Unsupervised learning

Federica Eduati

Eindhoven University of Technology
Department of Biomedical Engineering

2024

Learning goals

- **Understand Unsupervised Learning:** Define unsupervised learning and its objectives.
- **Dimensionality Reduction (PCA):** Explain PCA for reducing dimensionality while preserving variance. Understand the meaning and interpretation of principal components and loading vectors.
- **K-means Clustering:** Understand k-means algorithm steps: initialisation, assignment, and update. Learn techniques to select the number of clusters.
- **Hierarchical Clustering:** Understand agglomerative clustering and how dendrograms represent clusters. Compare similarity metrics (Euclidean distance, correlation) linkage methods (single, complete, average) and their impact.

Material

- Chapter 12 of "*An introduction to statistical learning with applications in python*, G. James, D. Witten, T. Hastie, R. Tibshirani, J. Taylor"

Overview

- Supervised vs unsupervised learning
- Dimensionality reduction
  - Principal component analysis (PCA)
- Clustering
  - K-means
  - Hierarchical clustering

Join at menti.com | use code 5399 7263

TU/e

Instructions

Go to

www.menti.com

Enter the code

5399 7263

Or use QR code

Difference between supervised and unsupervised learning

With both supervised and unsupervised learning we have a set of features $X_1, X_2, \ldots, X_p$ for each observation:

- With **supervised learning** (or learning with a teacher) we also have an associated variable Y (a label).
- With **unsupervised learning** (or learning without a teacher) we don't have the label.

Unsupervised learning

The goal of unsupervised learning is to find similarities among observations based on the set of features.

There are two categories of methods:

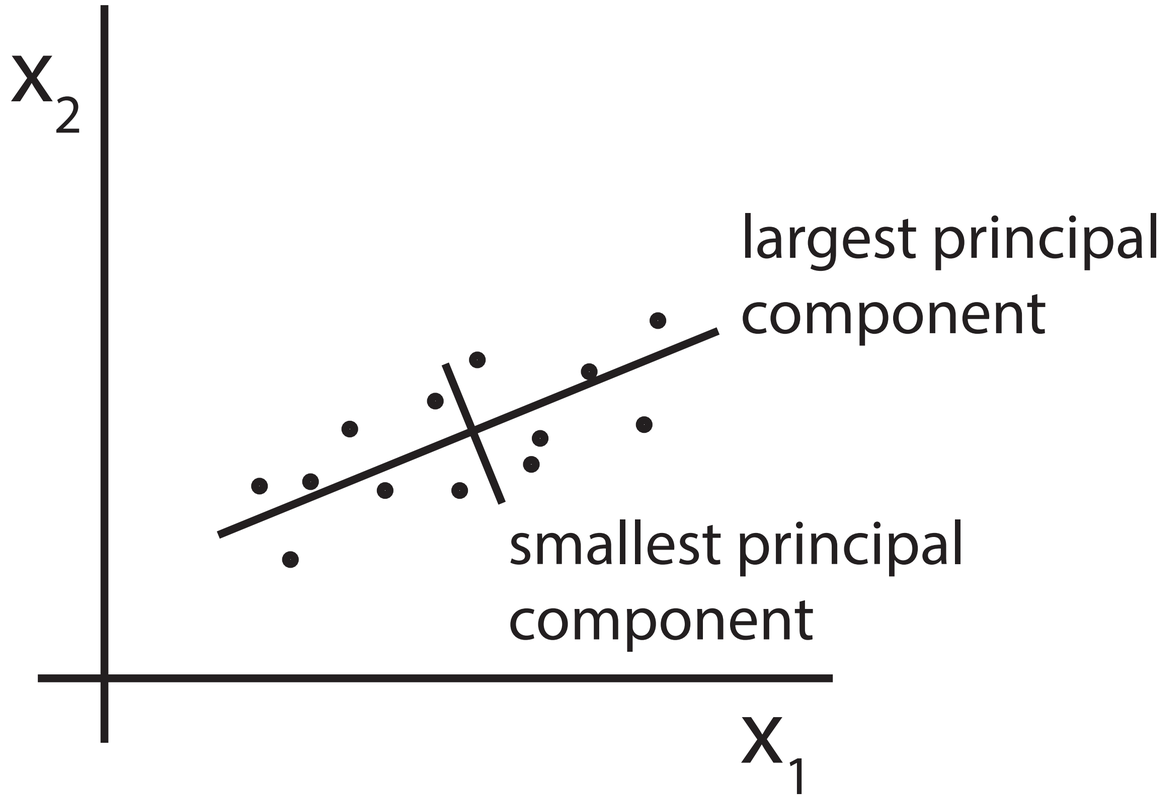
- **Dimensionality reduction:** methods to reduce the the dimensions of the input features to facilitate visualisation and identification of groups.
- **Clustering:** methods to discover unknown groups (clusters) in data.

Use of unsupervised learning

- No need for labels (often difficult to retrieve).
- Used for exploratory data analysis or for preprocessing before applying supervised learning techniques.
- No quantitative metrics to measure success; evaluation based on heuristic arguments.

Principal component analysis (PCA)

Idea: from the p variables (often correlated), derive a smaller subset of variables that explain most of the variability of the original set.



Principal component analysis (PCA)

Use of PCA:

- For feature reduction: derive a smaller subset of variables (the principal components) that are a good representation of the original data. These can then be used for supervised learning problems.
- For data visualisation: be able to visualise data in a smaller dimension. This can be used for exploratory data analysis or to identify groups of observations with similar characteristics.

First principal component

Starting from a set of features $X_1, X_2, \ldots, X_p$, the **first principal component** is the normalised linear combination of the features:

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p$$

Where:

- $\sum_{j=1}^{p} \phi_{j1}^2 = 1$
- $\phi_{11}, \phi_{21}, \ldots, \phi_{p1}$ are the **loadings** of the first principal component
- $\phi_1 = (\phi_{11}, \phi_{21}, \ldots, \phi_{p1})^T$ is the **first principal component loading vector**

Computing the first principal component

Consider a dataset with n observations $x_1, x_2, \ldots, x_n$, where each $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$, or in matrix form $\mathbf{X}$ of size $n \times p$.

Since we are only interested in the variance, we assume that each feature has 0 mean, i.e. the column means of $\mathbf{X}$ are 0.

We want to find the linear combination of the sample feature value:

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \ldots + \phi_{p1} x_{ip}$$

With the largest sample variance, subject to $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.

Computing the first principal component

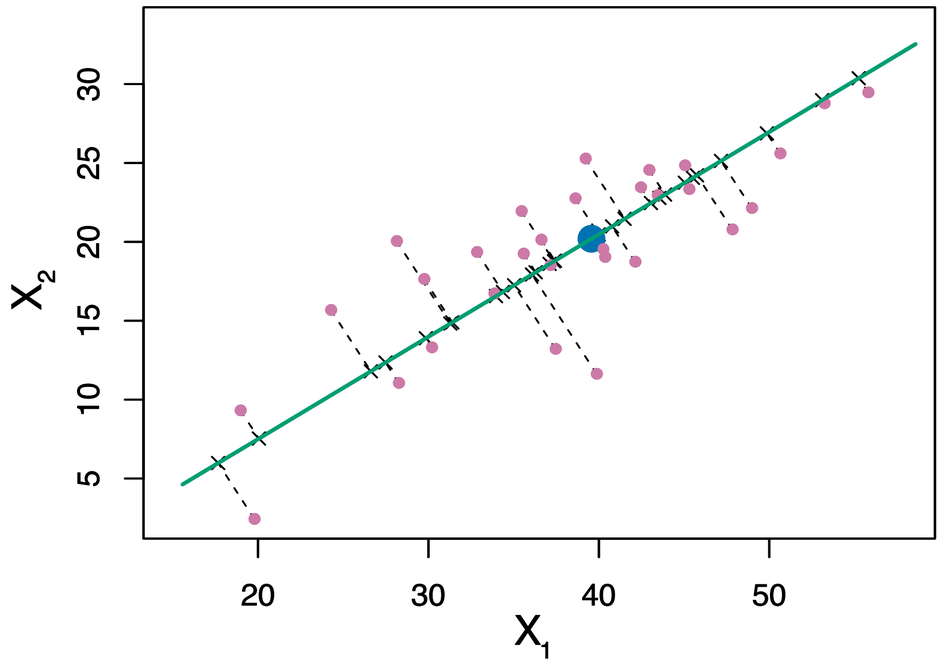
This is equivalent to solve the optimisation problem:

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \underbrace{\left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)}_{z_{i1}}{}^{2} \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \phi_{j1}^2 = 1.$$

Since each feature has zero mean (i.e. $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$), also the average of $z_{11}, \ldots, z_{n1}$ is zero. Hence what we are maximising is the sample variance of z_1, which is $\frac{1}{n}\sum_{i=1}^{n} z_{i1}^2$.

Interpreting the first principal component

- The **first principal component loading vector** $\phi_1 = (\phi_{11}, \phi_{21}, \ldots, \phi_{p1})^T$ define the direction along which the data vary the most.
- The **first principal component scores** are the projections of the n points $x_1, \ldots, x_n$ onto this direction.



Computing the second principal component

Given the first principal component Z_1, the second principal component Z_2 is the linear combination of the features that has maximal variance and is uncorrelated with Z_1.

The principal component scores take the form:

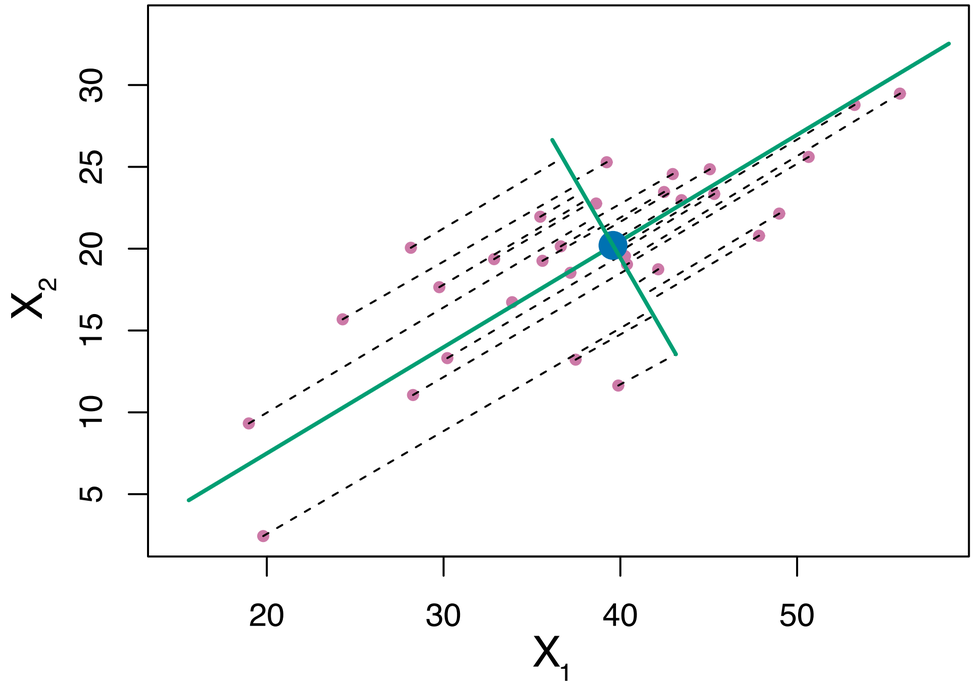
$$z_{i2} = \phi_{12}x_{i1} + \phi_{22}x_{i2} + \ldots + \phi_{p2}x_{ip}$$

That maximise the variance, under the constraints that:

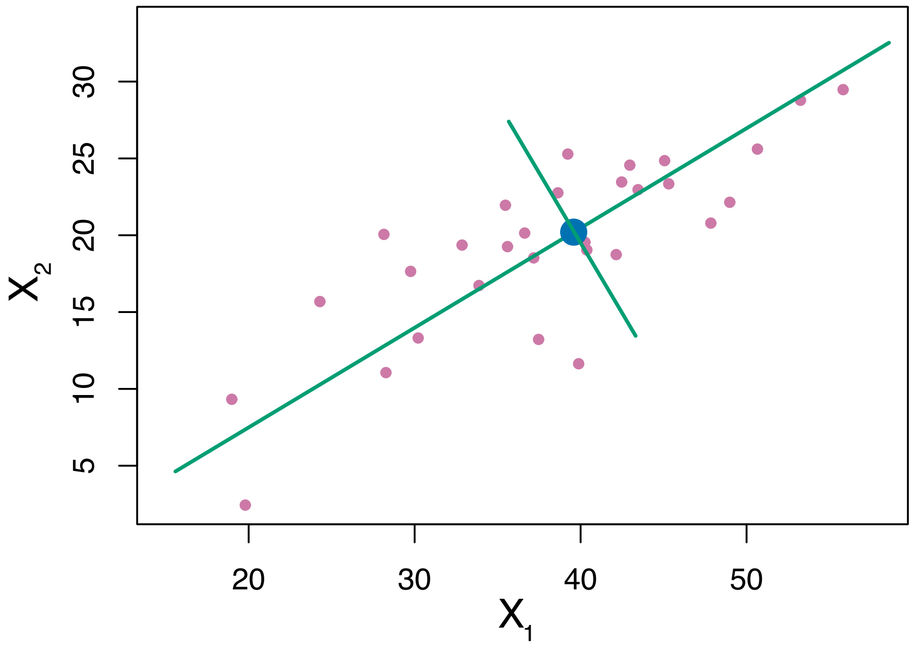
- $\sum_{j=1}^{p} \phi_{j2}^2 = 1$
- The direction of ϕ_2 orthogonal (perpendicular) to the direction of ϕ_1

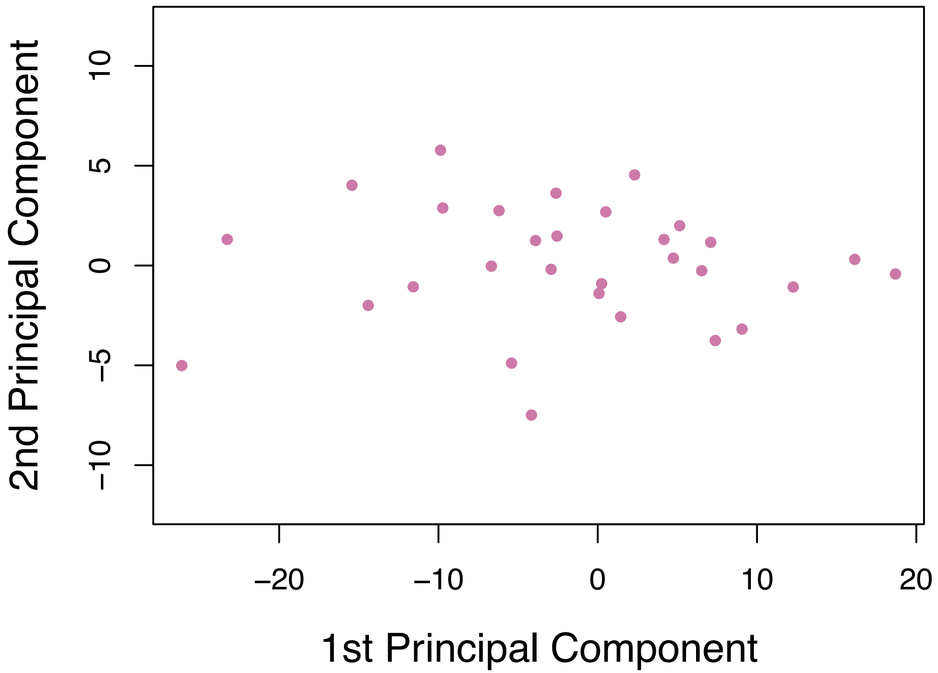
Interpreting the second principal component

- The direction of the **second principal component loading vector** ϕ_2 is orthogonal (perpendicular) to the direction of ϕ_1.
- The **second principal component scores** are the projections of the n points $x_1, \ldots, x_n$ onto the direction of ϕ_2.



Visualising the first two principal components





When do you expect PCA to have poor performance in feature reduction

(Multiple choices possible)

- When the features are highly correlated
- When the features are uncorrelated
- When the features are dominated by noise

PCA computation using singular value decomposition

The optimisation problem can be solved via singular value decomposition of the the $n \times p$ data matrix $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathbf{T}}$$

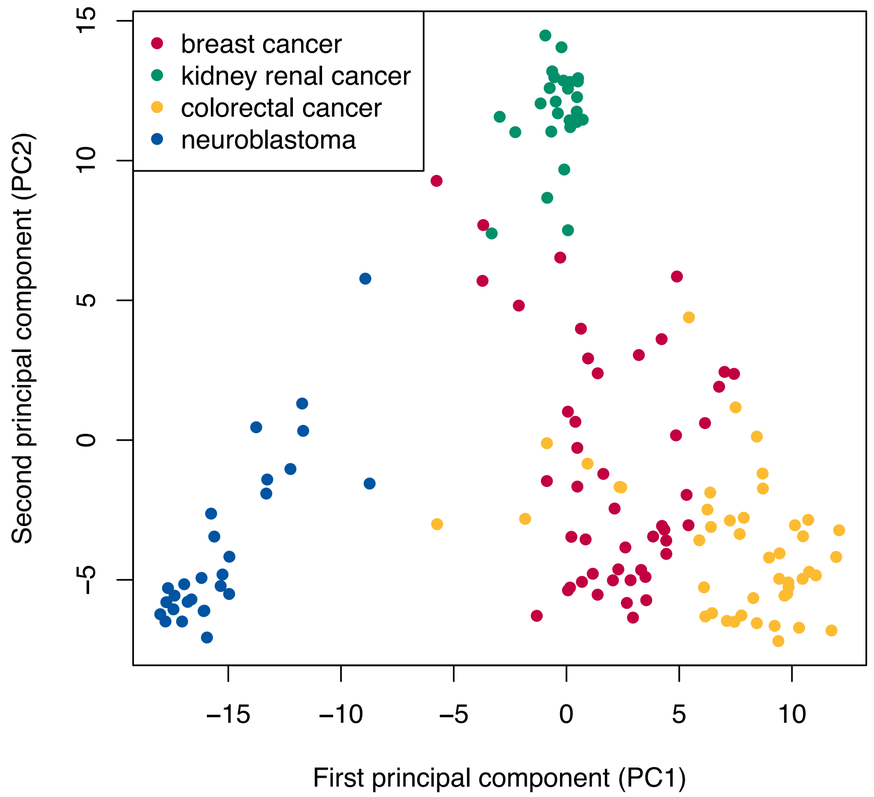
For $n > p$ this is a unique decomposition such that:

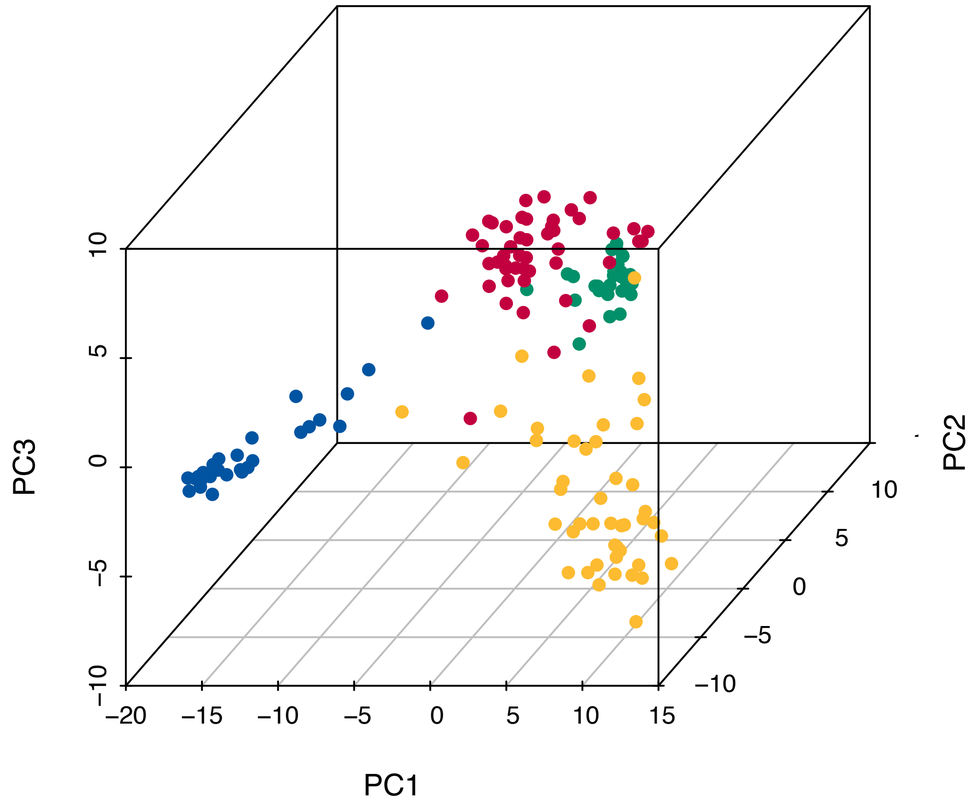
- $\mathbf{U}$ is a $n \times p$ orthogonal matrix (i.e. $\mathbf{U}^{\mathbf{T}}\mathbf{U} = \mathbf{I_p}$)
- $\mathbf{D}$ is a $p \times p$ diagonal matrix with $d_j > 0$ and $d_j \geq d_{j+1}$ known as singular values
- $\mathbf{V}$ is a $p \times p$ orthogonal matrix (i.e. $\mathbf{V}^{\mathbf{T}}\mathbf{V} = \mathbf{I_p}$)

The columns of $\mathbf{UD}$ are the principal components of $\mathbf{X}$ and $\frac{d_1^2}{n}, \frac{d_2^2}{n}, \ldots, \frac{d_p^2}{n}$ is the variance explained by each principal component.

Example: GDSC dataset - scores

RNA expression data (244 genes) for 148 cell lines from four cancer types.

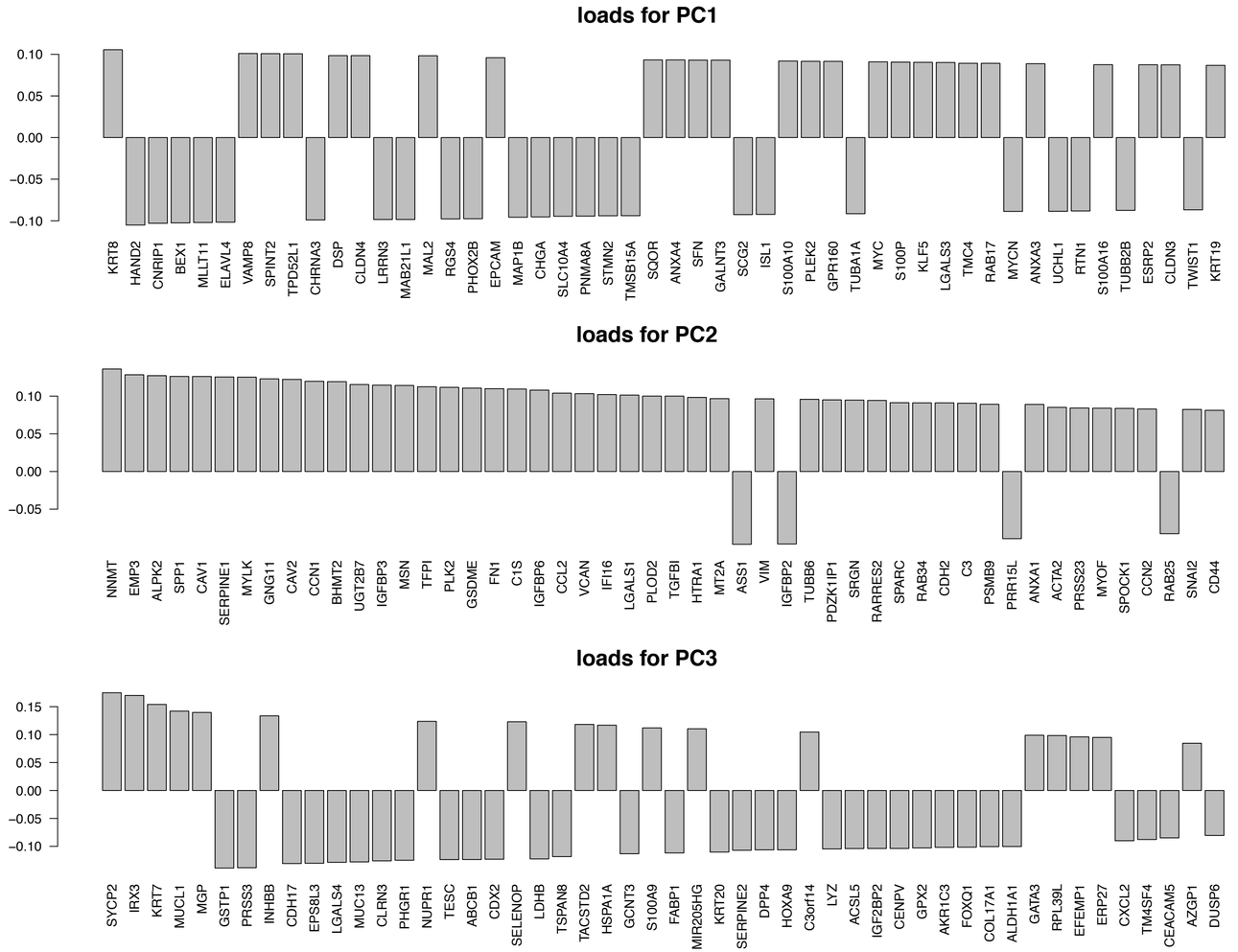




NOTE: this is an unsupervised learning problem, points are coloured *a posteriori* by cancer type

Example: GDSC dataset - loads

Loads for the first 3 principal components (only top 50 genes shown).

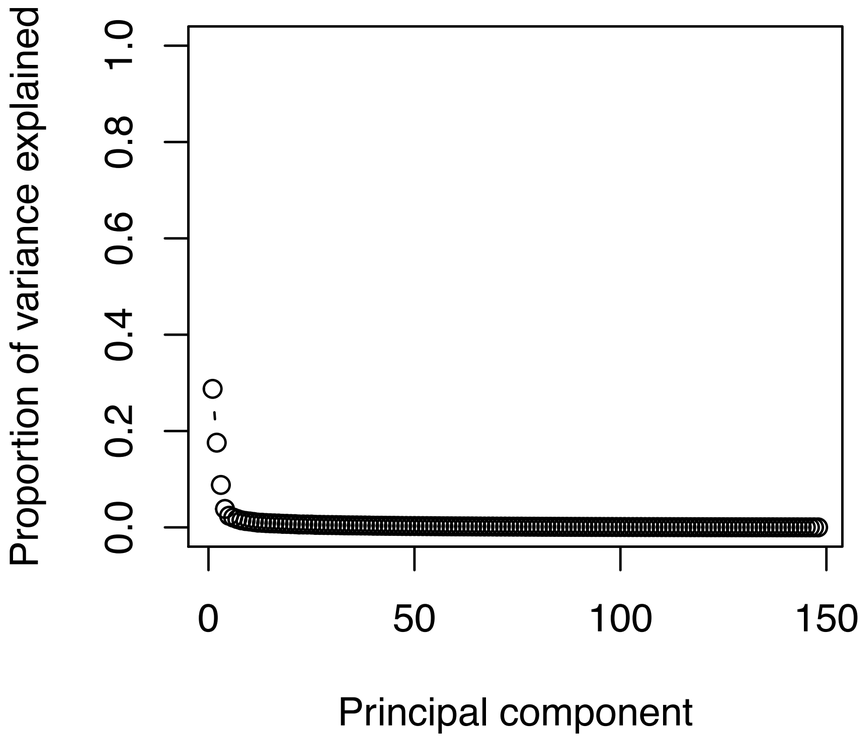


Deciding how many principal components to use

- In general an $n \times p$ data matrix $\mathbf{X}$ has $min(n-1,p)$ distinct principal components.
- We want to use the smallest number of principal components that give us a good understanding of the data.
- There is no unique way to define that, but approaches are in general based on the variance explained by the principal components.

Example: GDSC dataset - proportion of variance

We can look at the **proportion of variance explained** by each principal component.

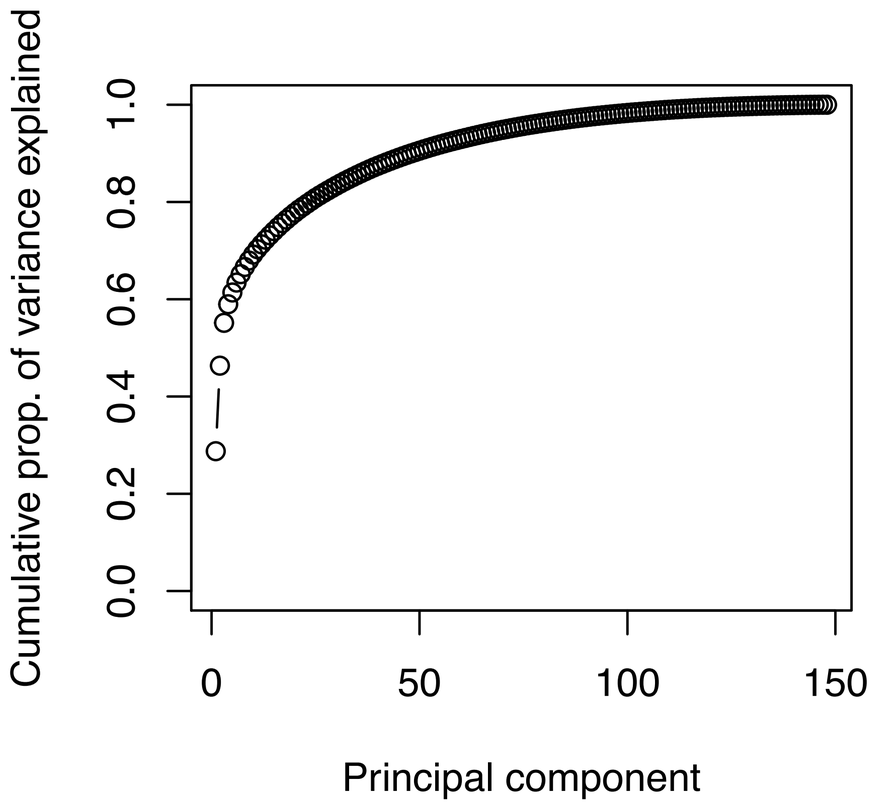


How to select the number of features?

Elbow method to select where there is a drop.

Example: GDSC dataset - cumulative proportion of variance

We can look at the **cumulative proportion of variance explained.**



How to select the number of features?

Set a threshold on what is the minimum desired cumulative variance explained.

Summary of dimensionality reduction with PCA

- **Goal:** Reduce the number of variables while retaining the most important information.
- **Key concepts:**
  - Projects data onto a smaller subspace.
  - Maximises variance in the data with fewer dimensions.
- **Interpretation:**
  - **Principal Components:** Ranked by how much variance they explain.
  - **Loadings:** Show how much each original variable contributes to each principal component.
- **Examples of biomedical Applications:**
  - Reducing dimensionality in gene expression data for cancer classification.
  - Simplifying complex patient measurements to identify dominant health factors.

Clustering

- Aim: Group observations into subsets, called clusters or segments, so that observations within a cluster are more similar to each other than observations assigned to different clusters.
- Requires a definition of *similarity* or *difference.*
- Unsupervised problem that try to find structures (clusters) based on the given data

Which of the following applications could make use of clustering?

(Multiple choices possible)

- Discover cancer patients subtypes based on clinical and genetic features.
- Reducing the number of variables in gene expression datasets to identify major patterns.
- Identifying groups of patients who respond similarly to a treatment based on biomarker changes.
- Categorising proteins based on structural similarities to understand functional relationships.
- Projecting high-dimensional biomarker data into 2D or 3D space for visualisation.

Note: These are all unsupervised learning problems that can be addressed either by PCA or clustering.

K-Means clustering

Given a desired number of clusters K, the K-means clustering assigns each observations to exactly one of the K clusters.

Mathematically, let $C_1, \ldots, C_K$ denote sets containing the indices of the observations in each cluster. These sets satisfy these two properties:

- $C_1 \cup C_2 \cup \ldots \cup C_K = \{1,\ldots,n\}$: each observation belongs to at least one of the K clusters
- $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$: the clusters are non-overlapping

K-Means clustering formulation

We want to define the clusters so that similar points are in the same cluster and dissimilar points in different clusters.

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

Where $W(C_k)$ is a measure of within-cluster variation for cluster C_k.
When using the squared Euclidean distance:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

Where $|C_k|$ denotes the number of observations in the kth cluster

K-Means clustering formulation

The optimisation problem becomes

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

This is equivalent to:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2 \right\}$$

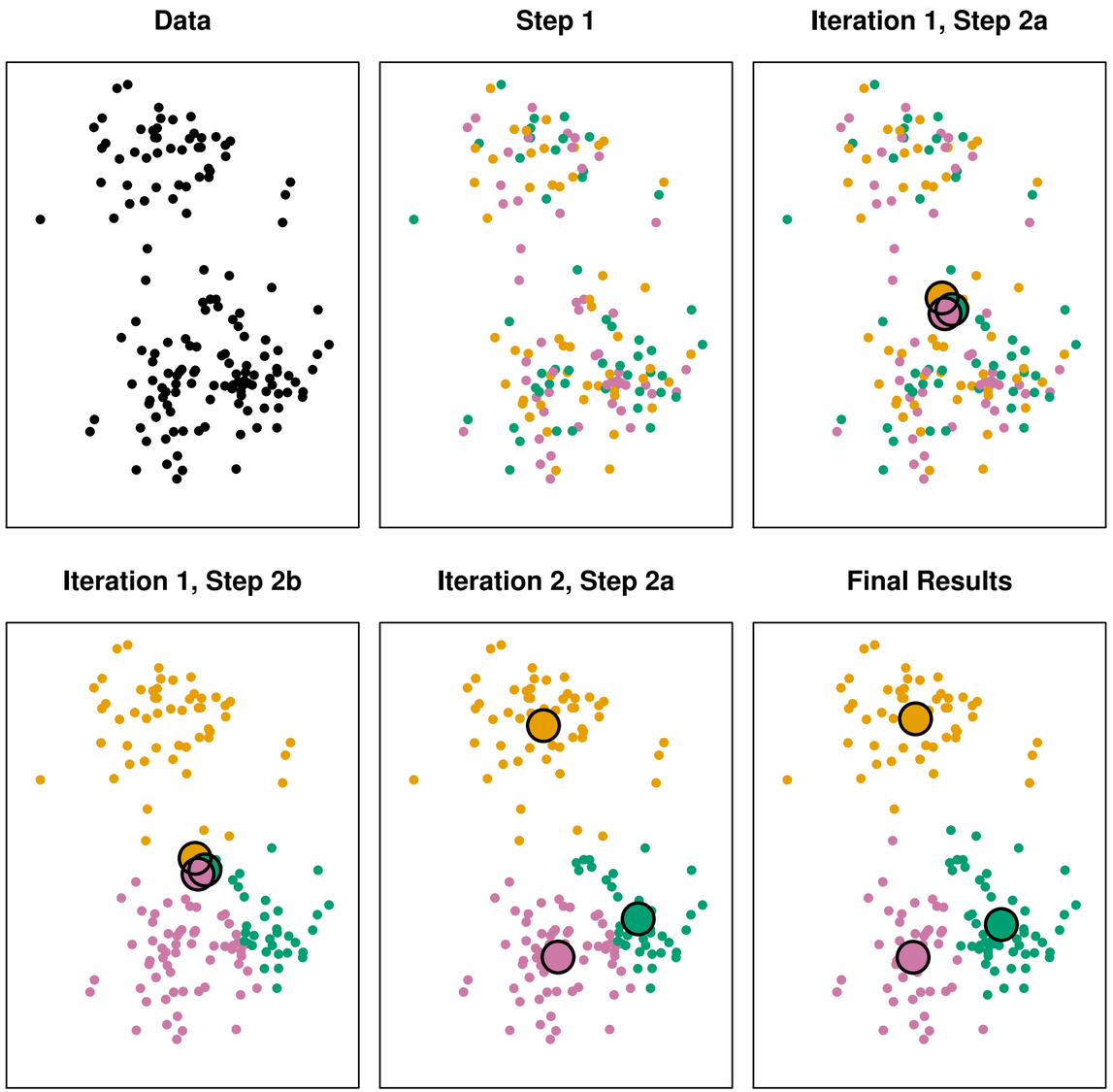
Where $\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$ is the mean for feature j in cluster C_k

K-Means clustering algorithm

Systematic assessment of all possible partitions is unfeasible. We need a smart optimisation algorithm.

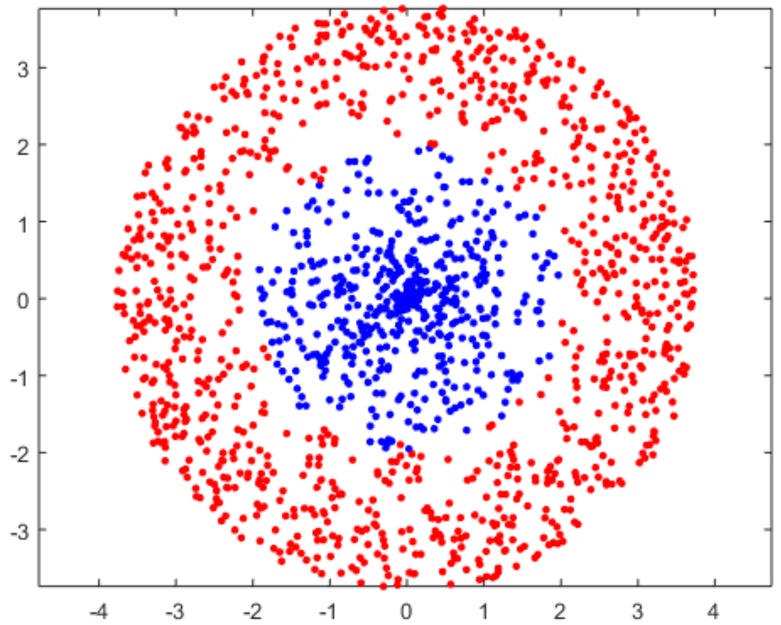
1. Randomly assign each observation to one cluster
2. Iterate the next two steps until cluster assignment stops changing
   - 2.a. For each cluster compute the mean vector $\overline{x}_k$ (i.e. the centroid)
   - 2.b. Assign each observation to the cluster whose centroid is closest (based on Eucledian distance)

K-Means clustering algorithm



What will happen if you apply K-means to this example with K=2

- It will correctly identify the two classes
- It depends on the initialisation
- It will not be able to correctly identify the two classes

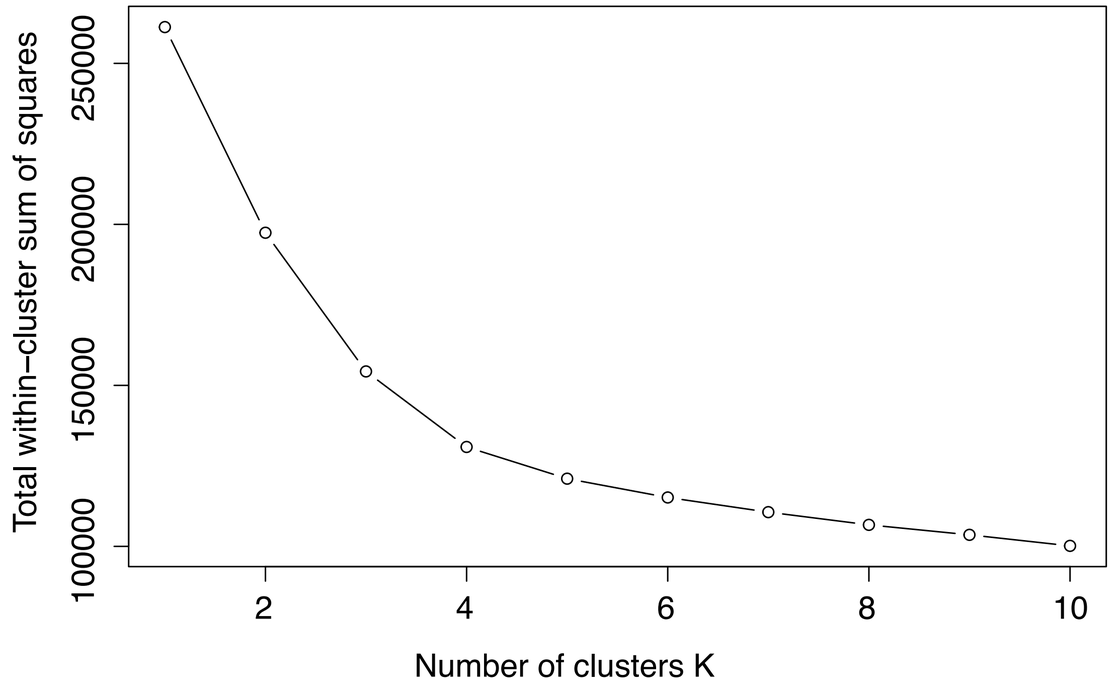


K-means algorithm problems and solutions

- Problem: Does not guarantee the global minimum. Solution: Check if switching single observations to a different group decreases the objective function.
- Problem: Different random initialisations can provide different solutions. Solution: Run it multiple times and select the solution with minimum objective function.
- We need to define the number of clusters.

K-means example: GDSC data

Compute K-means for different number of clusters and look at the total within-cluster sum of squares for each clustering.



Look at the elbow to define the optimal number of clusters.

K-means example: GDSC data

Number of cases of each cancer type (columns) in each cluster (rows), when using K=4.

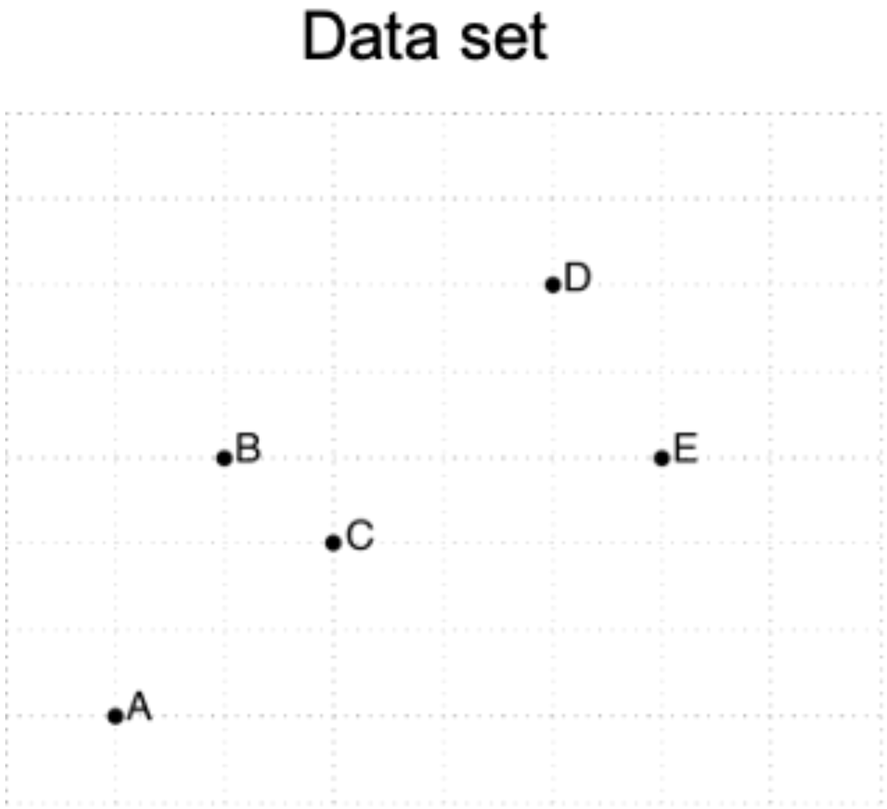
	breast	colorectal	kidney	neuroblastoma
1	6	0	28	1
2	41	7	0	0
3	0	37	0	0
4	0	1	0	27

Hierarchical clustering

- Differently from K-means, *hierarchical clustering* does not require to specify the number of clusters.
- It organises observations in a hierarchy, where clusters at each level of the hierarchy are created merging clusters at the next lower level.
- The approach that we will see is *bottom-up*:
  - It starts from the lowest level, where each observation is a singleton cluster.
  - For N-1 steps it merges a selected pair of clusters (i.e. the most similar) in a single cluster.
- The process can be visualised using a *dendrogram*.

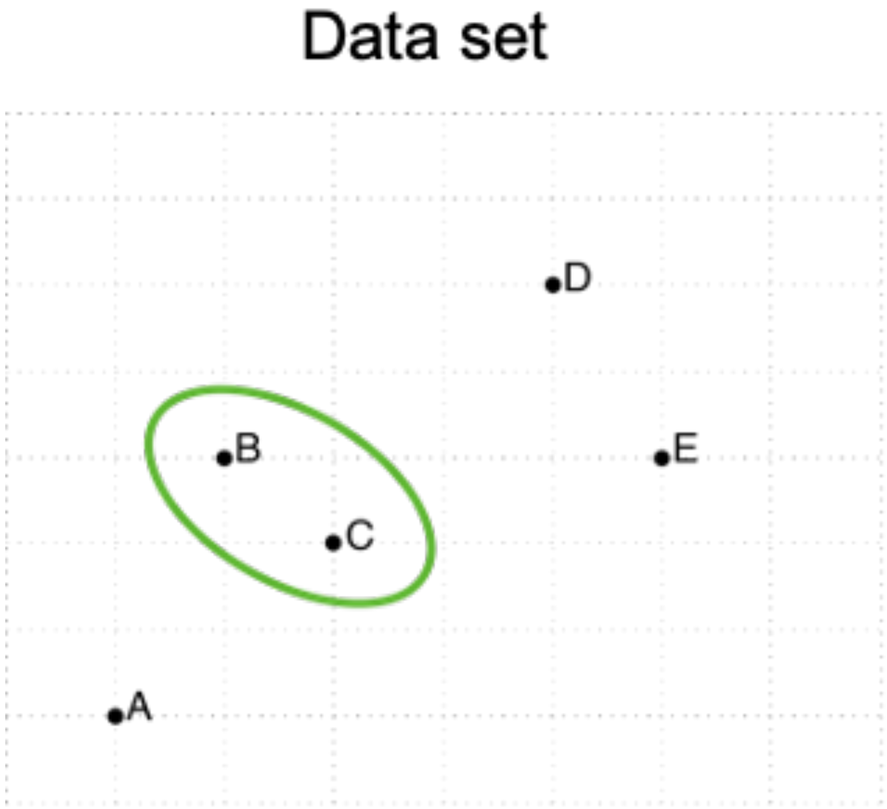
Hierarchical clustering - bottom-up approach

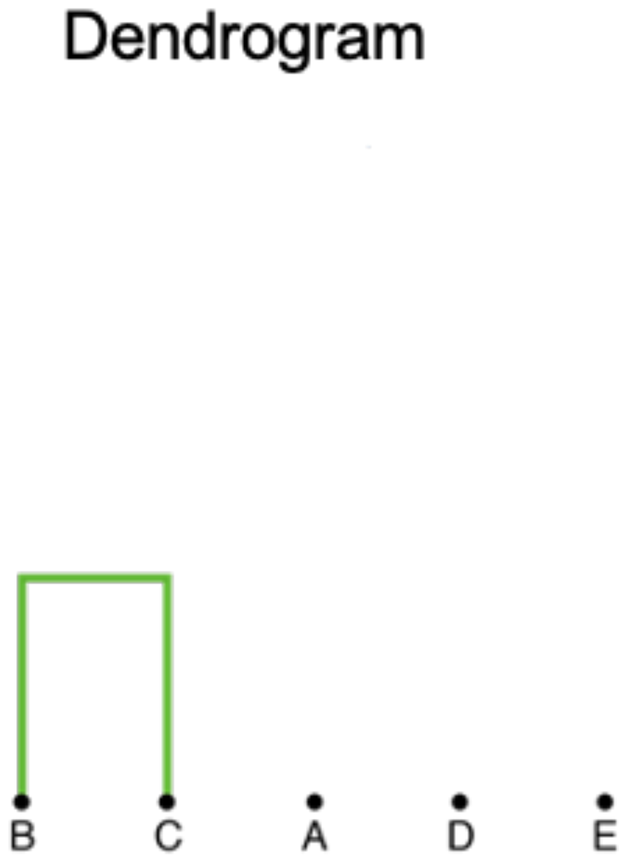
Each *leaf* of the dendrogram is an observation.



Hierarchical clustering - bottom-up approach

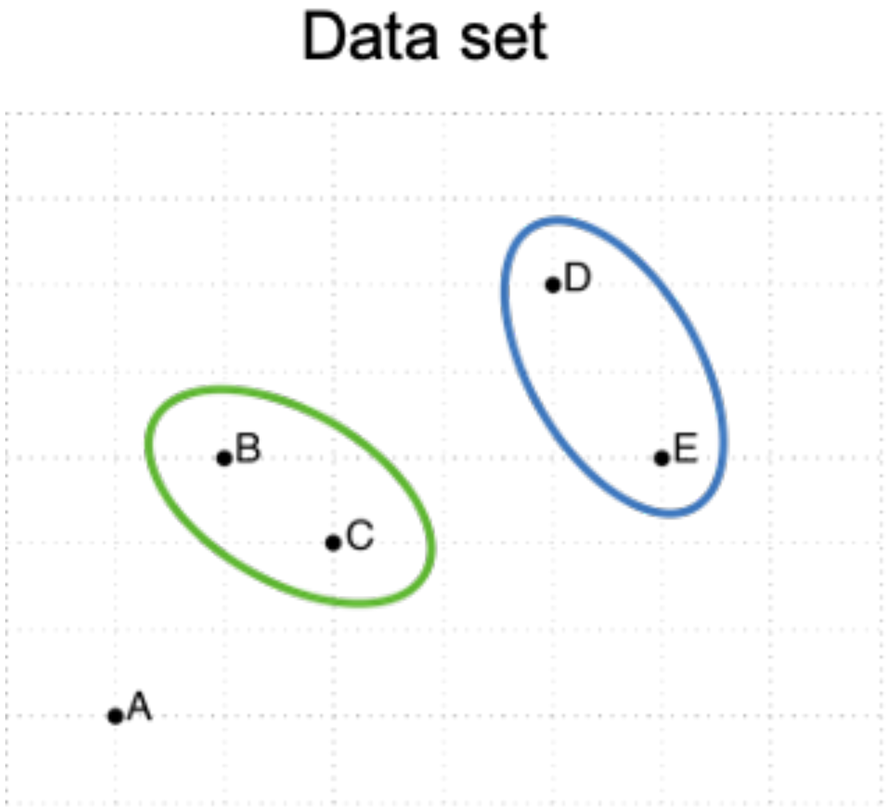
L*eaves* begin to fuse into *branches* for observations which are similar.

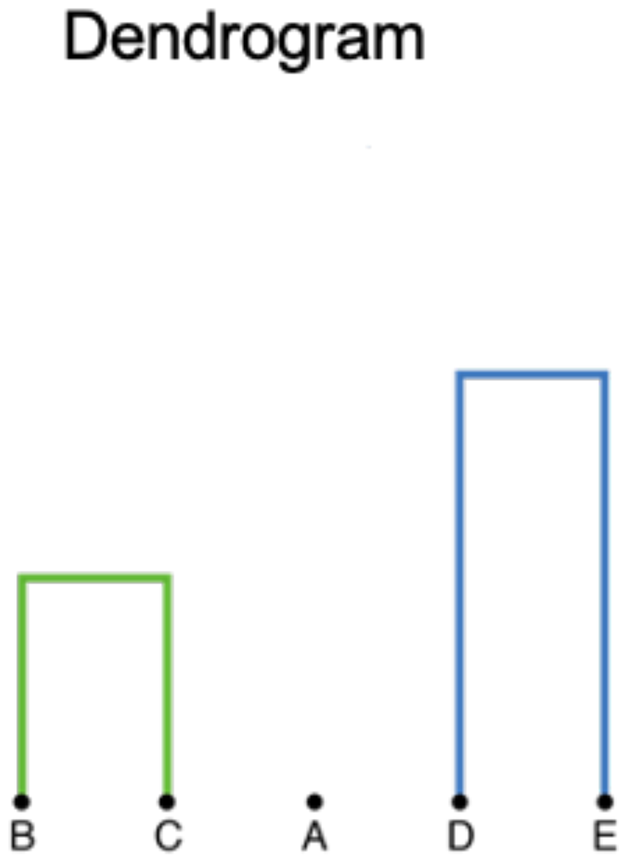




Hierarchical clustering - bottom-up approach

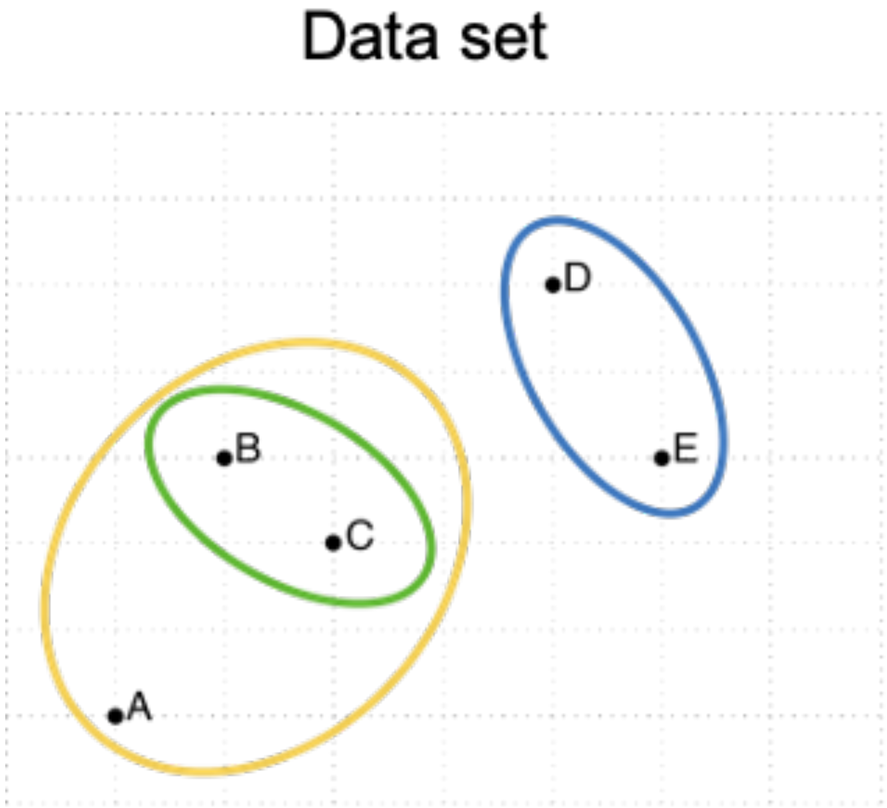
L*eaves* begin to fuse into *branches* for observations which are similar.

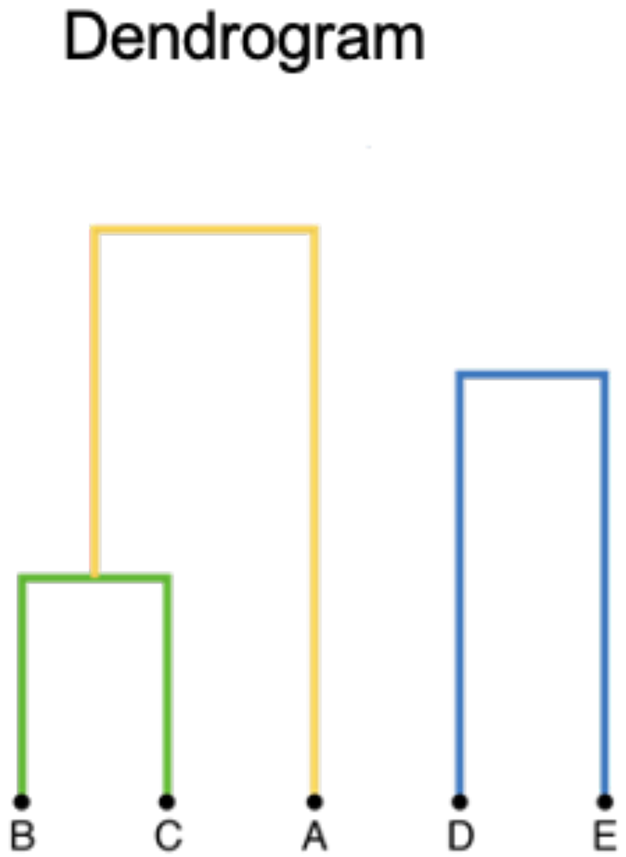




Hierarchical clustering - bottom-up approach

Branches start to fuse either with leaves...





Hierarchical clustering - bottom-up approach

Branches start to fuse either with leaves or with other branches.

Hierarchical clustering - Linkage

Given a metric of dissimilarity $d_{i,i'}$ between a pair of observations i and i' (e.g. Euclidean distance), we also need to define a measure of dissimilarity between clusters, i.e. the *linkage*.

Considering two clusters G and H, the most used types of linkage are:

- *Single linkage* $$d_{SL}(G,H) = \min_{i \in G, i' \in H} d_{i,i'}$$
- *Complete linkage* $$d_{CL}(G,H) = \max_{i \in G, i' \in H} d_{i,i'}$$
- *Average linkage* $$d_{AL}(G,H) = \frac{1}{n_G n_H} \sum_{i \in G} \sum_{i' \in H} d_{i,i'}$$

Where n_G and n_H are the number of observations in cluster G and H

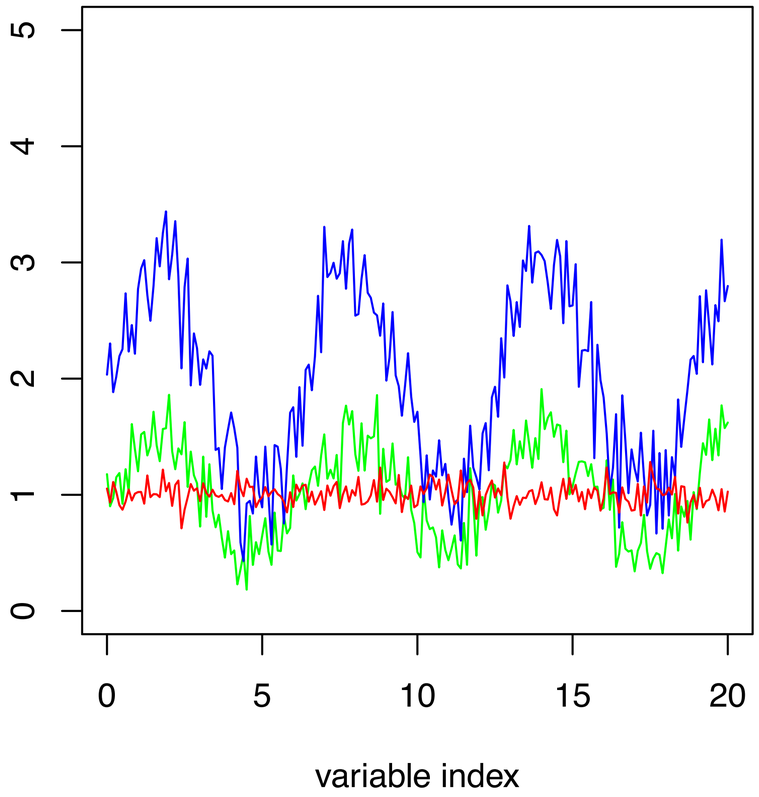
Hierarchical clustering - Dissimilarity metric

As metric of dissimilarity $d_{i,i'}$ between a pair of observations i and i' the most common approaches used are:

- Euclidean distance
- Correlation (of observations across features)

Which metric of dissimilarity between observations would you use?

These are time series data where each time point is a feature and we want to focus on similarity of the temporal profile.



- Euclidean distance
- Correlation

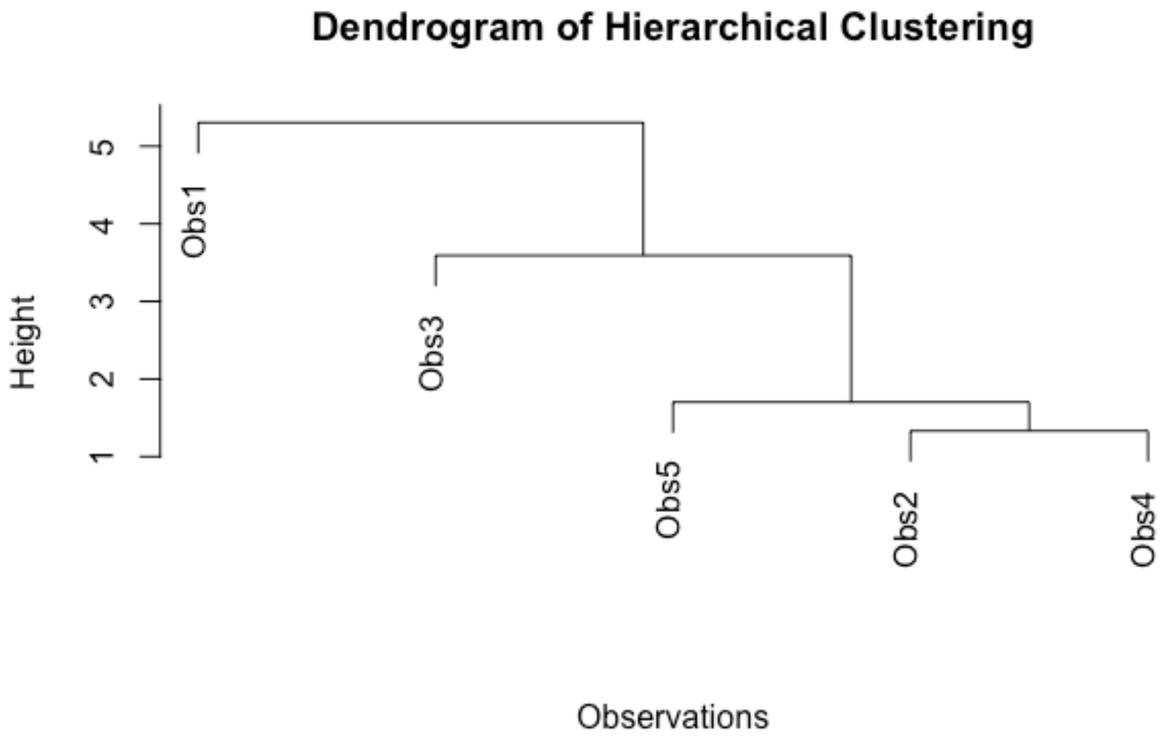
Hierarchical clustering - Interpreting the dendrogram

- Branches' high is proportional to the similarity between nodes
  - Observations that fuse at the bottom of the dendrogram are similar to each other
  - Observations that fuse at the top of the dendrogram are different from each other
- We can cut the dendrogram at a certain hight to obtain clusters

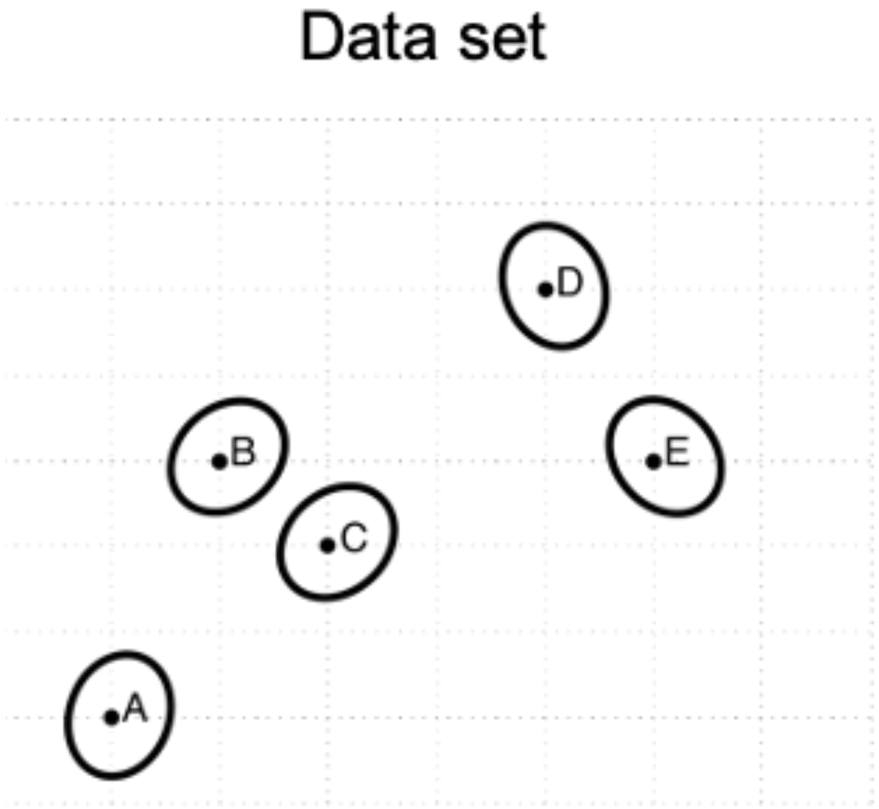
Which of the following two pairs of observations is more similar?

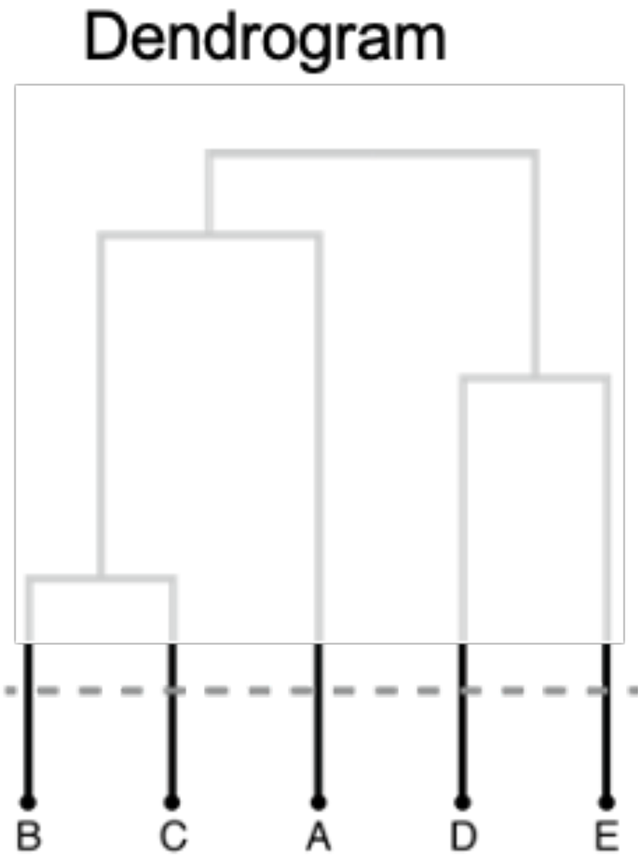
(Multiple choices possible)

- Observations 1 and 2
- Observations 1 and 3
- Observations 3 and 5
- Observations 3 and 2
- Observations 3 and 4

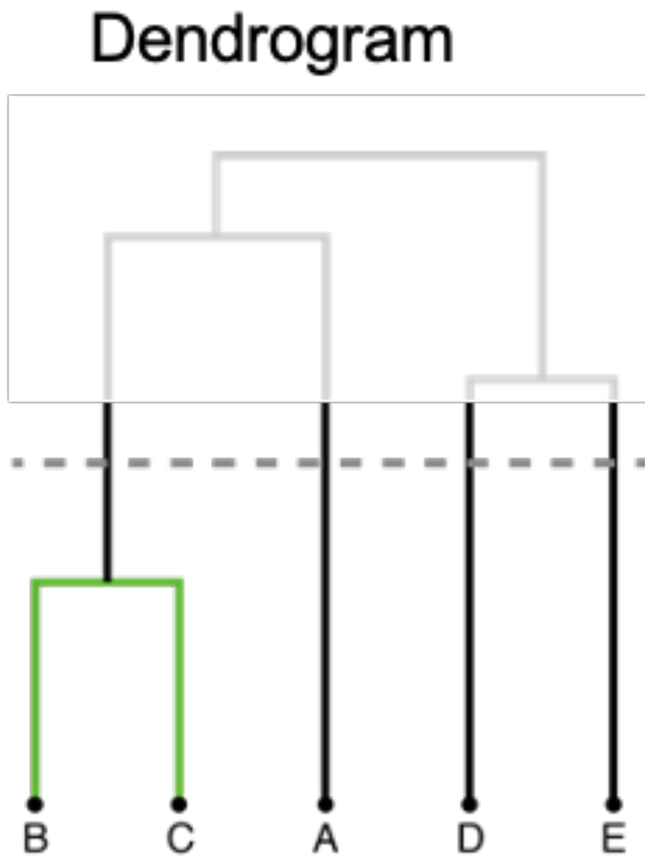


Hierarchical clustering - How to obtain clusters

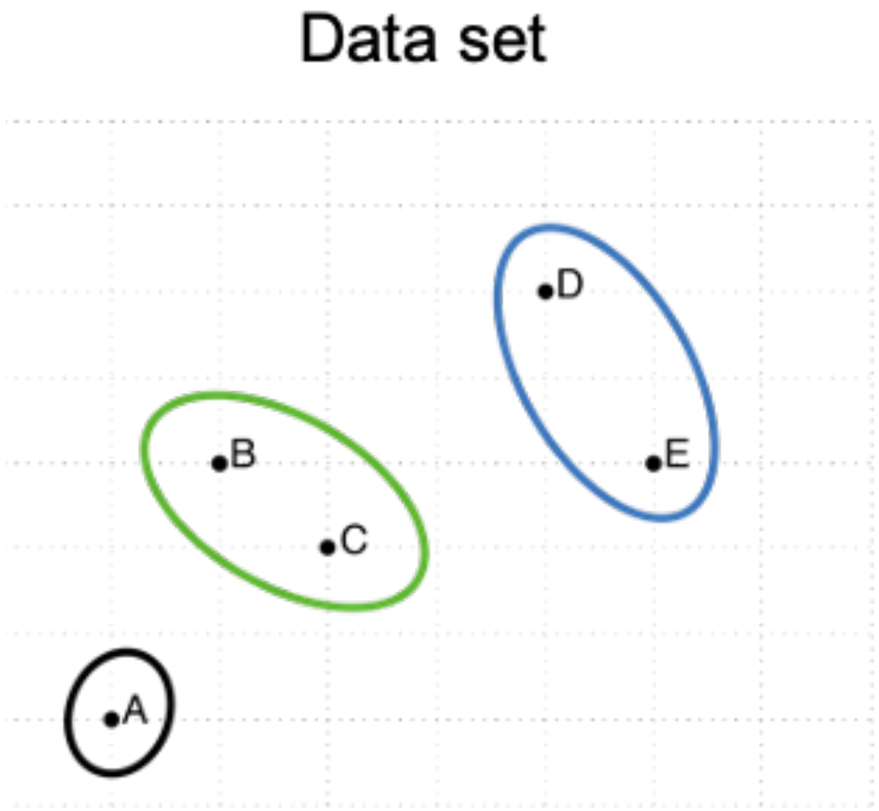


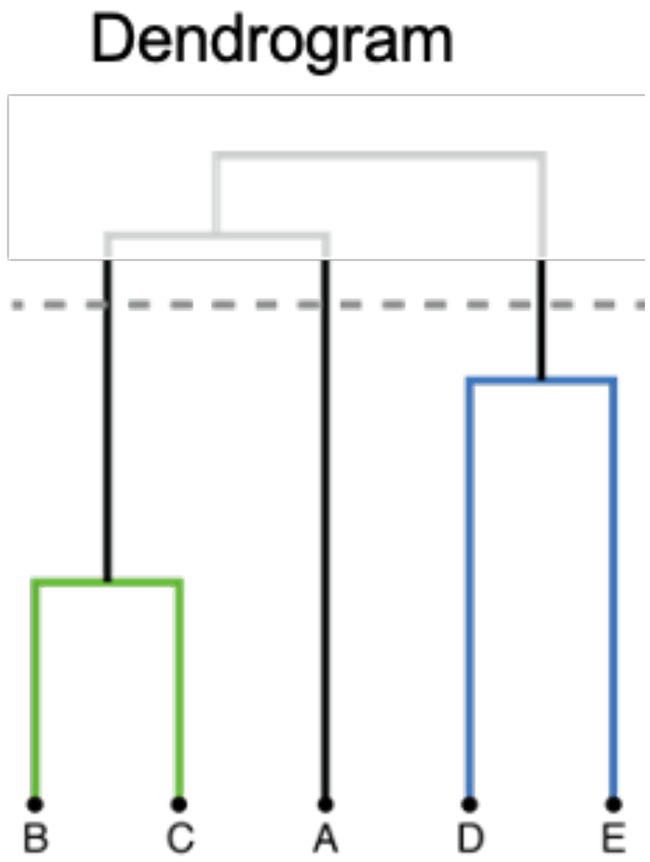


Hierarchical clustering - How to obtain clusters

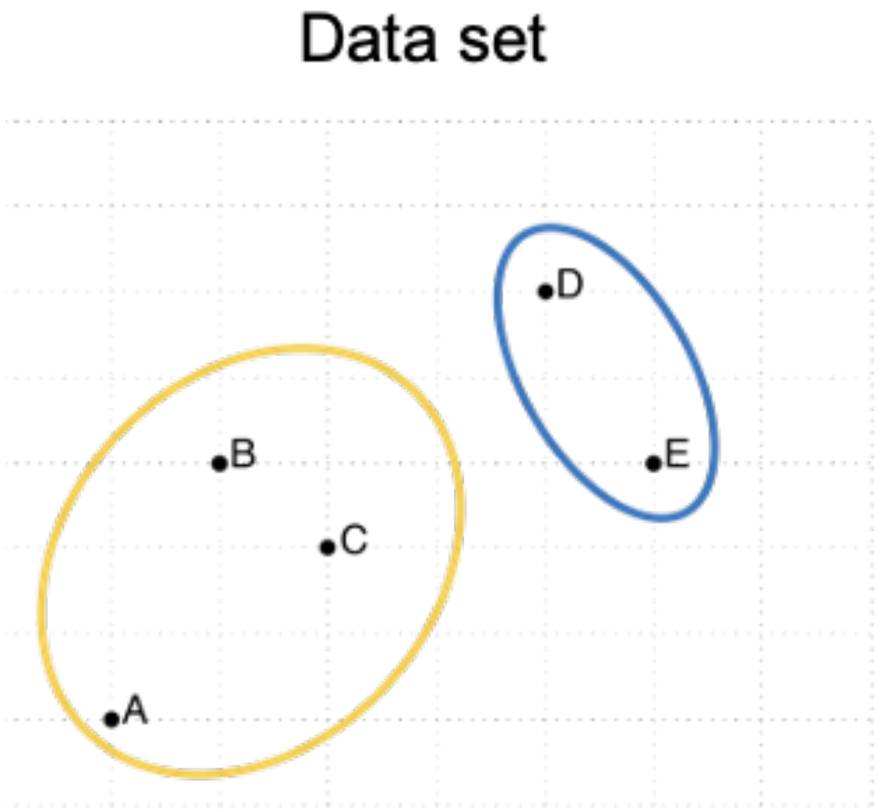


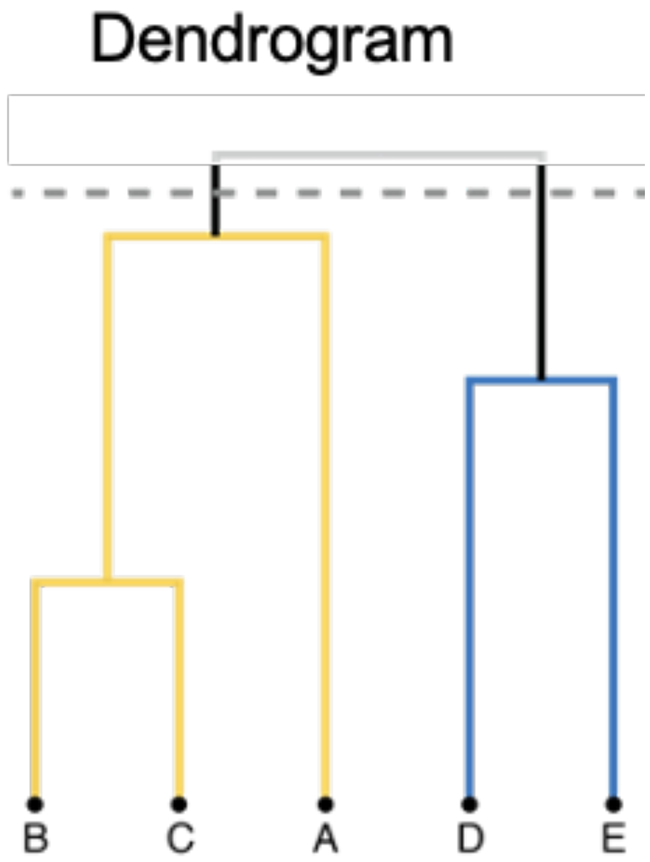
Hierarchical clustering - How to obtain clusters



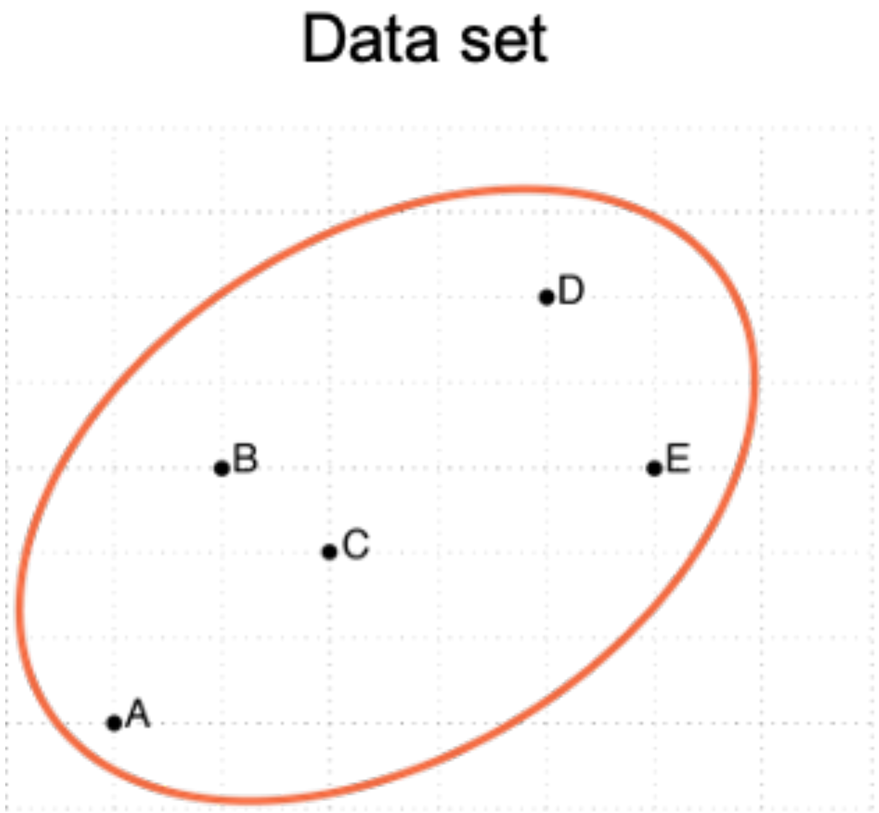


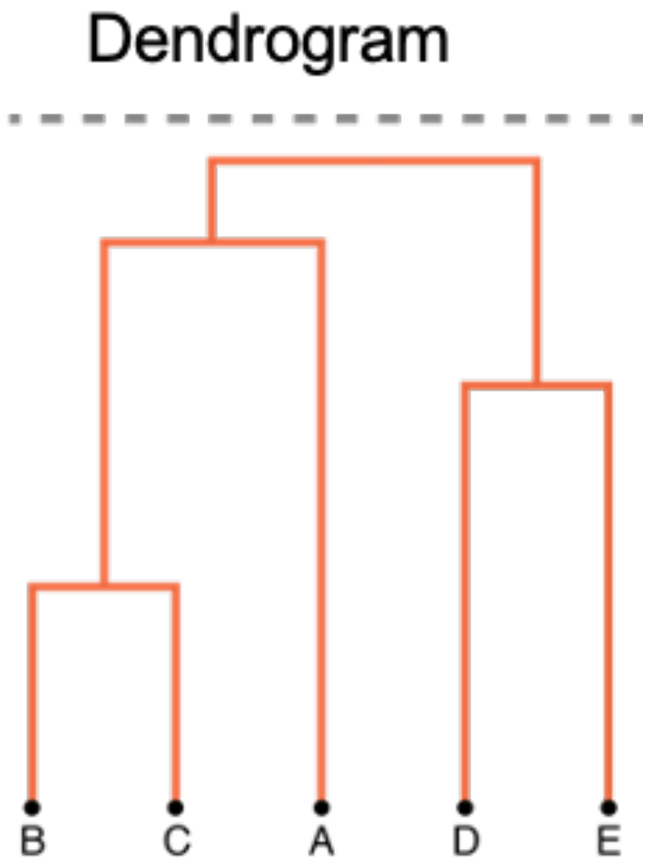
Hierarchical clustering - How to obtain clusters





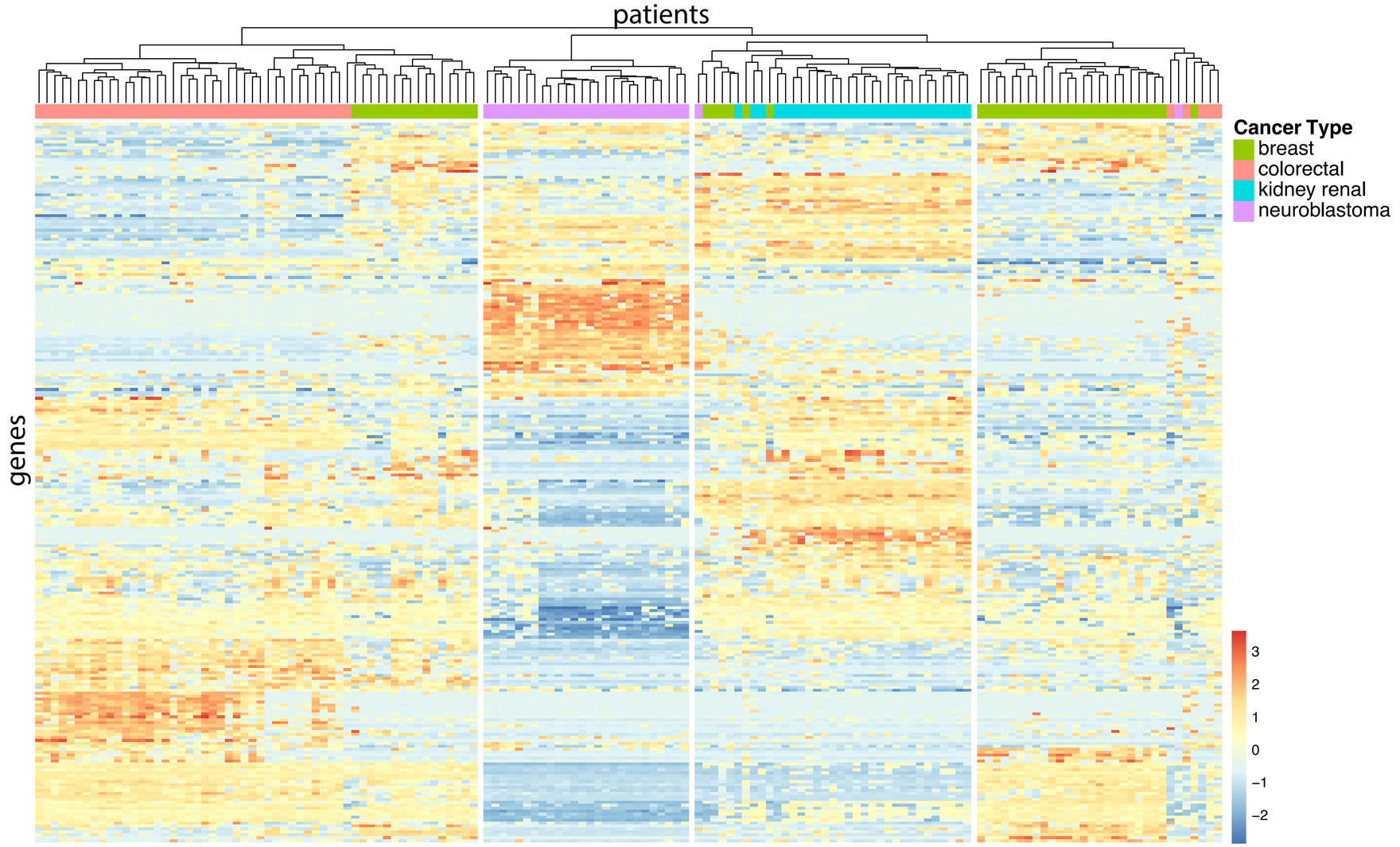
Hierarchical clustering - How to obtain clusters





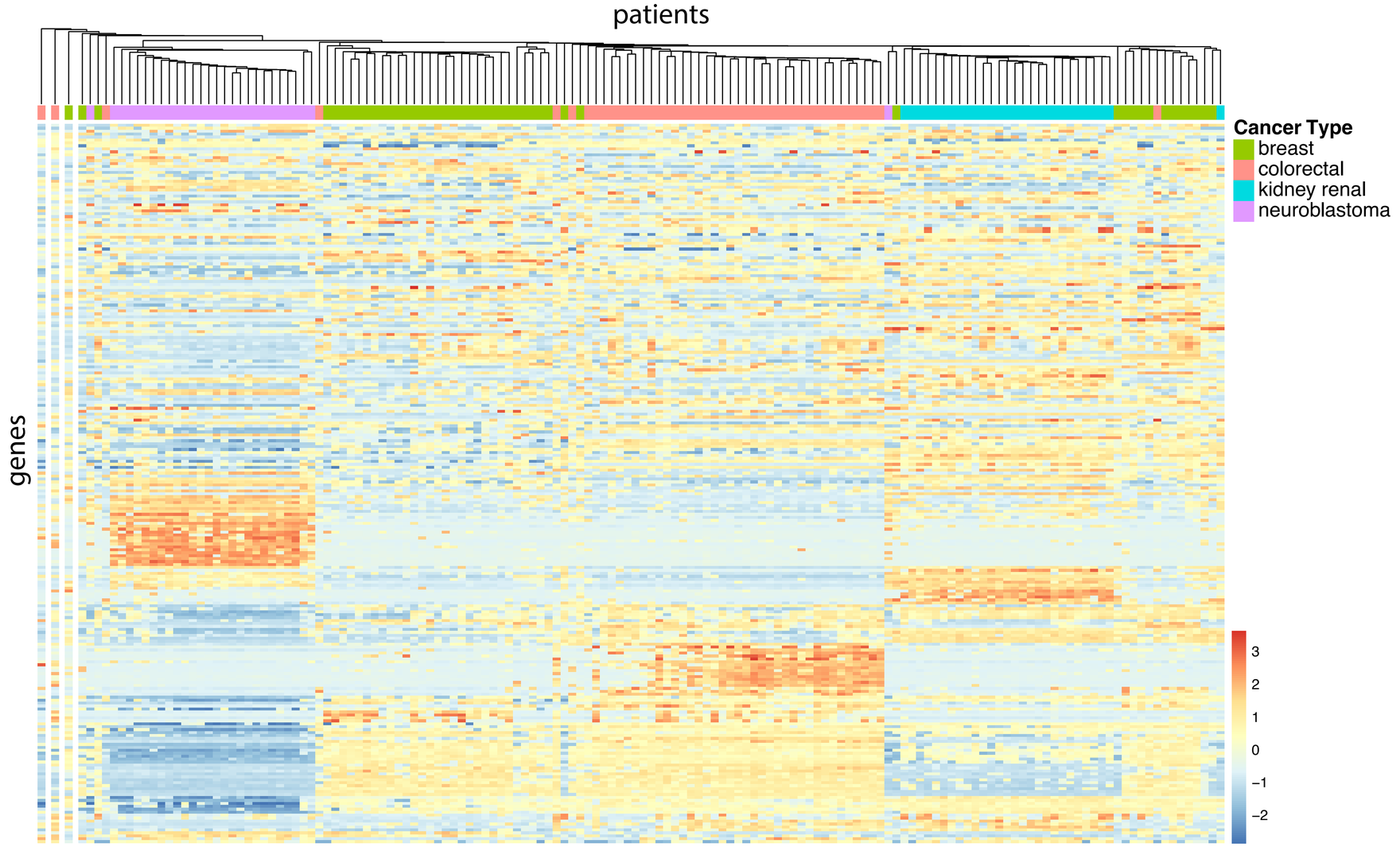
Hierarchical clustering - examples with GDSC data

Linkage: **Complete linkage** *Dissimilarity metric*: **Euclidean distance**



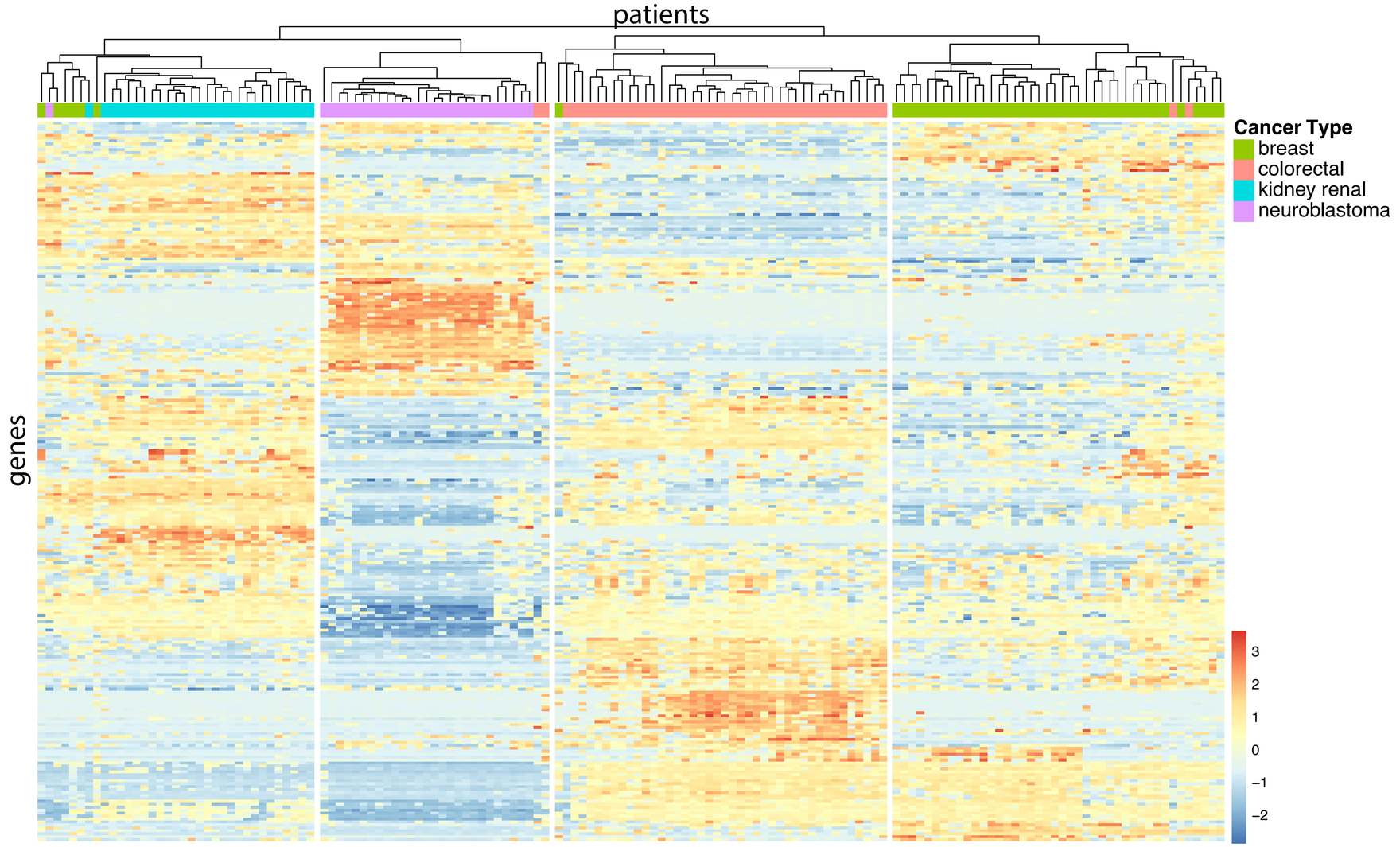
Hierarchical clustering - examples with GDSC data

Linkage: **Single linkage** *Dissimilarity metric*: **Euclidean distance**



Hierarchical clustering - examples with GDSC data

Linkage: **Complete linkage** *Dissimilarity metric*: **Correlation**



Summary of clustering

- **Goal:** Group data points based on similarity.
- **Key techniques:**
  - **K-means:**
    - Partition data into K clusters by minimising within-cluster variation.
    - Requires predefining the number of clusters.
  - **Hierarchical Clustering (bottom-up):**
    - Starts with each point as its own cluster and merges them step by step.
    - Produces a dendrogram that shows the merging process.
- **Examples of biomedical Applications:**
  - Grouping patients based on clinical measurements to identify subgroups with similar conditions.
  - Clustering genes based on expression patterns to reveal functional relationships.

Questions?