Unsupervised learning

Federica Eduati

Eindhoven University of Technology
Department of Biomedical Engineering

2020

Learning goals

At the end of this lecture you will:

- Have a general understanding of what is unsupervised learning.
- Have a general understanding what is dimensionality reductions and some methods to do it.
- Have a general understanding of what is clustering and some methods to do it.

Materials:

- Chapters 14 from Friedman et al., *The Elements of Statistical Learning*.

Overview

Topics of the lecture:

- What is unsupervised learning?
- Methods for dimensionality reduction
  - Principal components analysis
  - t-SNE
- Methods for clustering
  - K-means clustering
  - Hierarchical clustering

Difference between supervised and unsupervised learning

With both supervised and unsupervised learning we have a set of features $X_1, X_2, \ldots, X_p$.

- with **supervised learning** (or *learning with a teacher*) we also have a 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.

Two categories:

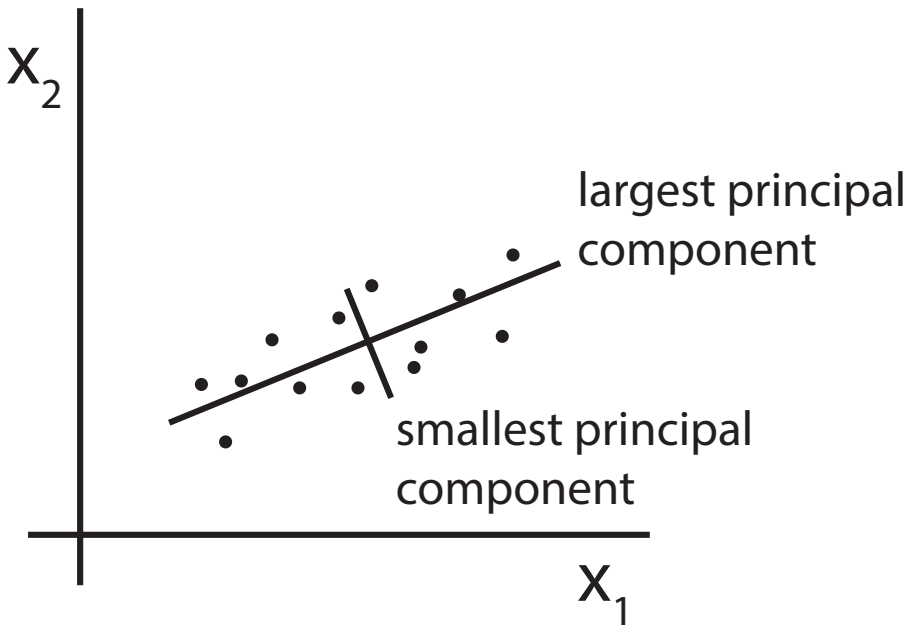
- **Dimensionality reduction**: reduce the the dimensions of the input features to facilitate visualisation and identification of groups. Also used as preprocessing before applying supervised techniques.
- **Clustering**: techniques to discover unknown groups (clusters) in data.

Compared to supervised learning:

- No need for labels (often difficult to retrieve)
- 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)

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.

$$Z_1 = v_{11}X_1 + v_{21}X_2 + \cdots + v_{p1}X_p$$

where $\sum_{j=1}^{p} v_{j1}^2 = 1$, and $v_{11}, v_{21}, \ldots, v_{p1}$ are the *loadings* of the first principal component and $v_1 = (v_{11}, v_{21}, \ldots, v_{p1})$ is the *first principal component loading vector*.

PCA computation - first principal component (PC1)

We have our 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 interested only in the variance we assume that each feature has 0 mean, i.e. $\mathbf{X}$ has columns with mean zero.

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

$$z_{i1} = v_{11}x_{i1} + v_{21}x_{i2} + \cdots + v_{p1}x_{ip}$$

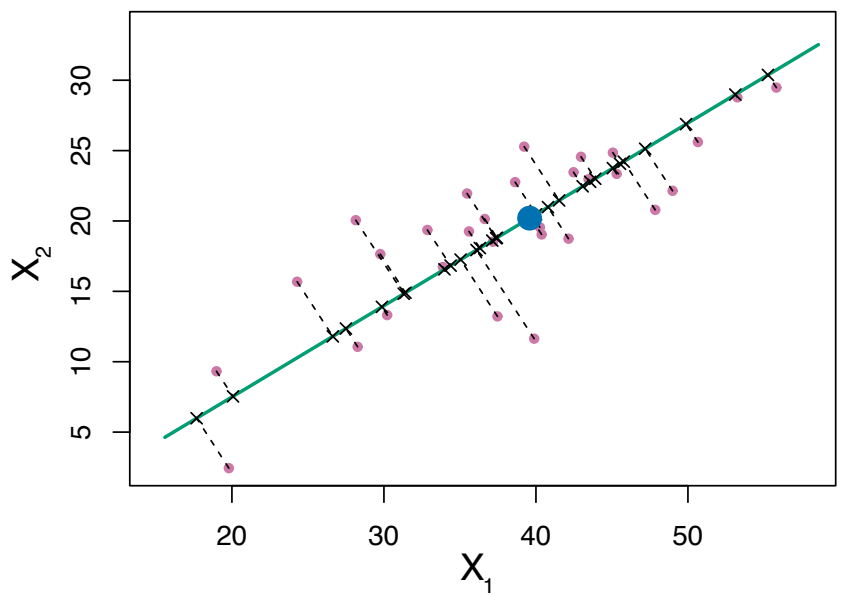
with the largest sample variance, i.e.

$$\max_{v_{11},\ldots,v_{p1}} \{\frac{1}{N}\sum_{i=1}^{N}(\sum_{j=1}^{p} v_{j1}x_{ij})^2\} \text{ subject to } \sum_{j=1}^{p} v_{j1}^2 = 1$$

Since each x_{ij} has mean zero, also does z_{ij}. Hence we are maximising the sample variance of Z_1 which is $\frac{1}{N}\sum_{i=1}^{N} z_{i1}^2$.

PCA computation - PC1 geometrical interpretation

- The loading vector v_1 with elements $v_{11}, v_{21}, \ldots, v_{p1}$ define the direction along which the data vary the most.
- The projections of the N points $x_1, x_2, \ldots, x_N$ onto this direction are the principal component scores $z_{11}, z_{21}, \ldots, z_{N1}$.

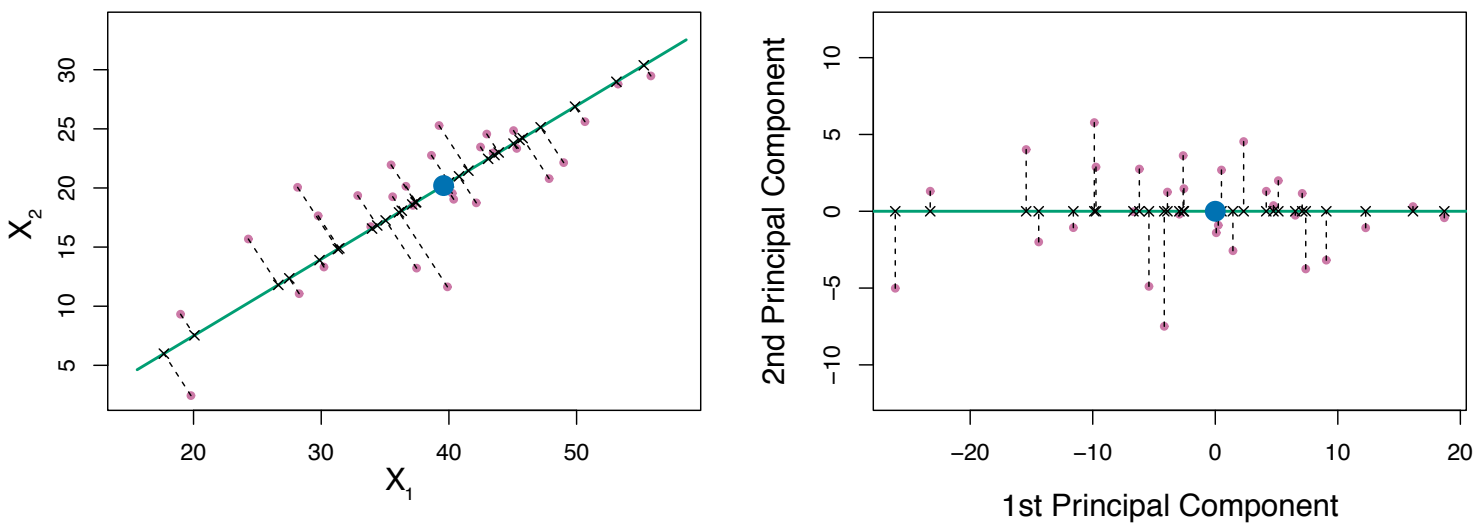


PCA computation - further principal components

Now that we have the fist principal component Z_1 we want to find the *second principal component*. Z_2.

Z_2 needs to be uncorrelated with Z_1. This is equivalent to constrain the direction of the loading vector v_2 is orthogonal to the direction of v_1.

Same for the following principal components.



PCA computation using SVD

This optimisation problem can be solved via *singular value decomposition* of the matrix $\mathbf{X}$:

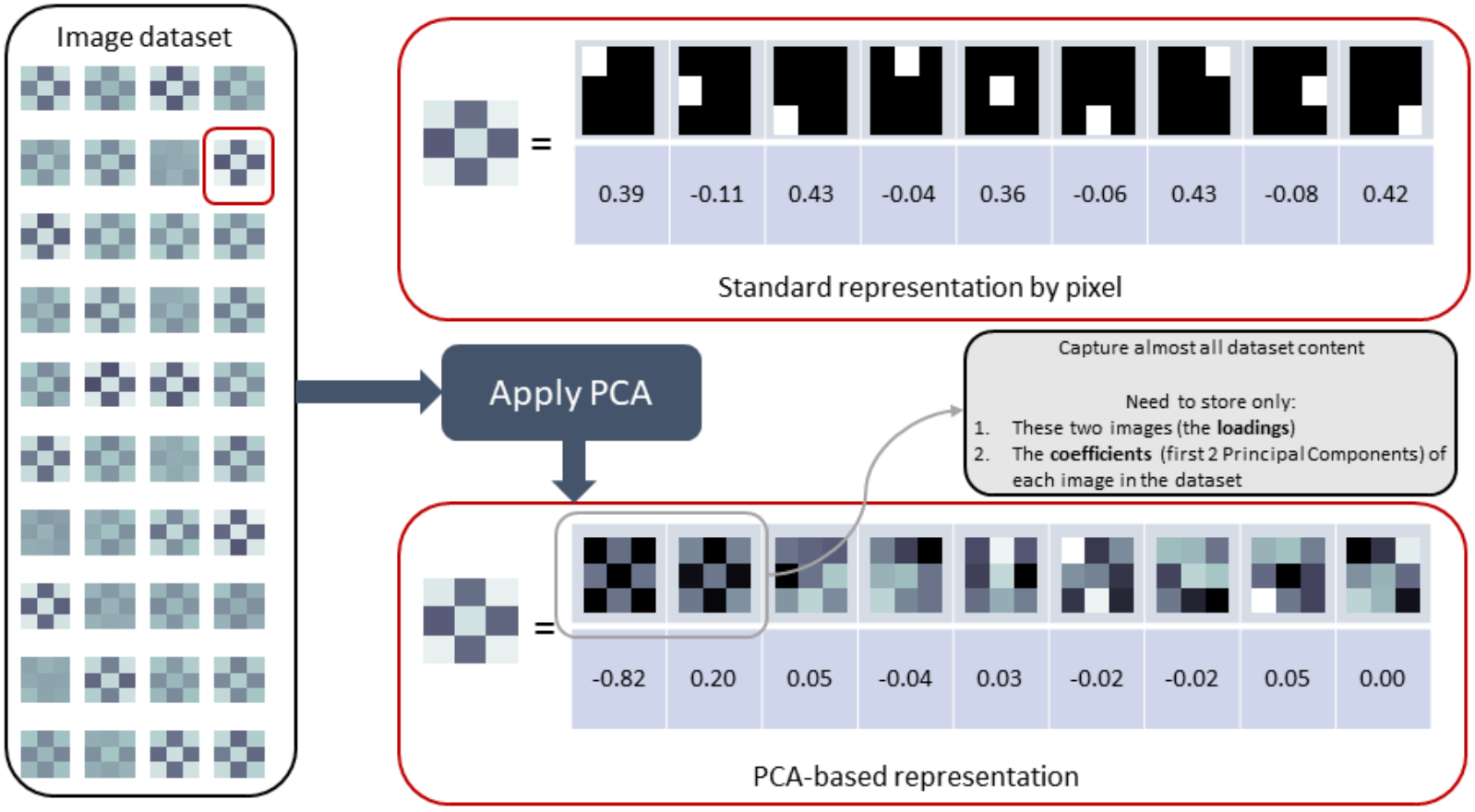
$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

which is a unique decomposition such that (for $N \geq p$):

- $\mathbf{U}$ is a $N \times p$ orthogonal matrix ($\mathbf{U}^T\mathbf{U} = \mathbf{I}_p$)
- $\mathbf{D}$ is a $p \times p$ diagonal matrix with $d_i \geq 0$ and $d_i \geq d_{i+1}$ known as the *singular value*
- $\mathbf{V}$ is a $p \times p$ orthogonal matrix ($\mathbf{V}^T\mathbf{V} = \mathbf{I}_p$)

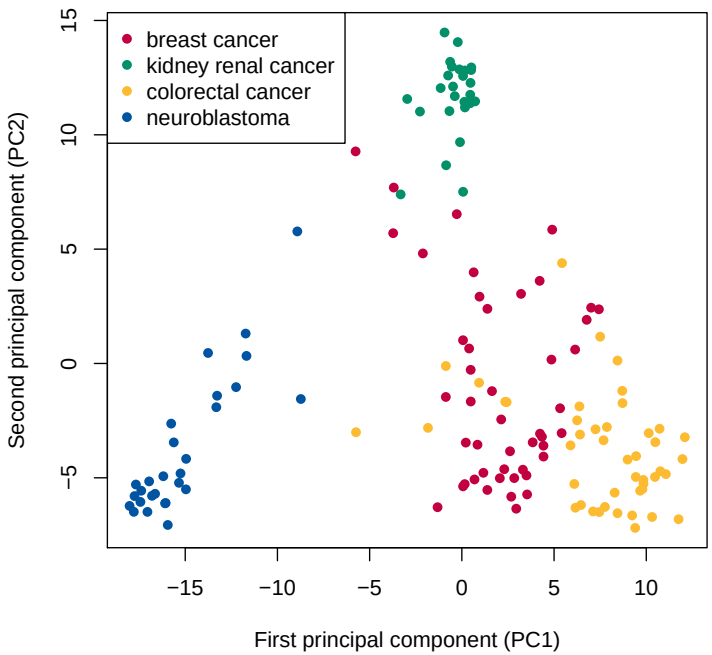
The columns of $\mathbf{UD}$ are the *principal components* of $\mathbf{X}$ ($PC1 = u_{i1}d_1, PC2 = u_{i2}d_2, \dots$) and $\frac{d_1^2}{N}, \frac{d_2^2}{N}, \dots$ is the variance explained by each principal component.

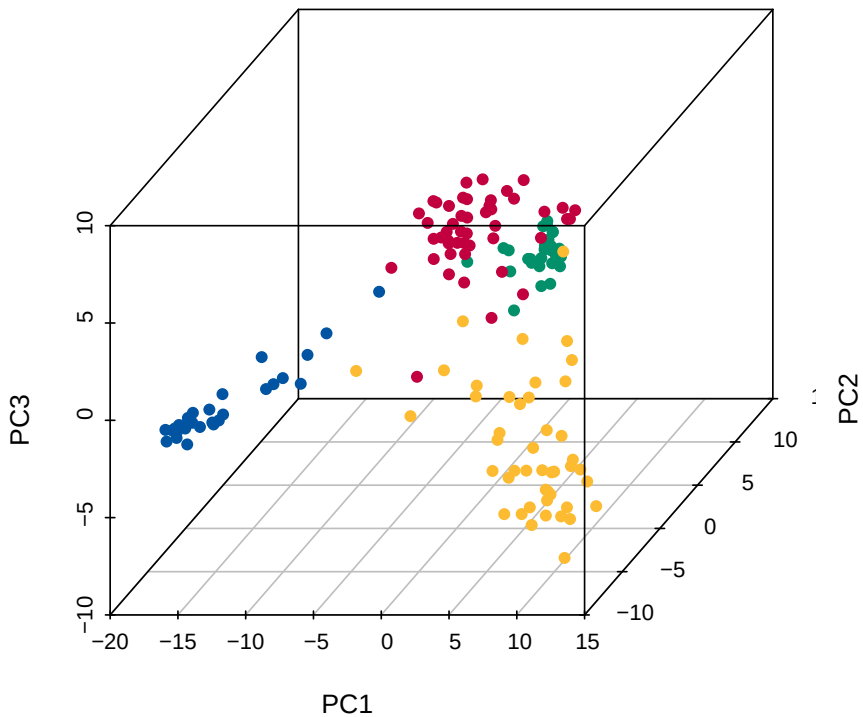
PCA example using images



PCA example using GDSC dataset

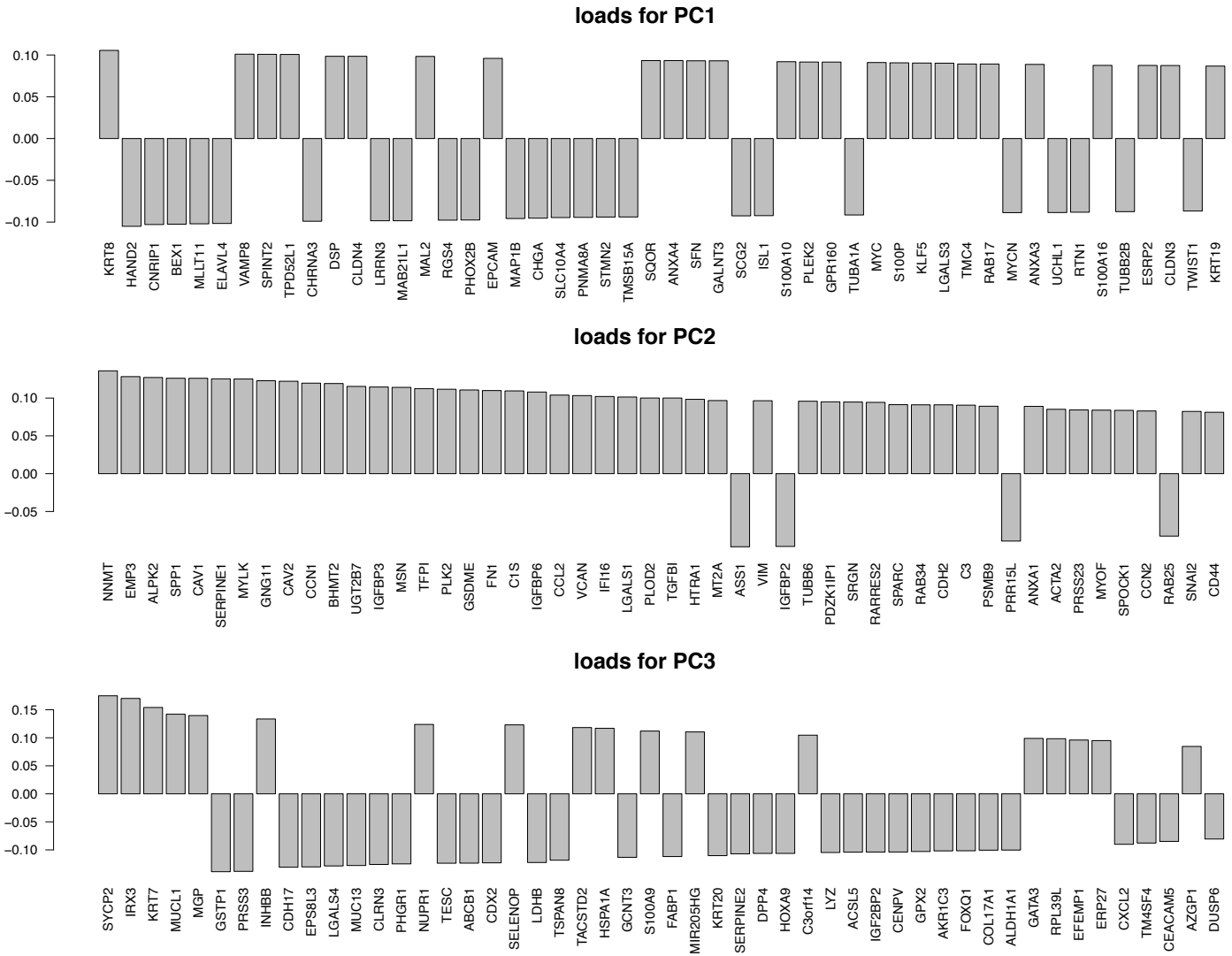
RNA expression data (244 genes) for 148 cell lines from four cancer types. Samples are *a posteriori* coloured by cancer type.





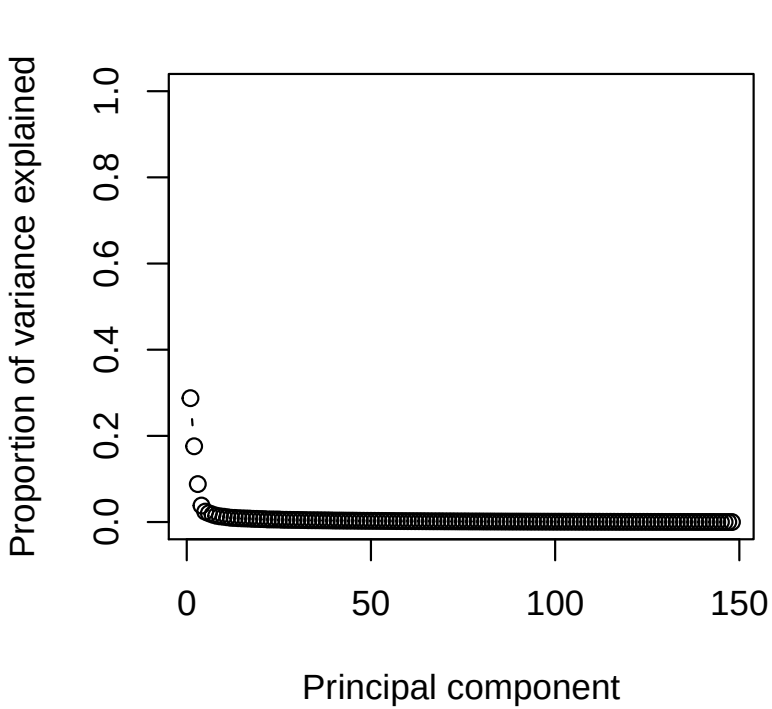
PCA example using GDSC dataset - loads

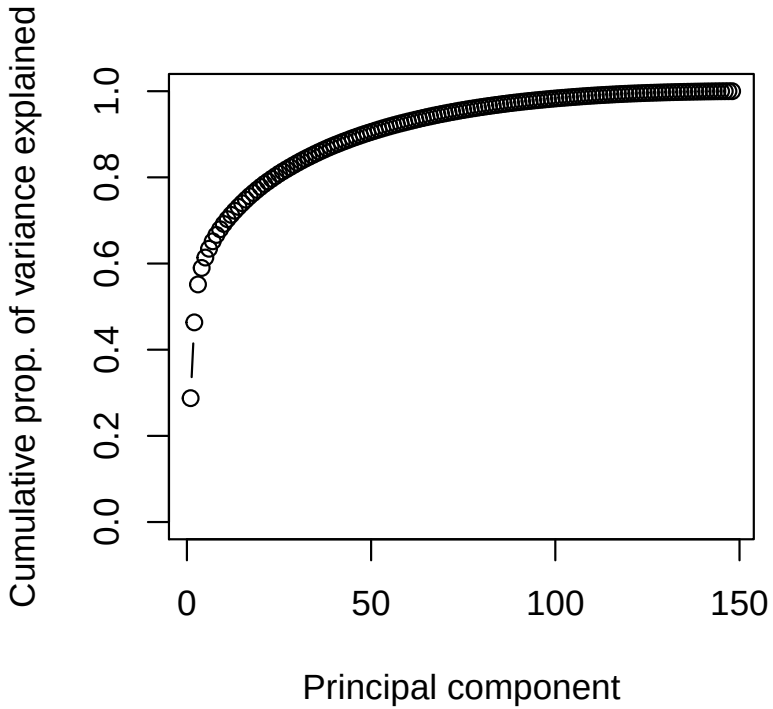
Loads of the first 3 principal components (only top 50 genes)



PCA example using GDSC dataset - variance explained

Variance explained by the principal components.





Elbow in the "proportion of variance explained" plot can be used as a criteria to decide how many principal components to use.

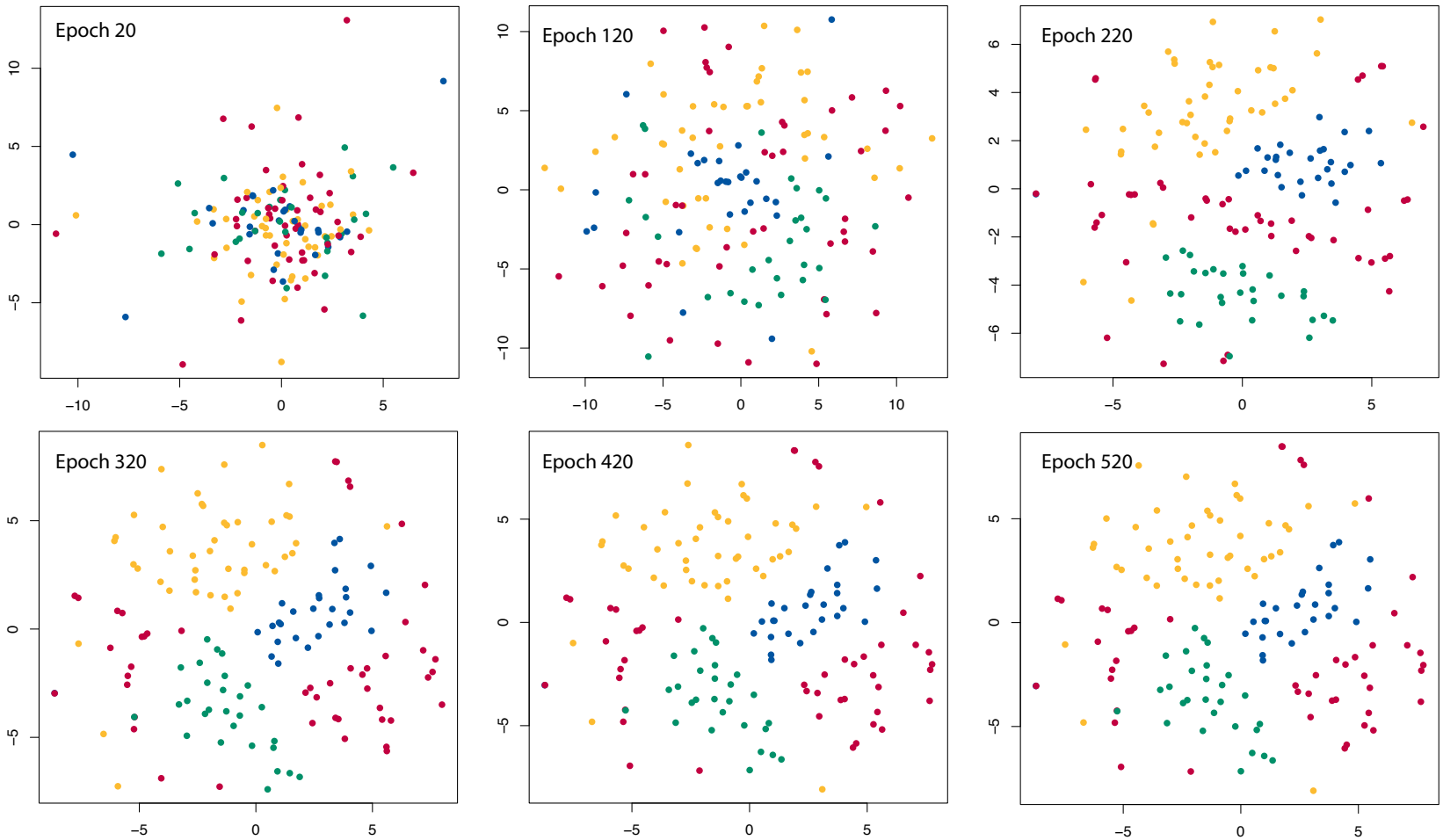
t-SNE

- PCA is a linear algorithm, i.e. principal components are linear combinations of the features.
- t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear techinque for dimensionality reduction.
- t-SNE works very well in high dimensional data.
- Cons: it is computationally demanding, it is stochastic and it is governed by hyperparameters.

t-SNE computation

- Computes a measure of pairwise similarity in the original (multi-dimentional) feature space;
- Tries to minimise the difference between the similarity in the high-dimensional space and the similarity in a lower-dimensional space (typically 2 or 3 dimensions);
- The measure of similarity in the high- and low- dimensional space is different and this allows to visualise the clusters as more homogeneous.

t-SNE example using GDSC data



Clustering

- Aim: Group observations into subsets or *clusters* or *segments* so that observations within a cluster are more similar to each other than observations assigned to different cluster.
- Requires a definition of *similarity* or *difference*.
- This is similar to the definition of the cost function for supervised learning, the most appropriate definition of *similarity* or *difference* depend on the type of data.

K-means clustering

- Assign each observation $i \in \{1, 2, \ldots, N\}$ to one cluster $k \in \{1, 2, \ldots, K\}$.
- K need to be predefined, and $K < N$
- This assignment correspond to a many-to-one mapping $k = C(i)$, which is an *encoder* that assigns the ith observation to the kth cluster.
- Each observation is assigned to one and only one cluster.

K-means clustering

We want to define the clusters so that similar points are in the same cluster and dissimilar points are in different clusters. Defining $d_{i,i'} = d(x_i, x_{i'})$ as a measure of dissimilarity between a pair of observations x_i and $x_{i'}$, the total point scatter T is :

$$T = \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N} d_{i,i'} = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\Big(\sum_{C(i')=k} d_{i,i'} + \sum_{C(i)\neq k} d_{i,i'}\Big)$$

Where:

- $W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k} d_{i,i'}$ is the *within-cluster* point scatter that we want to minimise
- $B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')\neq k} d_{i,i'}$ is the *between-cluster* point scatter that we want to maximise

Minimising $W(C)$ or maximising $B(C)$ is equivalent.

K-means clustering

With K-means all variables has to be quantitative and we use the Eucledian distance as a measure of similarity:

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = \|x_i - x_{i'}\|^2$$

The within-point scatter can be written as:

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} \|x_i - x_{i'}\|^2 = \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2$$

where N_k is the number of points in the kth cluster and $\bar{x}_k = (\bar{x}_{1k}, \ldots, \bar{x}_{pk})$ is the mean vector associated with the kth cluster

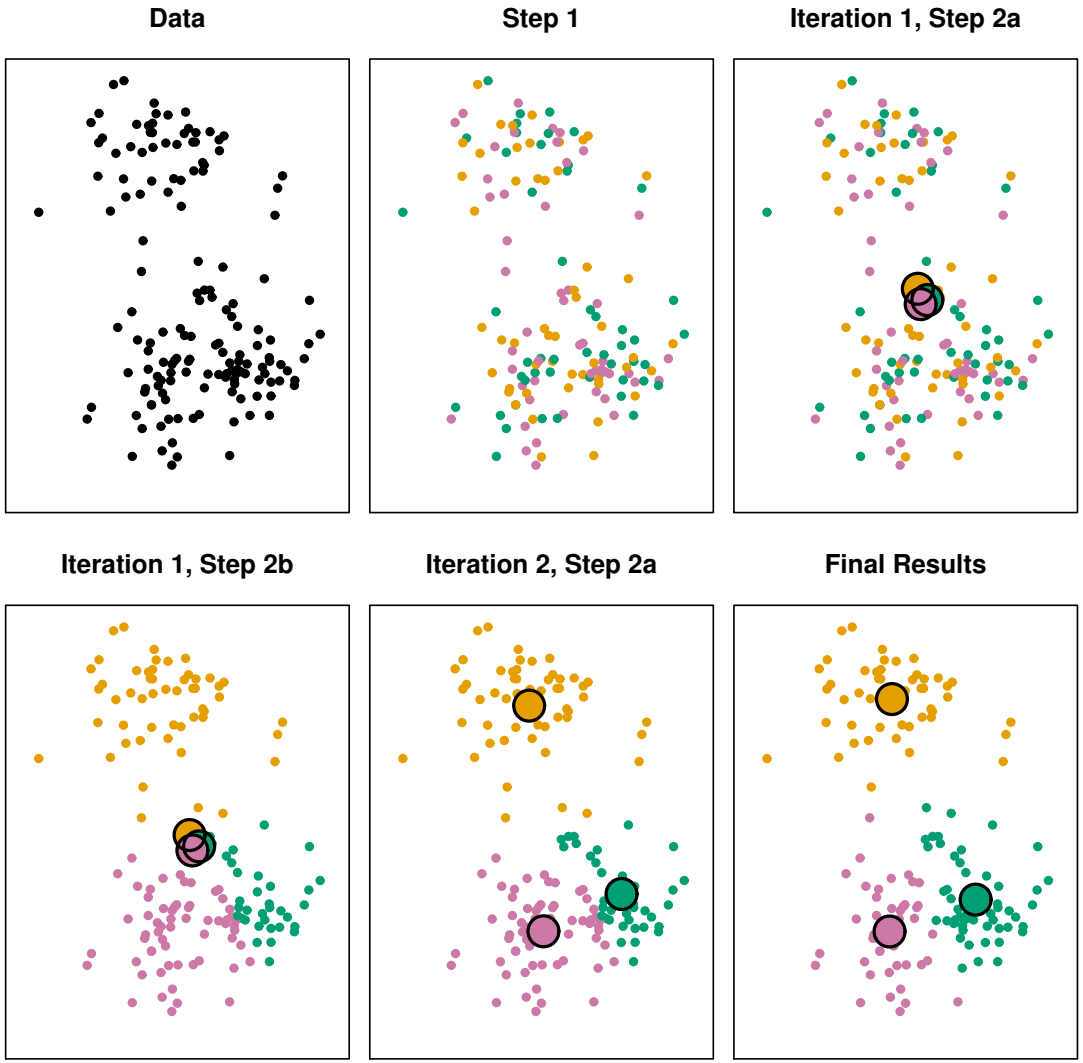
K-means clustering algorithm

For high number of points N it is infeasible to test all possible clustering assignments.

We need *iterative greedy descent* strategies to iteratively improve clustering assignments:

1. Randomly assign each observation to one cluster.
2. Iterate the next two steps until cluster assignment stops changing
   - 2.1 For each cluster compute the mean vector x_k (i.e. *centroid*)
   - 2.2 Assign each observation to the cluster whose centroid is closest (based on Eucledian distance).

K-means iterations

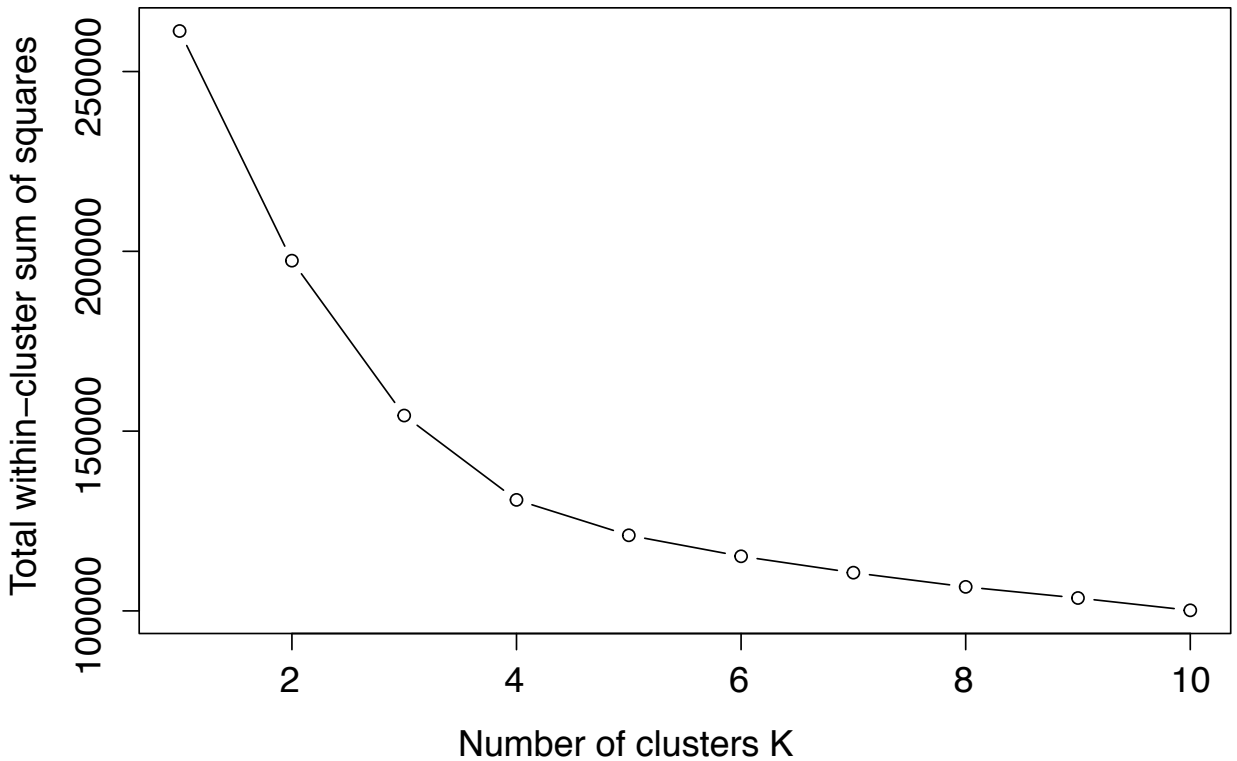


K-means problems and solutions

- Problem: Does not guarantee the global minimum. Solution: we can check that no single switch of an observation to a different group decreases the objective function.
- Problem: Different random initialisation can provide different solutions. Solution: We can run it multiple times and select the solution with minimum objective function.
- We need to define the number of clusters.

K-means example using GDSC data

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



Look at the elbow to define optimal number of cluster (not always possible).

K-means example using GDSC data

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

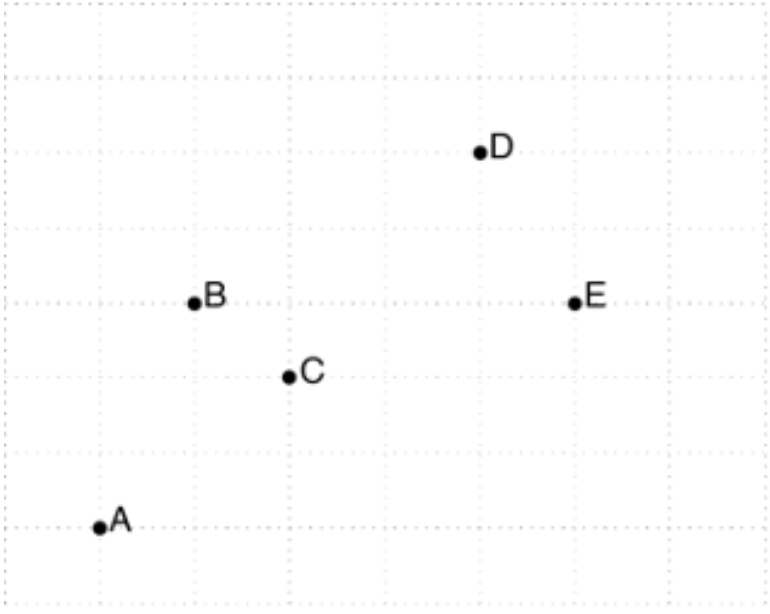
	breast	colorectal	kidney	neuroblastoma
1	6	0	28	1
2	41	7	0	0
3	0	37	0	0
4	0	1	0	27

Hierarchical clustering

- Differently from K-means, *hierarchical clustering* does not require to specify a 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*
  1. It starts from the lowest level, where each observation is a singleton cluster.
  2. For $N - 1$ steps it merges a selected pairs if clusters (i.e. the most similar) in a single cluster.
- This process can be visualized as a *dendogram*

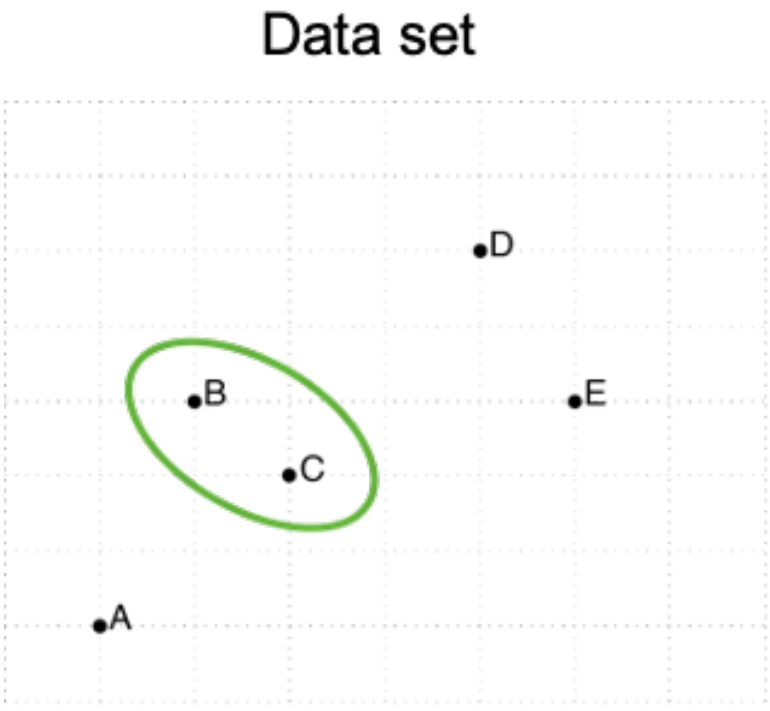
Hierarchical clustering - Bottom-up approach

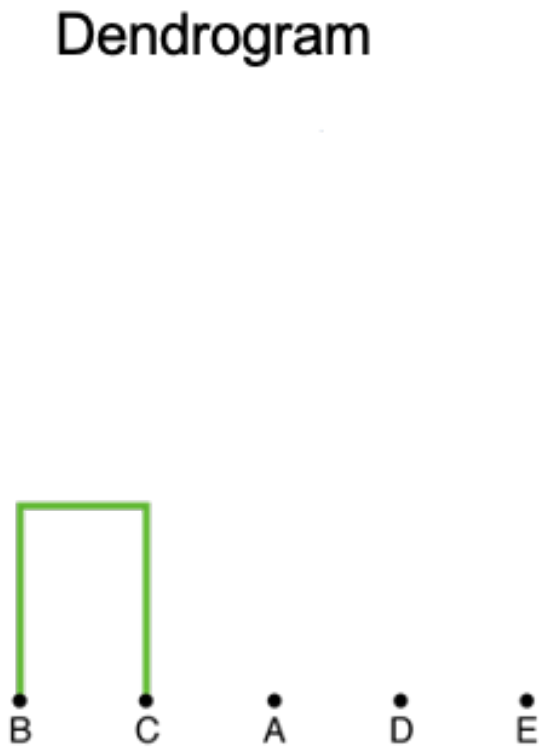
Data set



Dendrogram

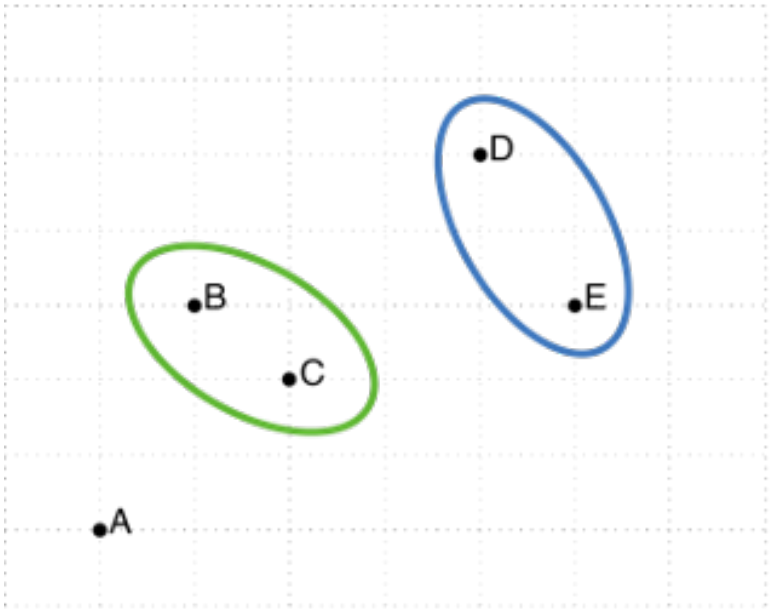
Hierarchical clustering - Bottom-up approach - step 1



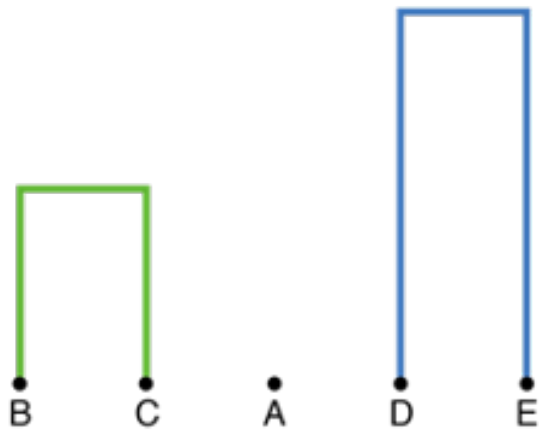


Hierarchical clustering - Bottom-up approach - step 2

Data set

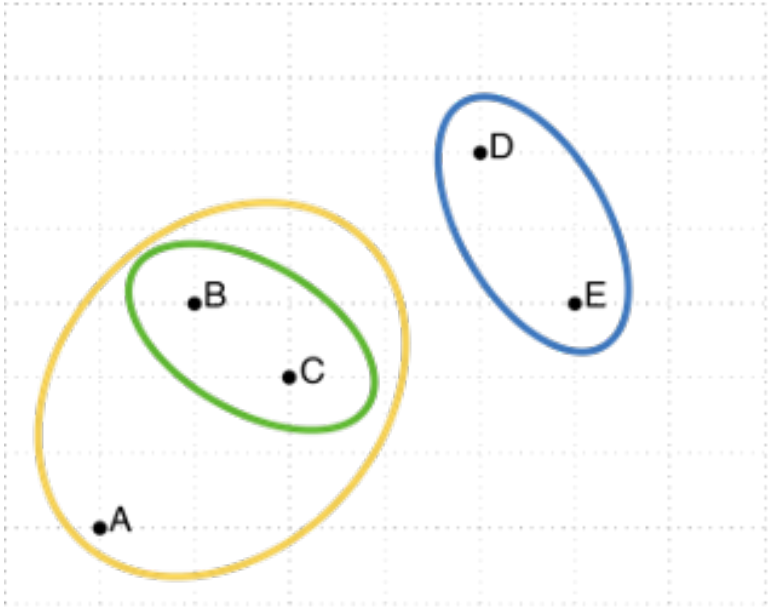


Dendrogram

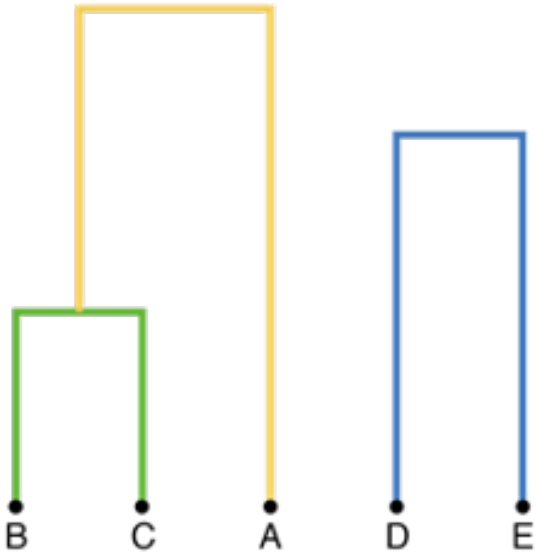


Hierarchical clustering - Bottom-up approach - step 3

Data set

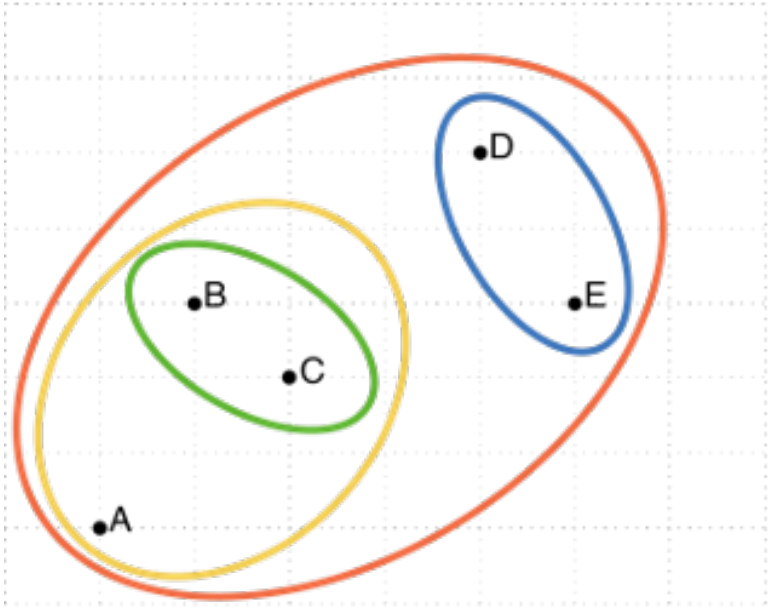


Dendrogram

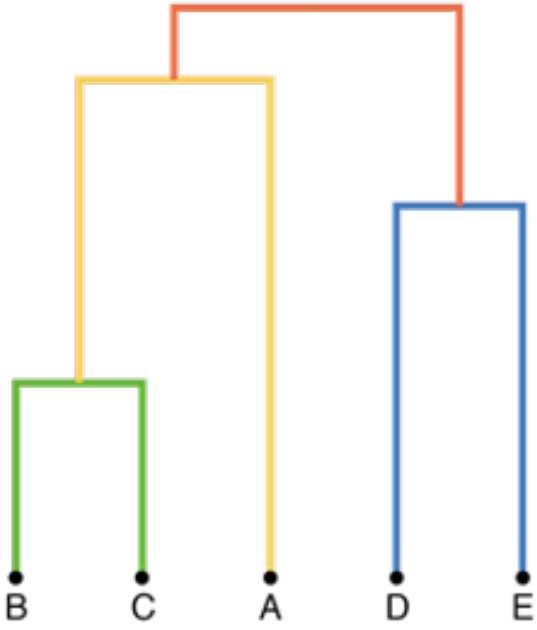


Hierarchical clustering - Bottom-up approach - step 4

Data set



Dendrogram



Hierarchical clustering - Linkage

How do we measure dissimilarity between two clusters (i.e. groups of observations)? Consider two clusters G and H, most used approaches are:

- *Single linkage*

$$d_{SL}(G, H) = \min_{i \in G, i' \in H} d_{ii'}$$

- *Complete linkage*

$$d_{CL}(G, H) = \max_{i \in G, i' \in H} d_{ii'}$$

- *Average linkage*

$$d_{CL}(G, H) = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{i' \in H} d_{ii'}$$

Hierarchical clustering - Dissimilarity measure

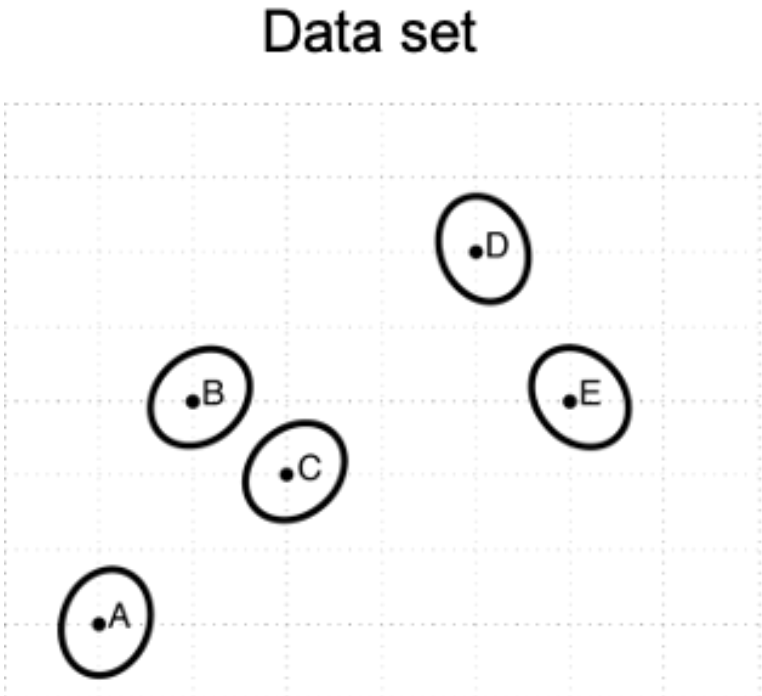
How do we measure dissimilarity between two points?

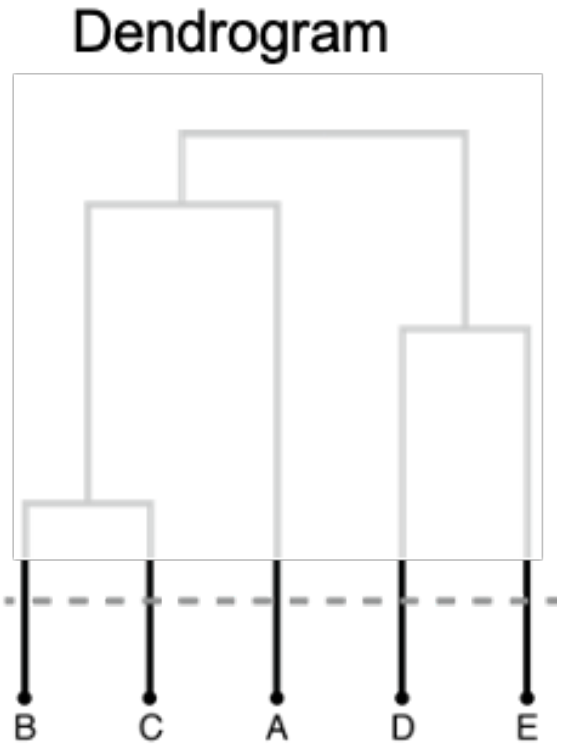
- Most commonly used is Euclidean distance.
- Another used metric is correlation (of observations across features).

Hierarchical clustering - How to interpret a dendogram

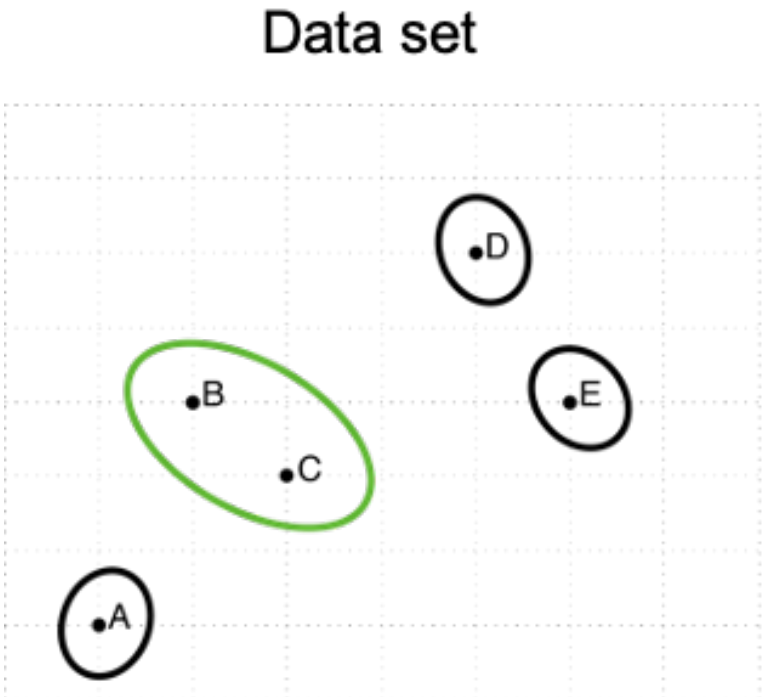
- Branches' height is proportional to the similarity between nodes.
  - Observations that fuse at the bottom of the dendogram are similar to each other.
  - Observations that fuse at the top of the dendogram are different from each other.
- We can cut the dendogram at a certain hight to obtain clusters.

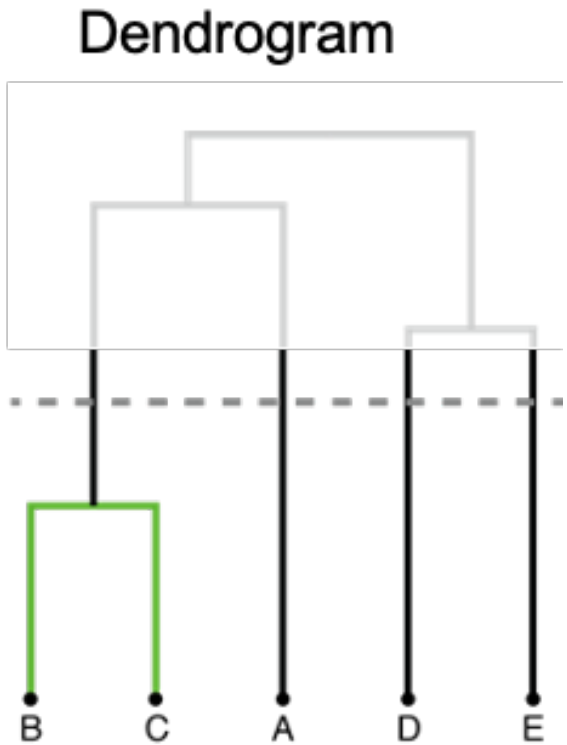
Hierarchical clustering - How to obtain clusters



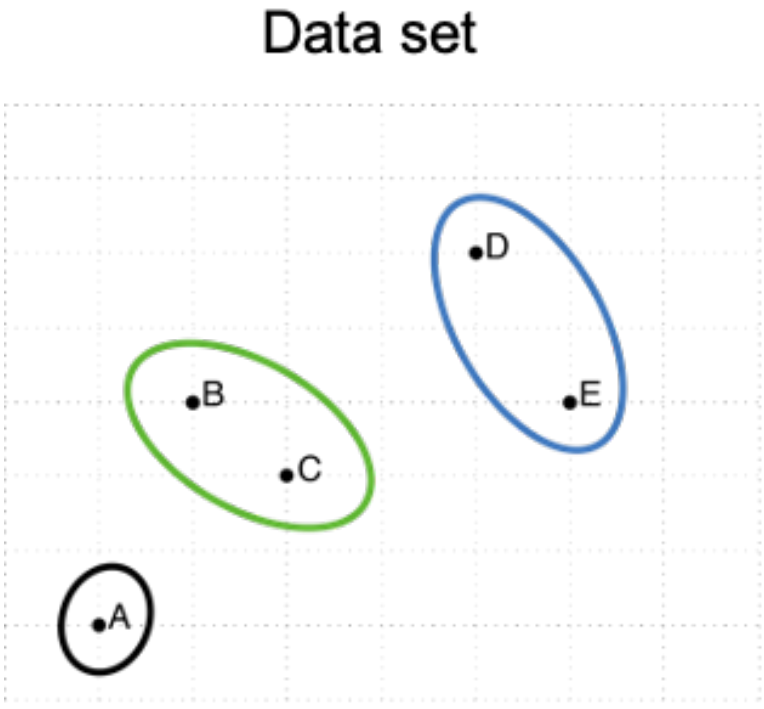


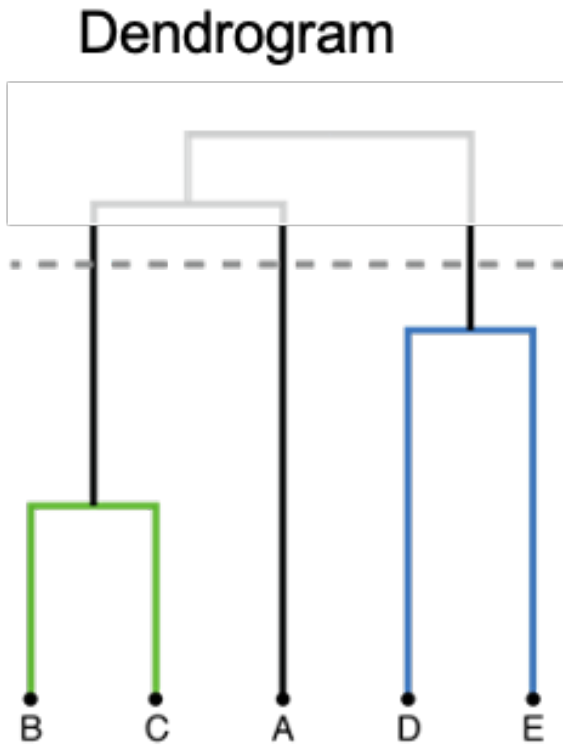
Hierarchical clustering - How to obtain clusters



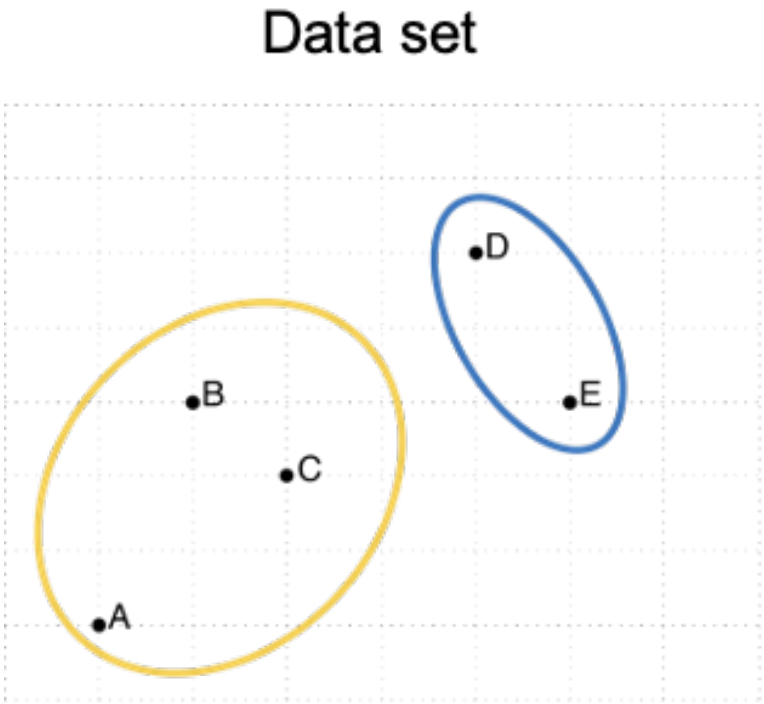


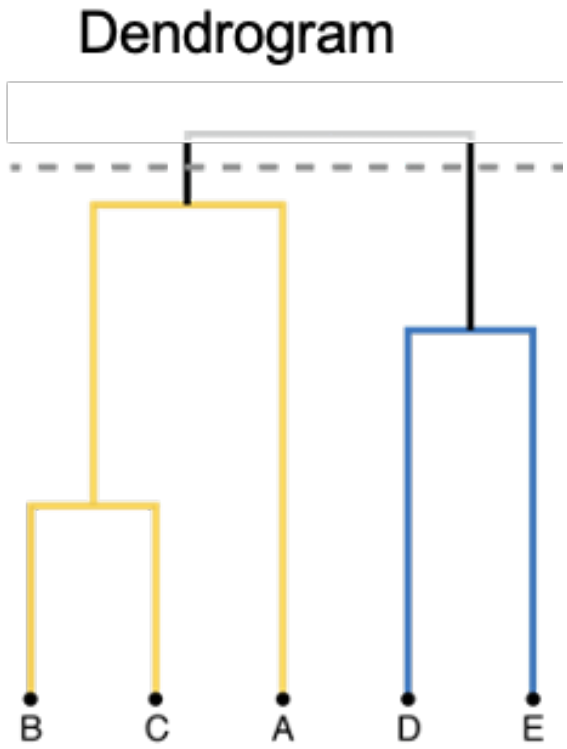
Hierarchical clustering - How to obtain clusters



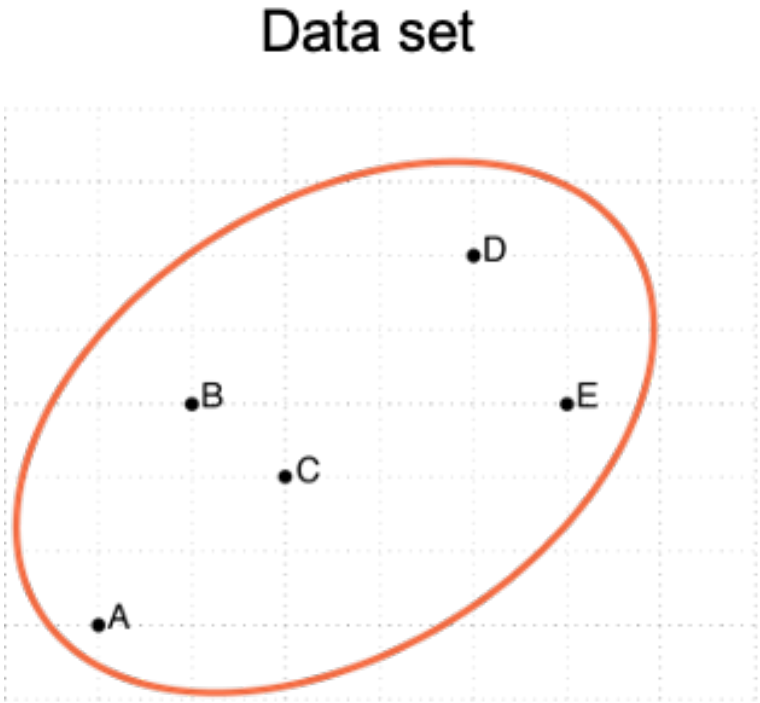


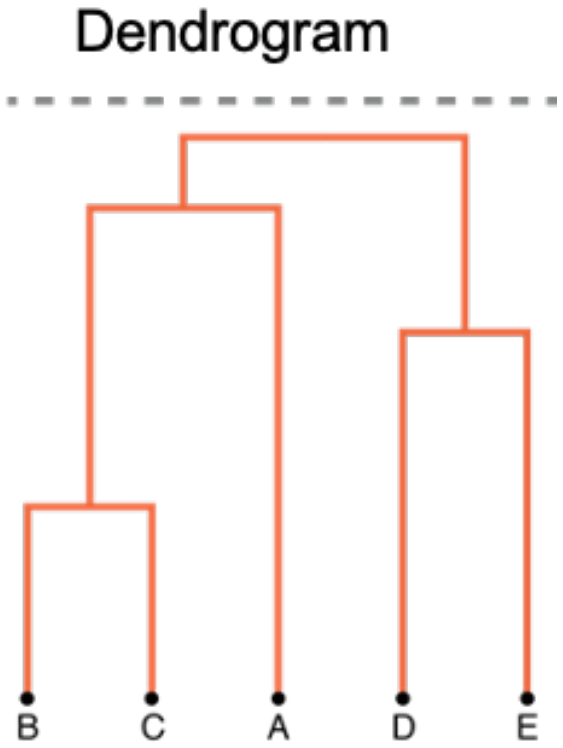
Hierarchical clustering - How to obtain clusters





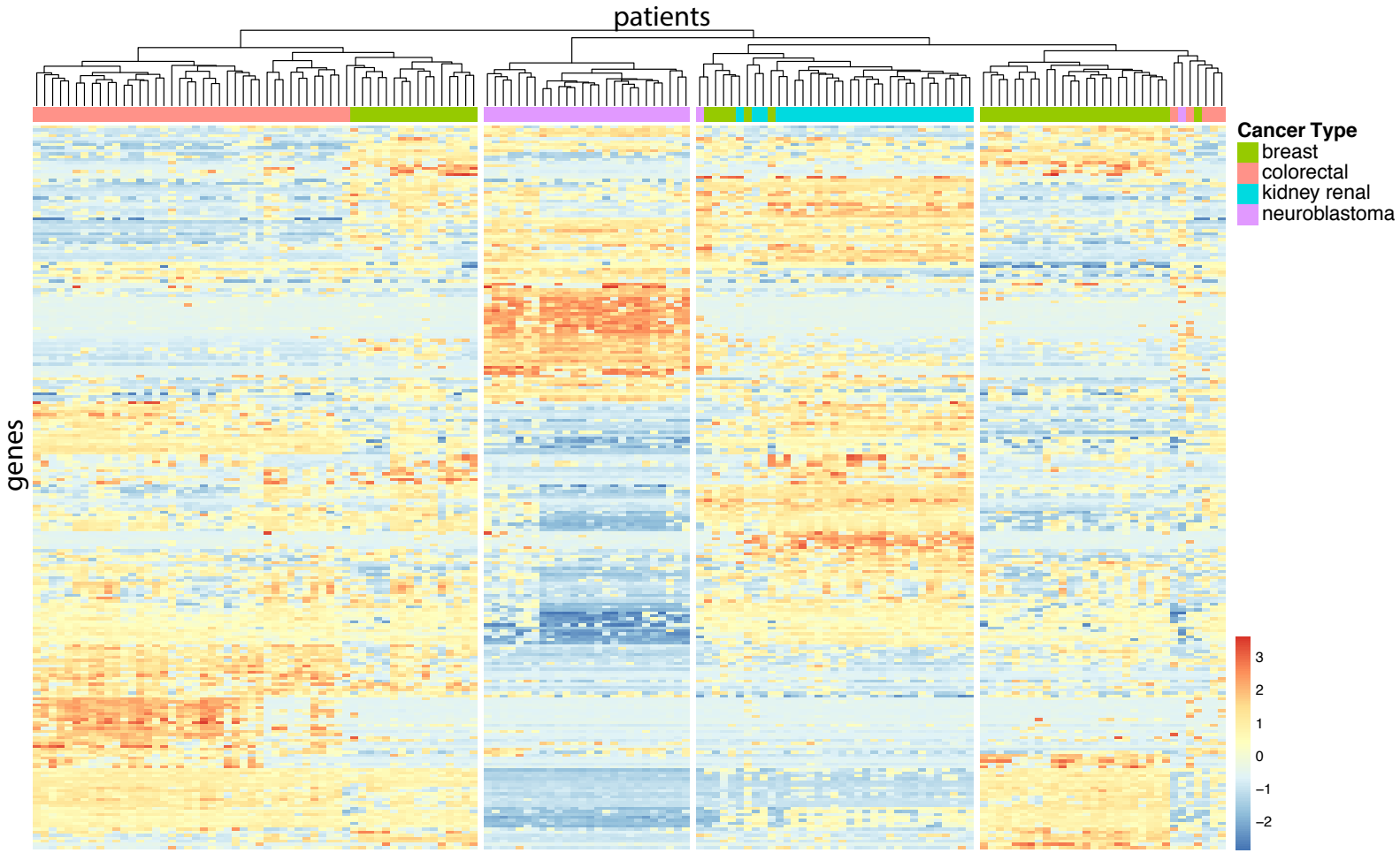
Hierarchical clustering - How to obtain clusters





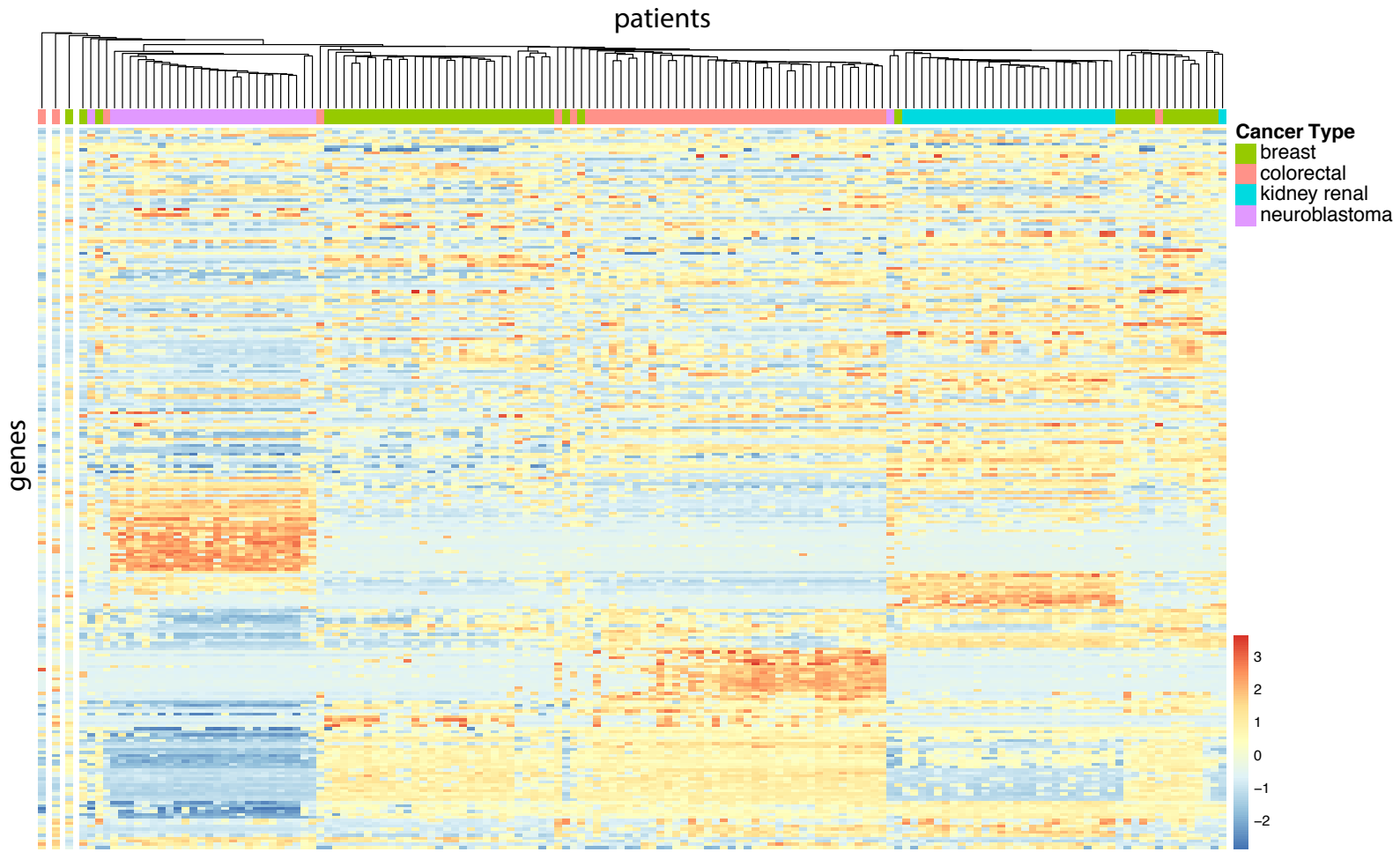
Hierarchical clustering - examples with GDSC data

Clustering using complete linkage for cluster similarity and Eucledian distance as observations similarity metric.



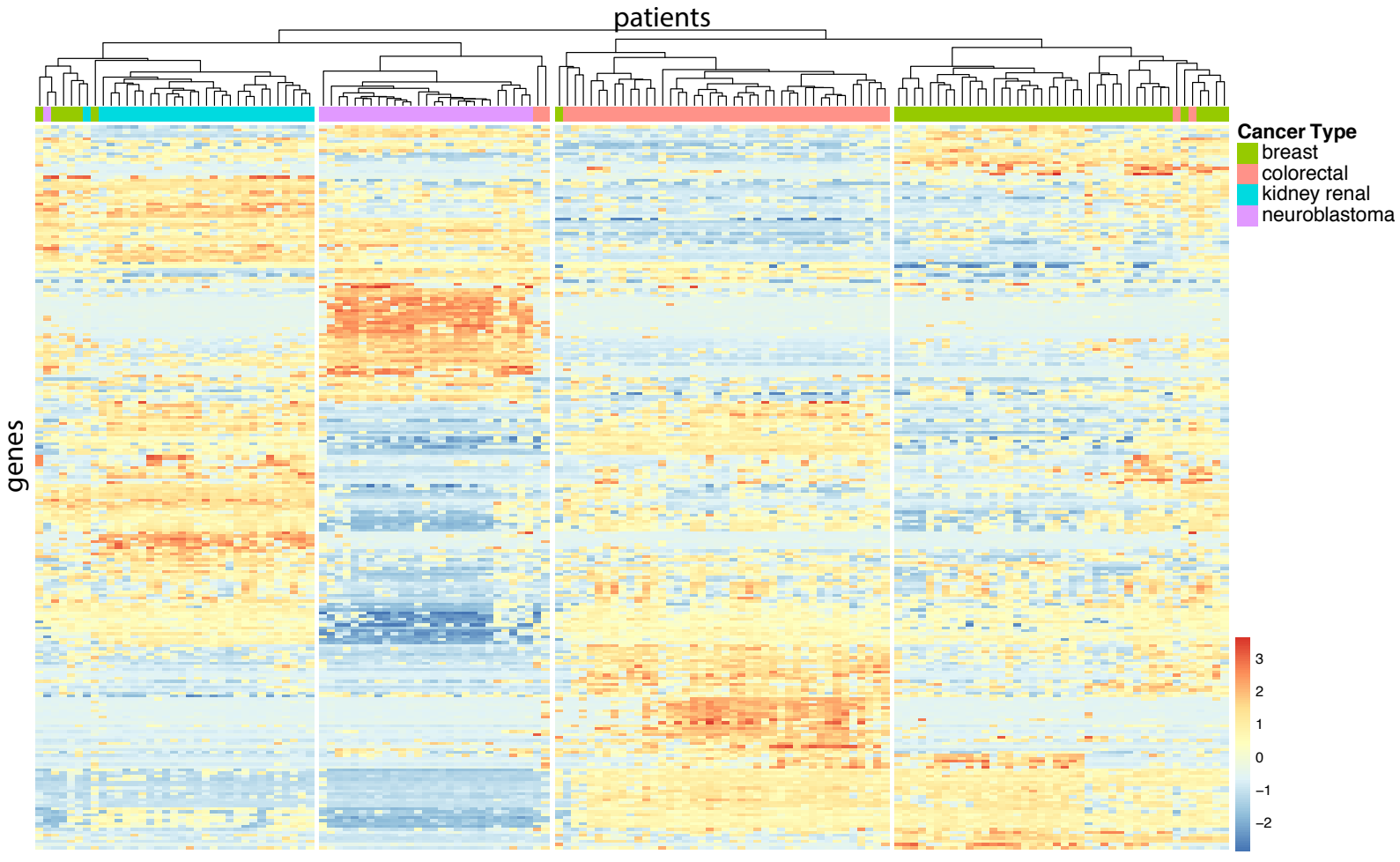
Hierarchical clustering - examples with GDSC data

Clustering using single linkage for cluster similarity and Eucledian distance as observations similarity metric.



Hierarchical clustering - examples with GDSC data

Clustering using complete linkage for cluster similarity and correlation as observations similarity metric.



Conclusions

- Unsupervised learning is useful to find inherent patterns in data.
- It does not use/require labels on data.
- No gold standard to assess performances.
- Used a lot for data exploration and/or as first step to then apply supervised learning.
- Active field of research, essential to explore increasingly available high-dimensional data.

References

- The figures showing geometrical interpretation of PCA and iterations of K-means are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani

Friedman, J., T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*. Springer series in statistics New York, 2001.