Principal Component Analysis and Clustering

Gioele La Manno

École Polytechnique Fédérale de Lausanne (EPFL)

School of Life Science (SV)

May 2025

EPFL - BMI - UPLAMANNO

Contents

PCA as Dimensionality Reduction

PCA enables dimensionality reduction by projecting data onto the first k principal components:

$$\mathbf{X}_c^{(k)} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$$

Where $\mathbf{U}_k$, $\mathbf{\Sigma}_k$, and $\mathbf{V}_k$ contain only the first k columns/entries.

The Eckart-Young theorem guarantees this is the best rank-k approximation in terms of minimizing the squared Frobenius norm $\|\mathbf{X}_c - \mathbf{X}_c^{(k)}\|_F^2$.

The proportion of variance explained by the first k principal components is:

$$\text{Proportion of variance explained} = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{\min(n,p)} \sigma_i^2} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{p} \lambda_i}$$

Proportion of Variance Explained

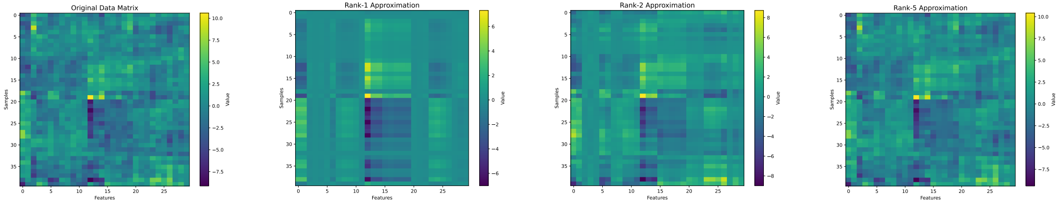
The proportion of variance explained measures how much of the total data variation is captured by the first k principal components:

$$\text{Proportion of variance explained} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{p} \lambda_i}$$

This metric helps determine how many principal components to retain:

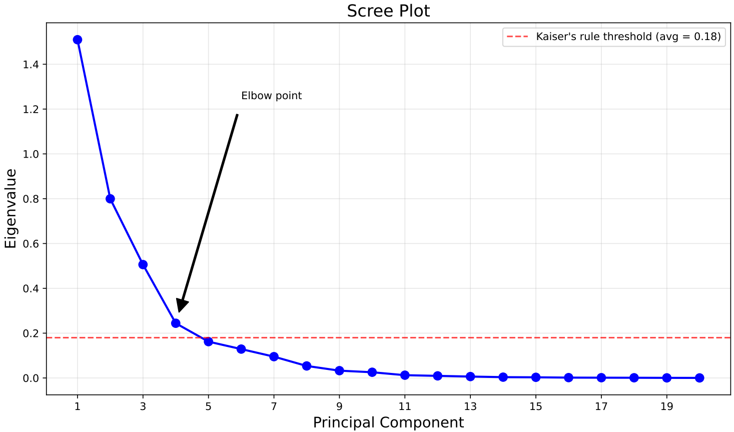
- Often visualized as a scree plot or cumulative variance plot
- Common thresholds: retain components explaining 80-90% of variance
- Alternatively, look for an "elbow" in the scree plot where additional components add minimal explanation

Rank-k Approximations of the Data Matrix



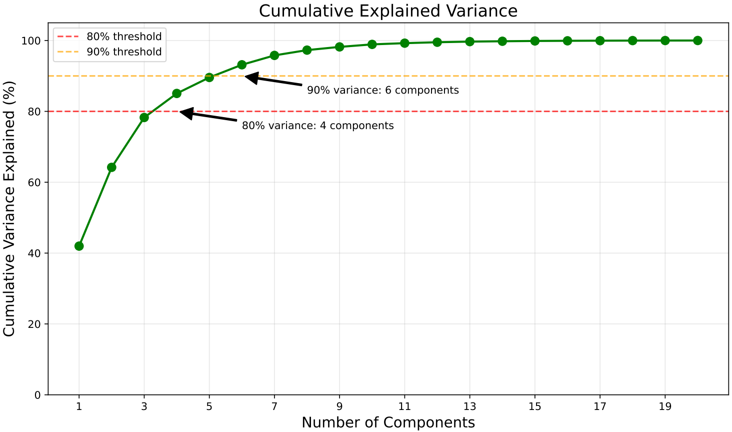
Original data matrix (top left) and its low-rank approximations: rank-1 (top right), rank-2 (bottom left), and rank-5 (bottom right). Note how adding components progressively captures more structure.

Using Scree Plots to Determine Dimensionality



A scree plot displays eigenvalues in descending order. The "elbow" where the curve levels off suggests how many components to retain.

Cumulative Variance Explained



Cumulative proportion of variance explained as components are added. A common heuristic is to retain enough components to explain 80-90

The Shape of Your Data: Geometric Intuition

PCA provides geometric intuition about data structure through eigenvalue patterns:

- **Spherical data**: All eigenvalues approximately equal - no preferred directions
- **Disk-shaped data**: Two dominant eigenvalues with similar magnitude - data concentrated in a plane
- **Cigarette-shaped data**: One dominant eigenvalue much larger than others - data elongated along a single direction
- **Rugby ball-shaped data**: Smoothly decreasing sequence of eigenvalues - complex, multi-factorial variation

This geometric language provides an intuitive way to discuss high-dimensional data structure.

Interpreting Principal Components

Principal components represent "latent factors" that explain patterns of variation:

Each principal component is a linear combination of the original variables:

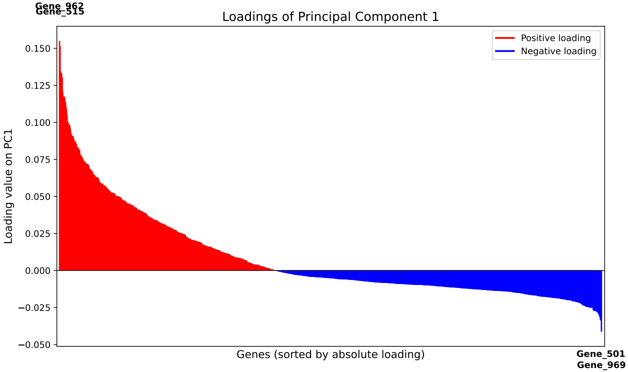
$$PC_i = \sum_{j=1}^{p} v_{ij} X_j$$

The challenge in interpretation comes from:

- Loadings having both positive and negative values
- Observations having both positive and negative coordinates

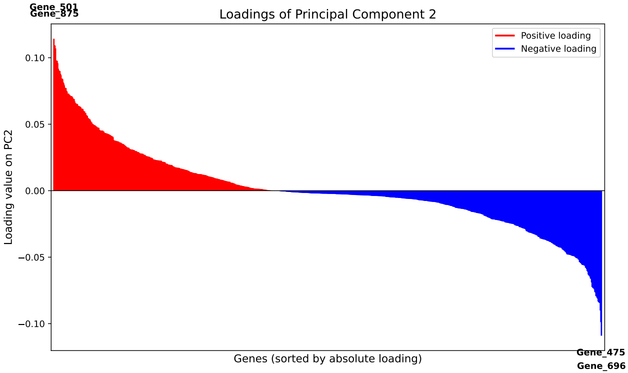
A component doesn't simply represent presence/absence of a biological process, but rather contrasting patterns - for example, high expression of proliferation genes and low expression of differentiation genes versus the opposite pattern.

Loadings of the First Principal Component



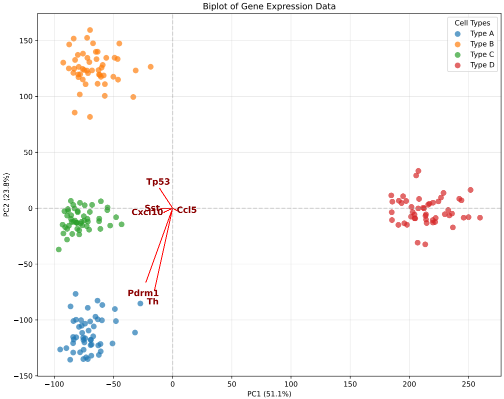
Loadings of PC1, with genes sorted by magnitude. Positive loadings (red) and negative loadings (blue) show which genes drive variation along this component in opposite directions.

Loadings of the Second Principal Component



Loadings of PC2. The different pattern compared to PC1 indicates this component captures a distinct axis of variation in the data.

Biplot of Gene Expression Data



Biplot of gene expression data. Points represent cells, while arrows show genes. This visualization reveals which genes drive the separation between cell groups.

Biplots: Visualization of Observations and Variables

A biplot displays both observations and variables in the same principal component space:

- Points represent observations (e.g., cells)
- Arrows represent variables (e.g., genes), with direction and length indicating loadings

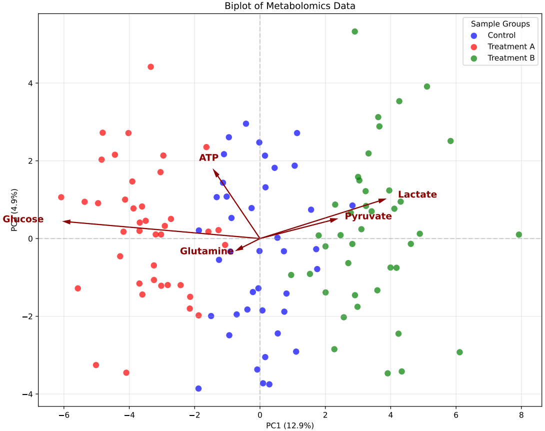
Construction:

- Observations: Plot the principal component scores (elements of $\mathbf{U} \cdot \mathbf{\Sigma}$)
- Variables: Represent as arrows with coordinates determined by loadings (elements of $\mathbf{V}$)

Interpretation:

- Observations positioned in direction of a particular variable arrow tend to have high values for that variable
- Length of variable arrows indicates how well that variable is represented in the displayed components

Biplot of Metabolomics Data



Biplot of metabolomics data. Reflecting the distinct correlation structure in metabolomic measurements.

Practical Considerations and Limitations

Despite its power and elegance, PCA has several limitations to consider:

- **Sensitivity to scaling**: Results depend crucially on variable scales. Variables with larger scales dominate the first components regardless of importance. Standardization (scaling to unit variance) is often used to address this.
- **Sensitivity to outliers**: As a least-squares method, PCA can be heavily influenced by outliers. Robust variants exist to address this issue.
- **Interpretation challenges**: Principal components are linear combinations of all original variables, which can make biological interpretation difficult.

Understanding these limitations helps ensure appropriate application of PCA in biological data analysis.

Clustering

From Continuous Variation to Discrete Groups

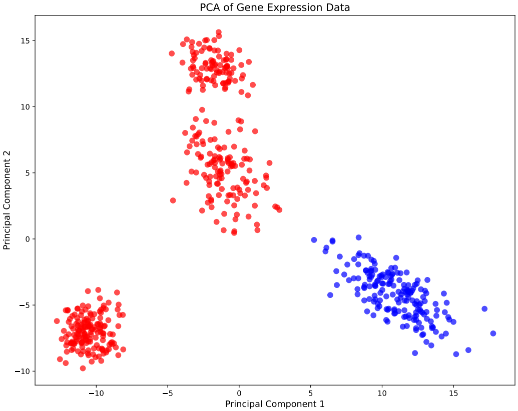
When we apply PCA to biological data, we often observe that individual data points naturally organize into distinct groups in the reduced dimensional space.

This clustering pattern suggests underlying biological structure:

- Discovering novel cell types in single-cell data
- Identifying patient subtypes with different disease prognoses
- Finding functional modules in gene regulatory networks

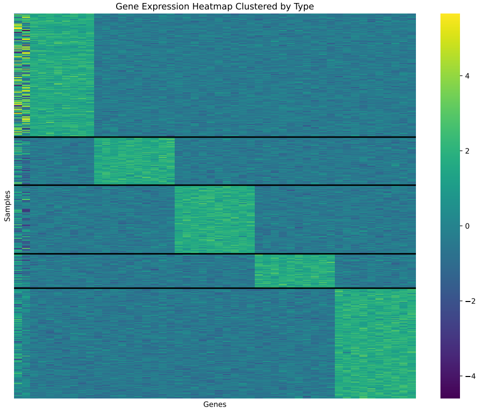
Clustering formalizes this process of identifying these groups objectively.

Clustering Patterns Emerge from PCA



Natural clusters emerge in PCA space, showing distinct cancer subtypes (red) separated from normal samples (blue).

The Data Matrix Reveals Cluster Structure



The same data as a heatmap: distinct expression patterns define different groups.

Clustering as Optimization

Clustering finds a partition of data that minimizes an objective function $J(\mathcal{C})$, balancing two competing goals:

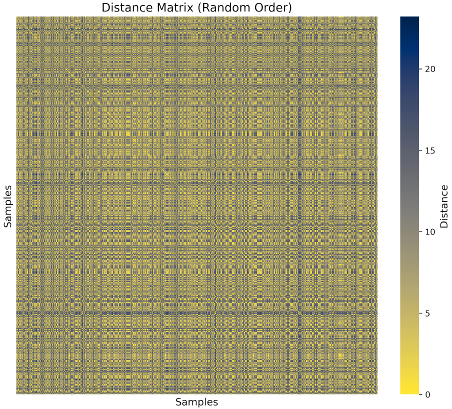
- **Intra-cluster similarity**: Observations within the same cluster should be similar
- **Inter-cluster dissimilarity**: Observations in different clusters should be dissimilar

Definition (Clustering as Optimization)

A clustering is a partition $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ that minimizes:

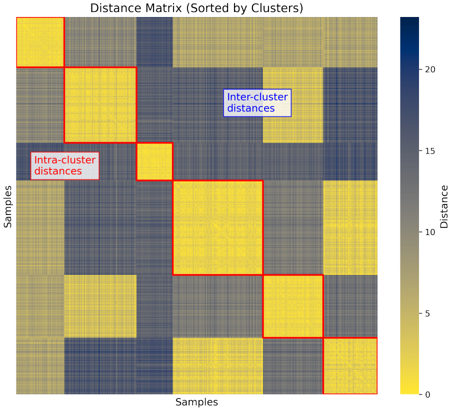
$$\mathcal{C}^* = \arg\min_{\mathcal{C}} J(\mathcal{C}) \tag{1}$$

Distance Matrices Visualize Cluster Quality



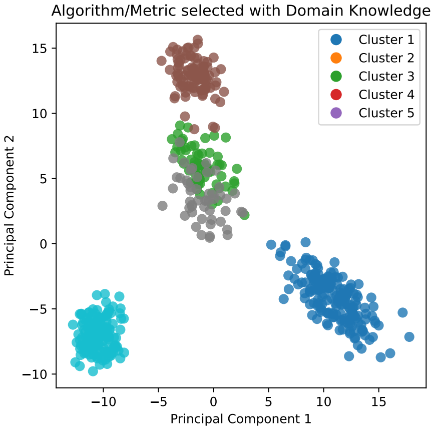
Distance matrix with random ordering shows no obvious structure.

Clusters Reveal Structure in Distance Matrices



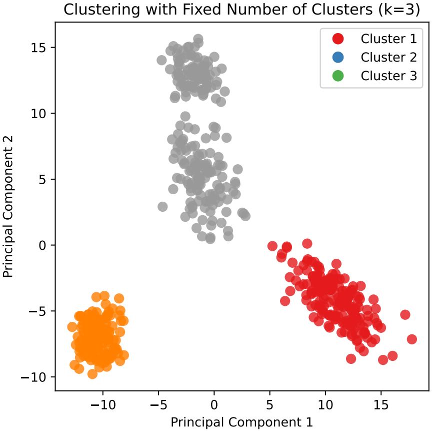
Same data sorted by clusters: small intra-cluster distances (red boxes), large inter-cluster distances (blue boxes).

Clustering: There’s No Single ”Right” Answer



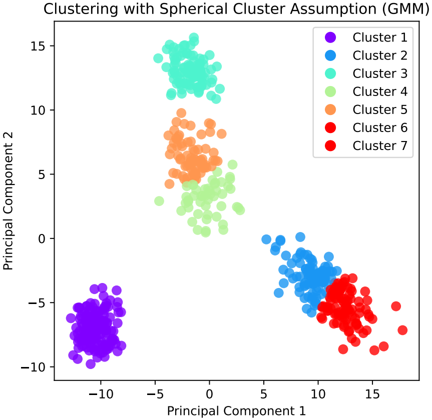
The ”true” clustering based on domain knowledge: 5 distinct groups.

Different Algorithms Give Different Results



Same data with clustering forced to 3 groups: merges related cancer types.

Algorithm Assumptions Matter



Method assuming spherical clusters: misses the natural structure completely.

Two Major Clustering Approaches

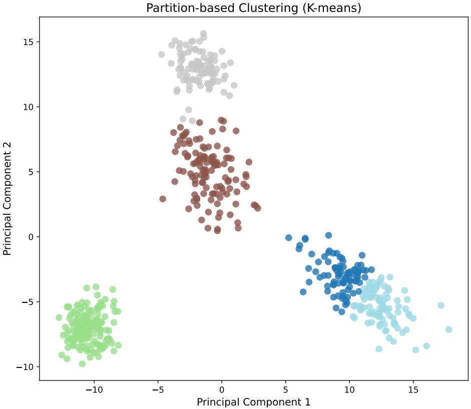
Partition-based methods (like k-means):

- Divide data into predetermined number of clusters
- Each observation belongs to exactly one cluster
- Good when expected number of groups is known

Hierarchical agglomerative methods:

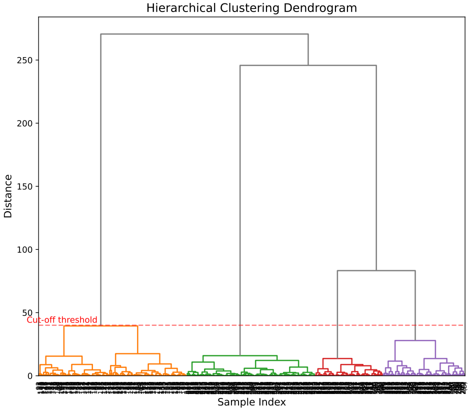
- Create nested sequence of partitions
- Reveal relationships at different scales through dendrograms
- Good when natural number of clusters is unclear

Partition-based Clustering Provides Flat Structure



K-means clustering partitions data into exactly k groups without hierarchy.

Hierarchical Clustering Reveals Relationships



Dendrogram shows how clusters merge, providing insight into relationships.

K-Means: Minimizing Within-Cluster Variance

K-means partitions data into k clusters by minimizing the within-cluster sum of squares:

$$J(\mathcal{C}) = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \tag{2}$$

where $\boldsymbol{\mu}_i$ is the centroid (mean) of cluster C_i.

This ensures points within each cluster are as close as possible to their cluster's center.

The K-Means Algorithm

K-means alternates between two key steps:

1. **Assignment Step**: Assign each observation to the nearest cluster centroid
2. **Update Step**: Recalculate cluster centroids as the mean of assigned points

This iterative process continues until the algorithm converges.

K-Means Algorithm

Algorithm: K-Means Clustering

Input: Dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, number of clusters k

Output: Clusters $C_1, C_2, ..., C_k$ and centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k$

1. Initialize centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k$ (e.g., randomly)
2. Repeat until convergence:
    - **Assignment step:** Assign each observation $\mathbf{x}_j$ to cluster C_i where:

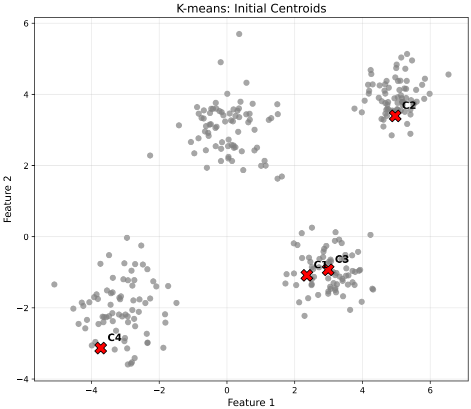
      $$i = \underset{i \in \{1,...,k\}}{\arg\min} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

    - **Update step:** Recalculate each centroid as the mean of its assigned points:

      $$\boldsymbol{\mu}_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$$

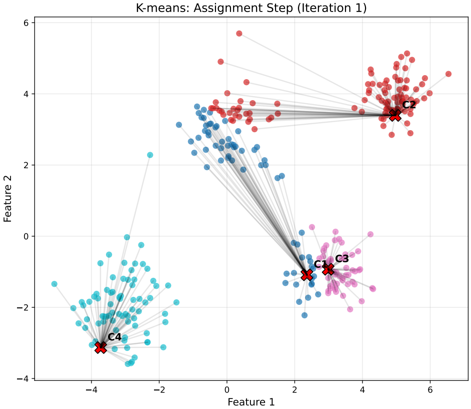
3. Return cluster assignments $C_1, C_2, ..., C_k$ and final centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k$

K-Means: Initial Random Centroids



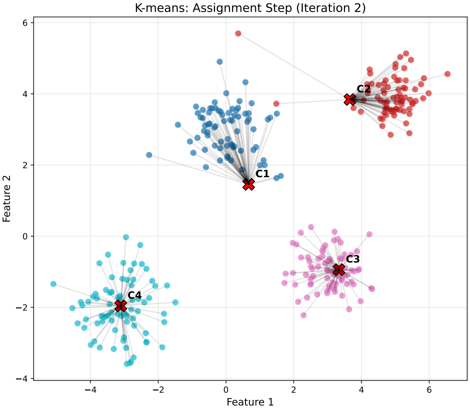
Algorithm starts with randomly chosen centroid locations.

K-Means: First Assignment Step



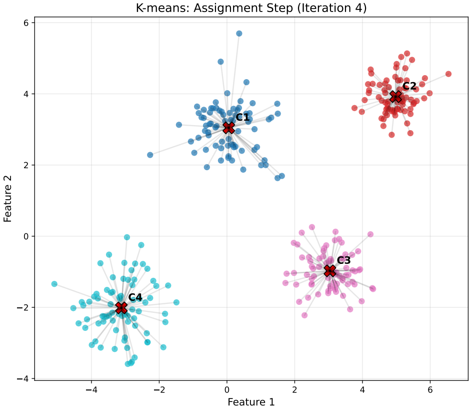
Each point is assigned to its nearest centroid, creating Voronoi regions.

K-Means: Second Assignment Step



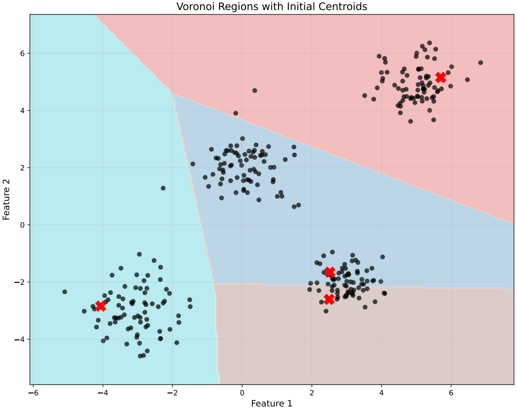
Centroids are updated to cluster means, changing point assignments.

K-Means: Approaching Convergence



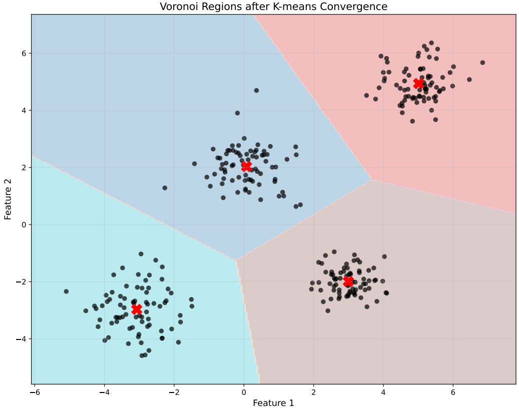
After several iterations, cluster assignments stabilize.

K-Means Creates Voronoi Regions



Initial Voronoi tessellation divides space based on proximity to centroids.

Voronoi Regions Evolve with Convergence



Final regions reflect the optimal partition of the data space.

Choosing k: The Challenge of Unknown Group Numbers

Determining the appropriate number of clusters, k, requires balancing:

- **Domain knowledge**: Expected cell types, disease stages
- **Model selection**: Avoid over- or under-clustering
- **Stability**: Consistent results across multiple runs

The silhouette score provides a quantitative measure of cluster quality.

The Silhouette Score: Measuring Cluster Quality

For each observation, the silhouette coefficient compares inter- and intra-cluster distances:

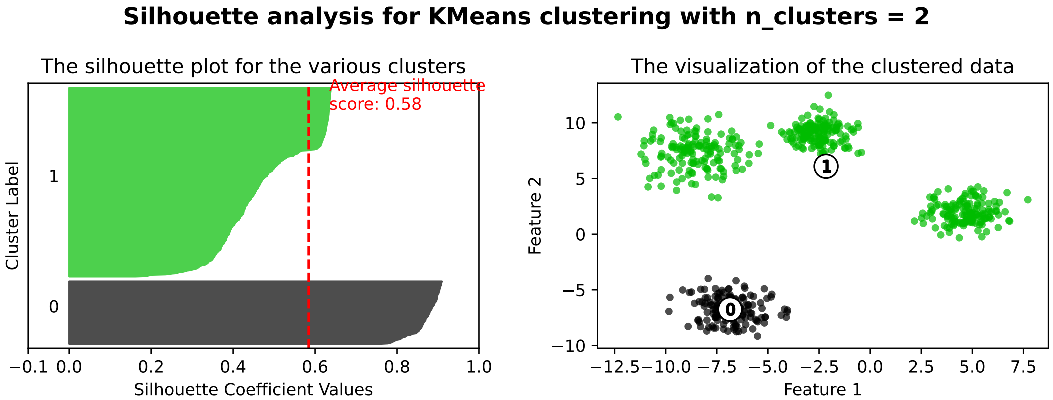
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{3}$$

where:

- $a(i)$: average distance to points in same cluster
- $b(i)$: average distance to points in nearest neighboring cluster

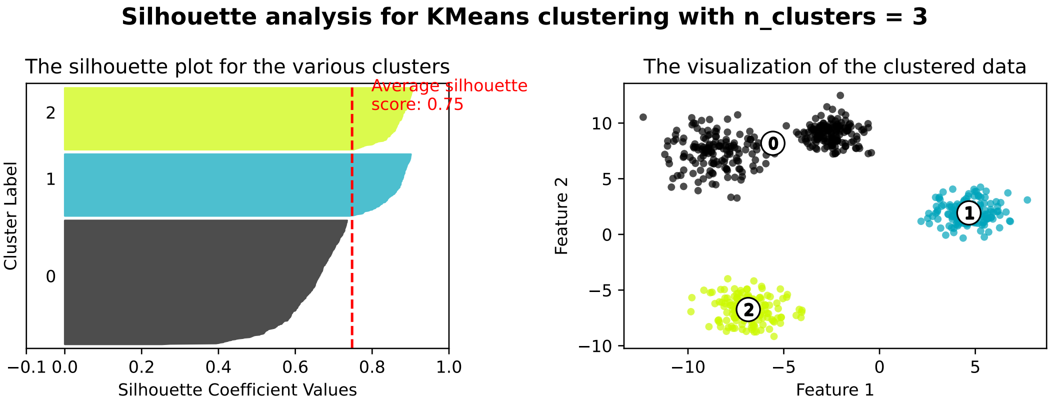
Values range from -1 (wrongly clustered) to $+1$ (well-clustered).

Silhouette Analysis for k=2



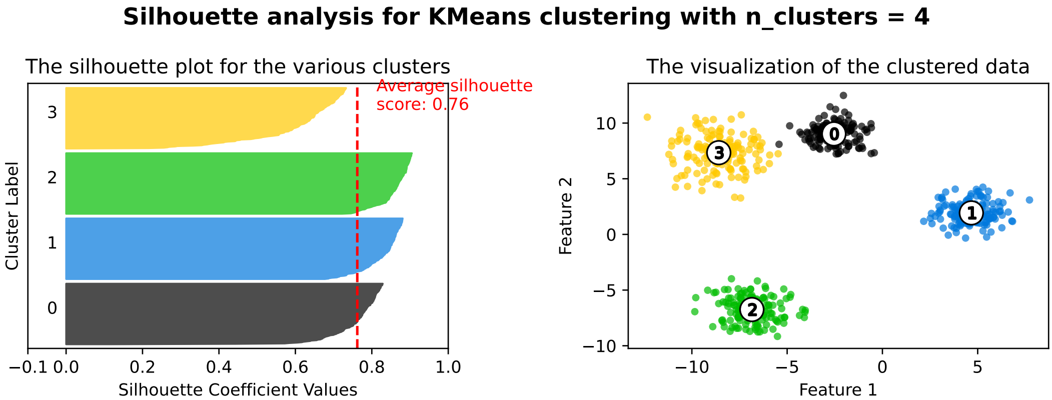
With two clusters, most points have positive silhouette values.

Silhouette Analysis for k=3



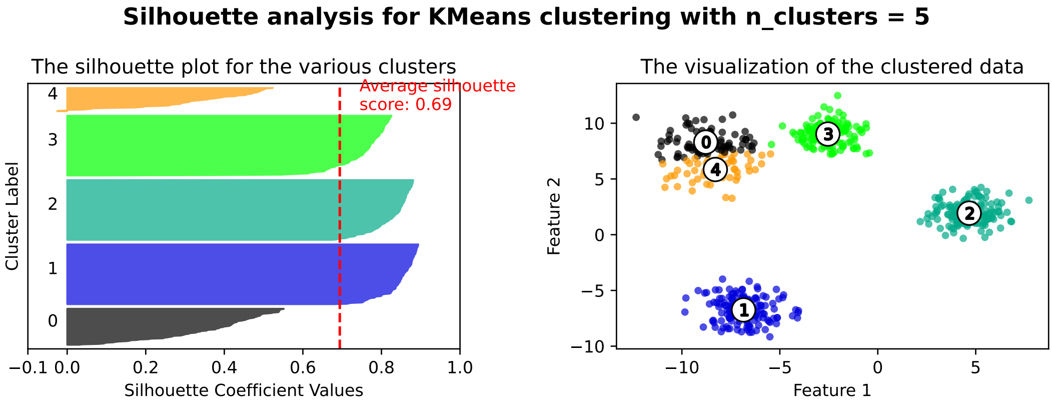
Three clusters show one problematic cluster with negative values.

Silhouette Analysis for k=4



Four clusters reveal better-defined groups with higher scores.

Silhouette Analysis for k=5



Five clusters maintain high silhouette scores and balanced group sizes.

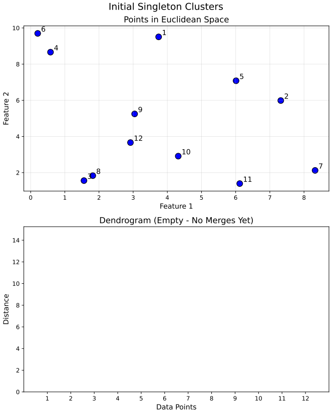
Agglomerative Clustering: Bottom-Up Approach

Hierarchical agglomerative clustering builds clusters progressively:

- Start with each observation as its own cluster
- Progressively merge the most similar pairs
- Continue until all observations belong to a single cluster

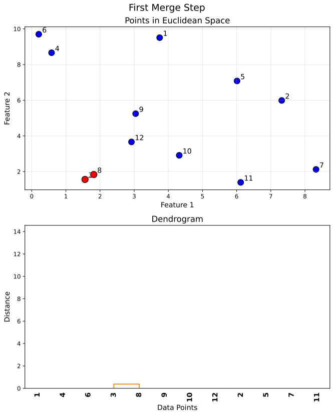
The result is a dendrogram showing the hierarchy of merges.

Agglomerative Process: Initial State



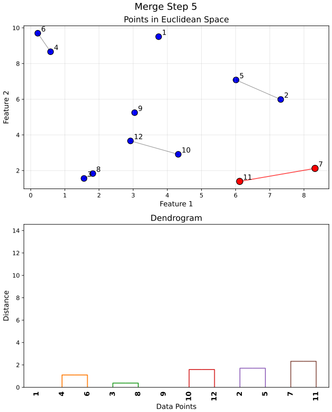
Each observation starts as its own singleton cluster.

Agglomerative Process: First Merge



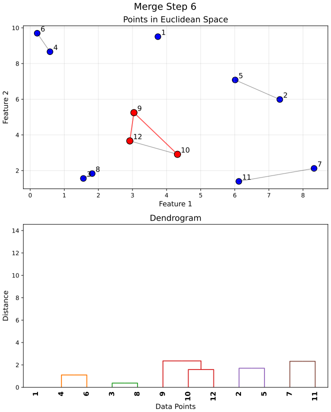
First merge combines the two most similar points, beginning the dendrogram.

Agglomerative Process: Progressive Merging



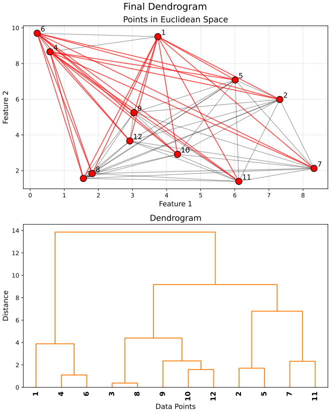
Additional merges form larger clusters, extending the hierarchy.

Building the Dendrogram Structure



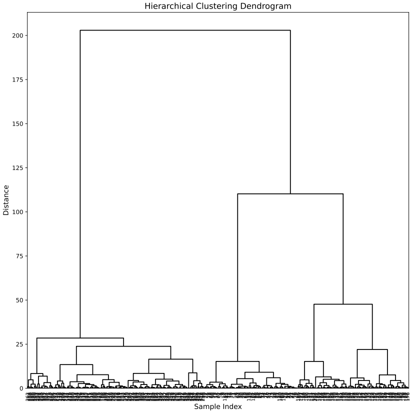
Higher merges correspond to greater dissimilarity between joined clusters.

Complete Dendrogram Reveals Hierarchy



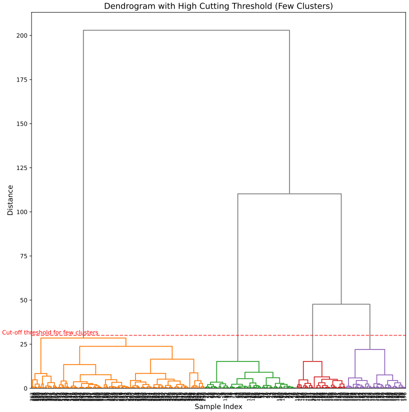
Completed dendrogram shows hierarchical relationships between all clusters.

Understanding Dendrograms



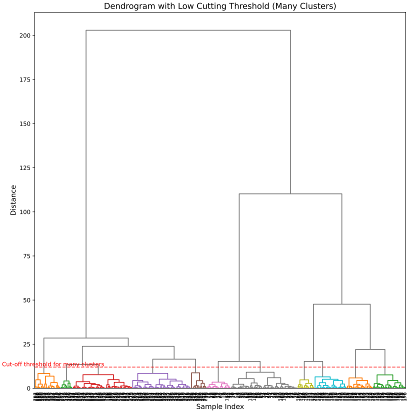
Tree structure encodes the sequence and distance of cluster merges.

Cutting Dendrograms to Create Clusters



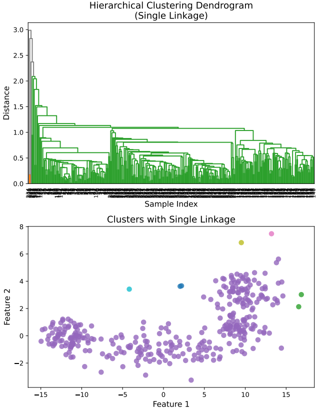
High cut produces few, large clusters.

Low Cuts Produce Many Clusters



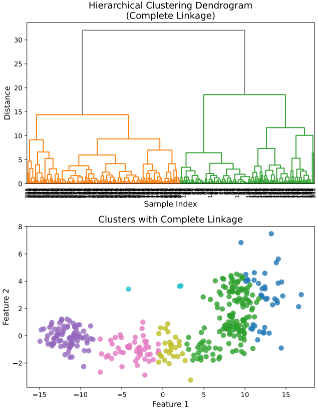
Low cut creates many small, fine-grained clusters.

Single Linkage: Nearest Points



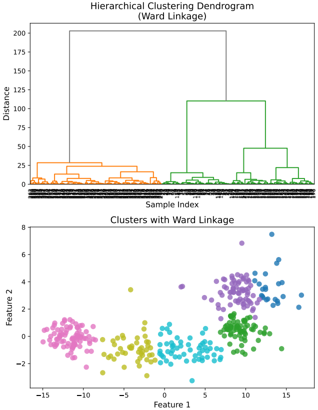
Single linkage connects clusters at their nearest points, prone to chaining.

Complete Linkage: Farthest Points



Complete linkage uses maximum distance, creating compact, similar-sized clusters.

Ward’s Method: Minimizing Variance



Ward’s method minimizes within-cluster variance, producing tight, homogeneous clusters.

Linkage Methods Affect Cluster Structure

Different linkage methods define inter-cluster distance differently:

Single Linkage: Distance between nearest points

$$d(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y})$$

Complete Linkage: Distance between farthest points

$$d(C_i, C_j) = \max_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y})$$

Ward's Method: Based on variance increase from merging

Agglomerative Clustering: Bottom-Up Approach

Hierarchical agglomerative clustering builds clusters progressively:

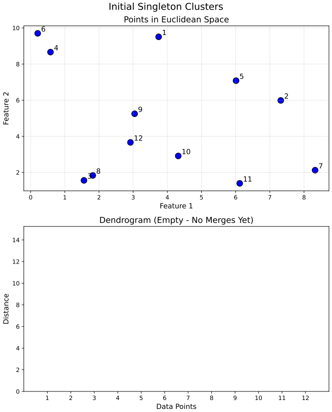
- Start with each observation as its own cluster
- Progressively merge the most similar pairs
- Continue until all observations belong to a single cluster

The result is a dendrogram showing the hierarchy of merges.

Agglomerative Hierarchical Algorithm

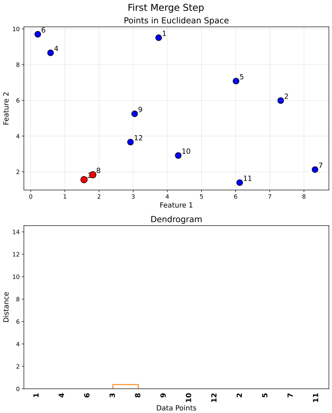
```
Algorithm: Agglomerative Hierarchical Clustering
Input: Dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, distance function $d(\cdot, \cdot)$
Output: Hierarchical cluster structure (dendrogram)
  1 Initialize singleton clusters $C_i = \{\mathbf{x}_i\}$ for $i = 1...n$
  2 Calculate distance matrix $D$ where $D_{ij} = d(C_i, C_j)$
  3 For $t = 1$ to $n - 1$:
        ▪ Find $(i, j) = \arg\min_{i,j} D_{ij}$ with minimum distance
        ▪ Merge $C_i$ and $C_j$ into new cluster $C_{new}$
        ▪ Record merge and distance for dendrogram
        ▪ Remove rows/columns $i, j$ from matrix $D$
        ▪ Calculate distances from $C_{new}$ to other clusters
        ▪ Add $C_{new}$ to distance matrix $D$
  4 Return dendrogram of cluster hierarchy
```

Agglomerative Process: Initialization



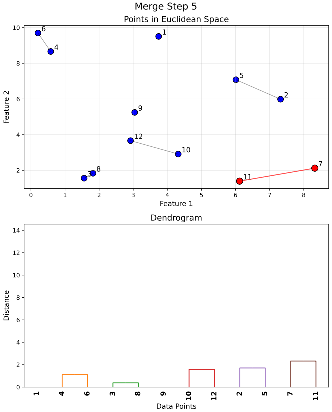
Each observation starts as its own singleton cluster.

Agglomerative Process: First Merge



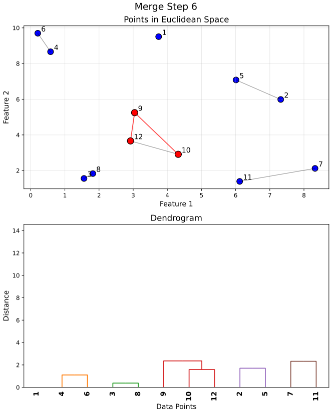
First merge combines the two most similar points, beginning the dendrogram.

Agglomerative Process: Progressive Merging



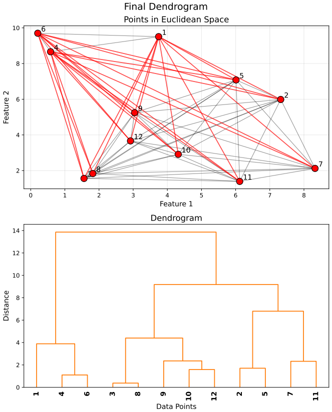
Additional merges form larger clusters, extending the hierarchy.

Building the Dendrogram Structure



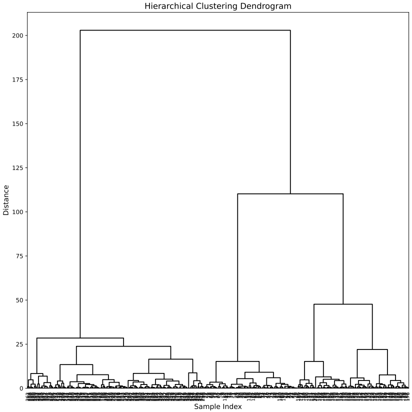
Higher merges correspond to greater dissimilarity between joined clusters.

Complete Dendrogram Reveals Hierarchy



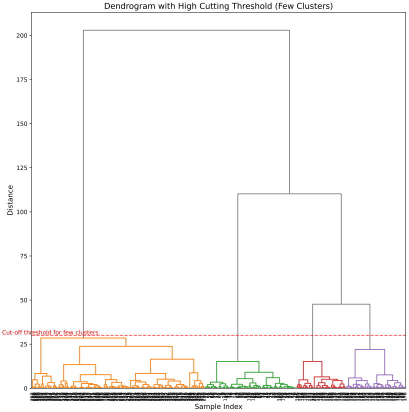
Completed dendrogram shows hierarchical relationships between all clusters.

Understanding Dendrograms



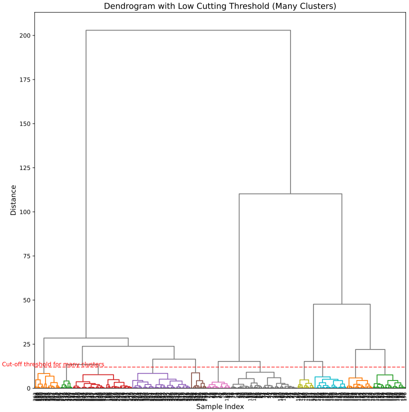
Tree structure encodes the sequence and distance of cluster merges.

Cutting Dendrograms to Create Clusters



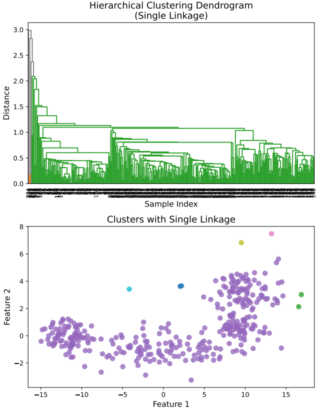
High cut produces few, large clusters.

Low Cuts Produce Many Clusters



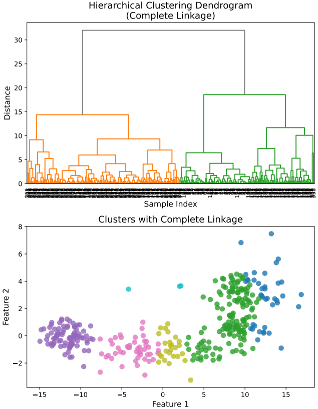
Low cut creates many small, fine-grained clusters.

Single Linkage: Nearest Points



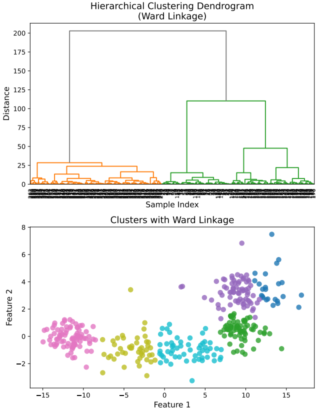
Single linkage connects clusters at their nearest points, prone to chaining.

Complete Linkage: Farthest Points



Complete linkage uses maximum distance, creating compact, similar-sized clusters.

Ward’s Method: Minimizing Variance



Ward’s method minimizes within-cluster variance, producing tight, homogeneous clusters.

Linkage Methods Affect Cluster Structure

Different linkage methods define inter-cluster distance differently:

Single Linkage: Distance between nearest points

$$d(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y})$$

Complete Linkage: Distance between farthest points

$$d(C_i, C_j) = \max_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y})$$

Ward's Method: Based on variance increase from merging

Question 1

What does a PCA *biplot* display?

A) Only the observations plotted on the first two principal components

B) Only the principal components of the variables, with arrows representing the observations

C) Both the observations (as points) and the original variables (as arrows) in the same principal-component space

D) Only the variables plotted as arrows on the PC axes

E) A heatmap of variable loadings across all PCs

Question 2

Which graphical tool helps determine the optimal number of principal components by plotting eigenvalues against their component index?

A) Biplot

B) Parallel coordinates plot

C) Scree plot

D) Loading plot

E) Heatmap of scores

Question 3

What is the main objective function minimized by the k-means clustering algorithm?

A) Total pairwise distances between all points

B) Between-cluster variance

C) Within-cluster sum of squared distances from cluster centroids (WCSS)

D) L1 distance from the global mean

E) Maximum distance between clusters

Question 4

Why might hierarchical clustering be preferred over k-means when analyzing biological data with unknown grouping structure?

A) It always runs faster than k-means for large datasets

B) It reveals relationships at multiple granularities without pre-specifying the number of clusters

C) It always produces spherical clusters

D) It requires fewer computational resources

E) It guarantees optimal cluster separation

Question 5

In biological data analysis, when should you prefer clustering over classification with pre-defined labels?

A) When predicting known disease outcomes from gene expression data

B) When assigning samples to established experimental conditions

C) When validating a diagnostic test against known disease status

D) When discovering novel cell types or patient subtypes without prior knowledge of categories

E) When replicating published gene expression signatures

Question 6

What is a dendrogram?

A) A statistical representation of dendrites in neurons

B) A tree diagram showing the hierarchical relationship between data points through successive merges

C) A cladistics representation useful in biological data analysis

D) A graphical display of the branching structure of blood vessel networks in medical imaging

E) A statistical plot showing the spread of data across population branches