Lecture 10

Introduction to bioinformatics (MVE510)

Autumn, 2020

Additional reading: Clustering. Naomi Altman and Martin Krzywinski. Nature Methods, 14(6) 2017 and Principal component analysis. Jake Lever, Martin Krzywinski, and Naomi Altman. Nature Methods, 14(7) 2017.

Todays agenda

- Unsupervised methods and explorative data analysis
- Clustering
  - Hierarchical clustering
  - K-means clustering
- Techniques for dimension reduction
  - Principal component analysis

Supervised and unsupervised methods

- Data analysis methods can be divided into supervised and unsupervised methods
- Supervised methods rely on *metadata*, i.e. data that provides information about the data. This can be any form of medical, biological or technical information about the samples.
- Linear models are a supervised method. Here we specifically look for differences associated with a covariate (i.e. metadata). This includes, for example, identification of differentially expressed genes between different groups of patients.

Unsupervised methods

- Unsupervised methods does not utilize metadata, i.e. no additional information about the samples are used.
- Unsupervised methods focus on the identification of patterns in the data. In bioinformatics this typically means patterns between samples or genes.
- Unsupervised methods are explorative and does not rely on any specific hypotheses. This means that we can identify almost any form of patterns.
- Another common application of unsupervised methods is dimension reduction, where high-dimensional data is visualized in e.g. two dimensions.

Unsupervised methods

- In this course we will look at two main methods for unsupervised data analysis
- Clustering
  - Aims find groups of samples or genes (‘clusters’) that are ‘similar’.
  - Two important methods: Hierarchical clustering and K-means clustering.
- Principal component analysis (PCA)
  - Used for exploration and visualization of high-dimensional data
  - Identifies the dimensions with highest variability. These are used to represent the data at a lower dimension (e.g. in two dimensions).

Hierarchical clustering

- Hierarchical clustering builds a hierarchy describing the similarity between clusters
- Agglomerative: All samples starts as individual clusters which are merged together (aggregated) according to
  - A distance measure describing the separation between data points
  - A linkage criterion measuring the distance between clusters
- Divisive: All samples starts in a single large cluster which is then successively divided
- The number of clusters are then defined based on a minimum similarity threshold

Hierarchical clustering - distances

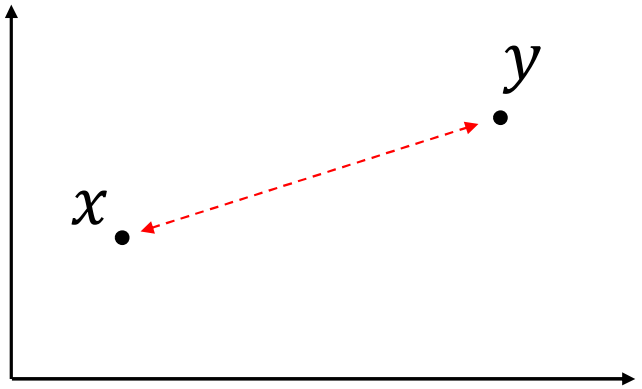
All clustering techniques are based on a distance metric that measures the similarity between two data points.

A few notations

We will use $x = (x_1, \dots, x_p)$ and $y = (y_1, \dots, y_p)$ to denote two data points defined in $\mathbb{R}^p$. Let $d(x, y)$ denote the distance between x and y.

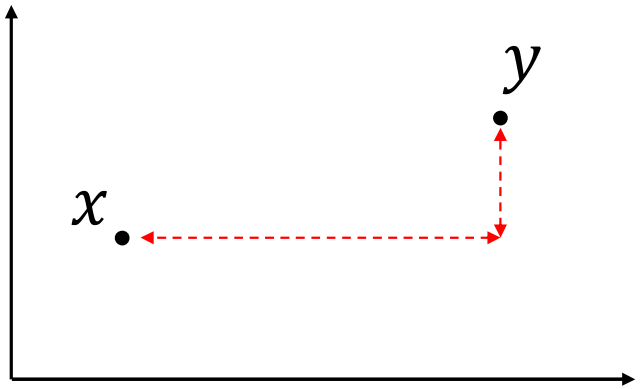
Euclidean distance

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$



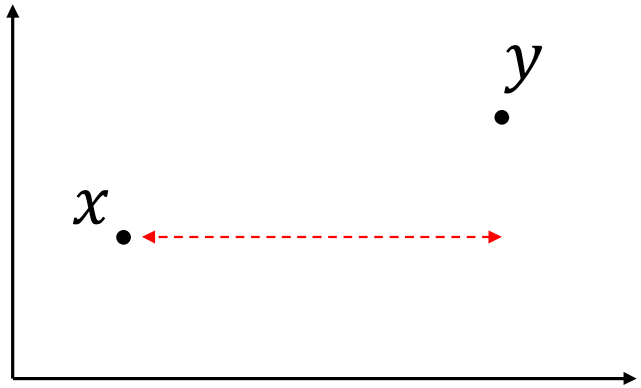
Manhattan distance

$$d(x,y) = \sum_{i=1}^{n}|x_i - y_i|$$



Maximum distance (L^{∞})

$$d(x,y) = \max_i |x_i - y_i|$$



Correlation distance

$$\text{corr}(x,y) = \frac{\text{cov}(x,y)}{\sqrt{\text{var}(x)\text{var}(y)}} \quad \text{and} \quad d(x,y) = \frac{1 - \text{corr}(x,y)}{2}$$

where $\text{cov}(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$, $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\text{var}(x) = \text{cov}(x,x)$.

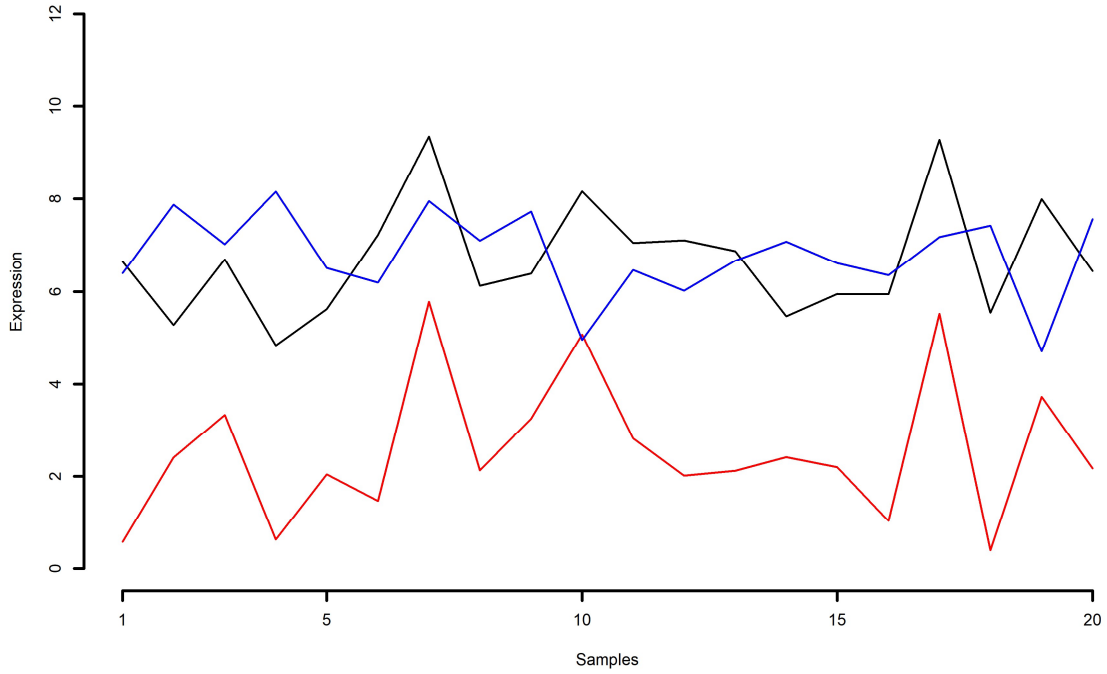
The distance matrix

- Contains all pair-wise distances between the data points.
- The distance matrix is symmetric since $\mathrm{d}(x, y) = d(y, x)$.
- This matrix can be really big since it contains n^2 elements (only $n(n-1)/2$ elements are actually needed).
- Some algorithms works instead with similarity matrices.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0.00	5.74	5.04	4.60	4.11	4.63	4.13	5.70	4.59	3.33
2	5.74	0.00	3.49	6.29	5.96	5.30	6.62	7.22	5.68	4.60
3	5.04	3.49	0.00	5.68	5.70	4.26	4.62	6.25	5.94	4.94
4	4.60	6.29	5.68	0.00	3.73	4.98	5.95	6.44	3.71	3.18
5	4.11	5.96	5.70	3.73	0.00	5.45	5.50	4.97	4.33	3.76
6	...									

Distances – which distance should I choose?

The choice of distance depends on the question. What similarities are you interested in?



Euclidean distance

$$d(x, y) = 59.4$$

$$d(x, z) = 360.0$$

Correlation distance

$$d(x, y) = 0.666$$

$$d(x, z) = 0.097$$

Hierarchical clustering - distances

Some more notation

We will use $x = (x_1, \ldots, x_p)$ and $y = (y_1, \ldots, y_p)$ to denote two data points defined in $\mathbb{R}^p$. Let $d(x, y)$ denote the distance between x and y.

Let X and Y denote two clusters with k and l members each, i.e. $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(k)}\}$ and $Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(l)}\}$. Let $D(X, Y)$ denote the distance between X and Y.

Hierarchical clustering

Algorithm: Agglomerative hierarchical clustering

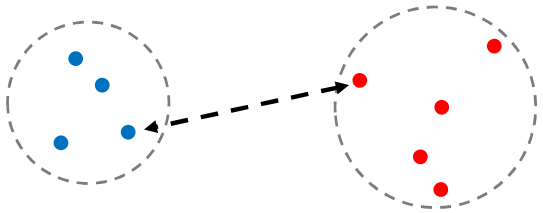
1. Calculate a distance matrix containing the pair-wise distance between all data point
2. Let each data point be a cluster
3. Identify the most similar clusters X and Y according to a *linkage criterion.*
4. Merge X and Y into a new cluster. Update the distance matrix.
5. Goto 3 if the number of clusters are more than 2.

Linkage criteria

The linkage criterion calculates the similarity between clusters.

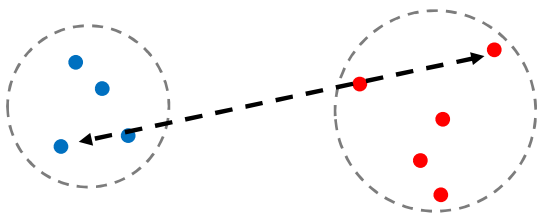
Single linkage

$$D(X,Y) = \min_{x \in X, y \in Y} d(x,y)$$



Complete linkage

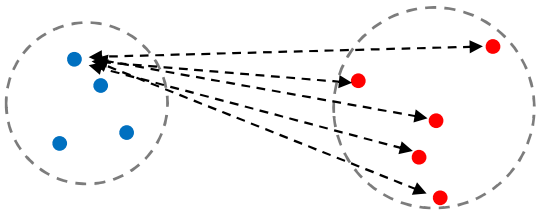
$$D(X,Y) = \max_{x \in X, y \in Y} d(x,y)$$



Linkage criteria

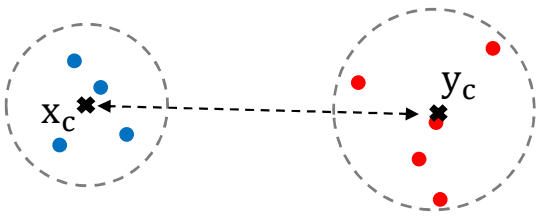
Average linkage

$$D(X,Y) = \frac{1}{kl} \sum_{\substack{\text{all pairs } (x,y), \\ x \in X, y \in Y}} d(x,y)$$



Centroid linkage

$$D(X,Y) = \mathrm{d}(\mathrm{x_c}, \mathrm{y_c})$$



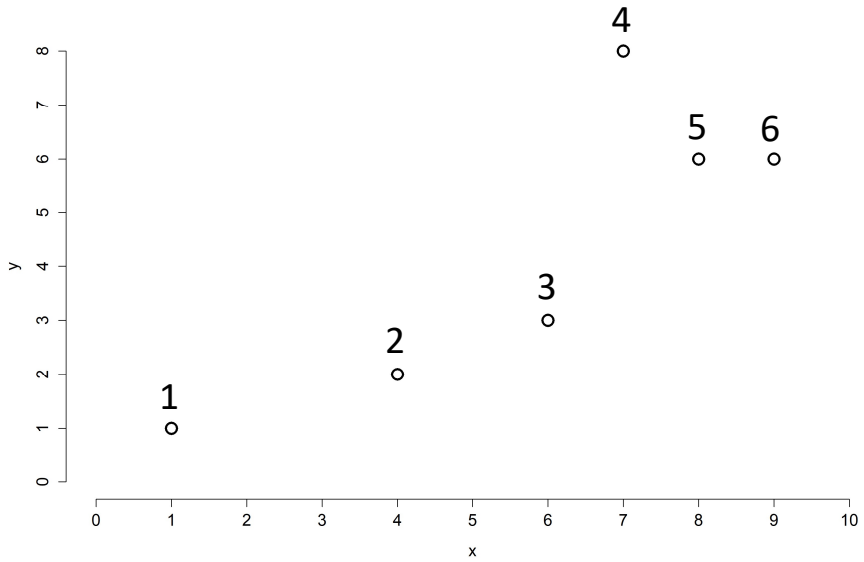
Hierarchical clustering

Algorithm: Agglomerative hierarchical clustering

1. Calculate a distance matrix containing the pair-wise distance between all data point
2. Let each data point be a cluster
3. Identify the most similar clusters X and Y according to a *linkage criterion.*
4. Merge X and Y into a new cluster. Update the distance matrix.
5. Goto 3 if the number of clusters are more than 2.

Hierarchical clustering

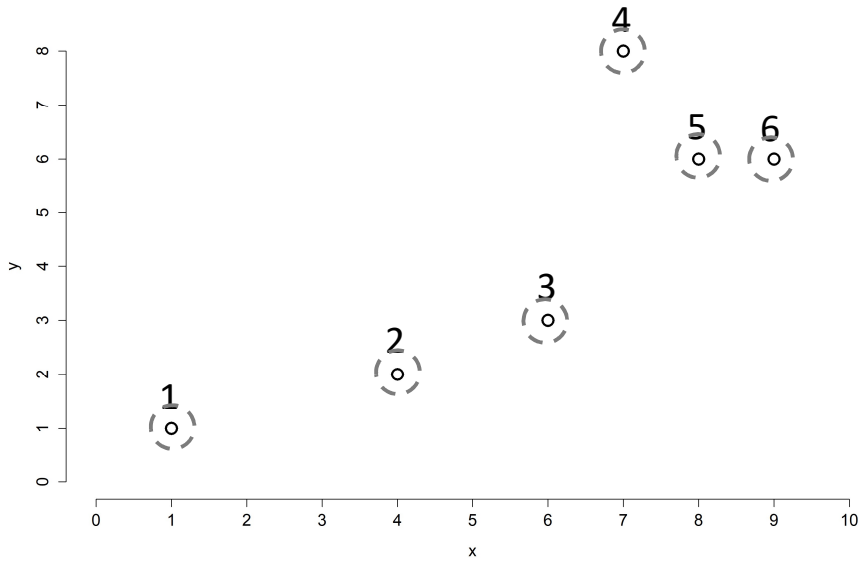
Example: Agglomerative clustering, Euclidean distance, complete linkage.



	1	**2**	**3**	**4**	**5**	**6**
1	0.00	3.16	5.39	9.22	8.60	9.43
2	3.16	0.00	2.24	6.71	5.66	6.40
3	5.39	2.24	0.00	5.10	3.61	4.24
4	9.22	6.71	5.10	0.00	2.24	2.83
5	8.60	5.66	3.61	2.24	0.00	1.00
6	9.43	6.40	4.24	2.83	1.00	0.00

Hierarchical clustering

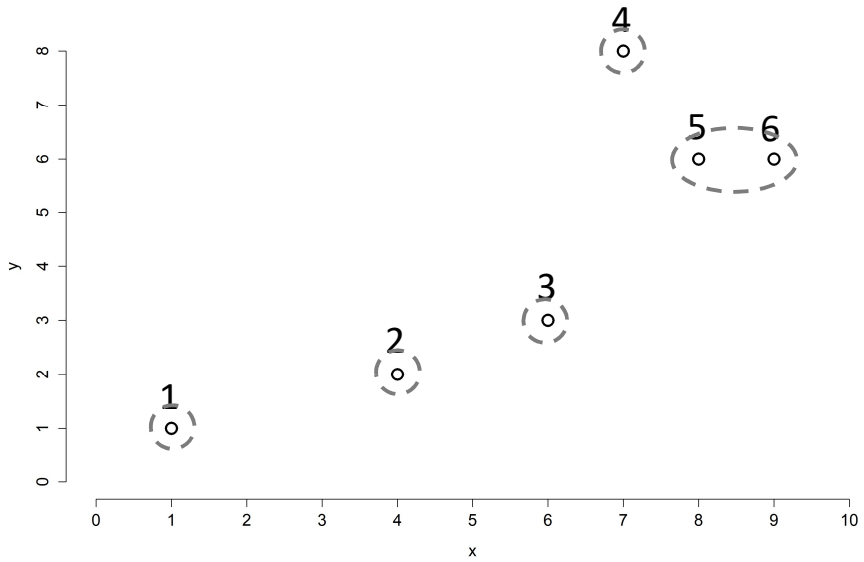
Example: Agglomerative clustering, Euclidean distance, complete linkage.



	1	2	3	4	5	6
1	0.00	3.16	5.39	9.22	8.60	9.43
2	3.16	0.00	2.24	6.71	5.66	6.40
3	5.39	2.24	0.00	5.10	3.61	4.24
4	9.22	6.71	5.10	0.00	2.24	2.83
5	8.60	5.66	3.61	2.24	0.00	1.00
6	9.43	6.40	4.24	2.83	1.00	0.00

Hierarchical clustering

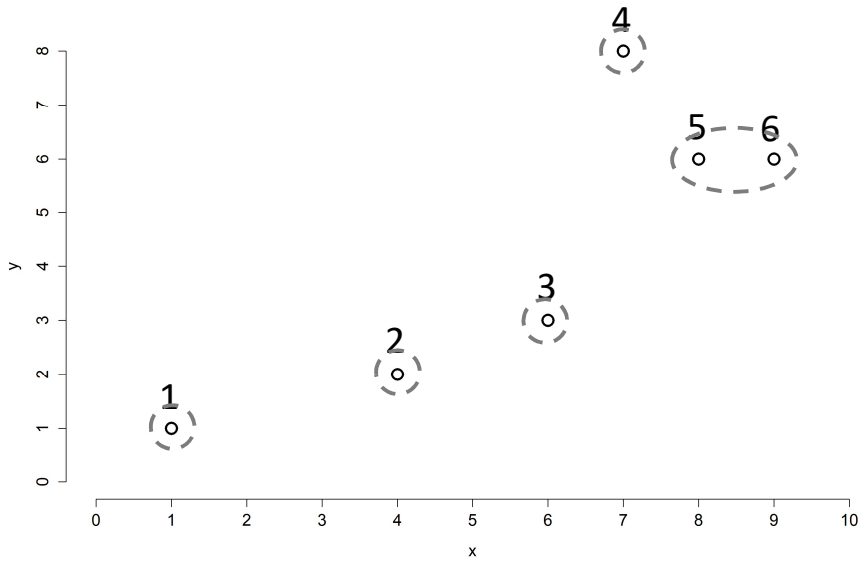
Example: Agglomerative clustering, Euclidean distance, complete linkage.



	1	**2**	**3**	**4**	**56**
1	0.00	3.16	5.39	9.22	9.43
2	3.16	0.00	2.24	6.71	6.40
3	5.39	2.24	0.00	5.10	4.24
4	9.22	6.71	5.10	0.00	2.83
56	9.43	6.40	4.24	2.83	0.00

Hierarchical clustering

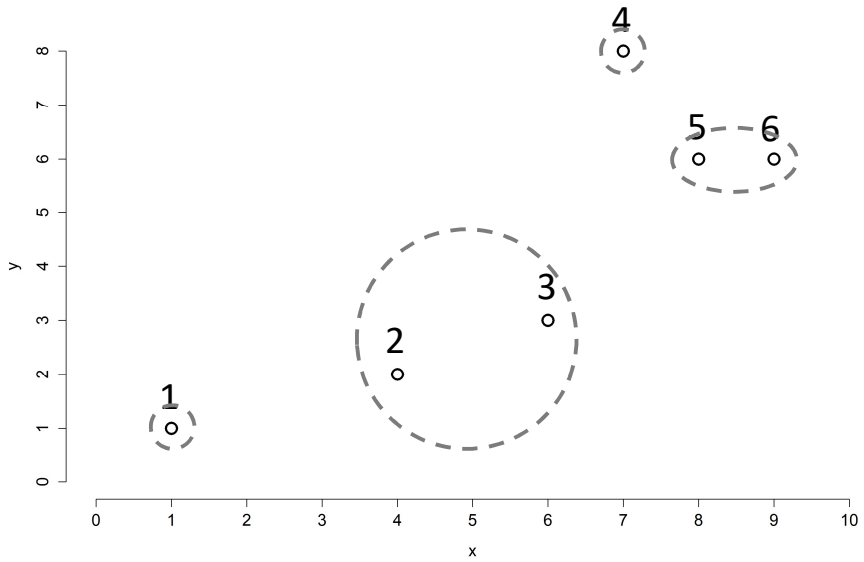
Example: Agglomerative clustering, Euclidean distance, complete linkage.



	1	2	3	4	56
1	0.00	3.16	5.39	9.22	9.43
2	3.16	0.00	2.24	6.71	6.40
3	5.39	2.24	0.00	5.10	4.24
4	9.22	6.71	5.10	0.00	2.83
56	9.43	6.40	4.24	2.83	0.00

Hierarchical clustering

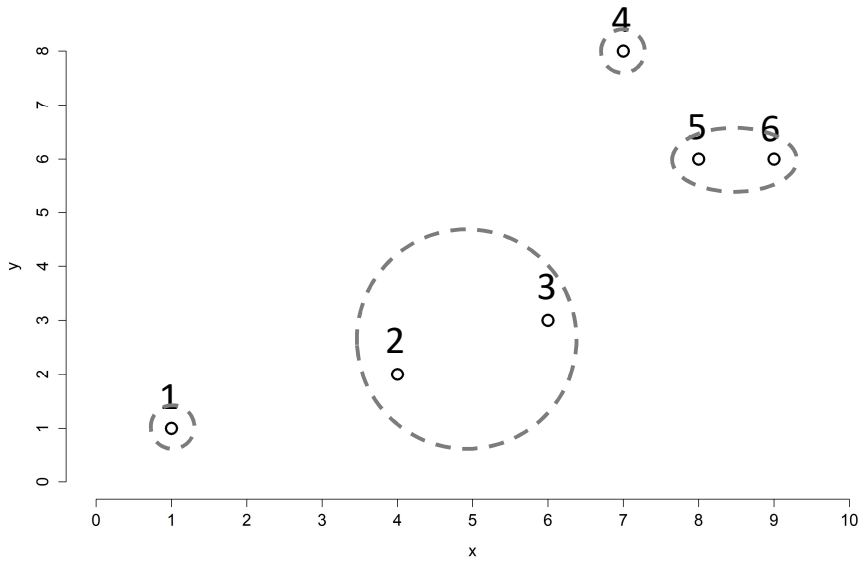
Example: Agglomerative clustering, Euclidean distance, complete linkage.



	1	**23**	**4**	**56**
1	0.00	5.39	9.22	9.43
23	5.39	0.00	6.71	6.40
4	9.22	6.71	0.00	2.83
56	9.43	6.40	2.83	0.00

Hierarchical clustering

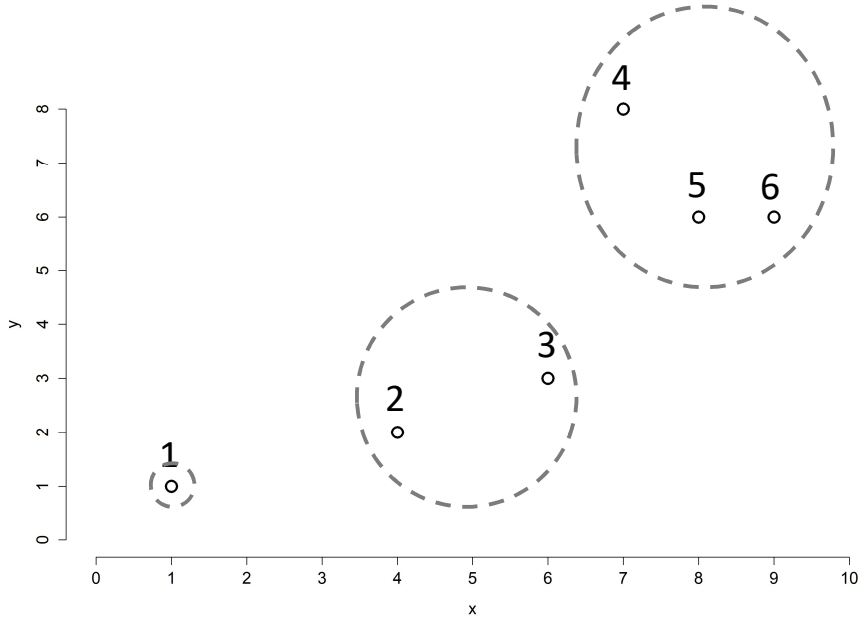
Example: Agglomerative clustering, Euclidean distance, complete linkage.



	1	**23**	**4**	**56**
1	0.00	5.39	9.22	9.43
23	5.39	0.00	6.71	6.40
4	9.22	6.71	0.00	2.83
56	9.43	6.40	2.83	0.00

Hierarchical clustering

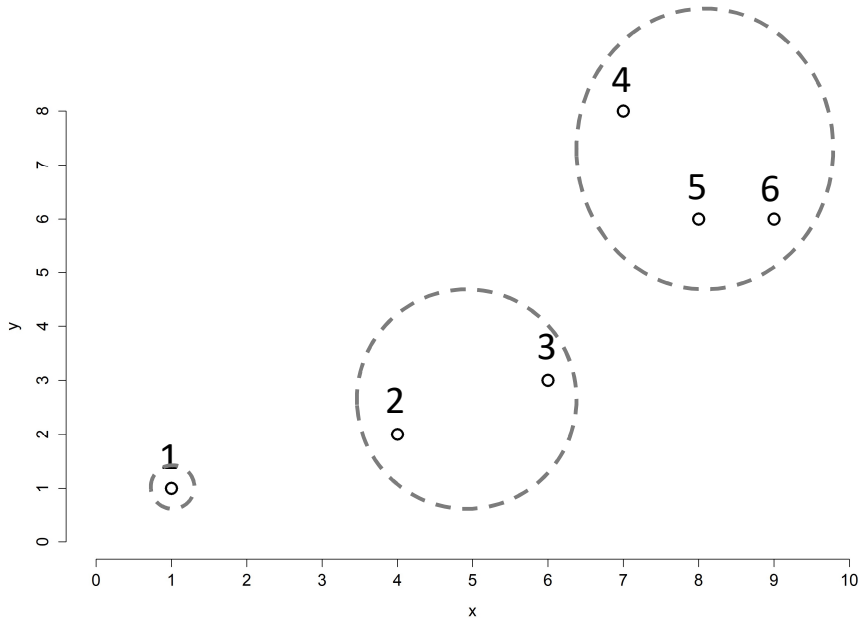
Example: Agglomerative clustering, Euclidean distance, complete linkage.



	1	**23**	**456**
1	0.00	5.39	9.43
23	5.39	0.00	6.71
456	9.43	6.71	0.00

Hierarchical clustering

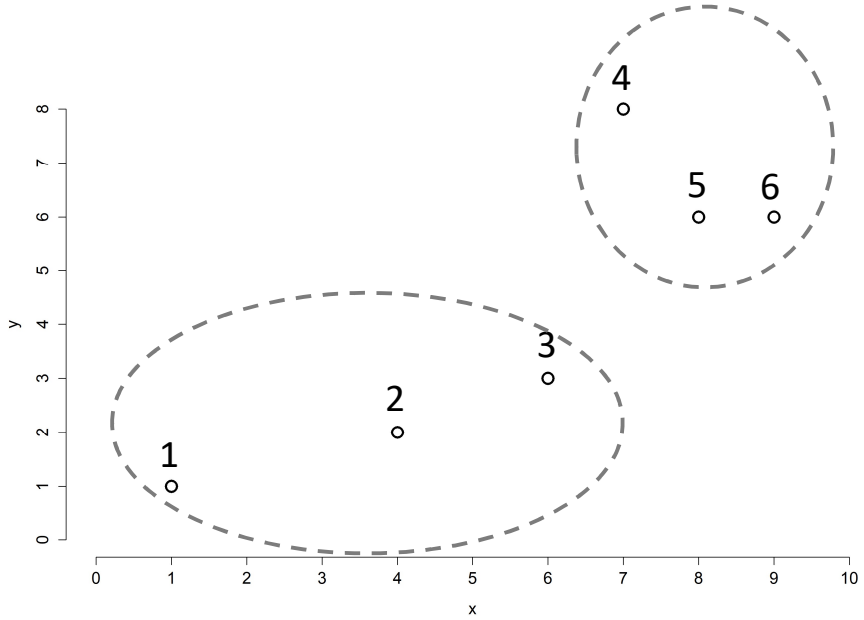
Example: Agglomerative clustering, Euclidean distance, complete linkage.



	1	**23**	**456**
1	0.00	5.39	9.43
23	5.39	0.00	6.71
456	9.43	6.71	0.00

Hierarchical clustering

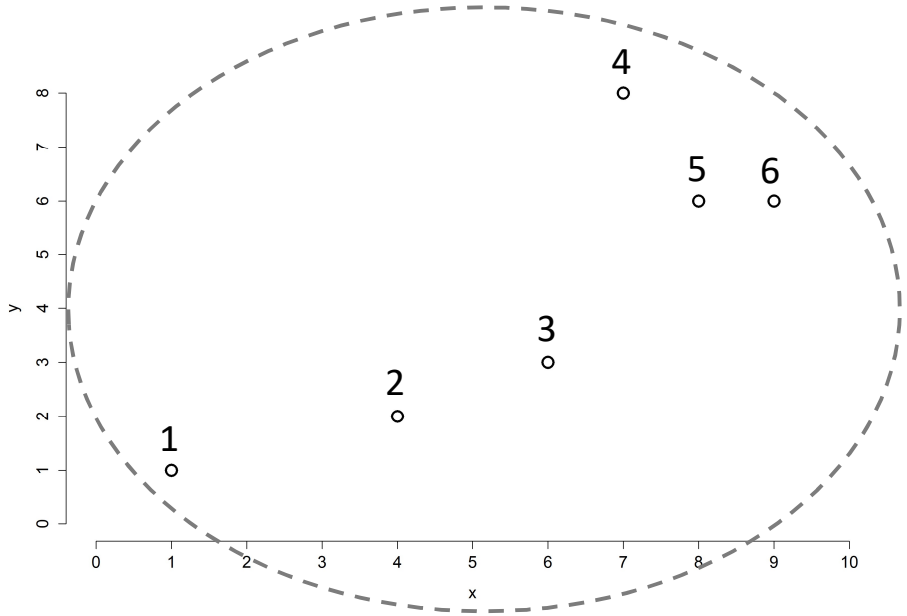
Example: Agglomerative clustering, Euclidean distance, complete linkage.



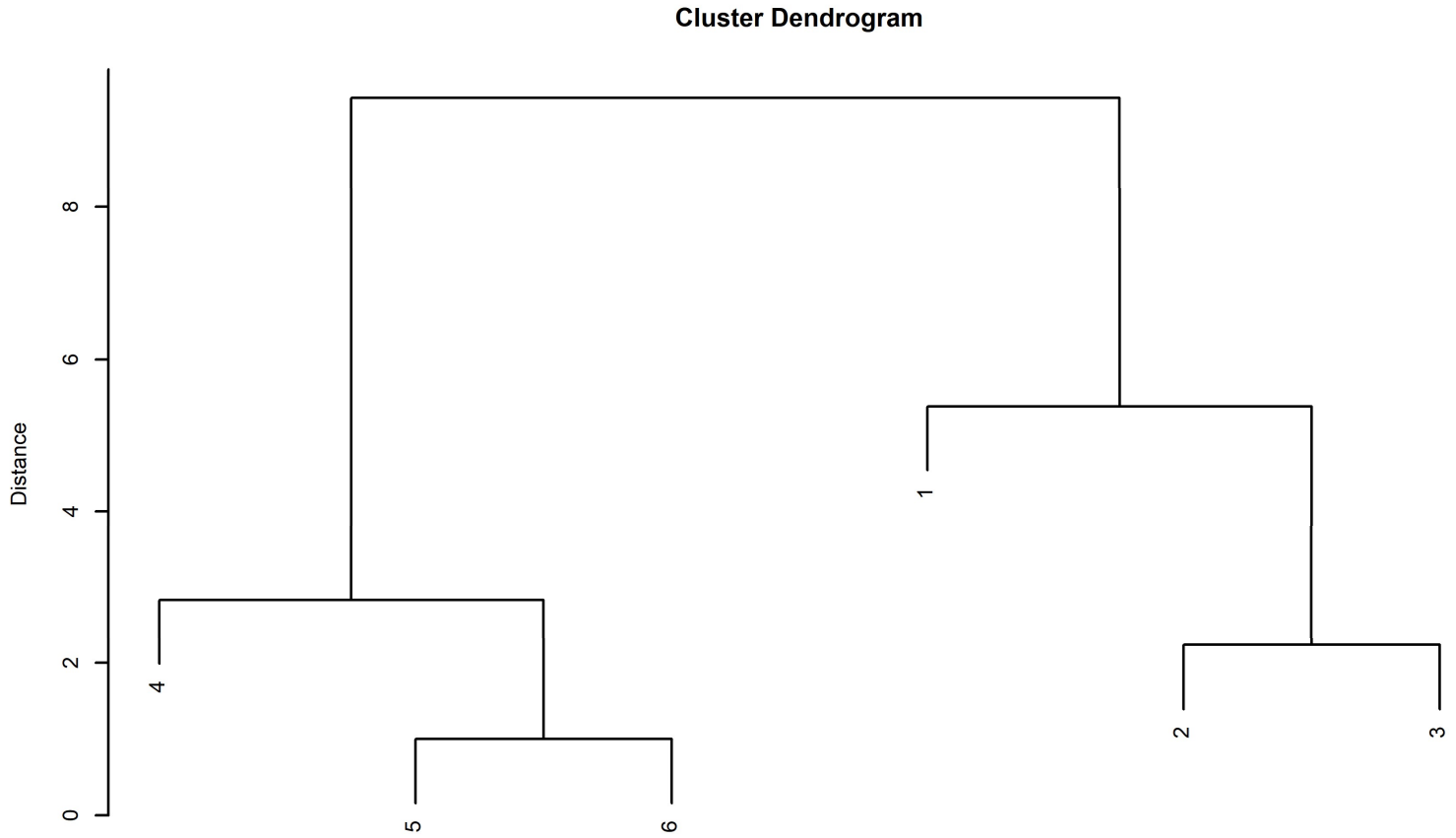
	123	**456**
123	0.00	9.43
456	9.43	0.00

Hierarchical clustering

Example: Agglomerative clustering, Euclidean distance, complete linkage.



Hierarchical clustering - dendrogram



Hierarchical clustering in R

Hierarchical clustering in R can be done using the **dist** and **hclust** commands.

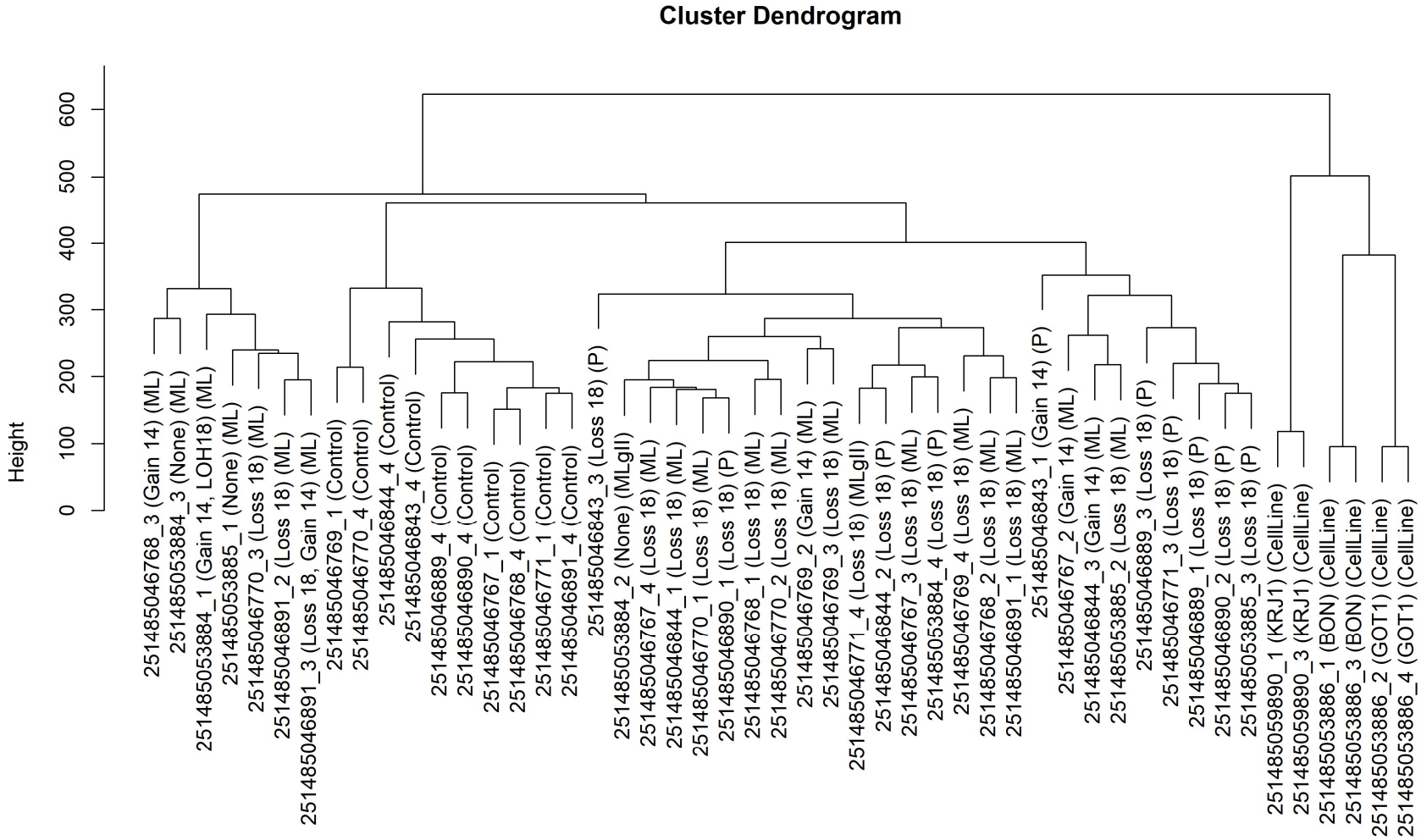
```
> dist.mat<-dist(data.matrix)      # Calculates a distance matrix. Default method is 'euclidean'
> dist.matrix
         1        2        3        4        5
2 3.162278
3 5.385165 2.236068
4 9.219544 6.708204 5.099020
5 8.602325 5.656854 3.605551 2.236068
6 9.433981 6.403124 4.242641 2.828427 1.000000
> res.cluster<-hclust(dist.matrix)  # Performs hierarchical clustering. Default method is 'complete'
> plot(res.cluster)                 # Plots the results
```

Clustering of gene expression data (RNA-seq)

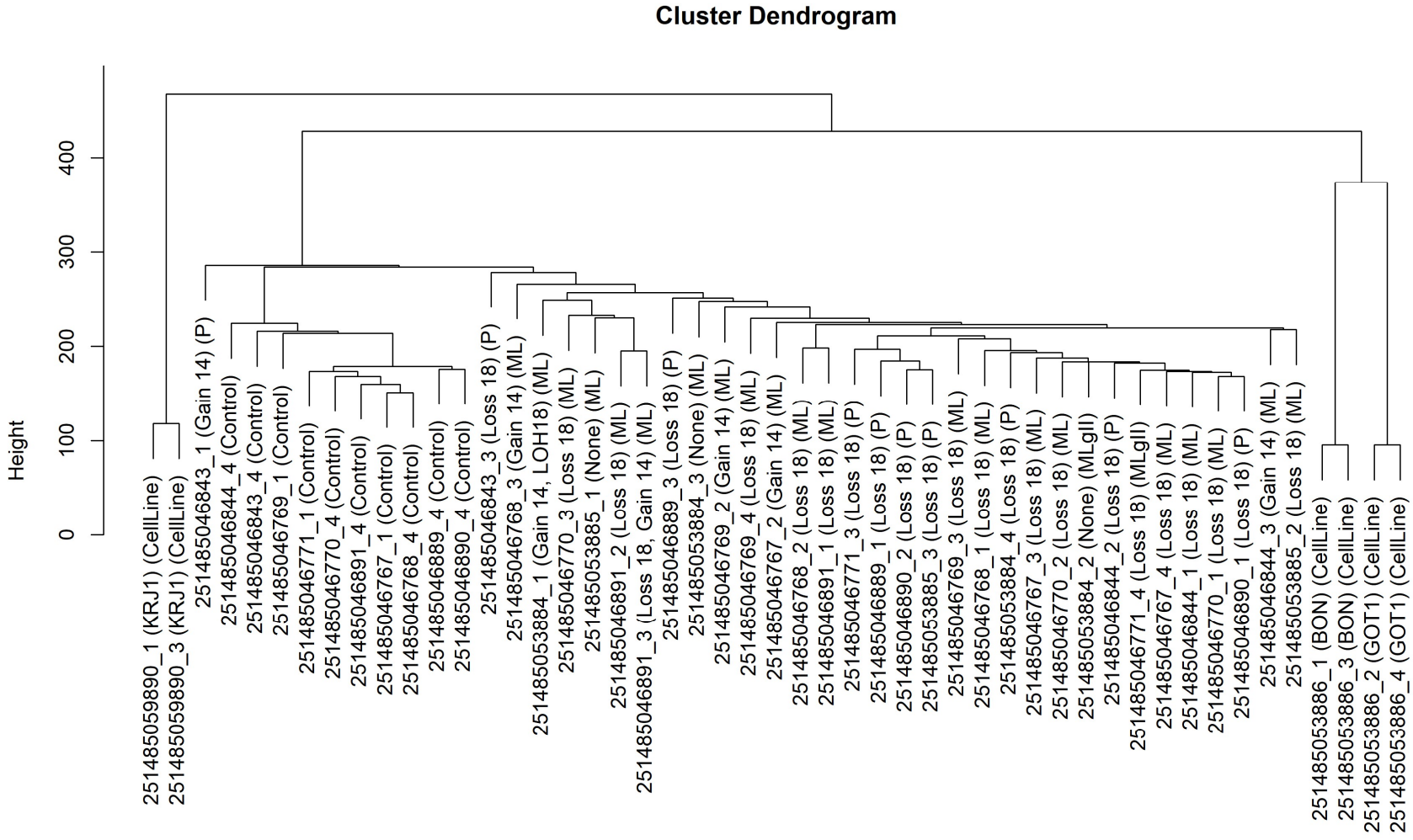
Data: Gene expression of 45,015 transcripts in 33 patients and 6 cell lines. Among the patients, 23 were from less aggressive group and 10 from more aggressive group. For each patient, a biopsy was sampled, the RNA extracted, sequenced using standard protocols, preprocessed and transformed and normalized so that the resulting data is approximately normal distributed. For each patient, the age and gender was also recorded.

Aim: Use clustering so investigate patterns between the patients.

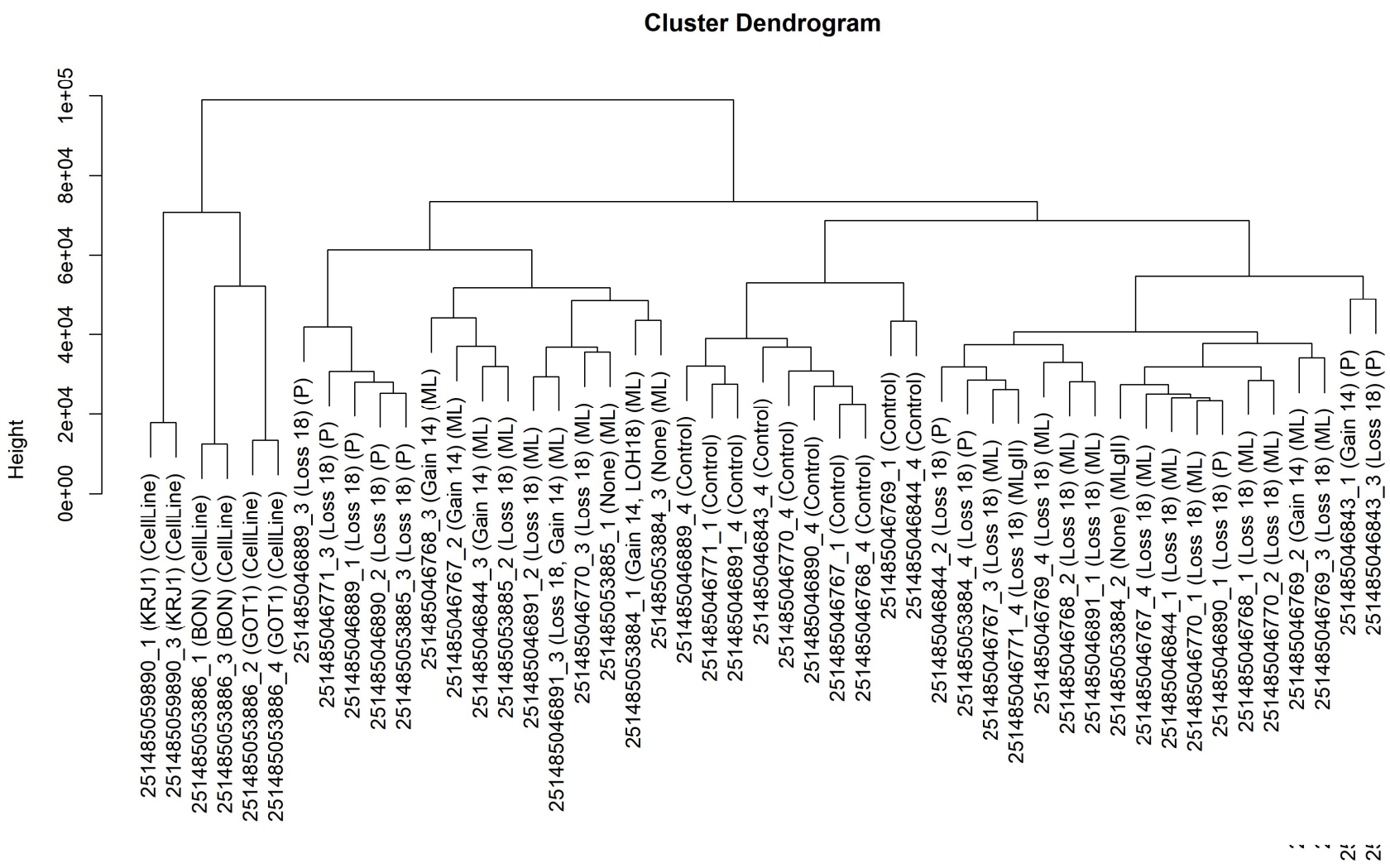
Clusters were calculated in R using **dist** and **hclust.**



Euclidean distance, complete linkage



Euclidean distance, single linkage



Manhattan distance, complete linkage

K-means clustering

- A method for clustering n observations in to s clusters. Note that s is pre-specified!
- We want assign each sample to one of the s clusters such that the variance within each cluster is minimized. This means that we want to minimize

$$\sum_{j=1}^{s} |c_j| Var(c_j).$$

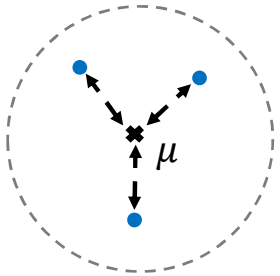
Cluster size

- K-means clustering implicitly assumes a Euclidean distance between points.

K-means clustering

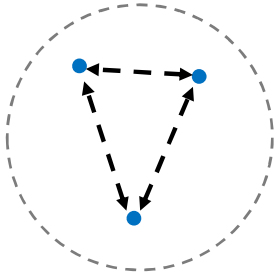
The variance within a cluster can be calculated by

$$Var(c_j) = \sum_{x \in c_j} \sqrt{\sum_{i=1}^{n} (x_i - \mu_i)^2}$$



where μ_i is the *mean point* of c_j. Note that minimizing $Var(c_j)$ is equivalent to minimizing the distance between all pairs of data points in the cluster, i.e.

$$\frac{1}{2} \sum_{x,y \in c_j} \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$



K-means clustering

The naïve algorithm

Start with an initial set of k randomly selected cluster means

$\mu = (\mu_1, \mu_2, \ldots, \mu_k)$.

Step 1: Assignment

Assign each data point to the closest cluster based on the Euclidean distance to the cluster mean.

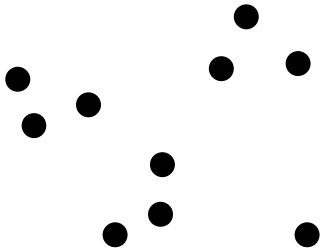
Step 2: Update

Recalculate the cluster mean values.

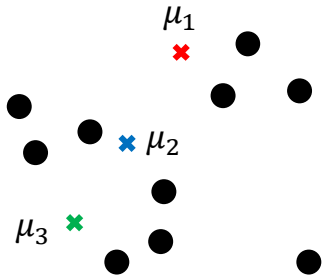
Repeat these steps until no more assignments are done.

K-means clustering

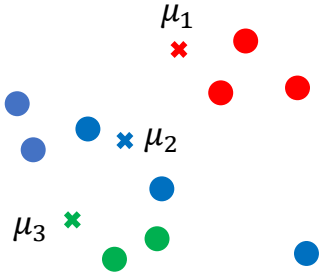
Data points



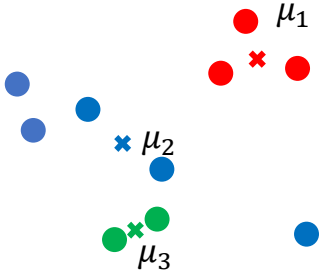
Initial mean values



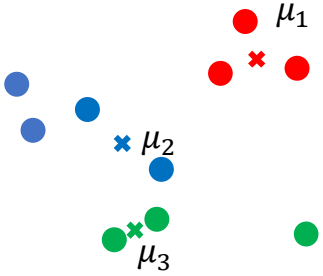
Iteration 1: Assignment



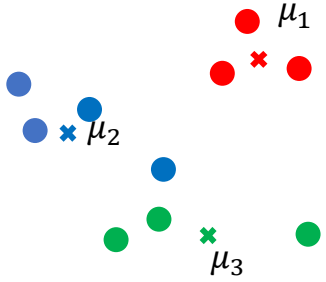
Iteration 1: Update



Iteration 2: Assignment

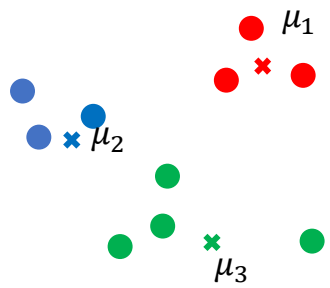


Iteration 2: Update

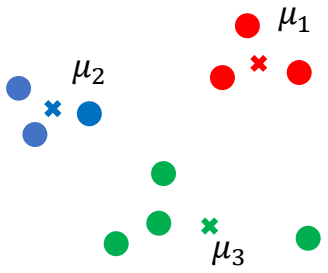


K-means clustering

Iteration 3: Assignment



Iteration 3: Update



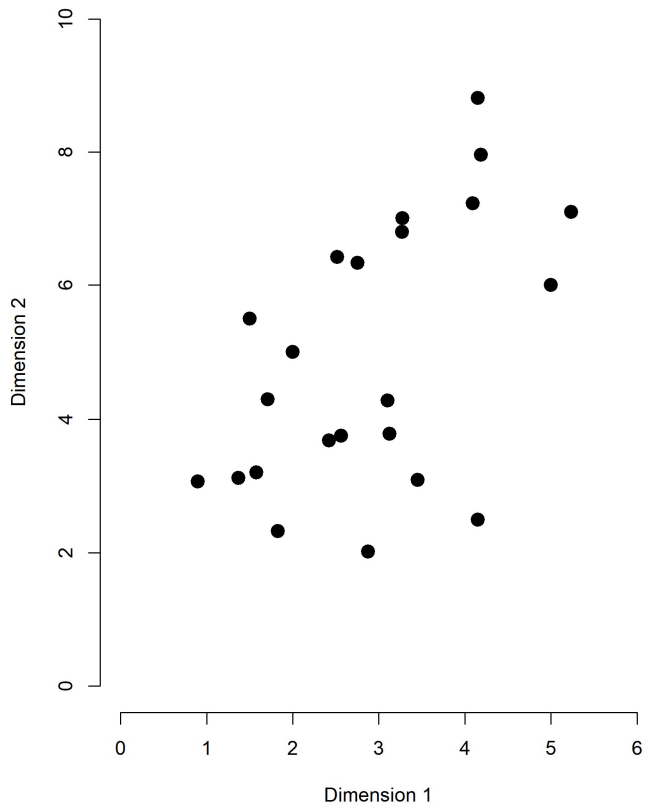
Since there are no changes in assignment, the algorithm has converged.

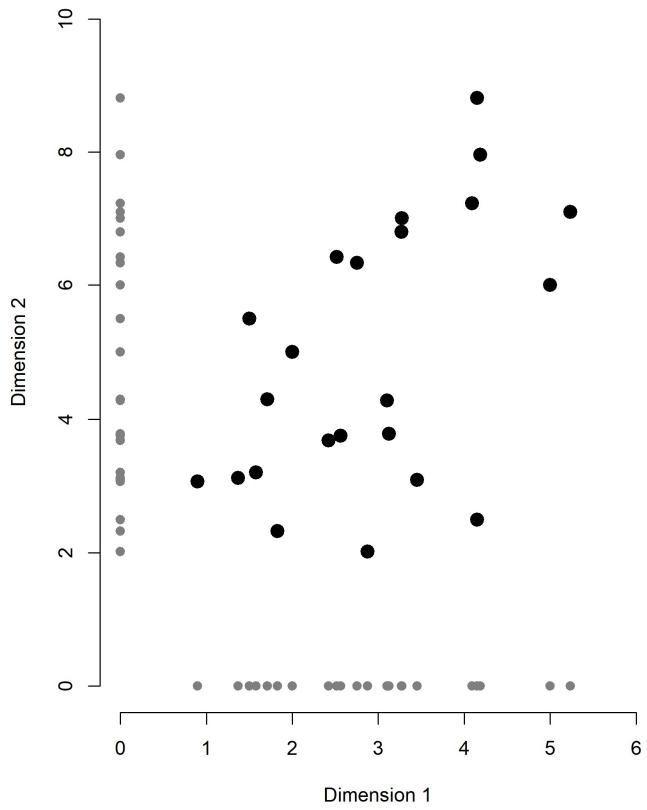
A drawback with K-means clustering is that it is often hard to decide the number of clusters (s).

Principal component analysis (PCA)

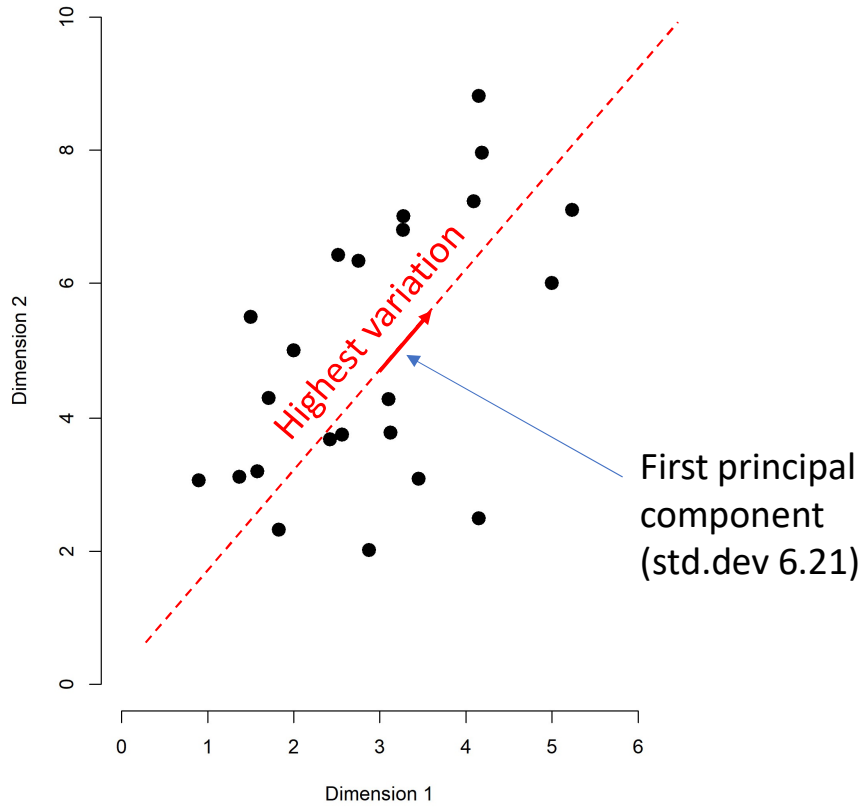
- Method used for exploration and visualization of high-dimensional data
- The main principle: Identify the dimensions of the data that explains as much variability as possible. These dimensions are defined by principal components.
- The data can then be projected on a set of the to principal components (often two) which results in a reduction of the dimensionality
- PCA is often used to identify patterns in the data. However, PCA does not assign each data point to a cluster.

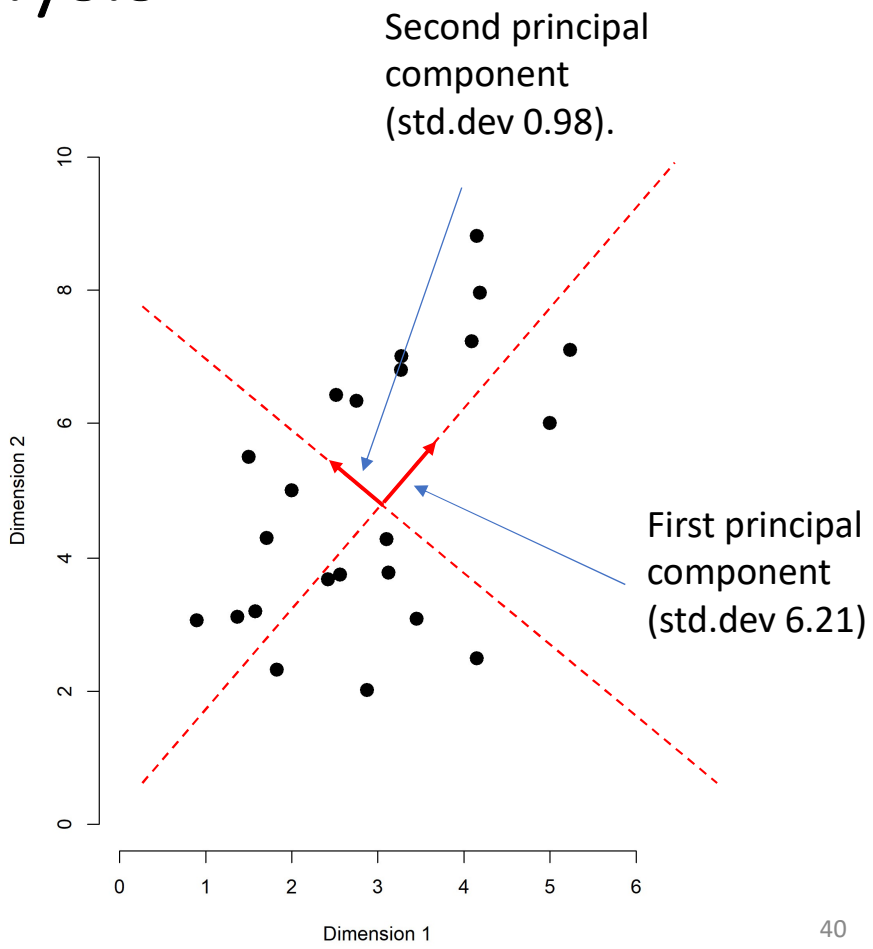
Principal component analysis



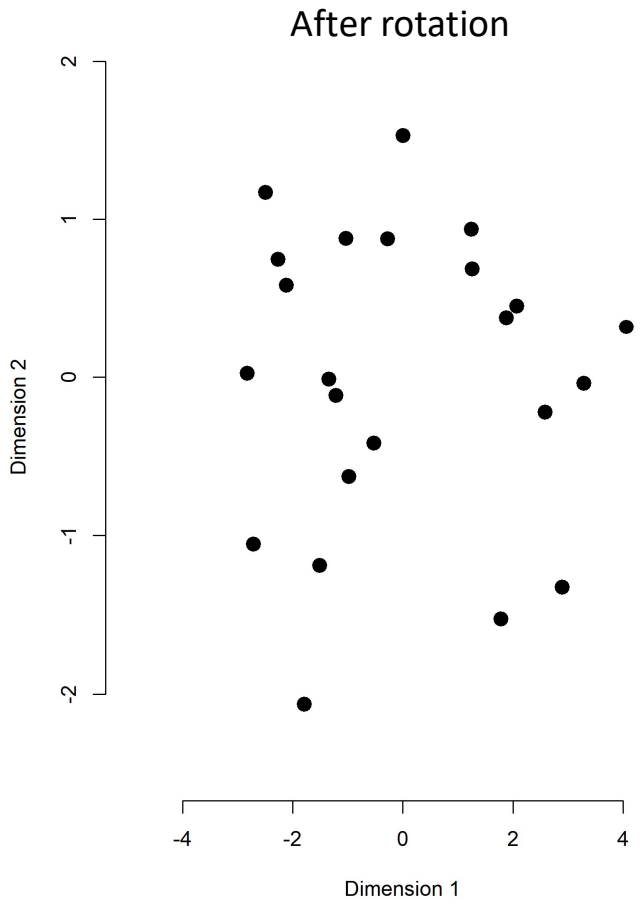


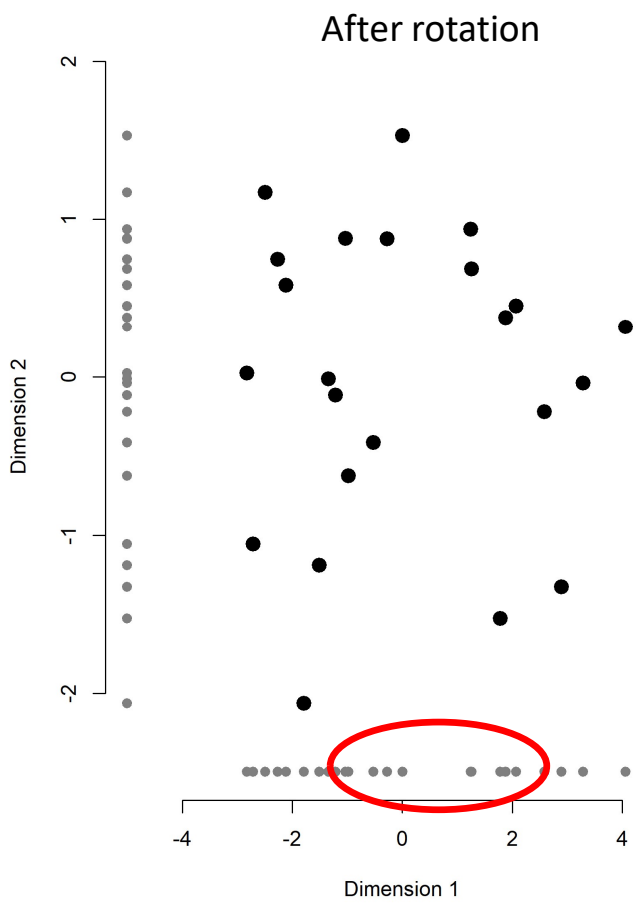
Principal component analysis





Principal component analysis





Principal component analysis

Some notation

We will, as before, use $x = (x_1, \dots, x_p)$ to denote a data point. Note that we will here regard x as a random variable.

Let $y_1, y_2, \dots y_n$ denote n linear combinations of the coordinates of x, i.e.

$$y_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = a_1^T x$$
$$y_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{1n}x_n = a_2^T x$$
$$\dots$$
$$y_n = a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = a_n^T x$$

Principal component analysis

The principal components will be identify be selecting $\{a_{ij}\}$ according to the following rules

The first principal component is the linear combination $y_1 = a_1^T x$ that maximizes the variability $\mathrm{Var}(y_1)$ under the condition that $a_1^T a_1$=1.

The second principal component is the linear combination $y_2 = a_2^T x$ that maximizes the variability $\mathrm{Var}(y_2)$ under the condition that $a_2^T a_2$=1 and is independent of y_1.

Principal component analysis

The *nth* principal component is the linear combination $y_n = a_n^T x$ that maximizes the variability $\mathrm{Var}(y_2)$ under the condition that $a_n^T a_n$=1 and is independent of $y_1, y_2, \ldots, y_{n-1}$.

The principal components corresponds to the eigenvectors of the covariance matrix. They are therefore often calculated by singular value decomposition (SVD).

Principal component analysis

A principal component analysis results

1. The principal components $y_1, \ldots, y_n$ constitutes a new orthogonal basis for the data. The number of principal components is the same as the dimension of the data (or, if lower, the number of samples).
2. The rotation matrix, which is formed from vectors $a_1, a_2, .., a_n$. This matrix describe how data can be rotated to the new basis. The individual elements are sometimes called loadings. Note that the rotation matrix is not unique – it can always be multiplied by -1.
3. The estimated variance for each principal component. These are always in decreasing order.

Principal component analysis in R

In R, principal component analysis can be done using the **prcomp** command.

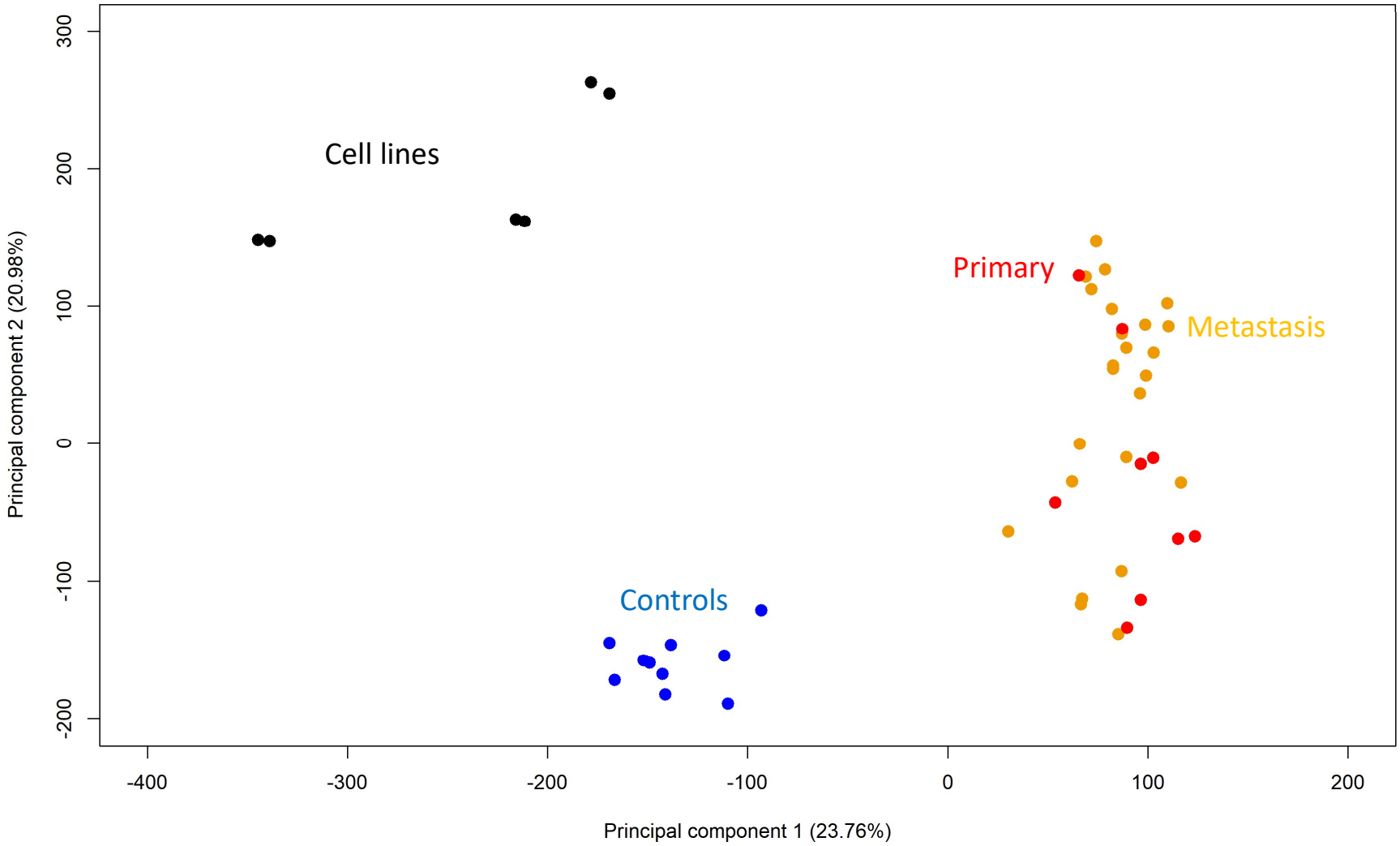
```
> res.pca<-prcomp(data.matrix, center=FALSE)   # 3x5 matrix

> res.pca$rotation                             # the rotation matrix defined by the loadings
             PC1         PC2         PC3
[1,] -0.26494186 -0.64099271  0.42258957
[2,] -0.96105263  0.14754678 -0.07881318
[3,]  0.07043733 -0.04676936  0.62289920
[4,] -0.01286506  0.02371949 -0.49200523
[5,] -0.03250730  0.75140422  0.43027198

> res.pca$x                                    # the transformed data calculated from the
           PC1        PC2        PC3           # rotation matrix
[1,] -1.084225 -1.4550113 -0.3345473
[2,]  1.559554 -1.6491774  0.1780445
[3,] -1.815806 -0.5476472  0.3526779

> res.pca$sdev                                 # the standard deviation of the three principal
[1] 1.8580773 1.6026168 0.3660624              # components
```

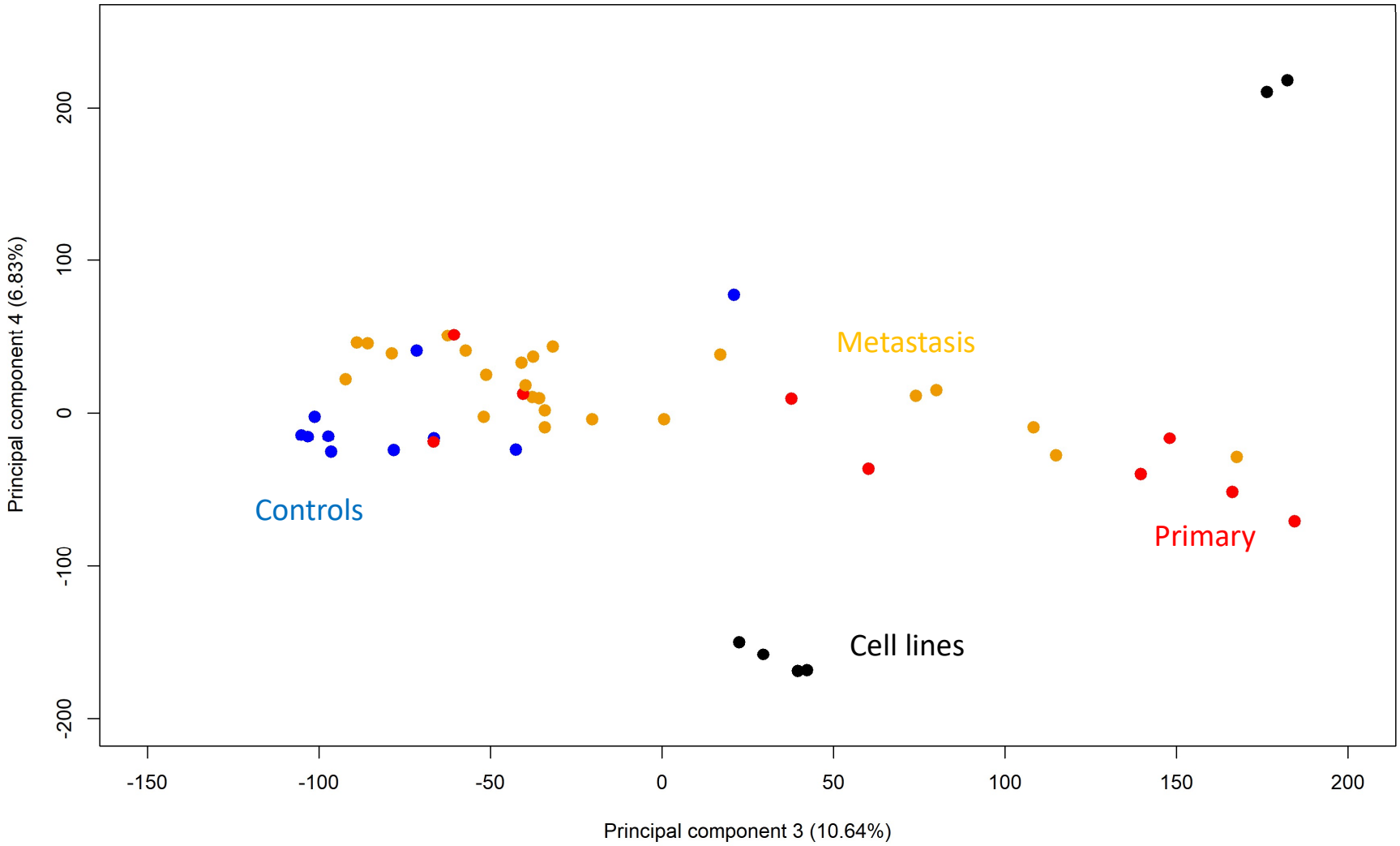
PCA of gene expression data (RNA-seq)

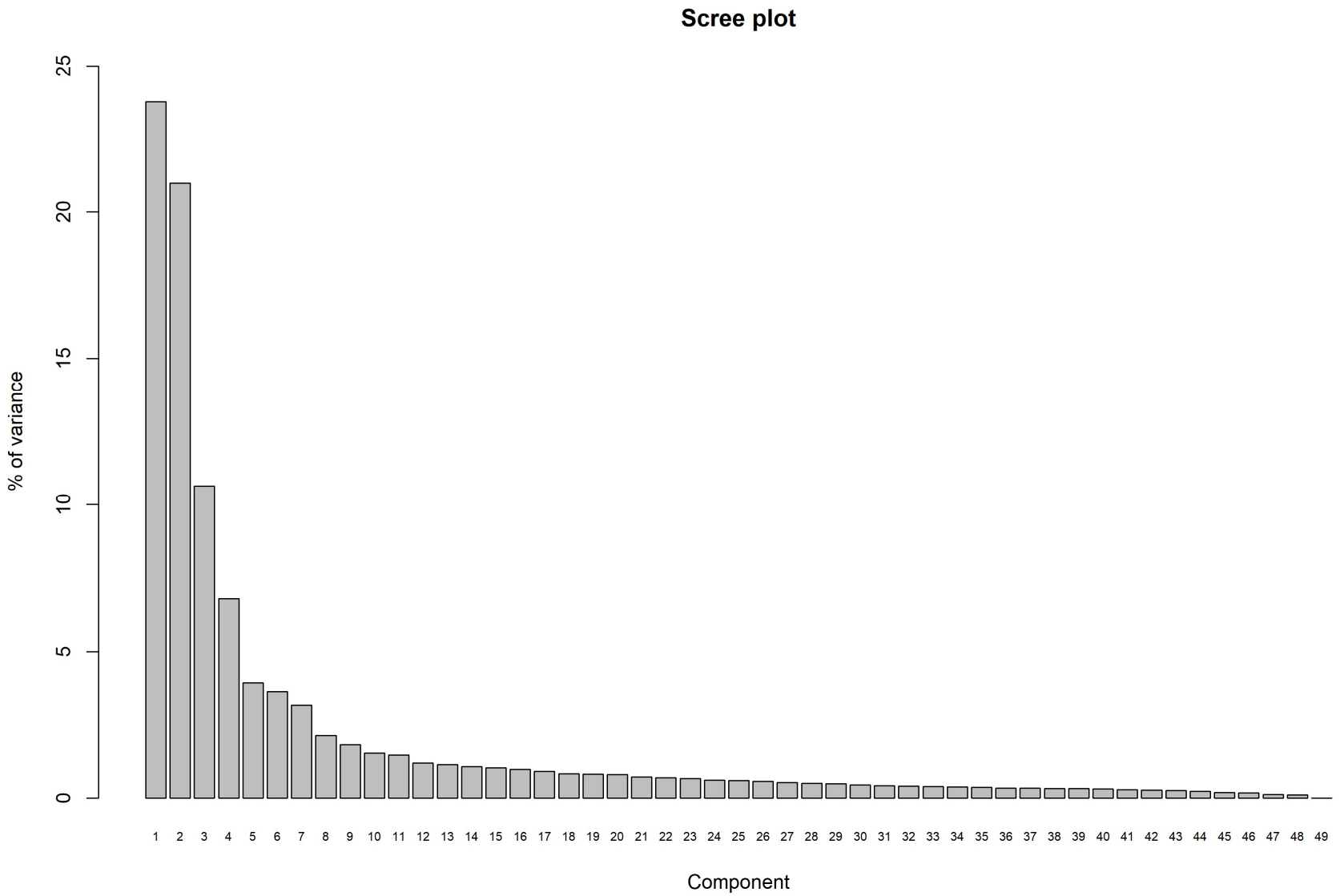
Data: Gene expression of 45,015 transcripts in 33 patients and 6 cell lines. Among the patients, 23 were from less aggressive group and 10 from more aggressive group. For each patient, a biopsy was sampled, the RNA extracted, sequenced using standard protocols, preprocessed and transformed and normalized so that the resulting data is approximately normal distributed. For each patient, the age and gender was also recorded.

Aim: Use PCA so investigate patterns between the patients.

The PCA was performed in R using **prcomp.** The principal components were plotted using the **plot** function.


Cell lines
Primary
Metastasis
Controls
Principal component 2 (20.98%)
Principal component 1 (23.76%)
300
200
100
0
-100
-200
-400
-300
-200
-100
0
100
200


Principal component 4 (6.83%)
200
100
0
-100
-200
-150
-100
-50
0
50
100
150
200
Principal component 3 (10.64%)
Controls
Metastasis
Primary
Cell lines


Scree plot
% of variance
0
5
10
15
20
25
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
Component

Clustering and PCA – some remarks

- Clustering and PCA are explorative methods. They can be used to visualize data and formulate hypothesis but they does not result in any statistical test (and thus no measure of the significance).
- Many methods automatically center and scale the data. You may need to manually turn such features off.
- There is a wide range of different tools for clustering and PCA. The command **heatmap** can for example cluster and display the results as a heatmap and a dendrogram.

Summary – Unsupervised methods

- Unsupervised methods can be used to explore and visualize high-dimensional data.
- Clustering is a technique used to find groups of data points (‘clusters’). Common clustering methods includes hierarchical clustering and k-means clustering. A distance metric and linkage criterion is needed to perform hierarchical clustering.
- Principal component analysis can be used to reduce the dimension of the data. The principal components are identified based on the dimension on where the variability of the data is as high as possible.