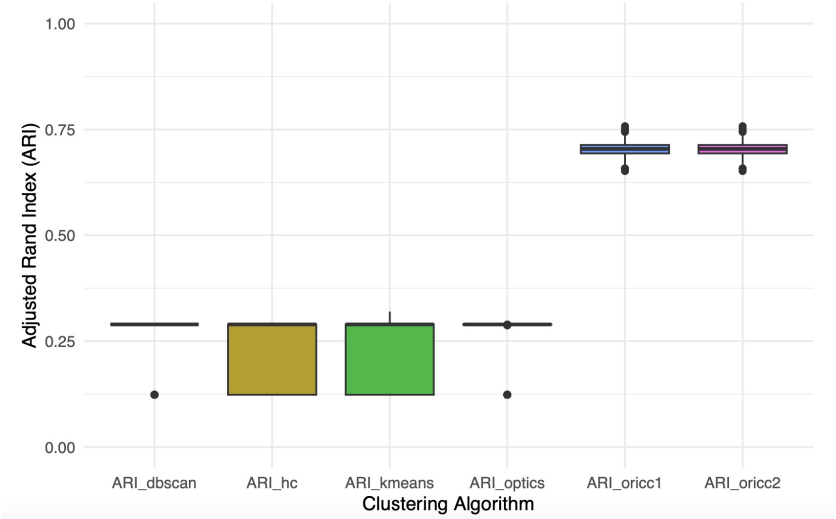


Figure 24: ARI value of 500 simulations for gene profile group 3 standard deviation 1.

All clustering algorithms were applied to 500 simulated datasets from Gene Profile Group 3, which includes 14 distinct gene profiles. Details (gene profile name, true cluster level, number of genes simulated at each gene profile) has already been discussed in previous section which is also summarized at Table 8. Despite the datasets having a standard deviation of 1, only ORICC1 and ORICC2 performed well, while the other clustering algorithms did not show satisfactory performance.

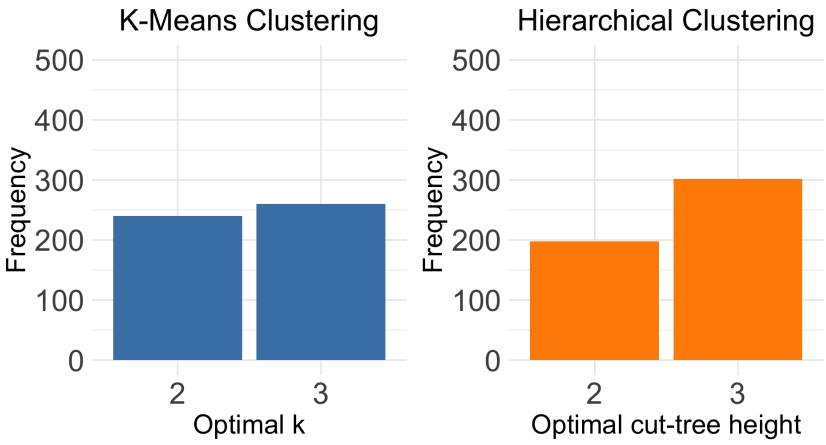


Figure 25: Selected Optimal k value based on Silhouette Coeffiecent

The poor performance of the k-means and hierarchical clustering algorithms can be explained by examining the barplot in Figure 25, which shows the chosen k-values and cut-tree heights for these algorithms. Instead of the actual number of clusters, which is 14, the optimal k-value and cut-tree height were frequently selected as 2 or 4.

	1	2	3	4
1	0	0	0	100
2	0	0	75	0
3	0	0	75	0
4	0	0	75	0
5	0	0	75	0
6	25	0	25	0
7	24	25	0	1
8	0	100	0	0
9	75	0	0	0
10	75	0	0	0
11	75	0	0	0
12	50	0	0	0
13	24	0	25	1
14	0	25	0	25

(a). K-means algorithm.

	1	2	3	4
1	0	0	0	100
2	0	75	0	0
3	0	75	0	0
4	0	75	0	0
5	0	75	0	0
6	0	25	25	0
7	25	0	25	0
8	100	0	0	0
9	0	0	75	0
10	0	0	75	0
11	0	0	75	0
12	0	0	50	0
13	0	25	25	0
14	25	0	0	50

(b). Hierarchical clustering algorithm.

Table 16: Confusion Matrix of true vs. predicted cluster level for gene profile group 3 with standard deviation 1.

The confusion matrices for k-means and hierarchical clustering algorithms, shown in Table 16(a) and 16(b), reveal that strictly increasing and decreasing gene profiles were correctly assigned to two separate clusters. However, all up-down gene profile groups were combined into a single large cluster, as were the down-up gene profile groups, for both k-means and hierarchical clustering.

In contrast, ORICC1 and ORICC2 performed exceptionally well with median ARI value of approximately 0.74 for every data simulation, compared to the other clustering algorithms. The boxplot in Figure 33(c) for standard deviation 1 further indicates that no genes were rejected by ORICC2 during the pre-selection process, which is expected since no flat genes were simulated for gene profile group 3 (refer to Table 8).

Dataset with standard deviation 2

Figure 26 illustrates the performance of clustering algorithms on 500 simulated datasets with a standard deviation of 2. It is evident that the ARI values for DBSCAN, hierarchical clustering, k-means, and OPTICS are around 1.8. In contrast, the ORICC1 and ORICC2 algorithms performed relatively better.

Examination of the barplot in Figure 27 reveals that the optimal k-value and cut-tree height for the dataset were consistently selected as 2 based on the silhouette coefficient, whereas the actual number of clusters is 14.

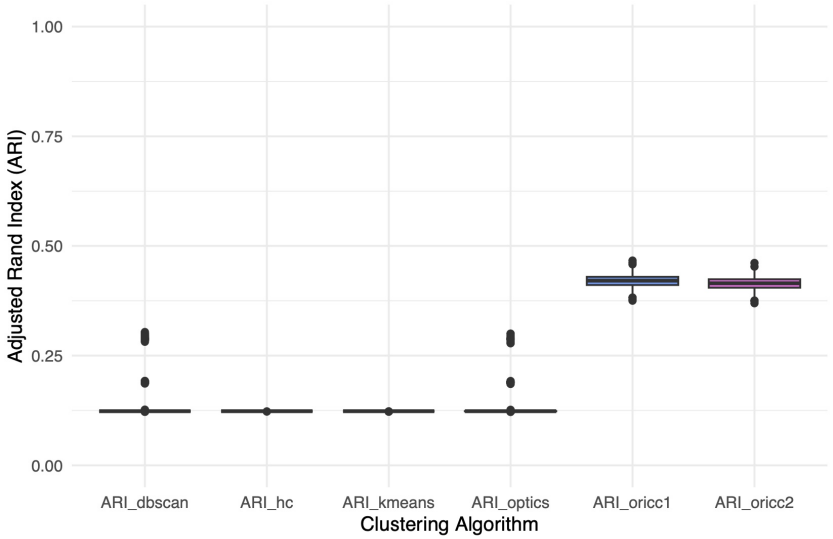


Figure 26: ARI value of 500 simulations for gene profile group 3 standard deviation 2.

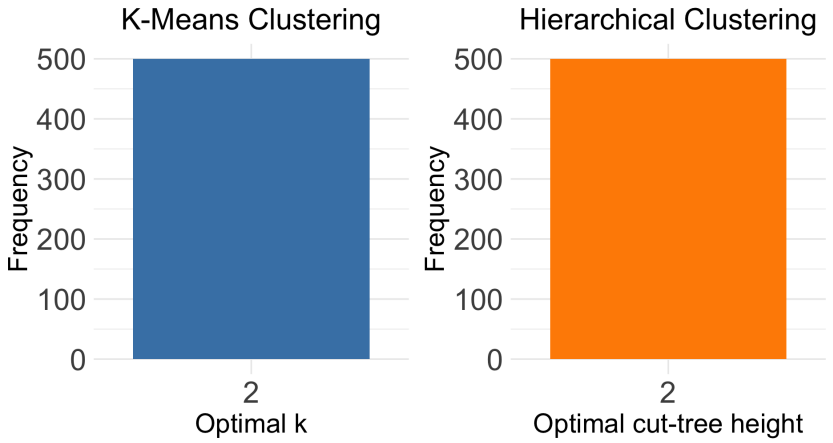


Figure 27: Frequency distribution of optimal parameter values across 500 simulations for two clustering algorithms based on silhouette coefficient.

The confusion matrices in Table 17(a) and 17(b) show that most of the clusters were combined into a single larger cluster, resulting in only 2 clusters being generated instead of the true 14 clusters. For both clustering algorithms, nearly all increasing genes (true cluster level: 8), as well as Up Down Max 2 (true cluster level: 2), Up Down Max 3 (true cluster level: 3), Up Down Max 4 (true cluster level: 4), Up Down Max 5 (true cluster level: 5), and about half of the genes from Up Down Max 6 (true cluster level: 6) and Up Down Max 7 (true cluster level: 7), were merged into a single large cluster (predicted cluster level: 1). The remaining genes were assigned to another cluster (predicted cluster level: 2).

	1	2
1	0	100
2	75	0
3	75	0
4	75	0
5	75	0
6	25	25
7	25	25
8	100	0
9	0	75
10	0	75
11	0	75
12	0	50
13	25	25
14	25	50

(a). K-means algorithm.

	1	2
1	0	100
2	75	0
3	75	0
4	75	0
5	75	0
6	25	25
7	25	25
8	100	0
9	0	75
10	0	75
11	0	75
12	0	50
13	25	25
14	25	50

(b). Hierarchical clustering algorithm.

Table 17: Confusion Matrix of true vs. predicted cluster level for gene profile group 3 with Standard deviation 2.

ORICC1 and ORICC2 both performed similarly, with nearly identical ARI values across the 500 simulated datasets. However, the boxplot in Figure 33(c) shows that approximately 15 genes were discarded by ORICC2 during the pre-selection stage, despite the simulated dataset containing no flat gene profiles. This rejection might be due to the slightly higher standard deviation of 2, which could have made some "non flat" genes appear flat.

Dataset with standard deviation 5

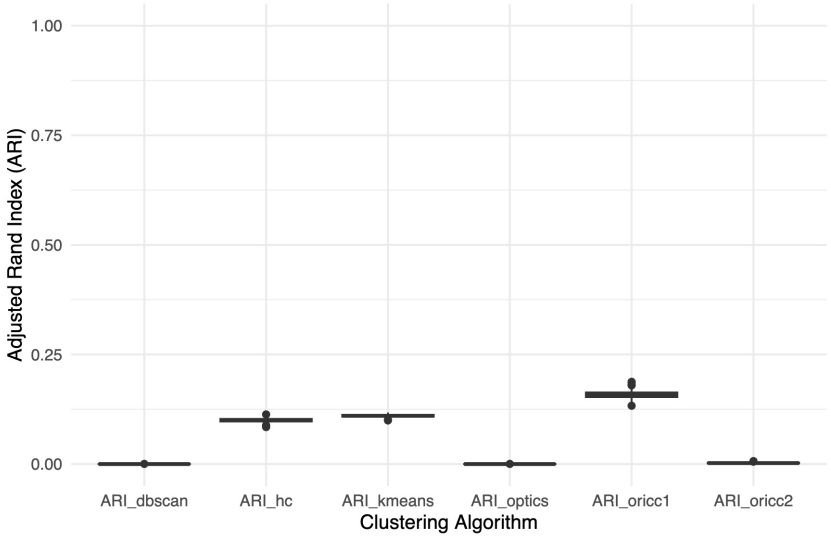


Figure 28: ARI value of 500 simulations for gene profile group 3 standard deviation 5.

The clustering algorithms were applied to the simulated datasets with a standard deviation of 5. According to the ARI values shown in Figure 28, DBSCAN, OPTICS, and ORICC2 have ARI value of 0.

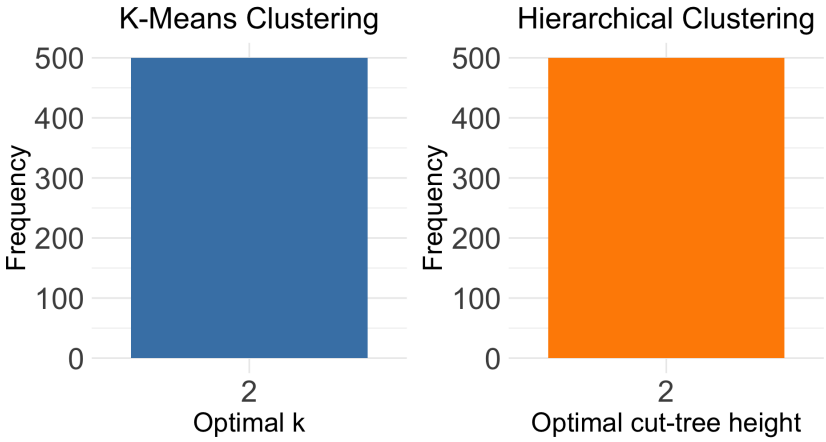


Figure 29: Frequency distribution of optimal parameter values across 500 simulations for two clustering algorithms based on silhouette coefficient.

	1	2
1	6	94
2	74	1
3	75	0
4	75	0
5	74	1
6	25	25
7	25	25
8	92	8
9	4	71
10	0	75
11	0	75
12	0	50
13	25	25
14	26	49

(a). K-means algorithm.

	1	2
1	5	95
2	70	5
3	75	0
4	74	1
5	73	2
6	25	25
7	25	25
8	90	10
9	6	69
10	0	75
11	0	75
12	3	47
13	25	25
14	26	49

(b). Hierarchical clustering algorithm.

Table 18: Confusion Matrix of true vs. predicted cluster level for gene profile group 3 with Standard deviation 5.

From the barplot in Figure 29 and the confusion matrices for k-means and hierarchical clustering algorithms shown in Figures 18(a) and 18(b), it is apparent that the same issues persist. Both k-means and hierarchical clustering algorithms performed poorly due to incorrect assignments of the initial k-value or cut-tree height.

Similarly, ORICC2 also performed poorly. The boxplot in Figure 33(c) from the additional figures section shows that approximately 820 out of 1000 genes were discarded by ORICC2 during the pre-selection stage, despite there being no flat genes simulated

in this gene profile group. Figure 34(c) in the Appendix also shows that all the genes rejected for a standard deviation of 5, for that dataset originate from gene profiles that were not simulated as flat profiles. This high rejection rate is likely due to the high variation in the dataset, which caused many gene profiles to be considered flat even though they were not simulated as flat genes. Since more than 80% of the data were excluded from ORICC2, the overall performance of the algorithm was adversely affected.

4.4 Summary and Interpretation of the performance of clustering algorithm on simulated dataset

1. **K-means and Hierarchical clustering algorithm**:

   Both algorithms performed effectively when the gene profiles had limited variety (e.g., gene profile group 1). However, as the diversity of gene profiles increased in the dataset, the performance of these two distance-based clustering algorithms declined.

   Additionally, both algorithms demonstrated strong performance on datasets with low variance (e.g., standard deviations of 1 and 2). Conversely, they struggled to perform well on sparser datasets with higher standard deviation.

2. **DBSCAN and OPTICS**: These two density-based clustering algorithms performed well only on dense datasets with low standard deviation and limited variety in gene structure, such as gene profile group 1 with a standard deviation of 1. For all other datasets and gene profiles, these algorithms did not deliver the expected performance.

3. **ORICC1 and ORICC2**: The two functions of the ORICC algorithm, ORICC1 and ORICC2, were consistently effective across all datasets with varying standard deviations and gene profile groups. Their performance was particularly strong when gene profiles were more diverse (e.g., gene profile group 3), where other clustering algorithms struggled. Although ORICC2 generally outperformed ORICC1, it failed on sparse datasets which had high standard deviation (e.g., 5). The underperformance occurred because, during the pre-selection stage of ORICC2, more than 80% of the genes that were not simulated as "flat profiles" were also rejected due to the high variance in the data. This led to poor performance, despite the diversity in gene profiles.

5 Summary

In the rapidly evolving field of genomics, analyzing how genes respond to varying treatment doses are essential for advancing personalized medicine and targeted therapies. High-dimensional dose-response data present significant opportunities for biomedical discoveries, but their complexity requires sophisticated methods for effective interpretation. Clustering algorithms offer a powerful approach by grouping genes with similar expression patterns, revealing underlying biological processes and drug mechanisms. This technique helps identify biomarkers, infer drug actions, reduce data complexity, and guide experimental design, ultimately contributing to more precise and personalized medical treatments (Johnson et al., 2021, p. 90).

Various clustering algorithms, including k-means, hierarchical clustering, DBSCAN, OPTICS, and ORICC, were applied to the VPA dataset, focusing on the top 1,000 most variable genes. For distance-based clustering, optimal parameters for k-means and hierarchical clustering were determined using the elbow plot and silhouette coefficient, with k = 2 and cut-tree height = 4 identified as the best settings. In density-based clustering, DBSCAN and OPTICS were optimized based on silhouette scores, but challenges were observed in effectively clustering sparse data. DBSCAN produced less meaningful clusters, while clearer groupings were obtained with OPTICS using MinPts = 10. The ORICC algorithm, which does not require user-defined parameters, was used to cluster genes based on their structure (profile) effectively. Overall, ORICC was highlighted for its ease of use and ability to produce meaningful clustering results without parameter tuning.

Simulated datasets with different characteristics were generated to evaluate the consistency and reliability of clustering algorithms. Each dataset contained 1,000 rows (each row representing a synthetic gene) and 27 columns (representing combinations of 8 concentrations and their replicates). The synthetic gene expression data was generated using three methods, having normal distributions with different standard deviations (1, 2, and 5).

Three gene profile groups were created, each consisting of different gene profiles. The total number of profiles varied across these groups, and each group represented diverse combinations of gene structures.

For each gene profile group and each data generation method, 500 datasets were simulated, making a total of 4,500 datasets. All five clustering algorithms were then applied

to each simulated dataset, and a comparative study of their performance, measured by ARI values, was discussed. The performance of the clustering algorithms varied based on dataset characteristics. K-means and hierarchical clustering algorithms performed well with low variance and limited gene profile diversity but struggled as diversity and variance increased. DBSCAN and OPTICS were effective only in dense, low-variance datasets, performing poorly in other scenarios. ORICC1 and ORICC2 consistently performed well across different datasets, particularly with diverse gene profiles, though ORICC2 had difficulty with high-variance datasets due to the rejection of a large portion of genes in pre-selection step of that algorithm.

Challenges are encountered specifying input parameters for clustering algorithms. For k-means, the selection of the number of clusters (k) and the appropriate initial centroids is recognized as a difficult task. In hierarchical clustering, the challenge lies in choosing the correct distance metric. Similarly, for DBSCAN and OPTICS, the process of determining optimal values for ϵ and MinPts is acknowledged as problematic, especially as these density-based algorithms tend to perform poorly on high-dimensional data (Steinbach et al., 2004, p. 299-300).

To address these challenges, several improvements are recommended. Enhancements should be made to methods for determining optimal initial centroids in k-means. Additionally, the identification of effective distance metrics for various datasets in hierarchical clustering is necessary. Refinements are needed in the techniques for setting parameters such as ϵ and MinPts in DBSCAN and OPTICS.

In conclusion, although well-known clustering algorithms such as K-Means, hierarchical clustering, DBSCAN, and OPTICS performed adequately under certain dataset configurations, they exhibited limitations, particularly when dealing with sparse data or datasets that required clustering based on structural features rather than mere distance. In contrast, the ORICC algorithm demonstrated strong and consistent performance across varying datasets and gene structures (profiles). Unlike the other four algorithms, ORICC successfully clustered data based on inherent structural patterns rather than relying solely on distance as the primary clustering factor. Therefore, the ORICC algorithm is better suited for clustering scenarios where structural characteristics, such as those found in gene expression data, are the primary criteria for cluster formation.

Bibliography

Alboukadel Kassambara, F. M. (2020). *Extract and Visualize the Results of Multivariate Data Analyses*. R package version 1.0.7.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2).

Baptiste Auguie, A. A. (2017). *Miscellaneous Functions for "Grid" Graphics*. R package version 2.3.

Bishop, C. M. (2006). Pattern recognition and machine learning. *Springer google schola*, 2.

Brittain, H. K., Scott, R., and Thomas, E. (2017). The rise of the genome and personalised medicine. *Clinical Medicine*, 17(6).

Charlon, T. (2019). *OPTICS K-Xi Density-Based Clustering*. R package version 0.1.

Charrad, M., Ghazzali, N., Boiteau, V., and Niknafs, A. (2015). *Determining the Best Number of Clusters in a Data Set*. R package version 3.0.1.

Han, J., Pei, J., and Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann.

Ian, H. W. and Eibe, F. (2005). *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann Publishers.

Ian, H. W. and Eibe, F. (2011). *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann Publishers.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2).

Johnson, K. B., Wei, W.-Q., Weeraratne, D., Frisse, M. E., Misulis, K., Rhee, K., Zhao, J., and Snowdon, J. L. (2021). Precision medicine, ai, and the future of personalized health care. *Clinical and translational science*, 14(1).

Kassambara, A. (2017). *Practical guide to cluster analysis in R: Unsupervised machine learning*, volume 1. Sthda.

Kassambara, A. (2021). rstatix: pipe-friendly framework for basic statistical tests. r package version 0.7. 0. *Computer software]. https://CRAN. R-proje ct. org/packa ge= rstatix*.

Kassambara, A. and Kassambara, M. A. (2020). Package 'ggpubr'. *R package version 0.1*, 6(0).

Krug, A. K., Kolde, R., Gaspar, J. A., Rempel, E., Balmer, N. V., Meganathan, K., Vojnits, K., Baquié, M., Waldmann, T., Ensenat-Waser, R., et al. (2013). Human embryonic stem cell-derived test systems for developmental neurotoxicity: a transcriptomics approach. *Archives of toxicology*, 87.

Liu, T., Lin, N., Shi, N., and Zhang, B. (2009). Information criterion-based clustering with order-restricted candidate profiles in short time-course microarray experiments. *BMC bioinformatics*, 10:1–20.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K., Studer, M., Roudier, P., and Gonzalez, J. (2013). *cluster: "Finding Groups in Data": Cluster Analysis Extended Rousseeuw et*. R package version 2.1.0.

Michael Hahsler, Sunil Arya, D. M. (2024). *Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Related Algorithms*. R package version 1.2-0.

R Development Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. R version 4.1.2.

Reddy, C. K. (2018). *Data clustering: algorithms and applications*. Chapman and Hall/CRC.

Steinbach, M., Ertöz, L., and Kumar, V. (2004). The challenges of clustering high dimensional data. In *New directions in statistical physics: econophysics, bioinformatics, and pattern recognition*. Springer.

Team, R. C. and contributors worldwide (2020a). *The R Graphics Package*. R package version 4.0.3.

Team, R. C. and contributors worldwide (2020b). *The R Stats Package*. R package version 4.0.3.

Tianqing Liu, Nan Lin, N. S. and Zhang, B. (2009). *Order-restricted Information Criterion-based Clustering Algorithm*. R package version 1.0-1.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Wickham, H. (2020). *Flexibly Reshape Data: A Reboot of the Reshape Package*. R package version 1.4.4.

Wickham, H. (2021). *Tidy Messy Data*. R package version 1.1.3.

Wickham, H., François, R., Henry, L., and Müller, K. (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.8.

Appendix

A Additional figures

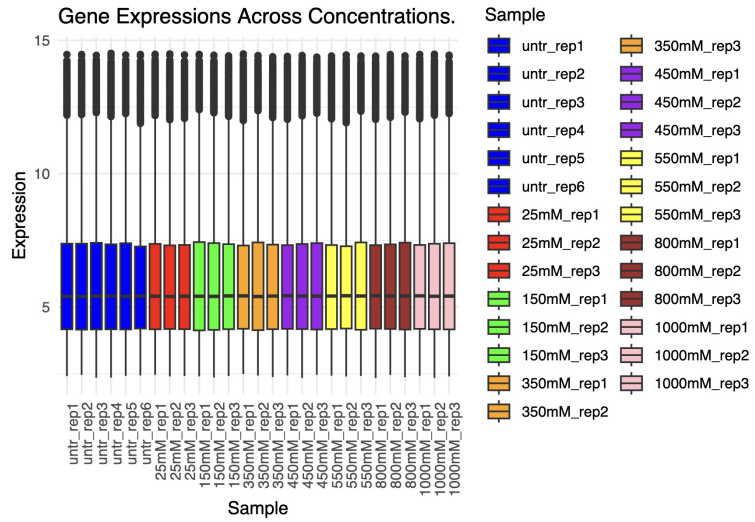


Figure 30: Boxplot of all the genes of VPA dataset.

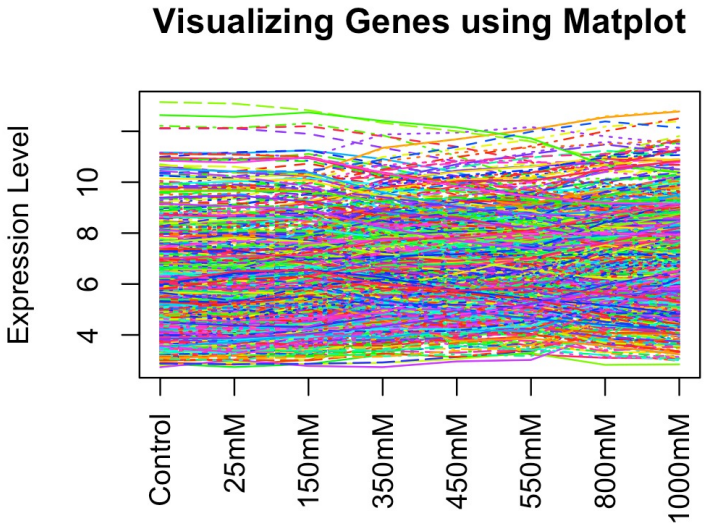


Figure 31: Matplot of top 1000 most variabled genes of VPA dataset.

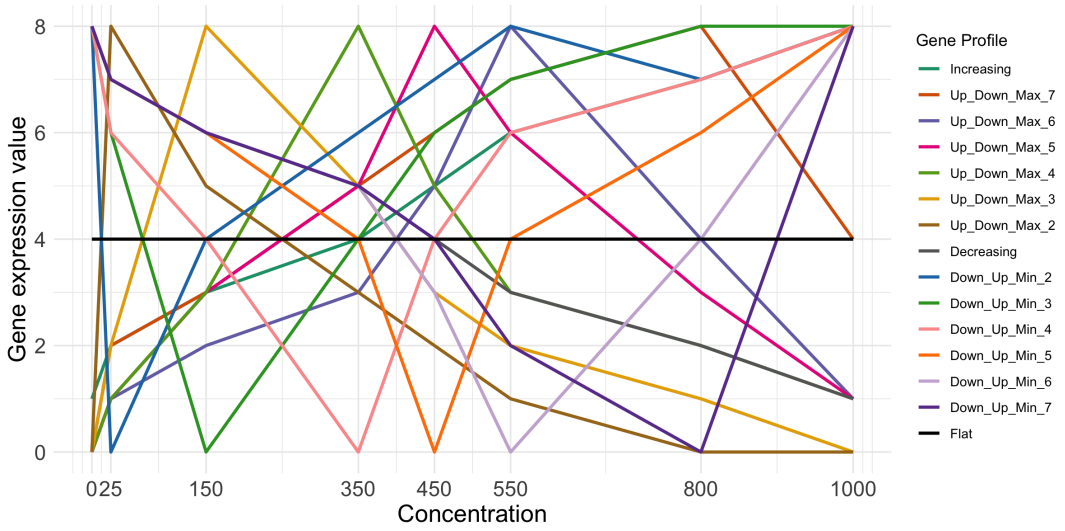
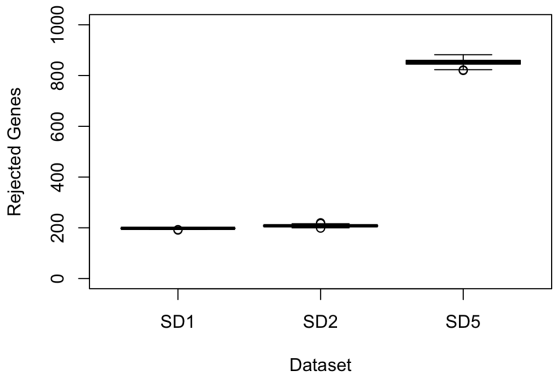
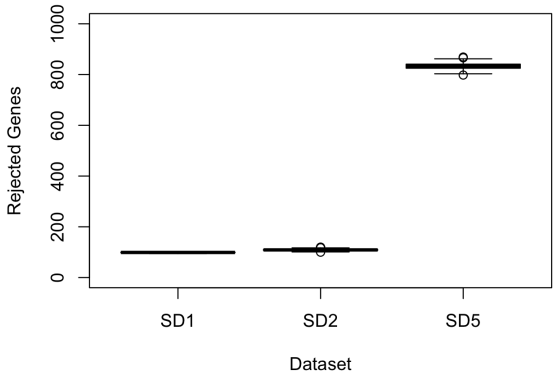


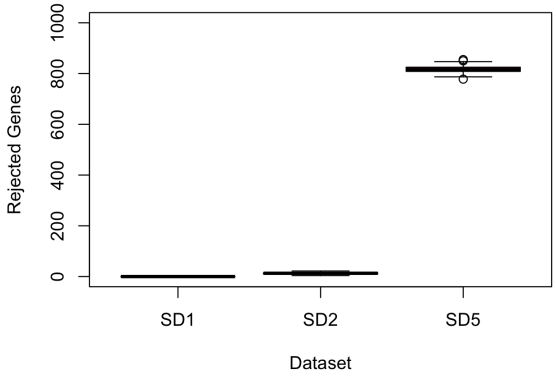
Figure 32: All simulated gene profiles.



(a) Gene profile group 1.

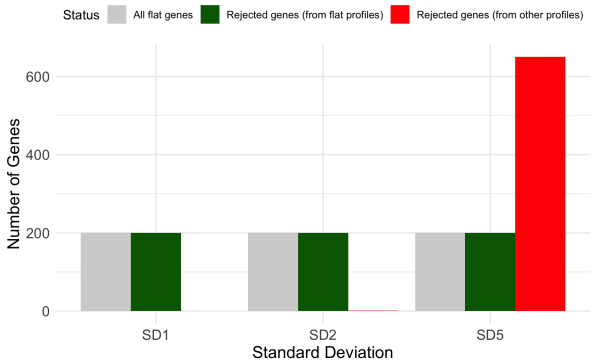


(b) Gene profile group 2.

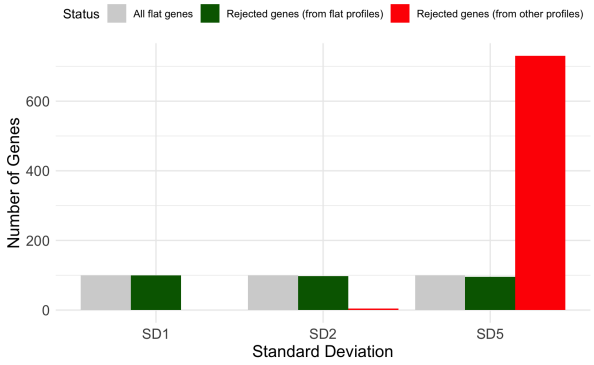


(c) Gene profile group 3.

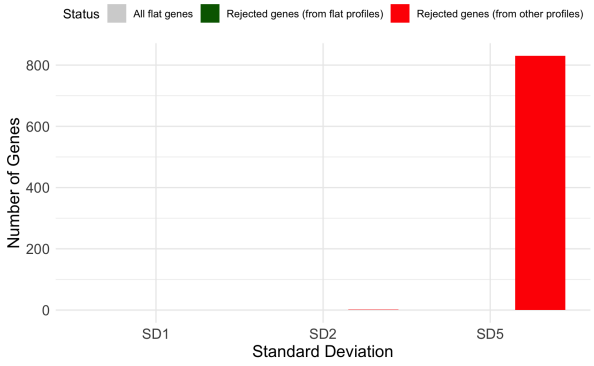
Figure 33: Rejected genes by ORICC2 at pre-selection stage for different gene profile groups with various standard deviations of the dataset.



(a) Genes profile group 1.



(b) Genes profile group 2.



(c) Genes profile group 3.

Figure 34: Gene Profiles of Rejected Genes during pre-selection stage by ORICC2 Across Different Dataset.