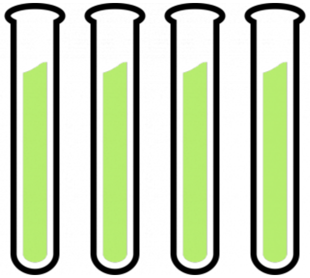
# Clustering

Dr. Holger Fröhlich

SS 2016

# OMICs-Data (e.g. gene expression)



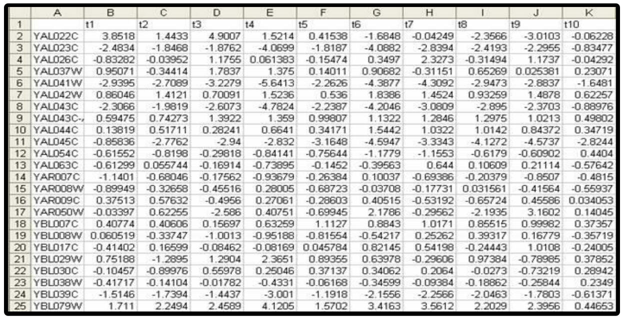
Biological Samples

High-Throughput Measurements
(NGS, Microarrays)

Preprocessing

n Samples

p Genes / Transcripts



|  | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 |  | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | t10 |
| 2 | YAL022C | 3.8518 | 1.4433 | 4.9007 | 1.5214 | 0.41538 | -1.6848 | -0.04249 | -2.3566 | -3.0103 | -0.06228 |
| 3 | YAL023C | -2.4834 | -1.8468 | -1.8762 | -4.0699 | -1.8187 | -4.0882 | -2.8394 | -2.4193 | -2.2955 | -0.83477 |
| 4 | YAL026C | -0.83282 | -0.03952 | 1.1755 | 0.061383 | -0.15474 | 0.3497 | 2.3273 | -0.31494 | 1.1737 | -0.04292 |
| 5 | YAL037W | 0.95071 | -0.34414 | 1.7837 | 1.375 | 0.14011 | 0.90682 | -0.31151 | 0.65269 | 0.025381 | 0.23071 |
| 6 | YAL041W | -2.9395 | -2.7089 | -3.2279 | -5.6413 | -2.2626 | -4.3877 | -4.3092 | -2.9473 | -2.8837 | -1.6481 |
| 7 | YAL042W | 0.86046 | 1.4121 | 0.70091 | 1.5236 | 0.536 | 1.8386 | 1.4524 | 0.93259 | 1.4878 | 0.62257 |
| 8 | YAL043C | -2.3066 | -1.9819 | -2.6073 | -4.7824 | -2.2387 | -4.2046 | -3.0809 | -2.895 | -2.3703 | -0.88976 |
| 9 | YAL043C- | 0.59475 | 0.74273 | 1.3922 | 1.359 | 0.99807 | 1.1322 | 1.2846 | 1.2975 | 1.0213 | 0.49802 |
| 10 | YAL044C | 0.13819 | 0.51711 | 0.28241 | 0.6641 | 0.34171 | 1.5442 | 1.0322 | 1.0142 | 0.84372 | 0.34719 |
| 11 | YAL045C | -0.85836 | -2.7762 | -2.94 | -2.832 | -3.1648 | -4.5947 | -3.3343 | -4.1272 | -4.5737 | -2.8244 |
| 12 | YAL054C | -0.61552 | -0.8198 | -0.29818 | -0.84141 | -0.75644 | -1.1779 | -1.1553 | -0.6179 | -0.60902 | 0.4404 |
| 13 | YAL063C | -0.61299 | 0.055744 | -0.16914 | -0.73895 | -0.1452 | -0.39563 | 0.644 | 0.10609 | 0.21114 | -0.57642 |
| 14 | YAR007C | -1.1401 | -0.68046 | -0.17562 | -0.93679 | -0.26384 | 0.10037 | -0.69386 | -0.20379 | -0.8507 | -0.4815 |
| 15 | YAR008W | -0.89949 | -0.32658 | -0.45516 | 0.28005 | -0.68723 | -0.03708 | -0.17731 | 0.031561 | -0.41564 | -0.55937 |
| 16 | YAR009C | 0.37513 | 0.57632 | -0.4956 | 0.27061 | -0.28603 | 0.40515 | -0.53192 | -0.65724 | 0.45586 | 0.034053 |
| 17 | YAR050W | -0.03397 | 0.62255 | -2.586 | 0.40751 | -0.69945 | 2.1786 | -0.29562 | -2.1935 | 3.1602 | 0.14045 |
| 18 | YBL007C | 0.40774 | 0.40606 | 0.15697 | 0.63259 | 1.1127 | 0.8843 | 1.0171 | 0.85515 | 0.99982 | 0.37357 |
| 19 | YBL008W | 0.060519 | -0.33747 | -1.0013 | -0.95188 | -0.81554 | -0.54217 | 0.25262 | 0.39317 | 0.16779 | -0.35719 |
| 20 | YBL017C | -0.41402 | 0.16599 | -0.08462 | -0.08169 | 0.045784 | 0.82145 | 0.54198 | -0.24443 | 1.0108 | -0.24005 |
| 21 | YBL029W | 0.75188 | -1.2895 | 1.2904 | 2.3651 | 0.89355 | 0.63978 | -0.29606 | 0.97384 | -0.78985 | 0.37852 |
| 22 | YBL030C | -0.10457 | -0.89976 | 0.55978 | 0.25046 | 0.37137 | 0.34062 | 0.2064 | -0.0273 | -0.73219 | 0.28942 |
| 23 | YBL038W | -0.41717 | -0.14104 | -0.01782 | -0.4331 | -0.06168 | -0.34599 | -0.09384 | -0.18862 | -0.25844 | 0.2349 |
| 24 | YBL039C | -1.5146 | -1.7394 | -1.4437 | -3.001 | -1.1918 | -2.1556 | -2.2566 | -2.0463 | -1.7803 | -0.61371 |
| 25 | YBL079W | 1.711 | 2.2494 | 2.4589 | 4.1205 | 1.5702 | 3.4163 | 3.5612 | 2.2029 | 2.3956 | 0.44653 |

...

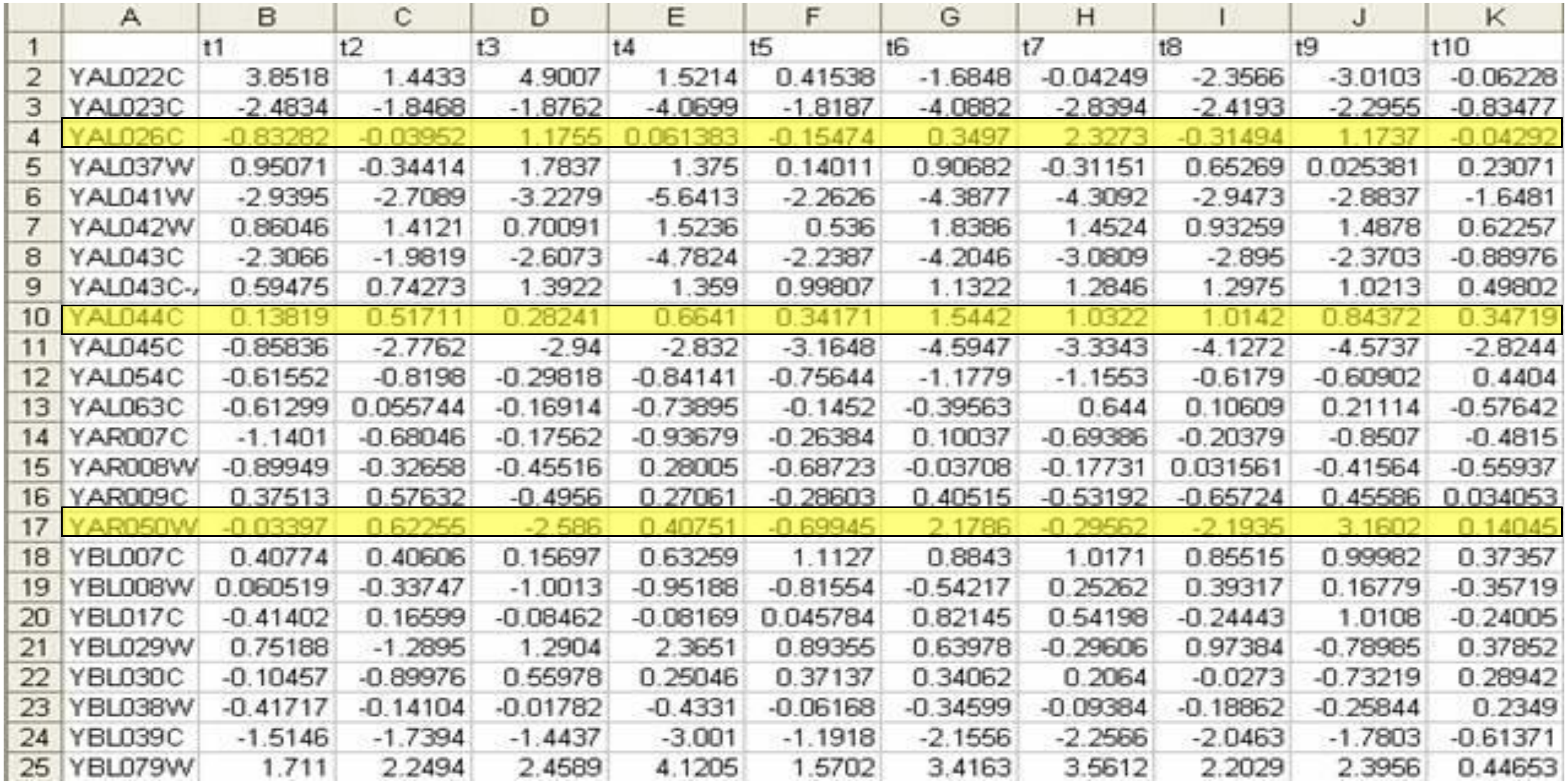
p = 20.000 – 50.000

**n << p**

patients: n <= few 100

Cells / cell lines: n <= 5 (per biol. condition)

# Cluster Analysis

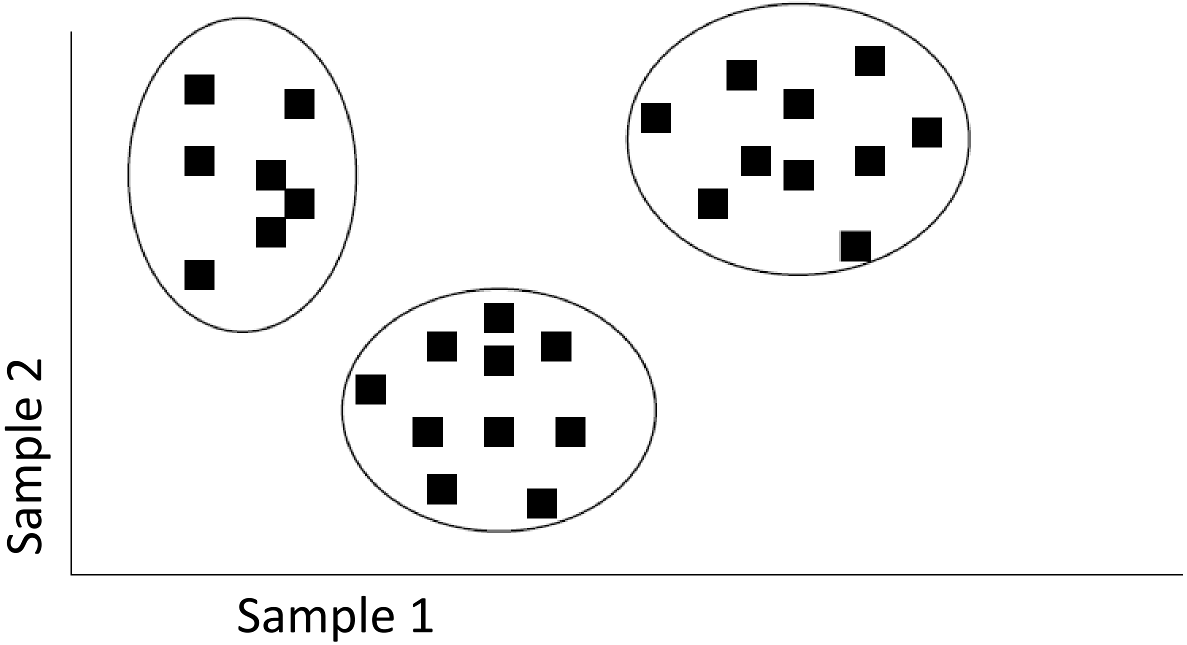


| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | t10 |
| 2 | YAL022C | 3.8518 | 1.4433 | 4.9007 | 1.5214 | 0.41538 | -1.6848 | -0.04249 | -2.3566 | -3.0103 | -0.06228 |
| 3 | YAL023C | -2.4834 | -1.8468 | -1.8762 | -4.0699 | -1.8187 | -4.0882 | -2.8394 | -2.4193 | -2.2955 | -0.83477 |
| 4 | YAL026C | -0.83282 | -0.03952 | 1.1755 | 0.061383 | -0.15474 | 0.3497 | 2.3273 | -0.31494 | 1.1737 | -0.04292 |
| 5 | YAL037W | 0.95071 | -0.34414 | 1.7837 | 1.375 | 0.14011 | 0.90682 | -0.31151 | 0.65269 | 0.025381 | 0.23071 |
| 6 | YAL041W | -2.9395 | -2.7089 | -3.2279 | -5.6413 | -2.2626 | -4.3877 | -4.3092 | -2.9473 | -2.8837 | -1.6481 |
| 7 | YAL042W | 0.86046 | 1.4121 | 0.70091 | 1.5236 | 0.536 | 1.8386 | 1.4524 | 0.93259 | 1.4878 | 0.62257 |
| 8 | YAL043C | -2.3066 | -1.9819 | -2.6073 | -4.7824 | -2.2387 | -4.2046 | -3.0809 | -2.895 | -2.3703 | -0.88976 |
| 9 | YAL043C-/ | 0.59475 | 0.74273 | 1.3922 | 1.359 | 0.99807 | 1.1322 | 1.2846 | 1.2975 | 1.0213 | 0.49802 |
| 10 | YAL044C | 0.13819 | 0.51711 | 0.28241 | 0.6641 | 0.34171 | 1.5442 | 1.0322 | 1.0142 | 0.84372 | 0.34719 |
| 11 | YAL045C | -0.85836 | -2.7762 | -2.94 | -2.832 | -3.1648 | -4.5947 | -3.3343 | -4.1272 | -4.5737 | -2.8244 |
| 12 | YAL054C | -0.61552 | -0.8198 | -0.29818 | -0.84141 | -0.75644 | -1.1779 | -1.1553 | -0.6179 | -0.60902 | 0.4404 |
| 13 | YAL063C | -0.61299 | 0.055744 | -0.16914 | -0.73895 | -0.1452 | -0.39563 | 0.644 | 0.10609 | 0.21114 | -0.57642 |
| 14 | YAR007C | -1.1401 | -0.68046 | -0.17562 | -0.93679 | -0.26384 | 0.10037 | -0.69386 | -0.20379 | -0.8507 | -0.4815 |
| 15 | YAR008W | -0.89949 | -0.32658 | -0.45516 | 0.28005 | -0.68723 | -0.03708 | -0.17731 | 0.031561 | -0.41564 | -0.55937 |
| 16 | YAR009C | 0.37513 | 0.57632 | -0.4956 | 0.27061 | -0.28603 | 0.40515 | -0.53192 | -0.65724 | 0.45586 | 0.034053 |
| 17 | YAR050W | -0.03397 | 0.62255 | -2.586 | 0.40751 | -0.69945 | 2.1786 | -0.29562 | -2.1935 | 3.1602 | 0.14045 |
| 18 | YBL007C | 0.40774 | 0.40606 | 0.15697 | 0.63259 | 1.1127 | 0.8843 | 1.0171 | 0.85515 | 0.99982 | 0.37357 |
| 19 | YBL008W | 0.060519 | -0.33747 | -1.0013 | -0.95188 | -0.81554 | -0.54217 | 0.25262 | 0.39317 | 0.16779 | -0.35719 |
| 20 | YBL017C | -0.41402 | 0.16599 | -0.08462 | -0.08169 | 0.045784 | 0.82145 | 0.54198 | -0.24443 | 1.0108 | -0.24005 |
| 21 | YBL029W | 0.75188 | -1.2895 | 1.2904 | 2.3651 | 0.89355 | 0.63978 | -0.29606 | 0.97384 | -0.78985 | 0.37852 |
| 22 | YBL030C | -0.10457 | -0.89976 | 0.55978 | 0.25046 | 0.37137 | 0.34062 | 0.2064 | -0.0273 | -0.73219 | 0.28942 |
| 23 | YBL038W | -0.41717 | -0.14104 | -0.01782 | -0.4331 | -0.06168 | -0.34599 | -0.09384 | -0.18862 | -0.25844 | 0.2349 |
| 24 | YBL039C | -1.5146 | -1.7394 | -1.4437 | -3.001 | -1.1918 | -2.1556 | -2.2566 | -2.0463 | -1.7803 | -0.61371 |
| 25 | YBL079W | 1.711 | 2.2494 | 2.4589 | 4.1205 | 1.5702 | 3.4163 | 3.5612 | 2.2029 | 2.3956 | 0.44653 |

- Find groups (clusters) of genes with similar expressions profile (*co-expressed* genes)
- No true grouping known: unsupervised learning problem

# Geometric Interpretation

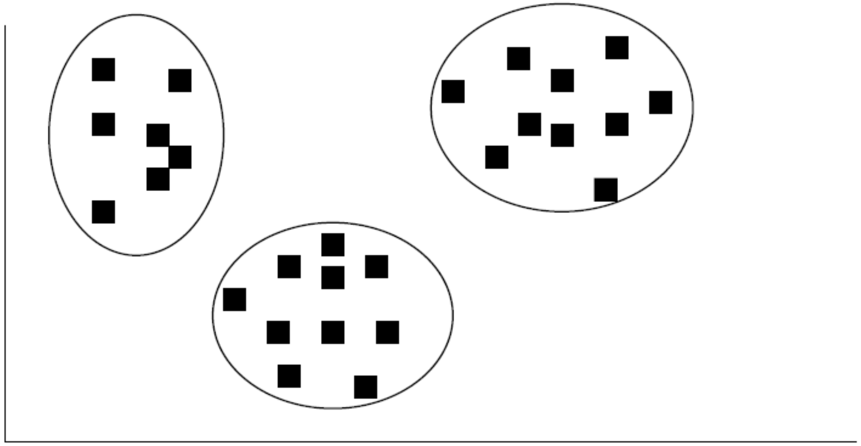
- Genes = points
- Find clusters of similar points
- No true clustering known
- How to find a „good“ clustering?



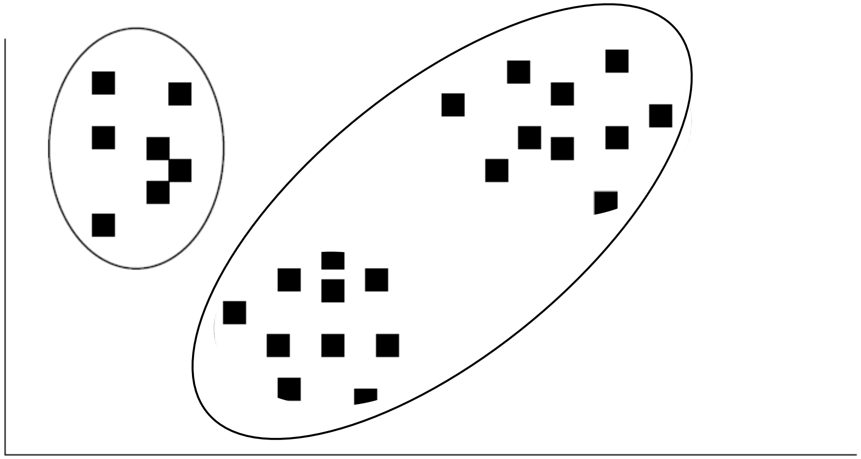
# Homogeneity and separation principle

- **Homogeneity:** Elements within a cluster are close to each other
- **Separation:** Elements in different clusters are further apart from each other

good



bad



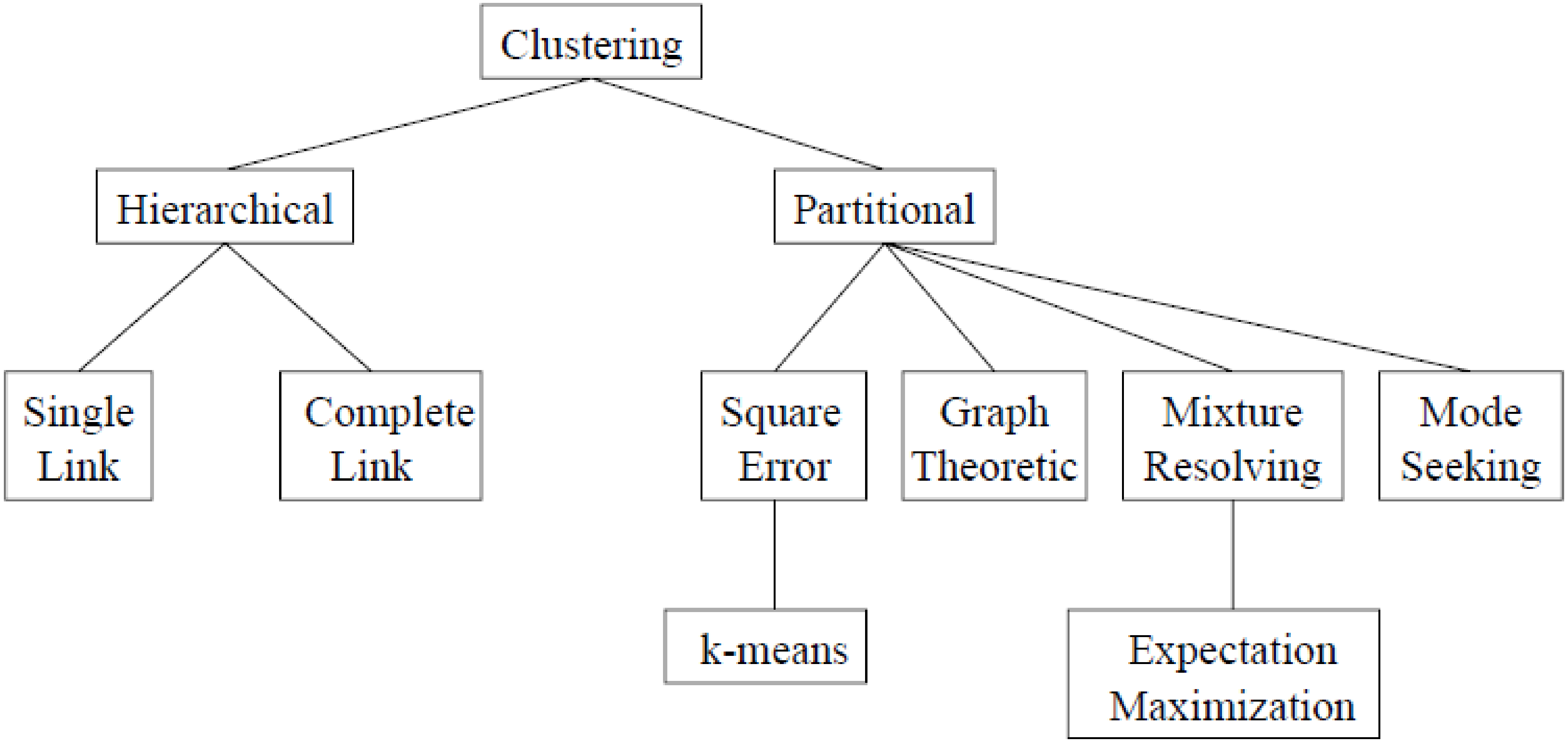
# Clustering algorithms

- No mathematically unique way of defining these two general criteria
  - *ill defined problem*

- Many clustering algorithms following different approaches
  - Success of individual algorithm is data dependent

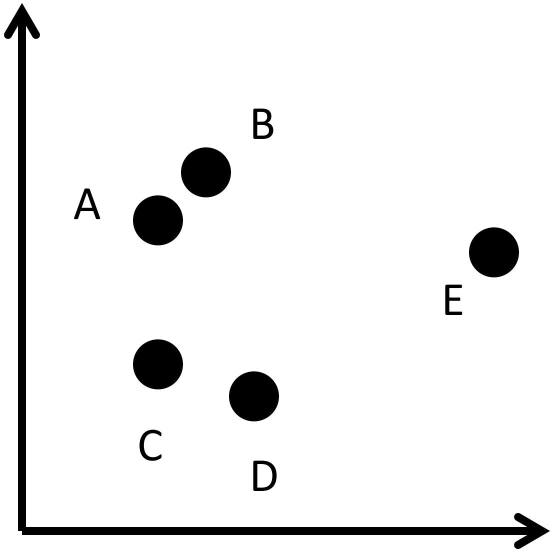
- Difficult to validate clustering results
  - Clustering is (usually) *unsupervised*: there is no objective ground truth or „true“ clustering

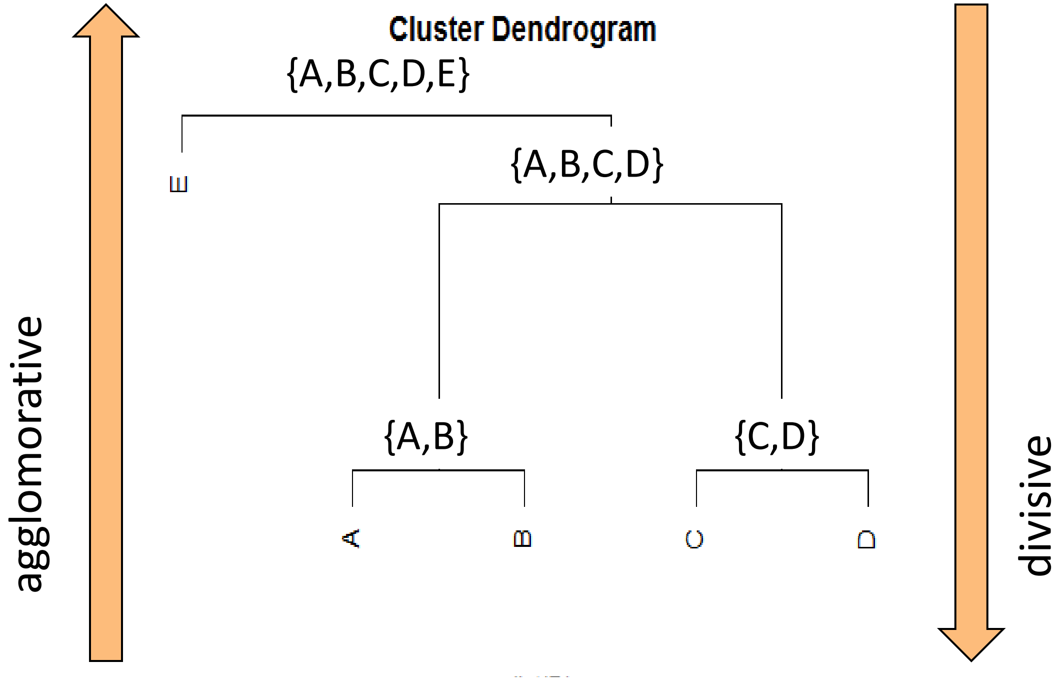
# Taxonomy of clustering techniques (Jain et al., 1999)



# Hierarchical vs. Partitioing Clustering

- **Partitioning**: Given goal function and fixed number k of clusters
  - Divide objects in k partitions, such that goal function is optimizied (e.g. mean distance to cluster centers)
- **Hierarchical:** agglomorative oder divisive
  - **Now**: agglomorative hierarch. clustering





# Proximity Measures

- Agglomorative hier. Clustering depends on:
  - How we measure the proximities of objects (genes)
  - How we measure the similarity of clusters (and thus fuse most similar ones)

- For metric data (such as gene expression) distances can be used for objects.
- Minkowski / p-norm distance / metric:

$$d(x,y) = \left( \sum_{i=1}^{n} | x[i] - y[i] |^{p} \right)^{1/p} = \| x - y \|_{p}$$

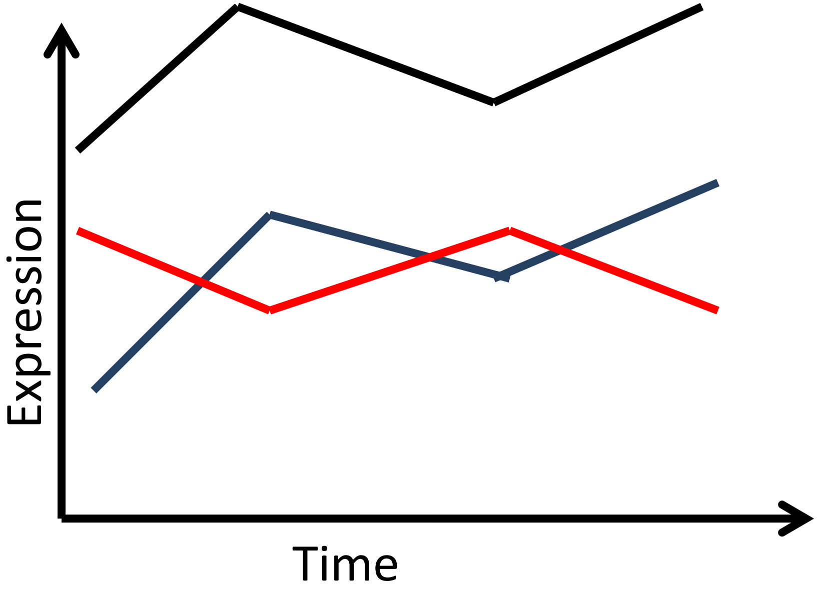
- Special cases:
  - Euclidian distance (p=2)
  - Manhattan distance (p=1)

# Distance measures

- Pearson correlation „distance“ (not a metric):

$$d(x,y) = 1 - \rho_{x,y} = 1 - \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

- Often used for time series clustering

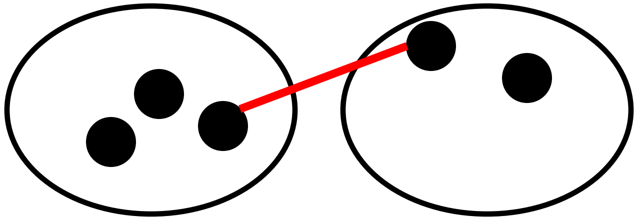


•Euclidean distance: green more similiar to rot

•Pearson distance: green more similiar to **black**

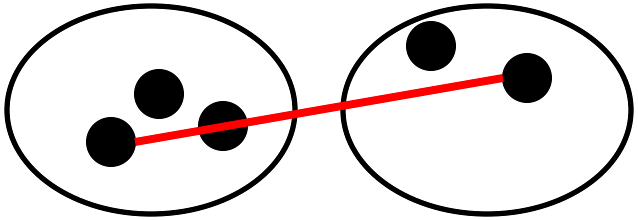
# Cluster fusion: How similar are two clusters?

## Single Linkage



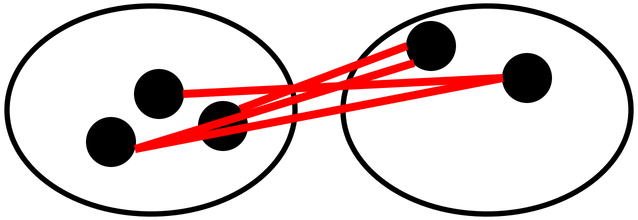
$$D_{SL}(A,B)=\min_{\substack{a\in A\\ b\in B}} d(a,b)$$

## Complete Linkage



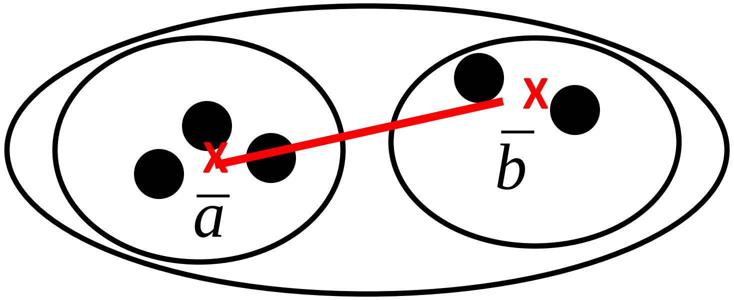
$$D_{CL}(A,B)=\max_{\substack{a\in A\\ b\in B}} d(a,b)$$

## Average Linkage



$$D_{AL}(A,B)=\frac{1}{|A||B|}\sum_{\substack{a\in A\\ b\in B}} d(a,b)$$

## Ward Criterion



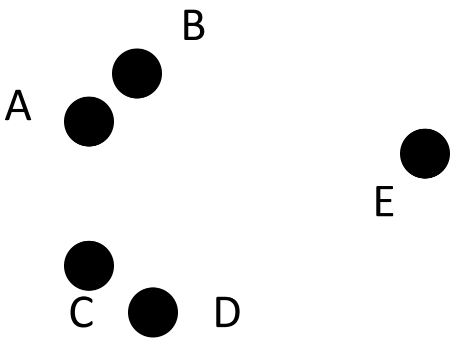
$$D_{Ward}(A,B)=\frac{d(\bar{a},\bar{b})}{1/|A|+1/|B|}$$

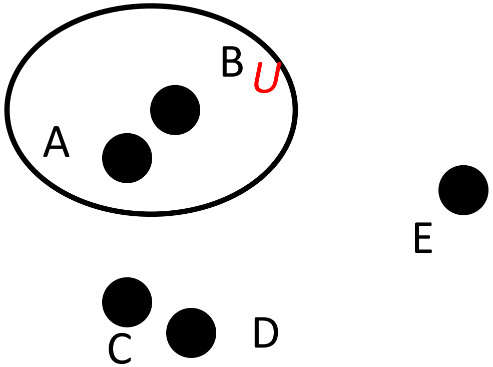
# Algorithm for agglomorative hierarchical Clustering

- Inputs:
  - n = Number of objects to cluster (genes)
  - D = n x n distance matrix
- Output: hierarchical clustering (*dendrogramm*)
- Example: **Complete Linkage Clustering**

|   | A | B | C | D | E |
|---|---|---|---|---|---|
| **A** | 0 | 1 | 2 | 3 | 5 |
| **B** | 1 | 0 | 3 | 4 | 4 |
| **C** | 2 | 3 | 0 | 1 | 4 |
| **D** | 3 | 4 | 1 | 0 | 2 |
| **E** | 5 | 4 | 4 | 2 | 0 |

|   | U | C | D | E |
|---|---|---|---|---|
| **U** | 0 | 3 | 4 | 5 |
| **C** | 3 | 0 | 1 | 4 |
| **D** | 4 | 1 | 0 | 2 |
| **E** | 5 | 4 | 2 | 0 |



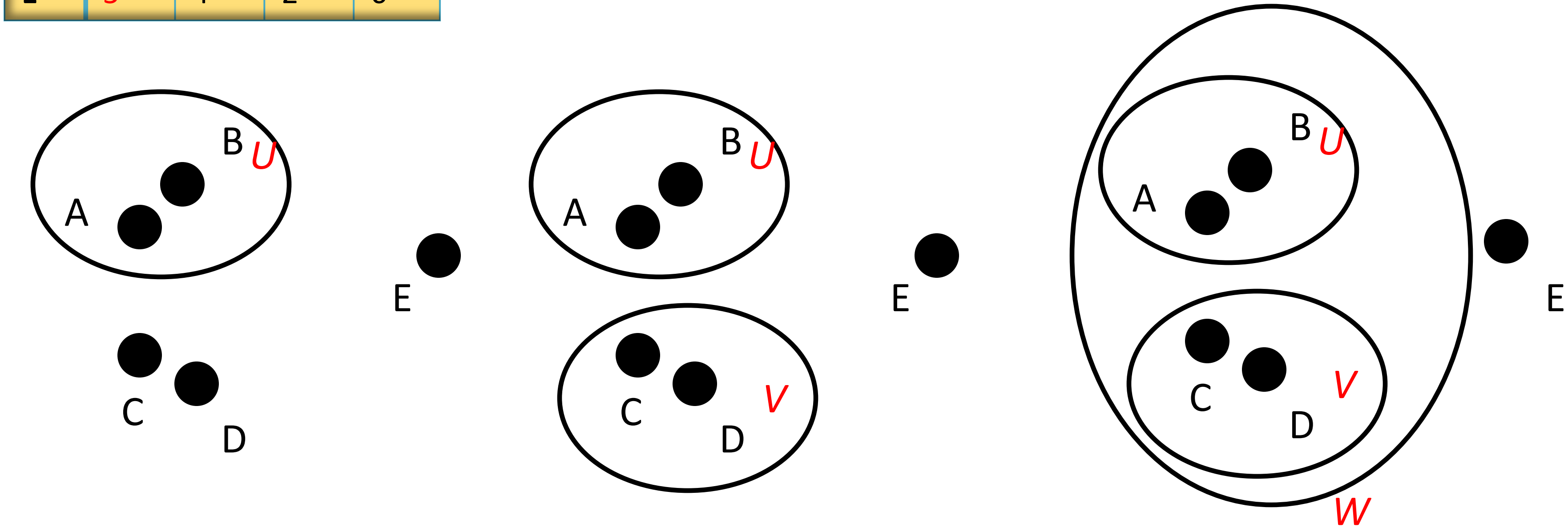


# Next Steps

| | U | C | D | E |
|---|---|---|---|---|
| **U** | 0 | 3 | 4 | 5 |
| **C** | 3 | 0 | 1 | 4 |
| **D** | 4 | 1 | 0 | 2 |
| **E** | 5 | 4 | 2 | 0 |

| | U | V | E |
|---|---|---|---|
| **U** | 0 | 4 | 5 |
| **V** | 4 | 0 | 4 |
| **E** | 5 | 3 | 0 |

| | W | E |
|---|---|---|
| **W** | 0 | 5 |
| **E** | 5 | 0 |



# Complexity

- n – 1 cluster merging steps
- In each of these steps O($n^2$) possibilities, to join clusters
- **(naive) overall complexity**: **O($n^3$)**
  - Using priority queues: O($n^2$ log n)
- For Single and Complete Linkage: improvement to O($n^2$) possible

- Agglomorative hierarchical clustering is relatively computational expensive
  - Typical applications: up to few thousand objects

# Lance & Willilams Formula

- Observation: algorithm only updates distances of new cluster U={A,B} to any existing cluster C
- General formula for update (**Lance & Williams, 1966**):

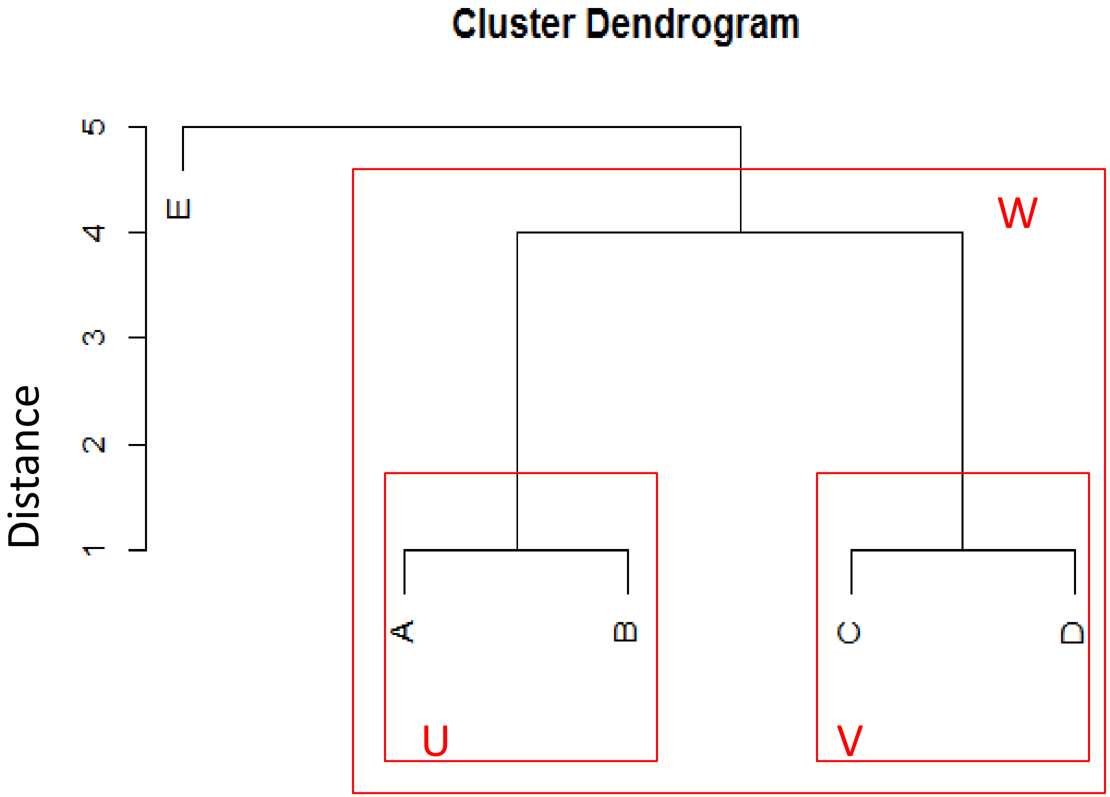
$$D(U,C) = \alpha_1 D(A,C) + \alpha_2 D(B,C) + \beta D(A,B) + \gamma \,|\, D(A,C) - D(B,C) \,|$$

| Methode | $\alpha_1$ | $\alpha_2$ | $\beta$ | $\gamma$ |
|---|---|---|---|---|
| Single L. | ½ | ½ | 0 | -½ |
| Complete L. | ½ | ½ | 0 | ½ |
| Weigthed group average L. | ½ | ½ | 0 | 0 |
| Unweighted group Av. L. | $\frac{\|A\|}{\|A\|+\|B\|}$ | $\frac{\|B\|}{\|A\|+\|B\|}$ | 0 | 0 |
| Ward | $\frac{\|A\|+\|C\|}{\|A\|+\|B\|+\|C\|}$ | $\frac{\|A\|+\|C\|}{\|A\|+\|B\|+\|C\|}$ | $-\frac{\|C\|}{\|A\|+\|B\|+\|C\|}$ | 0 |

**UPGMA**

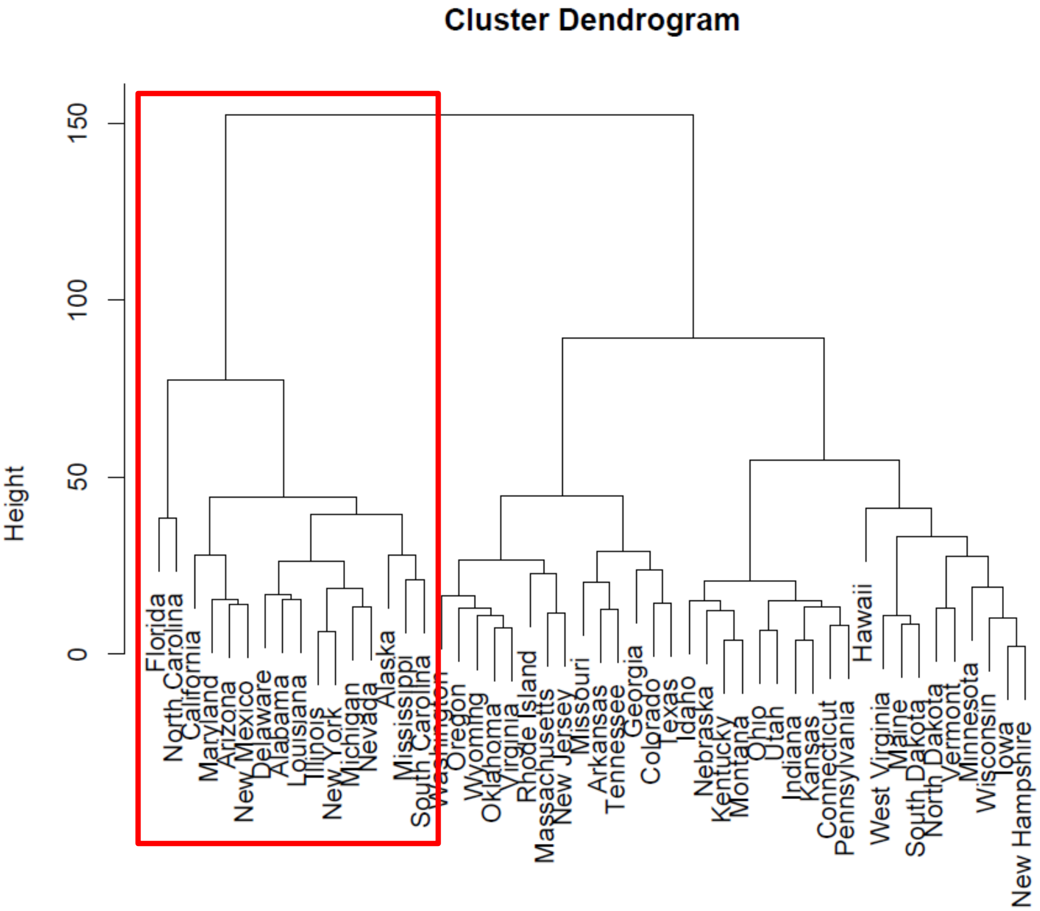
# Algorithm Output: The Dendrogramm

- Memorize during algorithm execution:
  - Which clusters were merged
  - Which distances they have



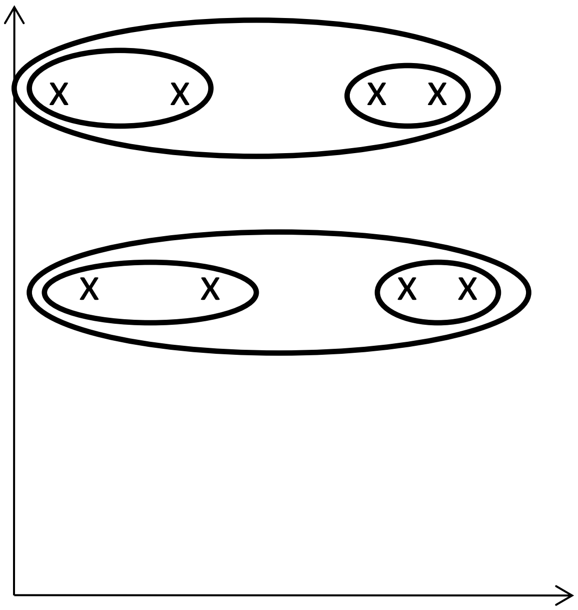
# Reading clustering dendrograms

- Clustering dendrogram is NOT unique!
- **Example**: subtree in red box could also be drawn on the right side!

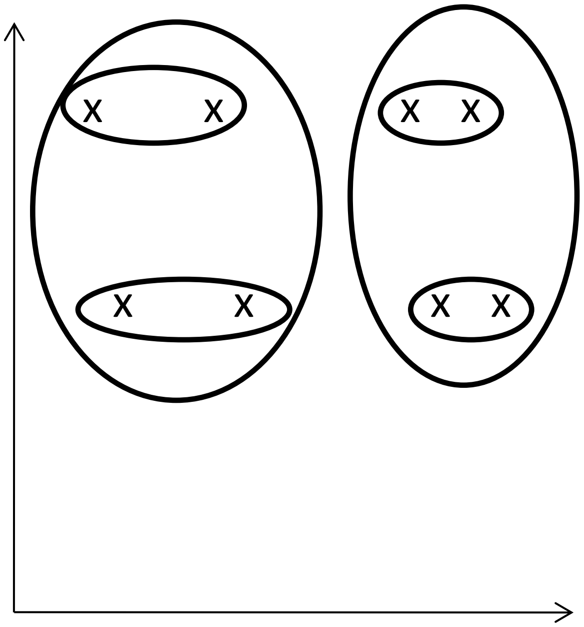


# Influence of cluster similarity measure on clustering result

Single linkage



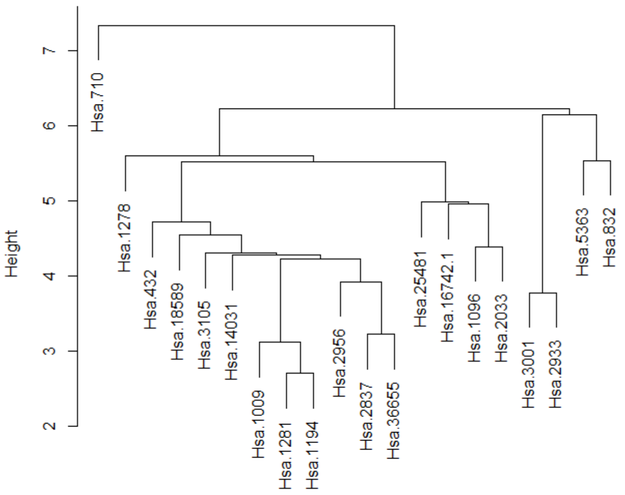
Complete linkage



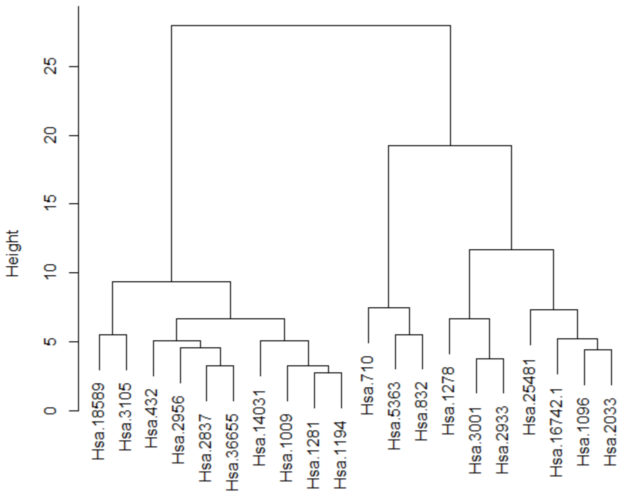
- Single linkage: long, chain-like clusters
- Complete linkage: small, ellipsoidal clusters
- Ward: spherical, isotropic clusters

# Example: Colon Cancer Data (Alon, 1999)

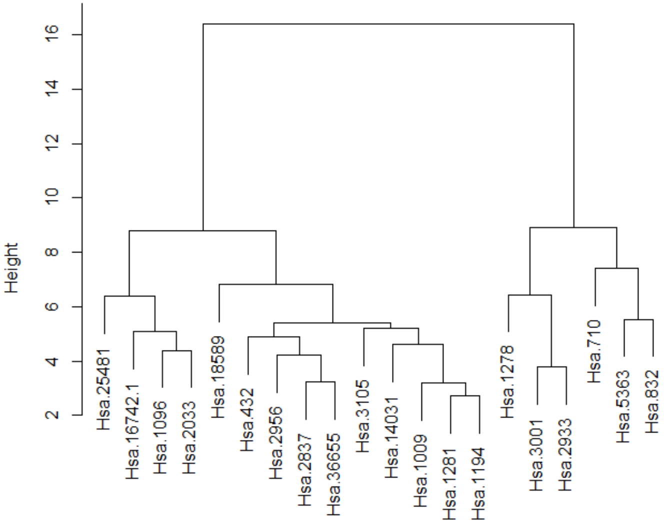
Single L.



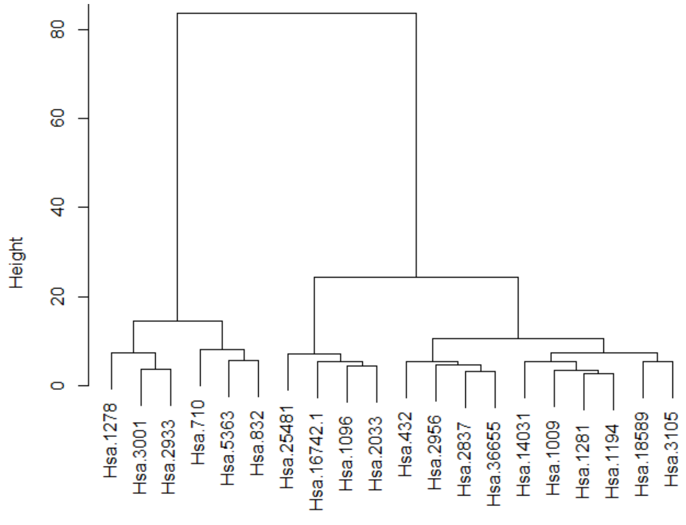
Complete L.



Average L.

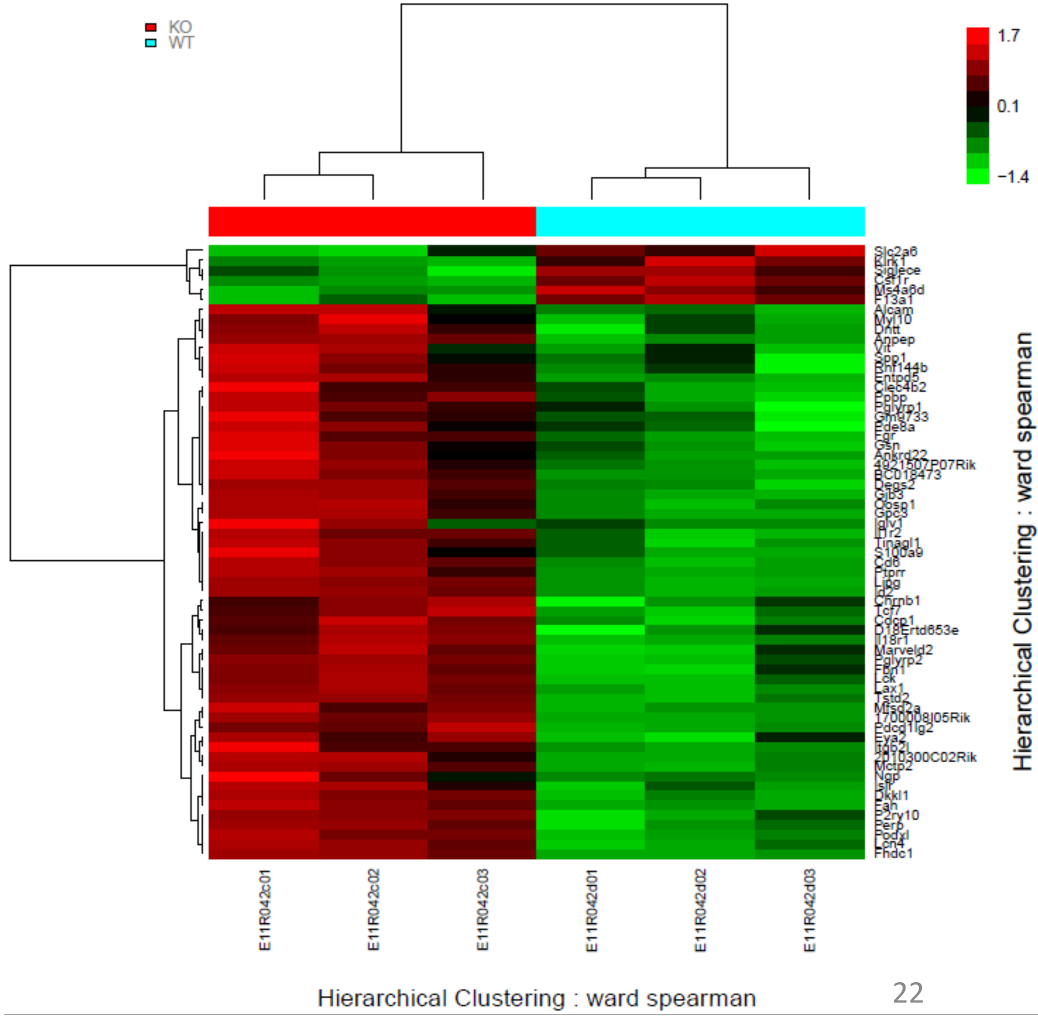


Ward



# Popular Application: Heatmaps

- Represent whole data matrix with false colors
  - Color code = expression level
- Group rows (genes) and columns (samples) via hierarchical clustering
  - Can see groups of related samples + genes at once



# Features of Hierarchical Clustering

- Advantages:
  - Visual data analysis: no pre-specified number of clusters
    - User can specify a cut point in the hierarchy, e.g. where intercluster distance exceeds some threshold
  - Organizes the clusters in a hierarchical way: dendrograms
    - BUT: dendrogram is not unique
  - Clustered heatmaps as application
- Drawbacks:
  - larger datasets: long run time, messy dendrograms

# Partioning Algorithms: K-means

- One of the most prominent **partioning** algorithms
- **Idea**: represent data in terms of K clusters, each summarized via prototype a $\mu_k$
- Each data point is assigned to one cluster
- Minimize distortion:

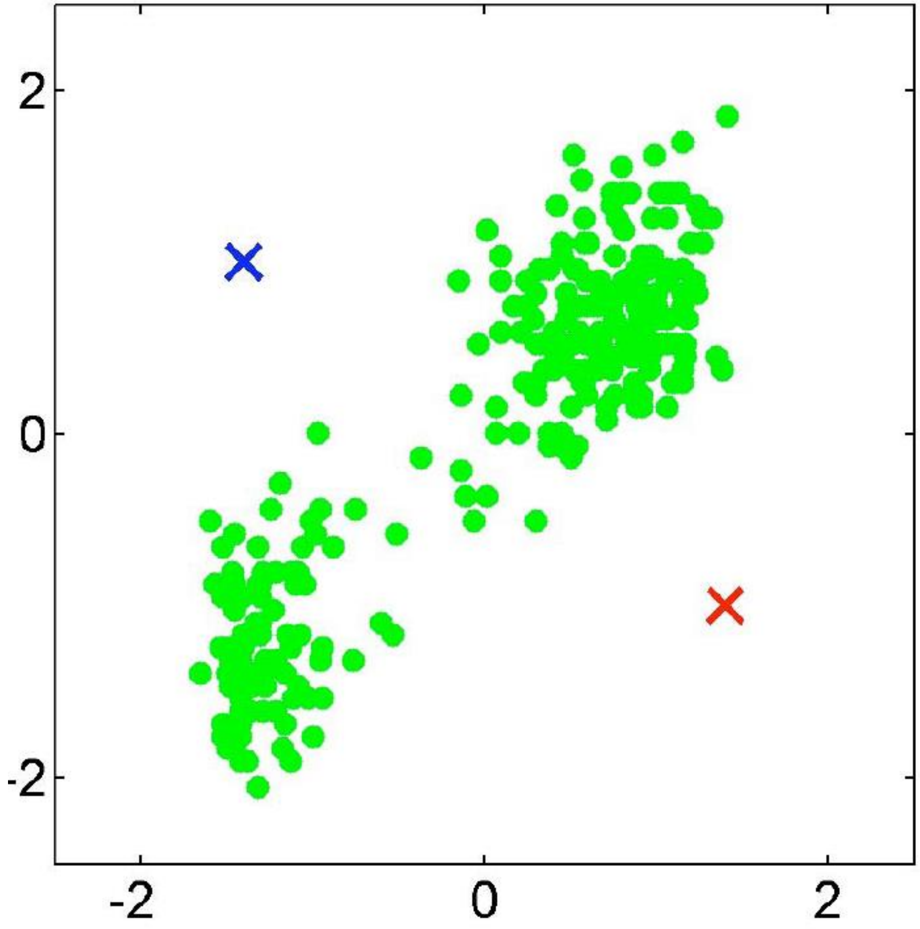
$$\sum_{k=1}^{K} \sum_{x_j \in C_k} \| x_j - \mu_k \|^2$$

# K-Means Clustering: Lloyd Algorithm

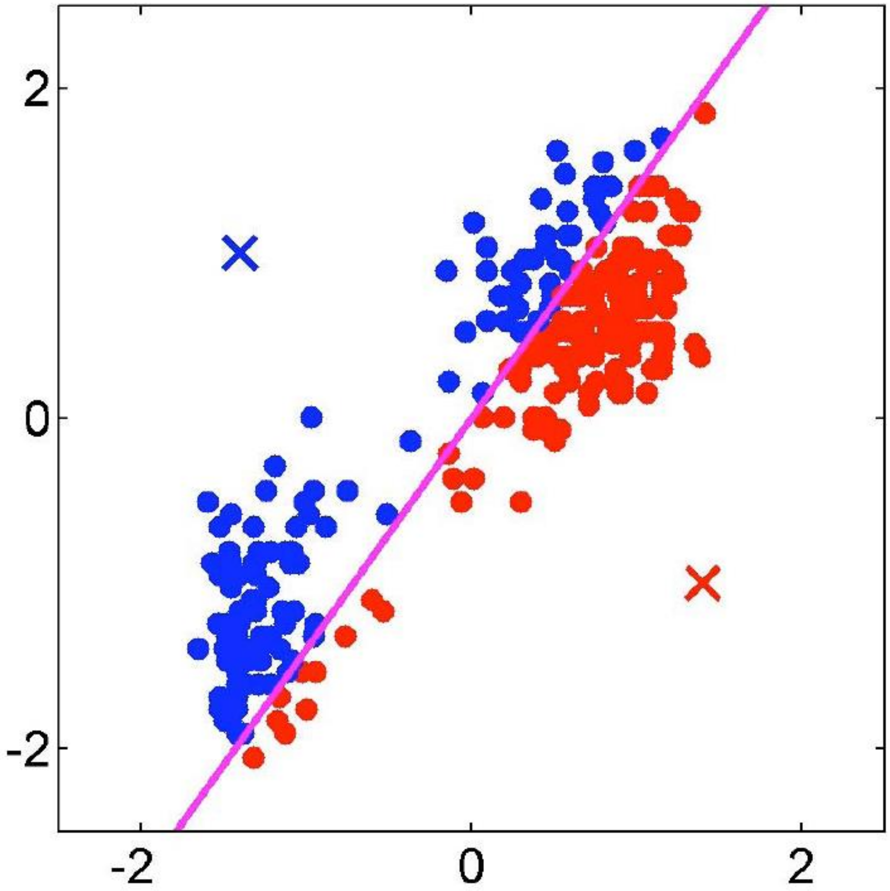
```
1. K-means(k)
2.    Arbitrarily assign the $k$ cluster centers
3.    while the cluster centers keep changing
4.       Assign each data point to the cluster $C_i$
         corresponding to the closest cluster
         representative (center)  ($1 \le i \le k$)
5.       After the assignment of all data points,
         compute new cluster representatives as
         $\mu_k = \frac{1}{|C_k|} \sum_{x_j \in C_k} x_j$
```

*Euclidean distance*

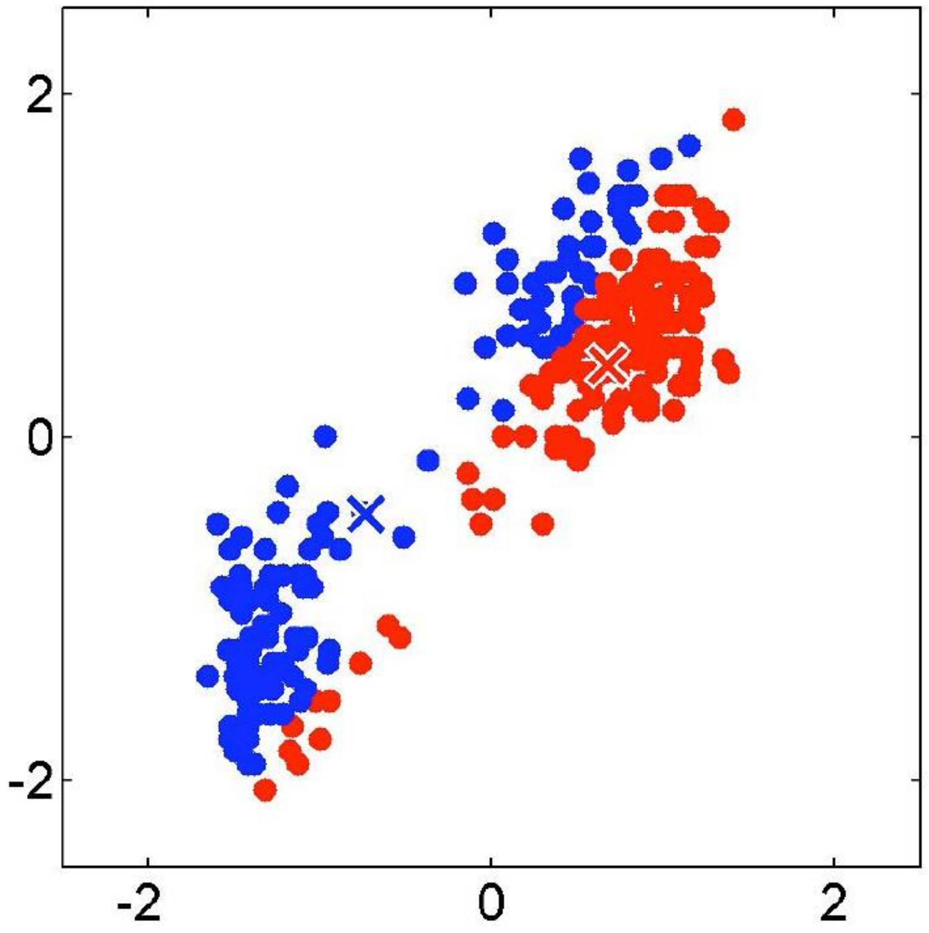
# Example: initialization



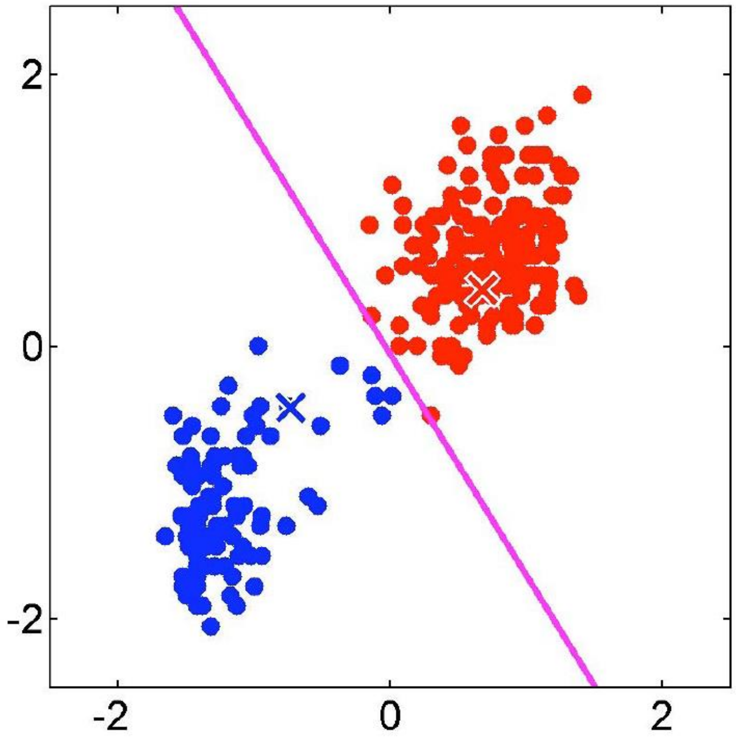
# Example: cluster assignment



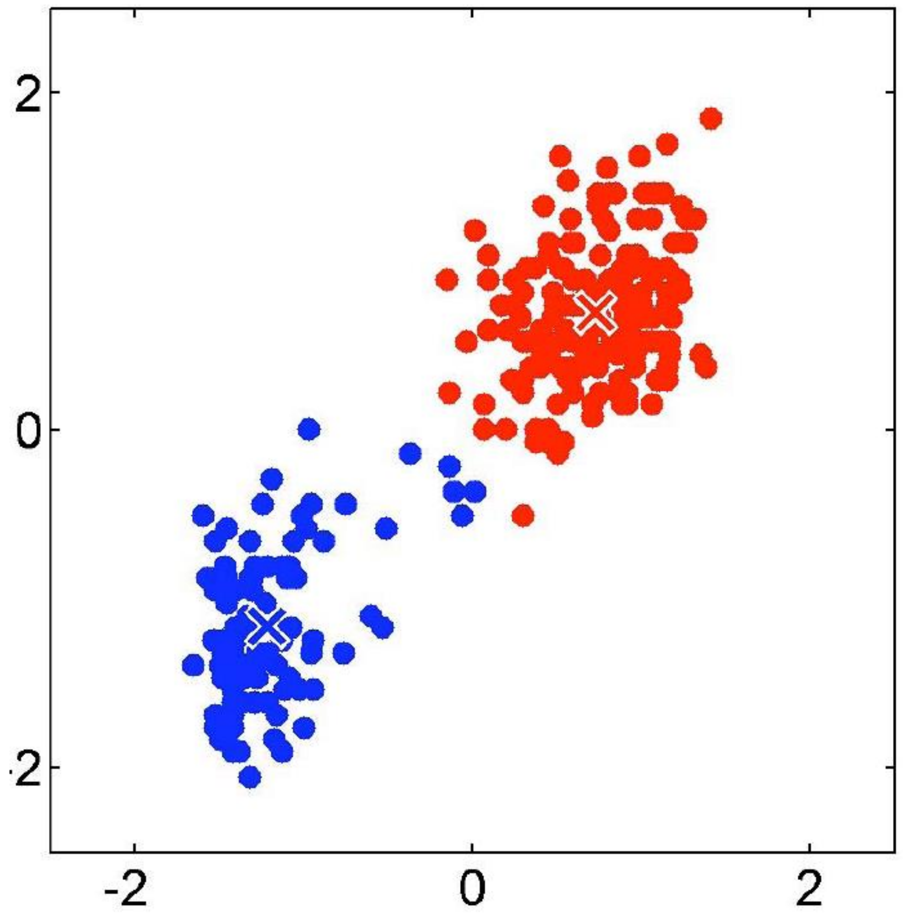
# Example: new cluster centers



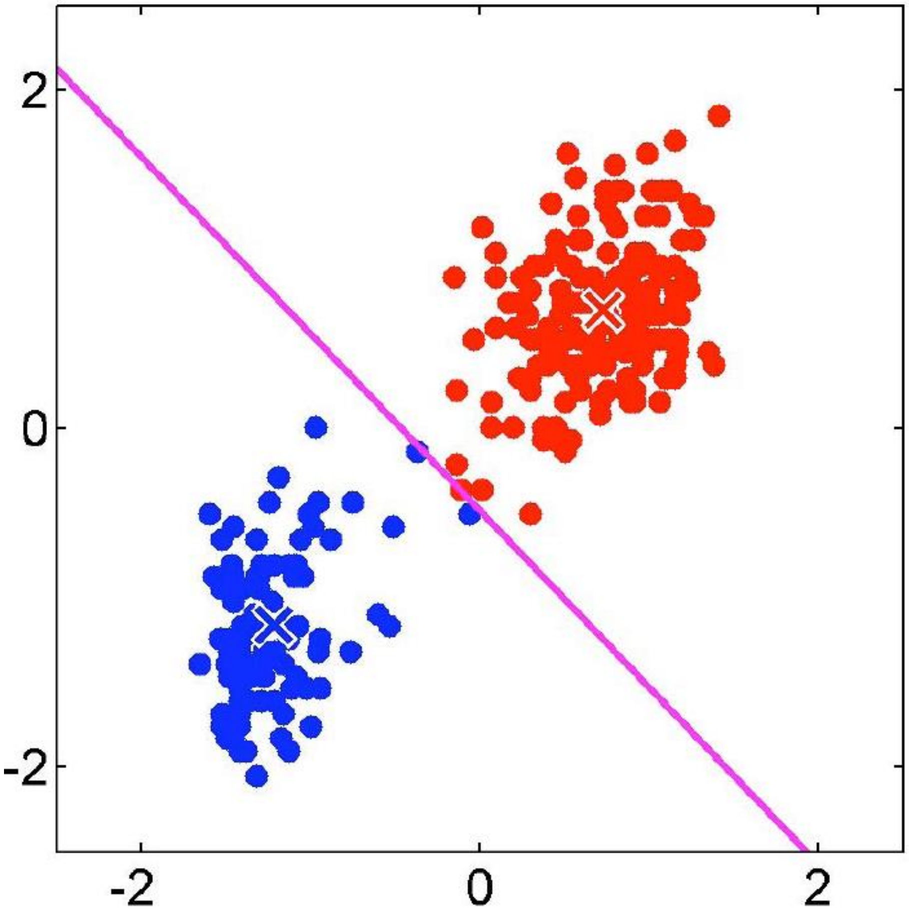
# Example: cluster re-assignment



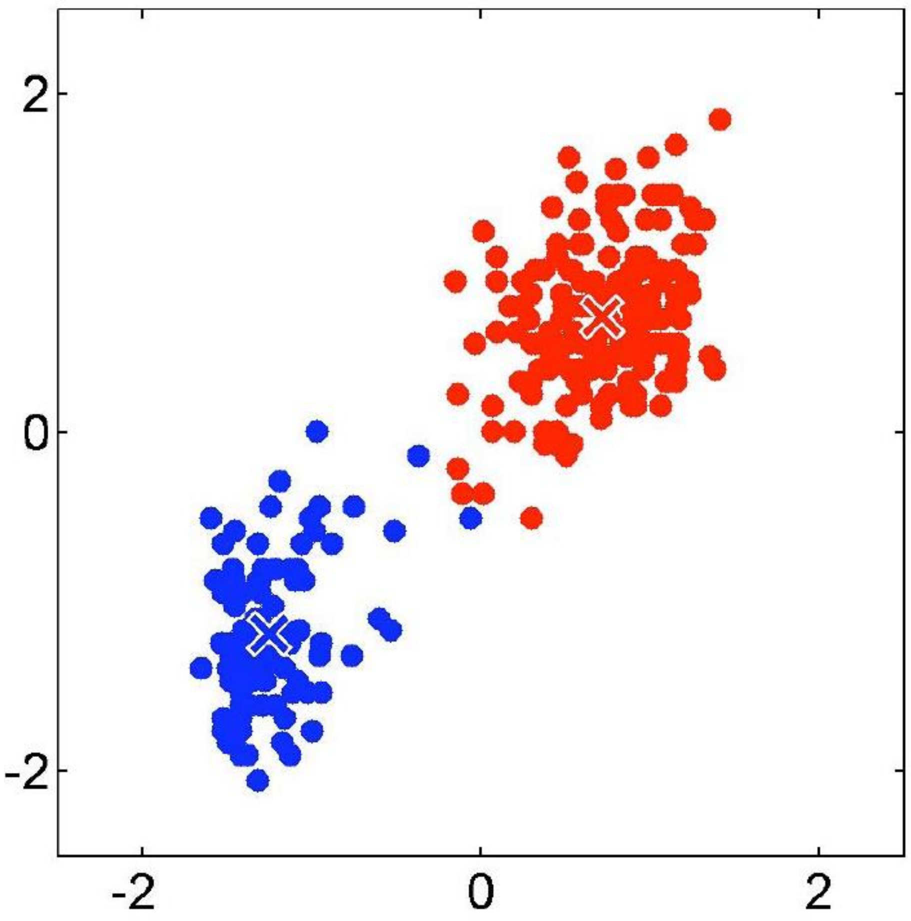
# Example: new cluster centers



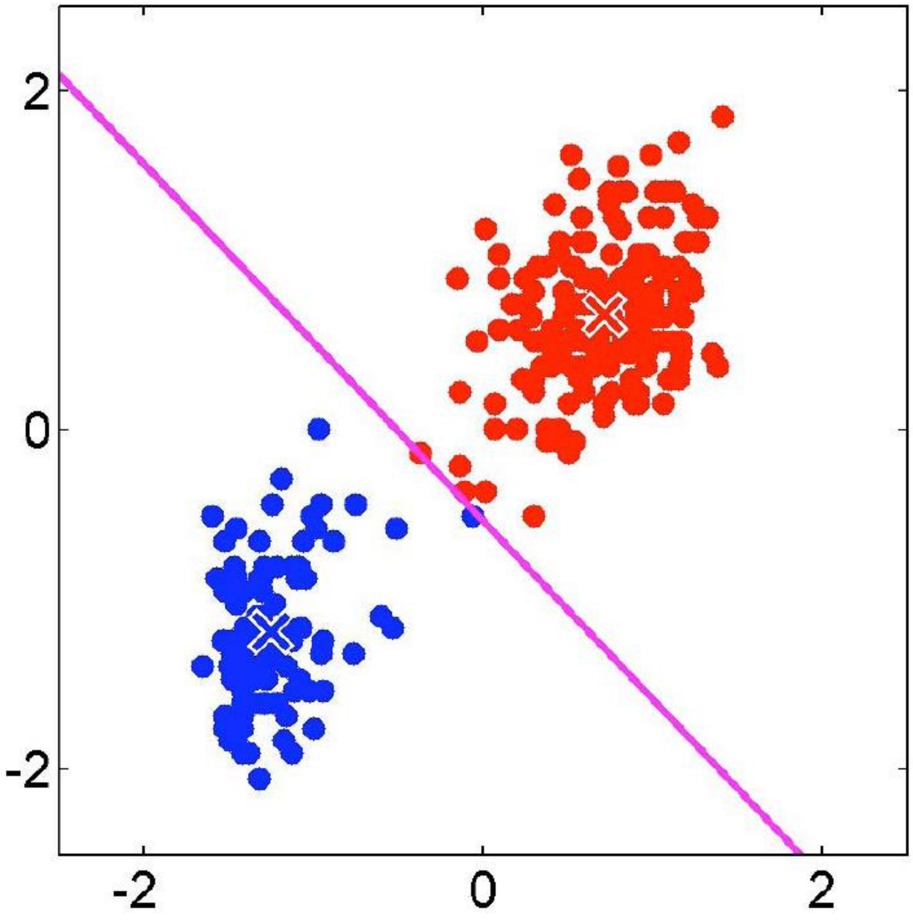
# Example: cluster re-assignment



# Example: final cluster centers



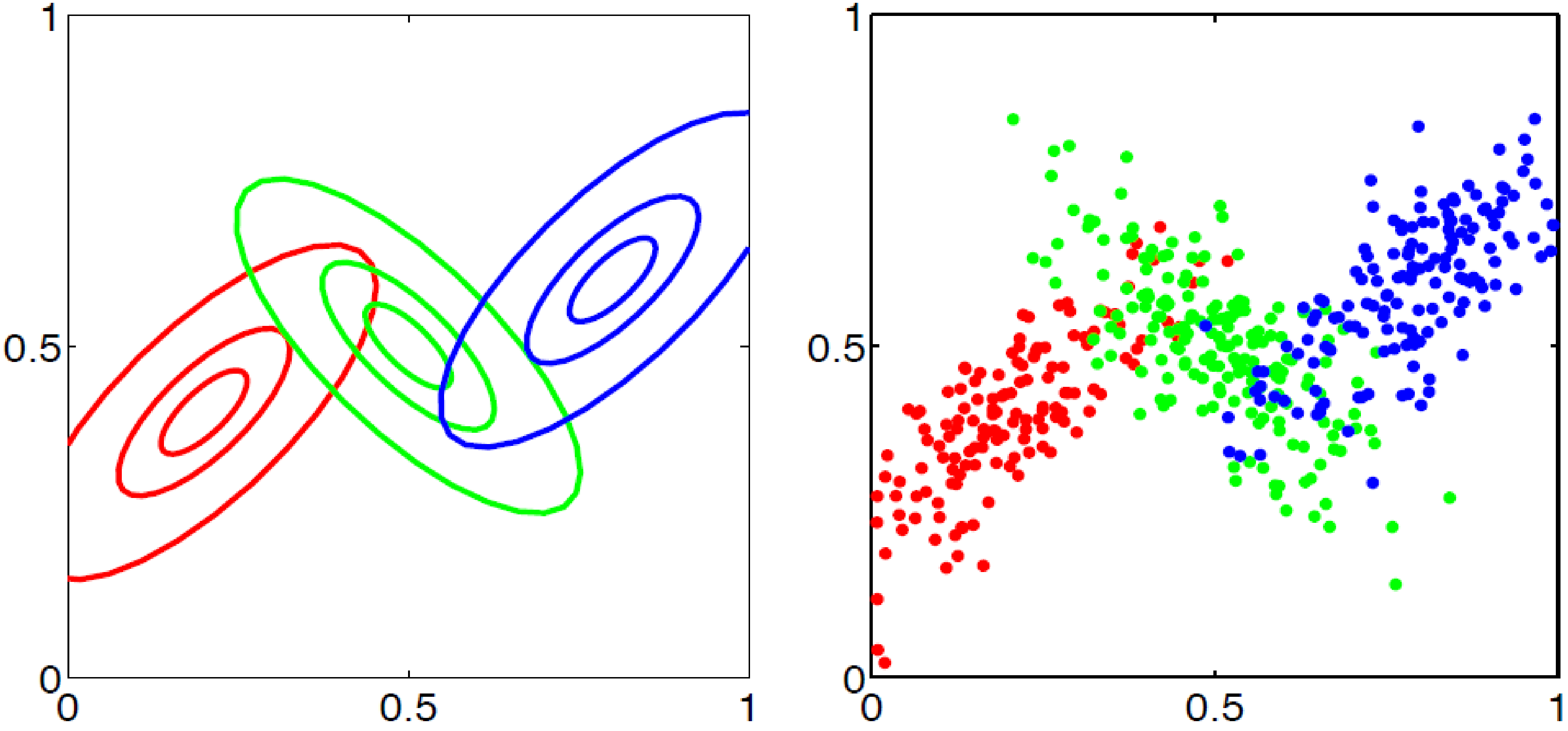
# Example: final cluster assignment



# Notes on k-means

- Algorithm usually converges very fast
  - complexity: O(n*k*d*i) with n = #samples; d = #variables; i = #iterations
- **BUT**: may lead to suboptimal solutions
- Clustering depends on initial conditions
  - Solution: repeat l times
- K-means can only detect spherical clusters!
- Hard assignment to clusters
  - Small shifts of a data point can flip cluster membership

# Example: Mixture of 3 Gaussians



# Multivariate Normal Distributions

$$N(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi \mid \boldsymbol{\Sigma} \mid)^{d/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

Mean vector
(d dimensions)

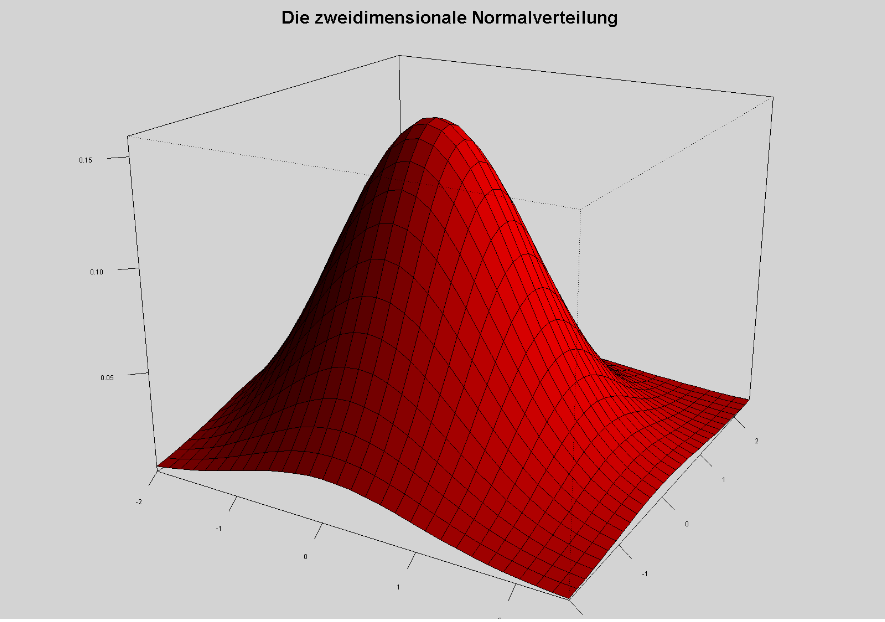
Coveriance matrix (d x d)

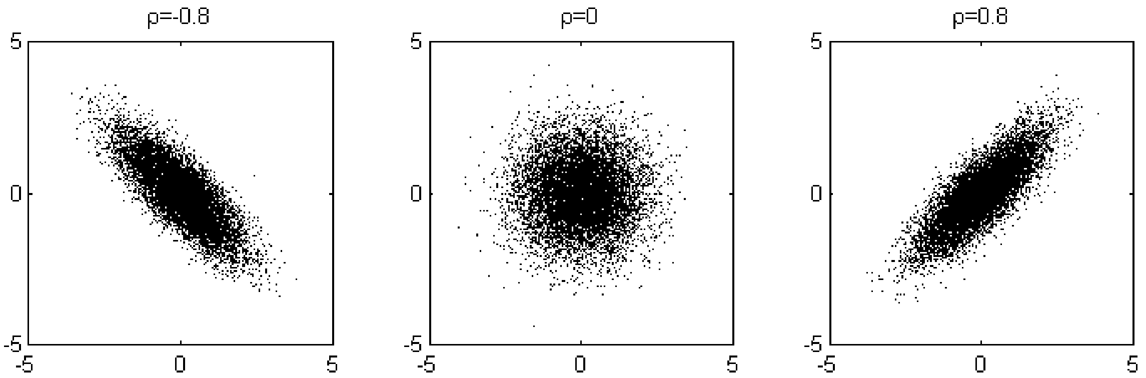
Maximum likelihood estimates:

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T$$

# Multivariate Normal Distributions - Example





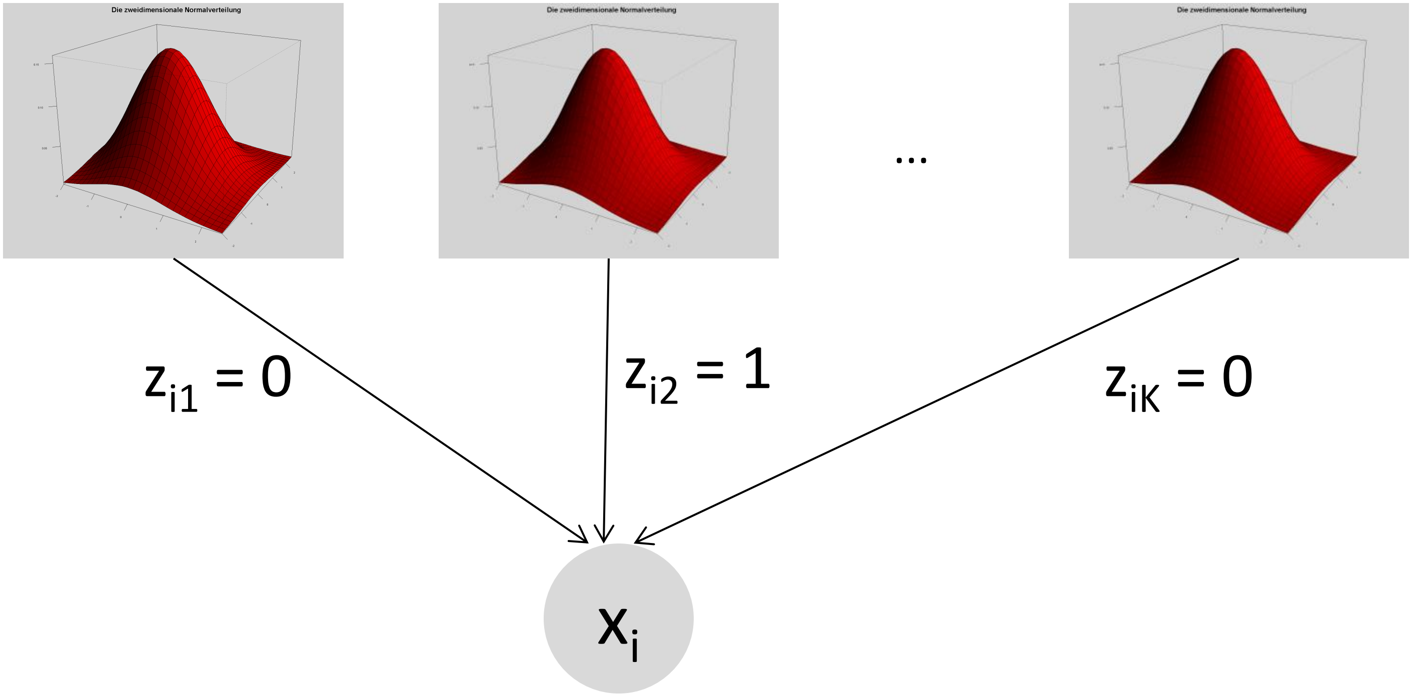
# Gaussian Mixtures Models (GMMs)

- We consider our data to be drawn from a *mixture* of multivariate normal distributions:

$$p(x) = \sum_{k=1}^{K} \pi_k N(x \mid \mu_k, \Sigma_k) \text{ with} \sum_{k} \pi_k = 1, \pi_{\mathrm{k}} \in [0,1]$$

- Interpretation of data generating process:
  - First pick a cluster (component) with probability $\pi_k$
  - Then draw a sample $\mathbf{x}_i$ from that component
- Each data point is generated by one of K multivariate normal distributions

# Gaussian Mixture Models: Schematic View


Die zweidimensionale Normalverteilung
Die zweidimensionale Normalverteilung
Die zweidimensionale Normalverteilung

$z_{i1} = 0$

$z_{i2} = 1$

…

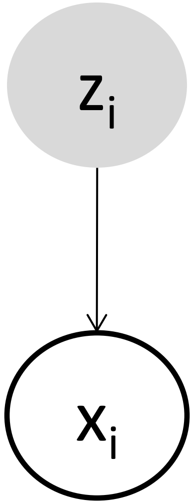
$z_{iK} = 0$

$x_i$

- We formally describe cluster membership of each data point via an indicator variable

# GMMs (cont‘d)

- Problem: true cluster membership of each data point unknown
- ➔ indicator variables are **hidden / latent**
- We would like to make inference on $z_i = (z_{i1},\dots,z_{iK})$ given observed data



# Likelihood

- We suppose data points to be drawn iid
- **Complete likelihood** of observed and unobserved variables:

$$p(\{x_i\},\{z_i\} \mid \mu_1,\ldots,\mu_K,\Sigma_1,\ldots,\Sigma_k) = \prod_{i=1}^{n}\prod_{k=1}^{K} p(x_i \mid \mu_k,\Sigma_k,z_{ik}=1)\,\mathrm{Pr}(z_{ik}=1)$$

- **Problem**: direct maximization w.r.t. parameters AND unobserved variables not possible

# Expectation Maximization (EM) algorithm – initialization

- Start with some cluster assignment
- Estimate parameters of each Gaussian via ML

$$\hat{\mu}_k = \frac{\sum_{i=1}^{n} z_{ik} x_i}{\sum_{i=1}^{n} z_{ik}}$$

$$\hat{\Sigma}_k = \frac{\sum_{i=1}^{n} z_{ik} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}{\sum_{i=1}^{n} z_{ik}}$$

$$\hat{\pi}_k = \frac{\sum_{i=1}^{n} z_{ik}}{n}$$

# Expectation Maximization (EM) algorithm – E-step

- Given parameters of each Gaussian: compute *expected* cluster assignment of each data point:

$$E[z_{ik}] = \Pr(z_{ik} = 1 \mid x_i, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K) * 1$$
$$+ 0 * \Pr(z_{ik} = 0 \mid x_i, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K)$$

$$\overset{\text{Bayeslaw}}{=} \frac{p(x_i \mid \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K, z_{ik} = 1) \Pr(z_{ik} = 1)}{\sum_{k=1}^{K} p(x_i \mid \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K, z_{ik} = 1) \Pr(z_{ik} = 1)}$$

# Expectation Maximization (EM) algorithm – M-step

- Given: expected cluster assignments of data points
- Recompute ML estimates for parameters for each Gaussian

$$\hat{\mu}_k = \frac{\sum_{i=1}^{n} E[z_{ik}] x_i}{\sum_{i=1}^{n} E[z_{ik}]}$$

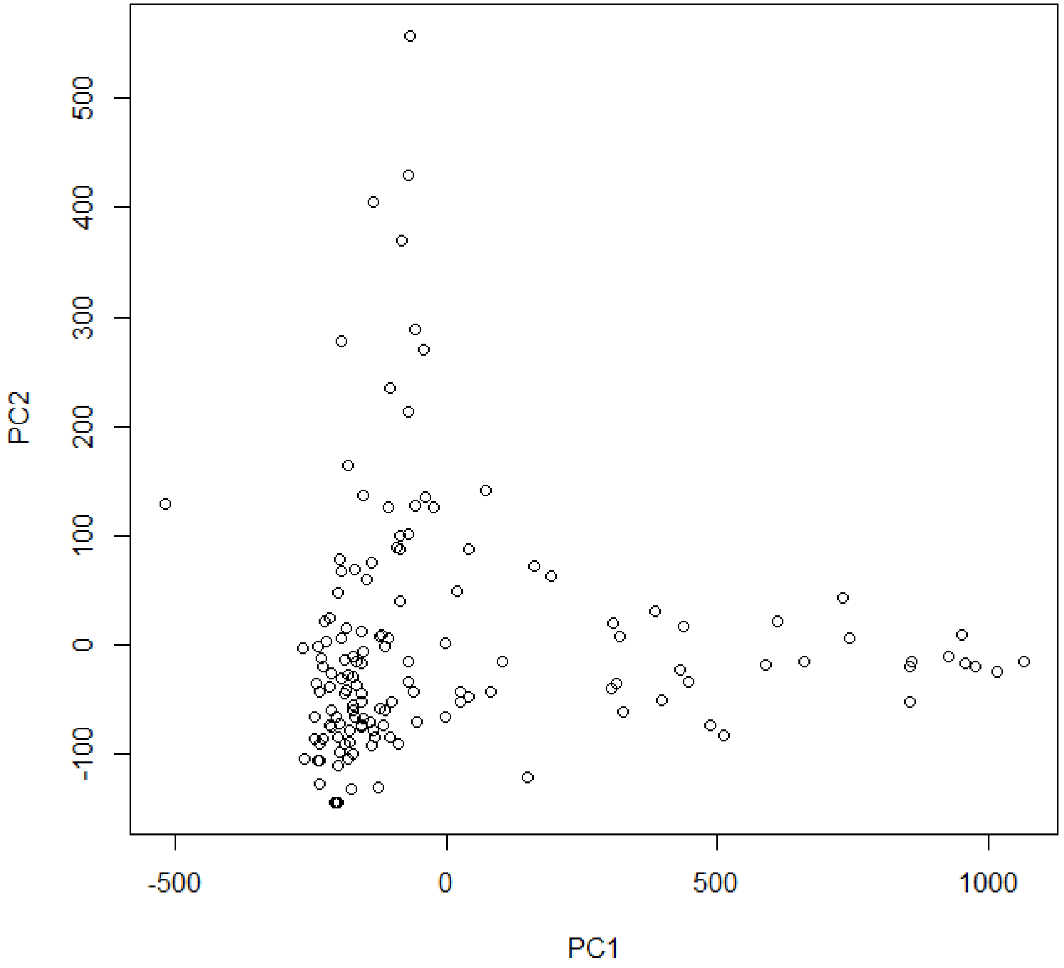
$$\hat{\Sigma}_k = \frac{\sum_{i=1}^{n} E[z_{ik}] (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}{\sum_{i=1}^{n} E[z_{ik}]}$$

$$\hat{\pi}_k = \frac{\sum_{i=1}^{n} E[z_{ik}]}{n}$$

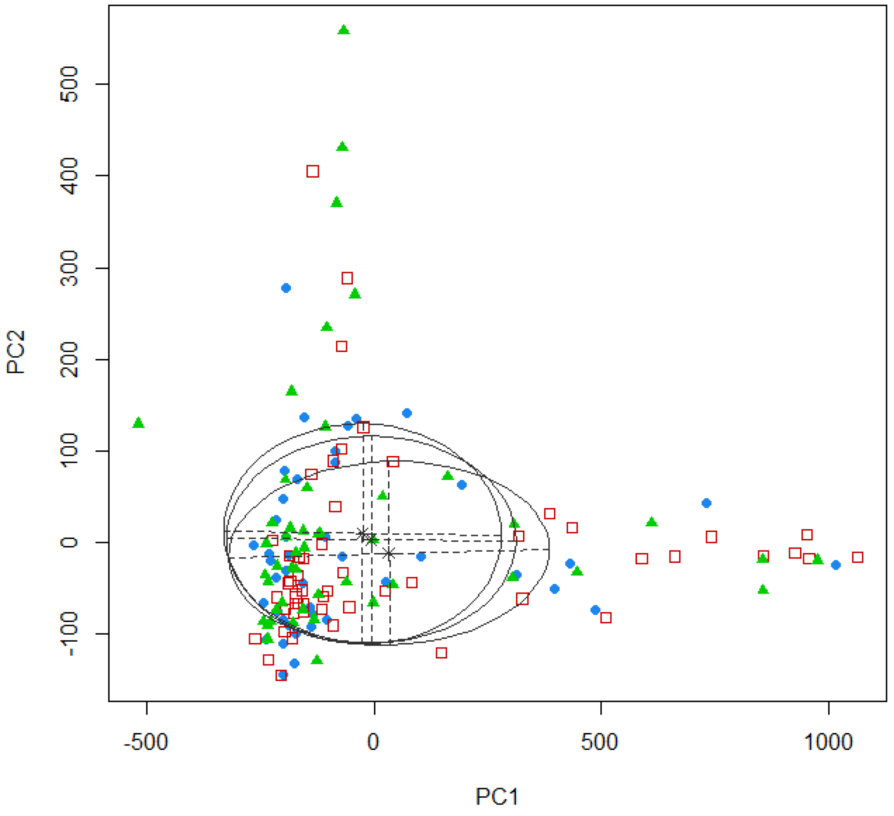
# Complete EM Algorithm

- Initialize cluster assignment (e.g. random or via k-means)
- Iterate until convergence (complete likelihood does not increase significantly):
    - E-step
    - M-step
- Algorithm is guaranteed to increase the complete likelihood in each iteration
- BUT: may get stuck into local optima
    - Sensitive to initialization

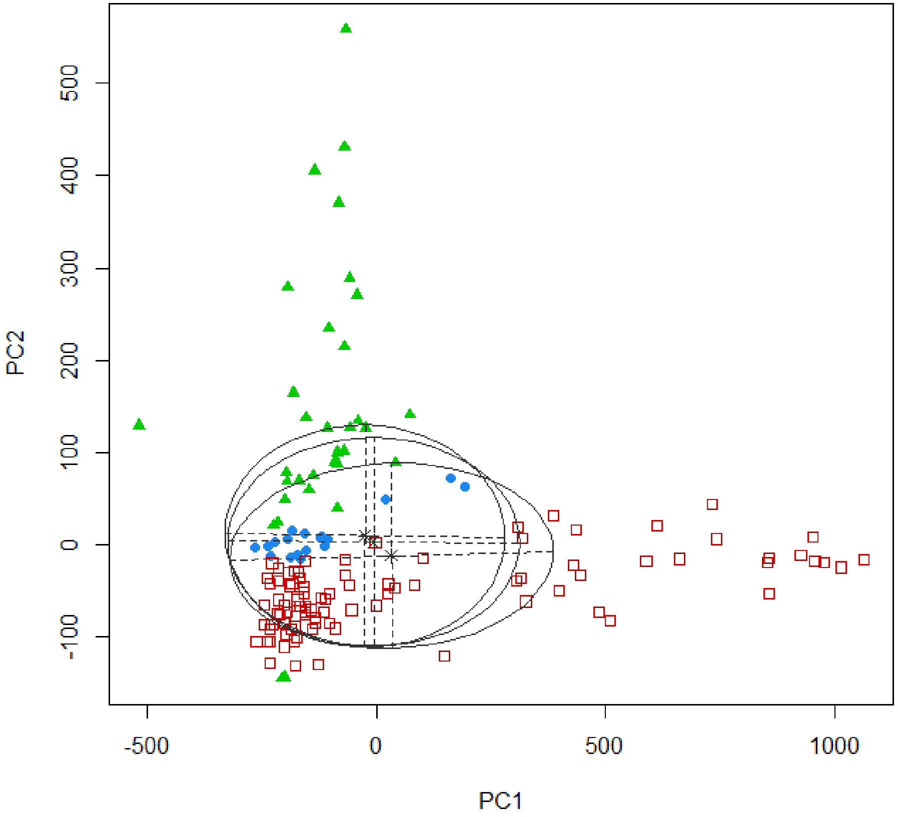
# Example: Diabetes dataset (8 variables) – PCA plot



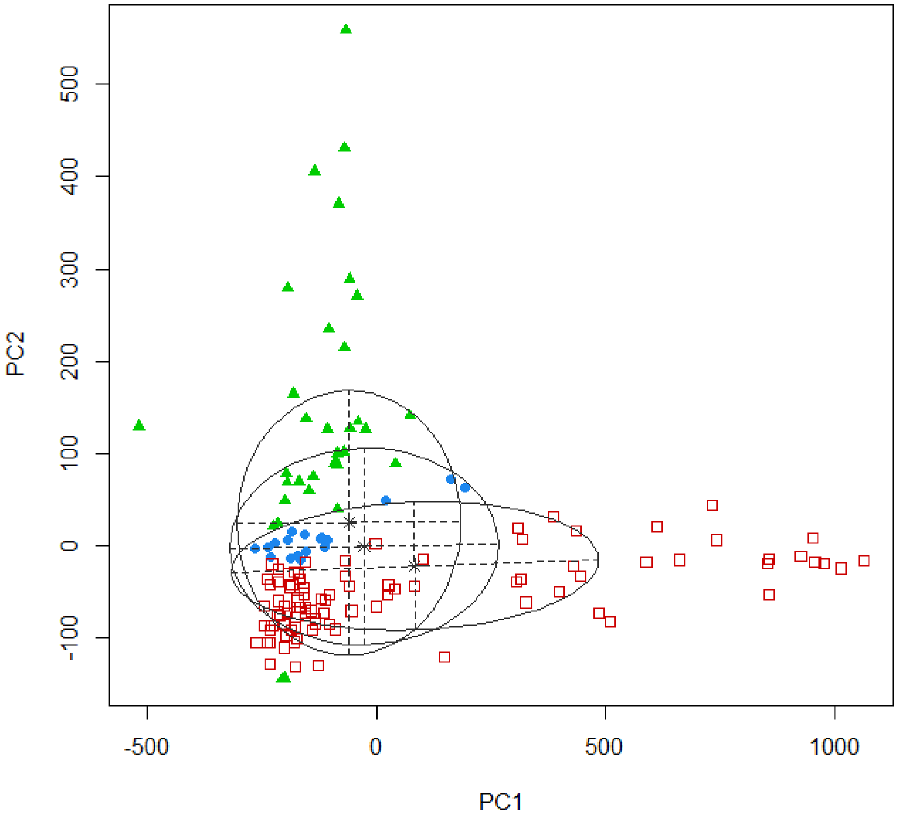
# Initialization (random cluster assigment)



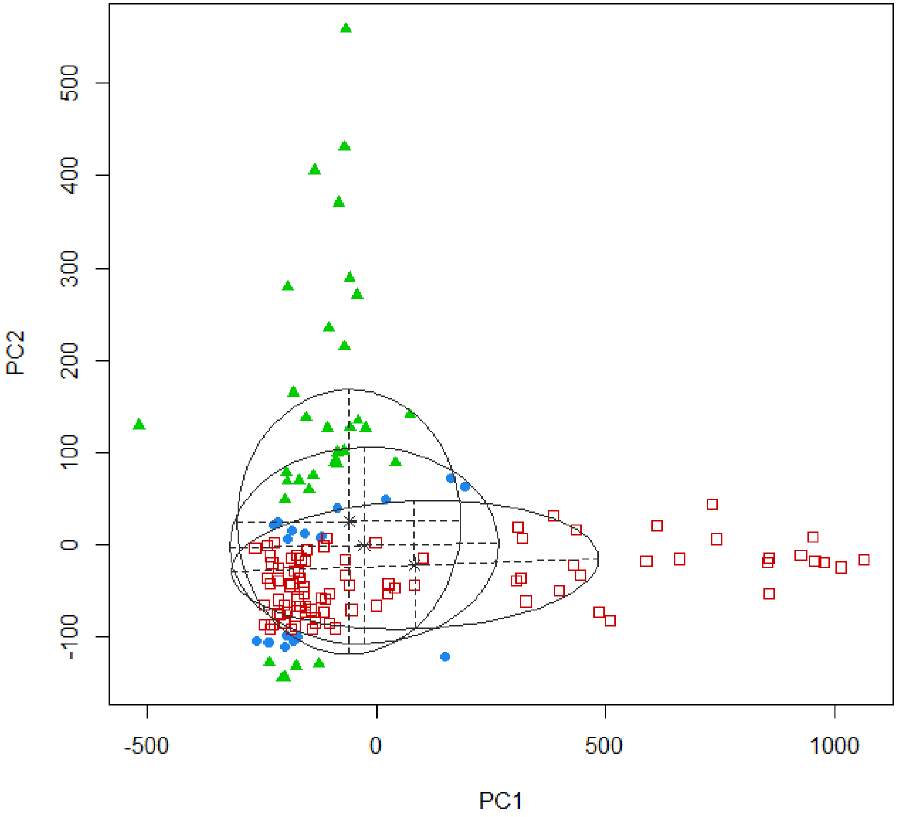
# E-Step



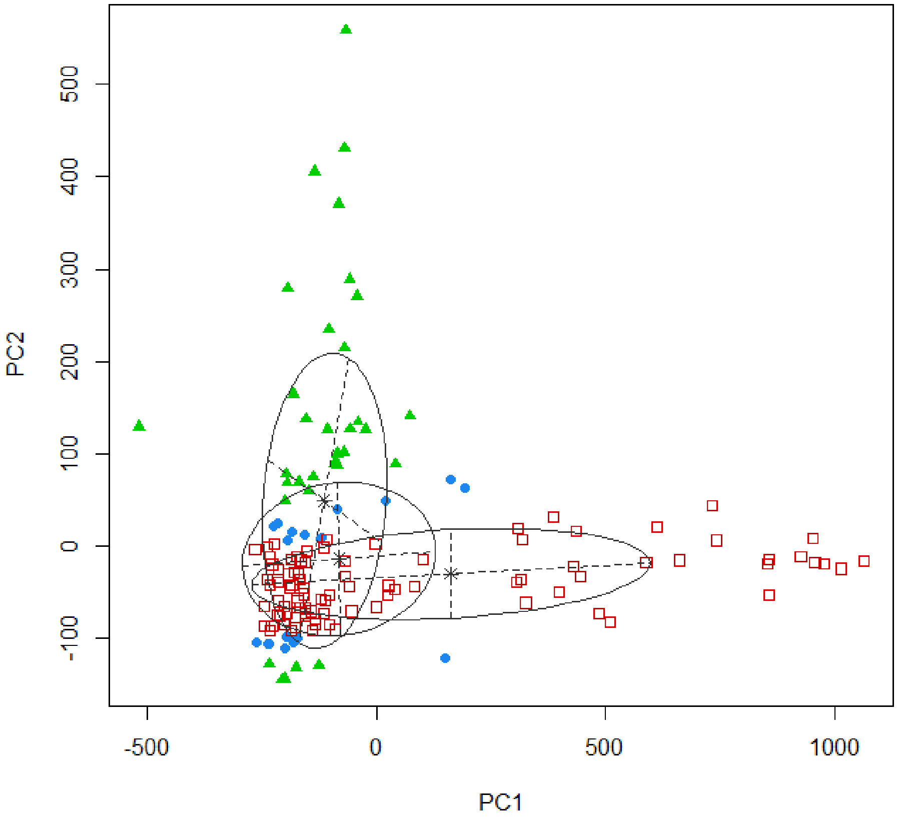
# M-Step



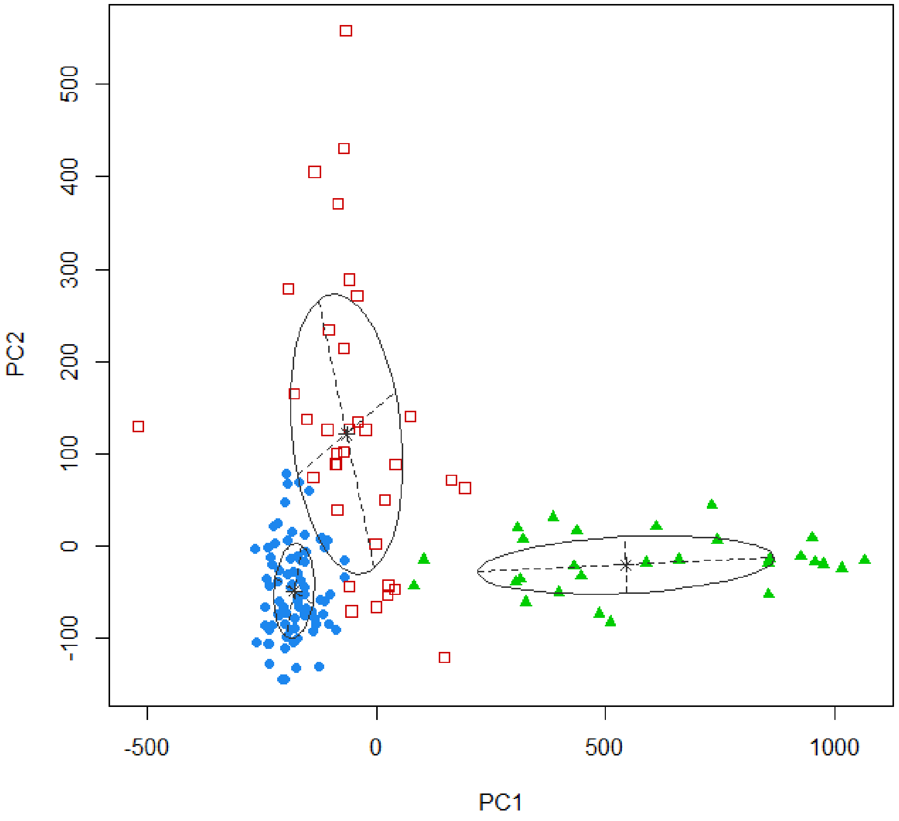
# E-Step (2)



# M-Step (2)



# After convergence



# EM Algorithm in general

- Let $\Theta$ denote the vector of all parameters and let $\Theta^t$ denote its current estimate.
- E-step: find expected value of complete log-likelihood

$$Q(\Theta \mid \Theta^t) := E_{p(z|x,\Theta^t)}[\log p(x, z \mid \Theta)] := \sum_{k=1}^{K} \log p(x, z = k \mid \Theta) P(z = k \mid x, \Theta^t)$$

- M-step: find the parameters that maximize Q:

$$\Theta^{t+1} = \arg\max_{\Theta} Q(\Theta \mid \Theta^t)$$

# Q-function for GMMs

$$Q(\Theta \mid \Theta^t) = \sum_{i=1}^{n} \sum_{k=1}^{K} \log p(x_i, z_{ik} = 1 \mid \Theta) \Pr(z_{ik} = 1 \mid x, \Theta^t)$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \log p(x_i, z_{ik} = 1 \mid \Theta) \Pr(z_{ik} = 1 \mid x, \Theta^t)$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \left( \log N(x_i \mid \underset{\uparrow}{\mu_k}, \Sigma_k, z_{ik} = 1) + \log \underset{\uparrow}{\pi_k} \right) \Pr(\underset{\uparrow}{z_{ik}} = 1 \mid x_i, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K)$$

Normal distribution density for point i

Prior probability for point i to belong to cluster k

Expected cluster membership for point i (slide 43)

# Features of GMMs

- Advantages:
  - Probabilistic cluster assignment, clusters may overlap
  - Algorithm can detect ellipsoidal clusters of different sizes
  - GMMs are **model based**: model may be used to assign future data points to the most likely cluster or to impute missing values
- Disadvantages:
  - need to know number of clusters / mixture components K in advance
    - BUT: existant statistical heuristics (→ model selection) to estimate K from data
  - Slow for large amounts of data

# Selecting the number of clusters k

- For GMMs we can use the so-called model selection criteria, e.g. Bayesian Information Criterion (BIC) to determine a good number of clusters from data:

$$BIC = -\log-\text{likelihood} + 0.5 * \log(n) * npar$$

- **Rational (informal)**: the more clusters we have, the more parameter we need to estimate effectively from data, i.e. the GMM model gets more and more complex

# Selecting the number of clusters k (cont‘d)

- **Problem with overly complex model**:
  - overfitting – other data drawn from the same distribution may not be explained well
  - GMM forms clusters, which are of minor information and do not help to interpret the data
- **Okkams razor principle:** Try to find a model, which is as simple as possible to explain your data sufficiently
- ➔ Clustering is a way to reduce data complexity. If too many clusters, nothing is won

# Bayesian Information Criterion (BIC)

$$BIC = -\log-\text{likelihood} + 0.5 * \log(n) * npar$$

data fit

Complexity penalty

- npar = number of parameters in the model to be estimated
  - Depends on number of clusters
- BIC balances fit to the data and model complexity
- Heuristic approach!

# Using BIC in practice

1. Define a set of cluster numbers K
2. for each k in K:
   a. Run GMM clustering
   b. Determine BIC
3. Select clustering with the lowest BIC

# Clustering Validity

- How can we check the quality of a clustering?
  - GMMs: log-likelihood (depends on k), BIC
  - k-means: distortion (depends on k)
- Problem: values are not on a normalized scale, can only be used in a relative sense
- Visual inspection
  - Plot the data itself (difficult for > 2 dimensions)
  - Plot distance structure (→ clustering silhouettes)
- Clustering indices to measure validity

# Cluster Silhouettes (Rousseeuw, 1987)

- For each observation i assigned to cluster C the silhouette s(i) is defined as:

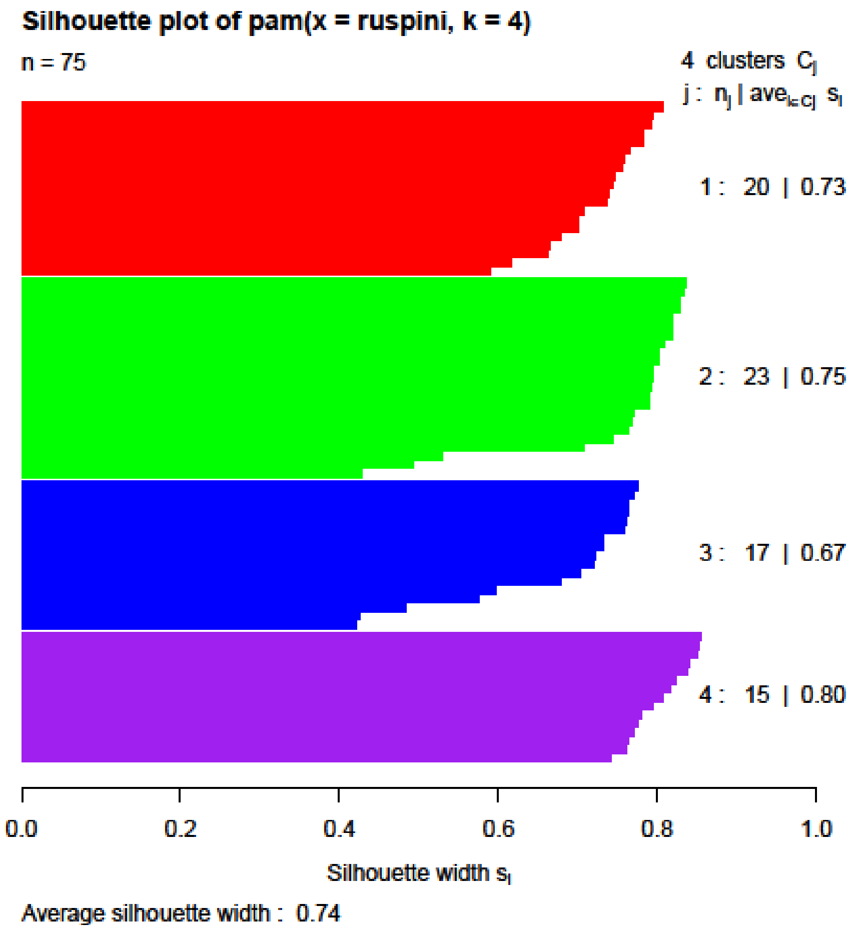
$$s(i) := \begin{cases} 0 & D(i,C) = 0 \\ \dfrac{D(i,B) - D(i,C)}{\max(D(i,C), D(i,B))} & otherwise \end{cases} \in [\text{-}1,1]$$

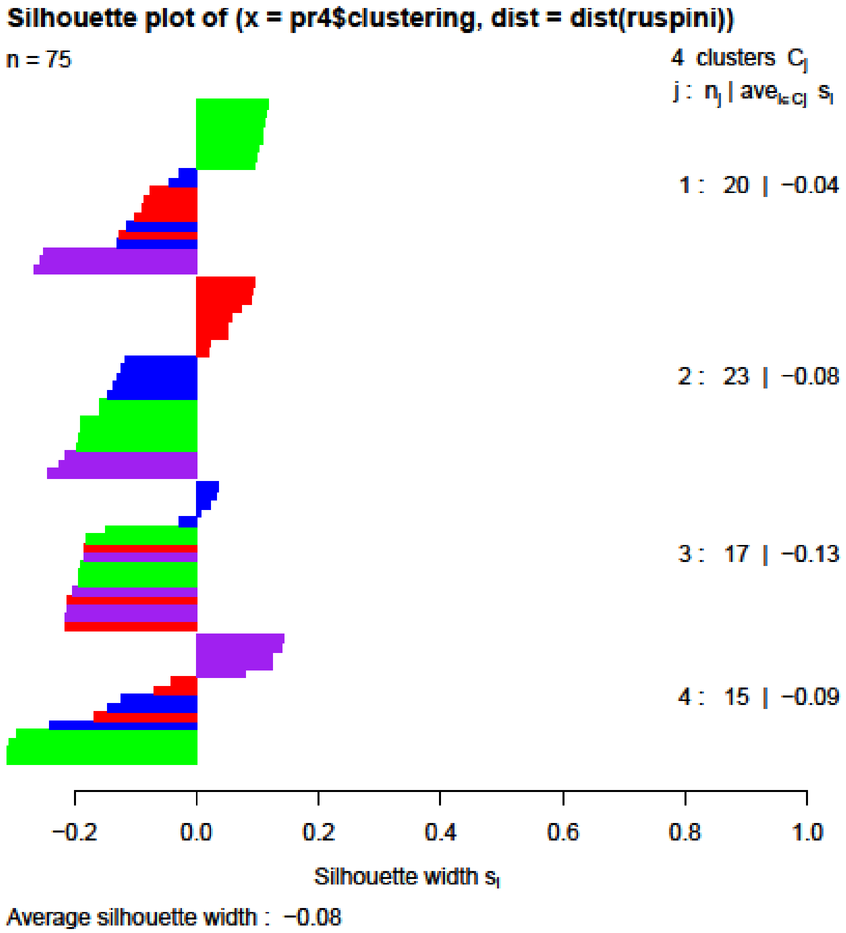
$$D(i,C) := \frac{1}{|C|} \sum_{a \in C} d(a,i)$$

$$D(i,B) := \min_{C' \neq C} D(i,C')$$

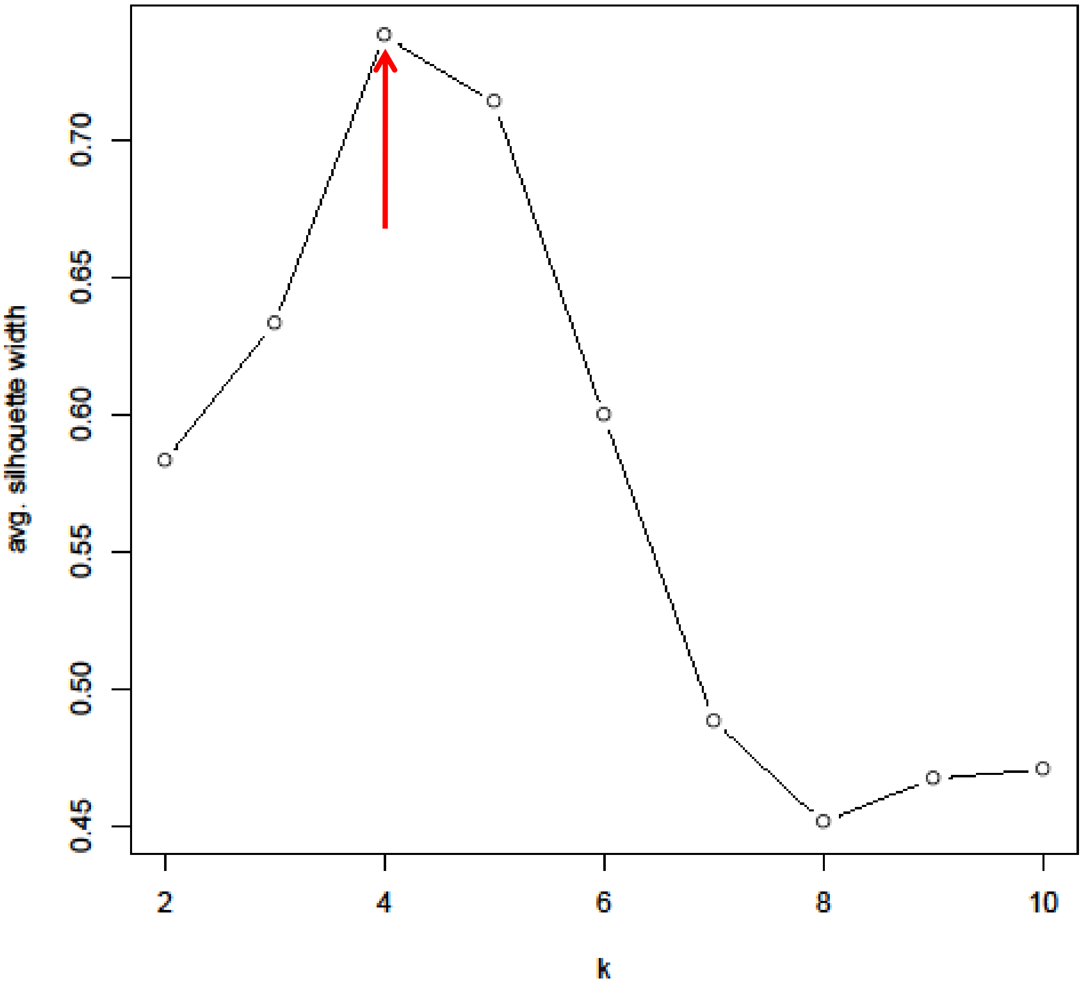
- Interpretation:
  – s(i) close to 1: i lies in cluster C
  – s(i) = 0: i lies between two clusters
  – s(i) < 0: i is close to B than to C

# Examples: good and bad clustering





# Selecting the number of clusters



# Clustering High Dimensional Samples

- **So far**: main focus on clustering of features
- How about clustering of samples?

- **Example**: Can we identify patient sub-populations from molecular data?

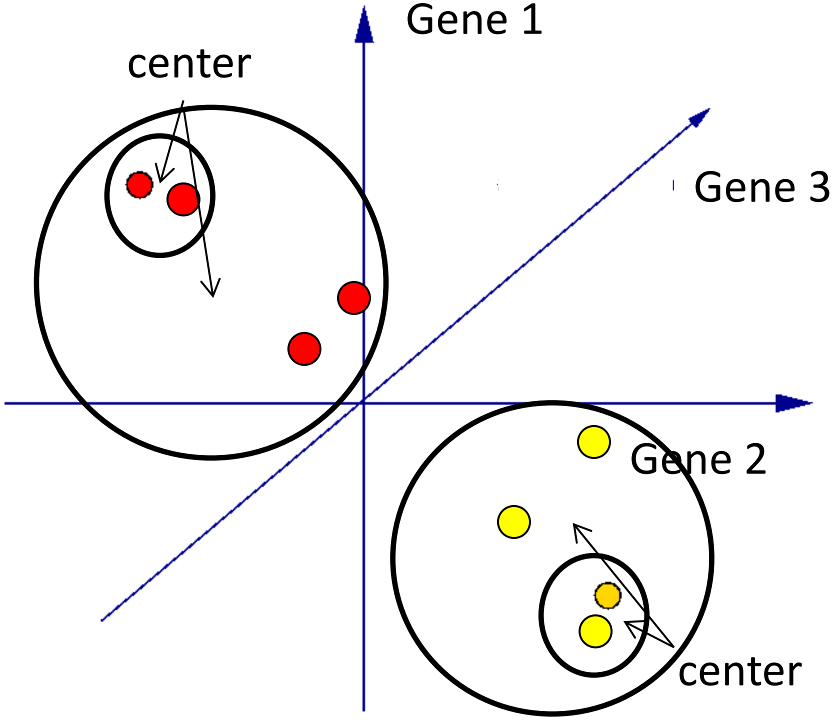
- Problem: far more features than samples

# Clustering in High Dimensions

- Data points are sparsely distributed in a high dimensional space
  - Biology: only few features (< 5%) expected to show significant differences between samples
  - (Euclidean) distance dominated by noise features

$$d(x,y) = \left( \sum_{i=1}^{n} |x[i] - y[i]|^2 \right)^{1/2}$$

  - All data points (patients) become almost equally distant from each other
  - We are clustering noise!
- Consequence: Leaving out or adding a few samples could drastically change clustering (statistical instability)



# Clustering in High Dimensions

1. Reduce features
   - Pre-filter features
   - Use clustering algorithm with in-built selection of most relevant features
2. Look for a pattern that remains statisticaly stable, even if sample set changes slightly
   - Consensus over different clusterings

# Example for Pre-filtering: Verhaak et al., Cancer Cell, 2010

- Data: gene expression profiles of 206 patients with Glioblastoma Multiforme (brain tumor), 3 different technical platforms

- Prefiltering of most variable genes:

1. High correlation across platforms ➔ 9,255 genes
2. High variability (MAD) on each platform ➔ 1,903 genes
3. Exclusion of genes with large differences in MAD across platforms ➔ 1,740 genes

- Consensus average linkage clustering (Monti et al., 2003) shows 4 cancer sub-types
  - Motivation: address statistical instability

# Consensus Clustering (Monti et al., Machine Learning, 2003)

- Idea: assess clustering stability via sub-sampling
  1. Sample p% (default: 80%) of the data points without replacement
  2. Optional: do the same for the features
  3. Run base clustering algorithm (e.g. k-means, average linkage)
  4. Repeat H times

- Main question: how to form consensus out of H clusterings?

- Define connectivity matrix for clustering h:

$$M^{(h)}(i, j) = \begin{cases} 1 & \text{if items } i \text{ and } j \text{ belong to the same cluster,} \\ 0 & \text{otherwise .} \end{cases}$$

- Let $\boldsymbol{I}^{(\boldsymbol{h})}$ be a N x N matrix indicating the presence of i and j:

$$I^{(h)} = \begin{cases} 1 & i \text{ and } j \text{ in subsample } h \\ 0 & \text{otherwise} \end{cases}$$

# Consensus Clustering

Consensus matrix is defined as properly normalized sum of all connectivity matrices:

$$\mathcal{M}(i, j) = \frac{\sum_h M^{(h)}(i, j)}{\sum_h I^{(h)}(i, j)}$$

Normalization takes into account whether both, i and j, are present

Observations:

- $M$ is symmetric
- $M$ has values in [0,1] where 1 means perfect consensus
- $M$ may be viewed as a similarity measure between items
- Reordering rows and columns in $M$ according to true clustering yields a block-diagonal matrix

Consequence: Final clustering can be achieved via hierarchical clustering using $\mathbf{1}-\boldsymbol{M}$ as distance matrix

# Summary Statistics

Consensus matrix provides information about

- Stability of overall clustering
- Stability of individual clusters
- Cluster representatives and outliers

Cluster k consensus:

$$m(k) = \frac{1}{N_k(N_k - 1)/2} \sum_{\substack{i,j \in I_k \\ i<j}} \mathcal{M}(i, j)$$

- Measures average frequency, how often objects in cluster k group together

Consensus item for cluster k:

$$m_i(k) = \frac{1}{N_k - 1\{e_i \in I_k\}} \sum_{\substack{j \in I_k \\ j \neq i}} \mathcal{M}(i, j)$$

- Measures average consensus of item $e_i$ with all other items in cluster k
- **Representative**: item with highest consensus
- **Outliers**: opposite

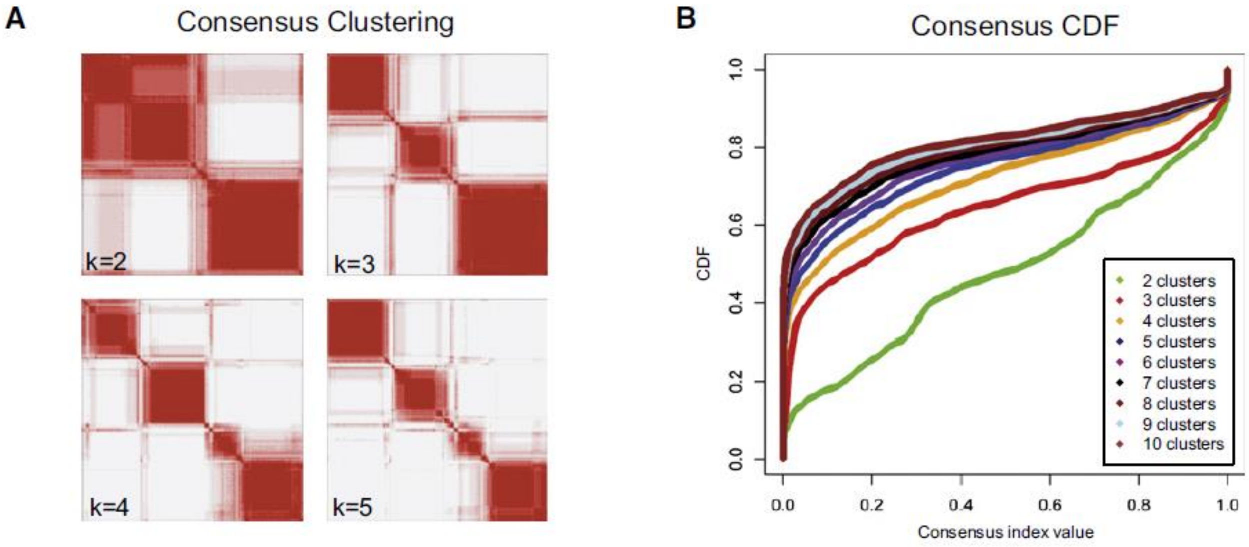
# Determining the Number of Clusters

Summary statistics depend on chosen number K of clusters: How to find a good K?

Idea: consider the distribution of values in consensus matrix

- Stronger skew to 1 ➔ higher stability

Plot empirical CDF of this distribution for different K



# Determining the Number of Clusters

Consider the area under the CDF curves

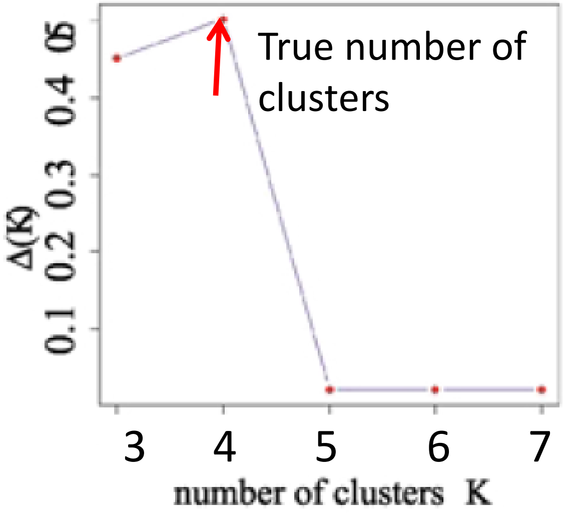
$$A(K) = \sum_{i=2}^{m} [x_i - x_{i-1}]\,\mathrm{CDF}(x_i)$$

Observation (example): area increases significantly from 3 to 4 clusters and then stabilizes

- More clusters cannot be detected stably

Thus: plot K against proportional change of A(K)

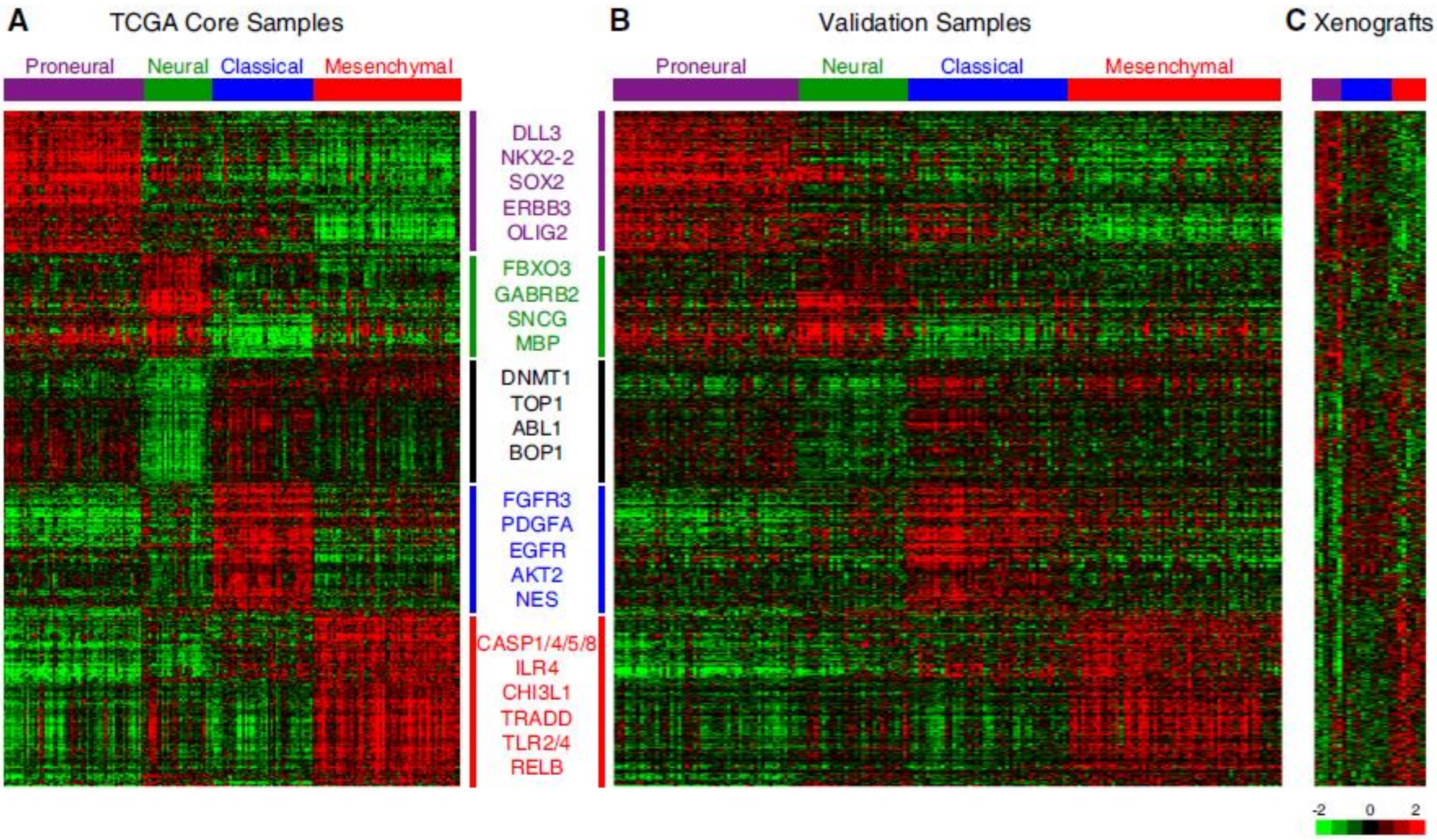
$$\Delta(K) = \begin{cases} A(K) & \text{if } K = 2 \\ \dfrac{A(K+1) - A(K)}{A(K)} & \text{if } K > 2, \end{cases}$$



True number of clusters

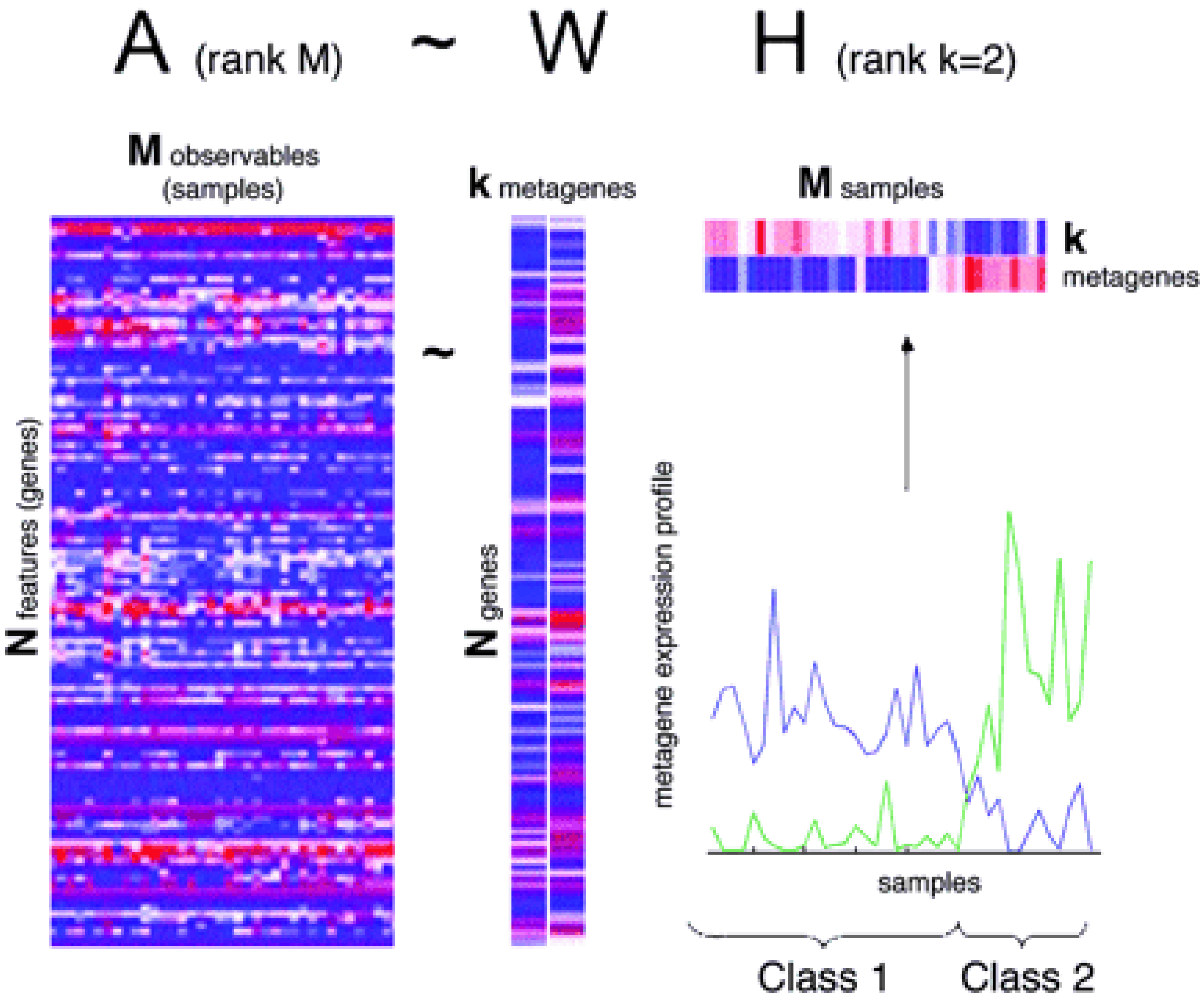
(c)

# Consensus Clustering Result



# Alternative Strategy: Non-Negative Matrix Factorization (Brunet et al., PNAS, 2004)

- Can different leukemia sub-types be identified from gene expression profiles?

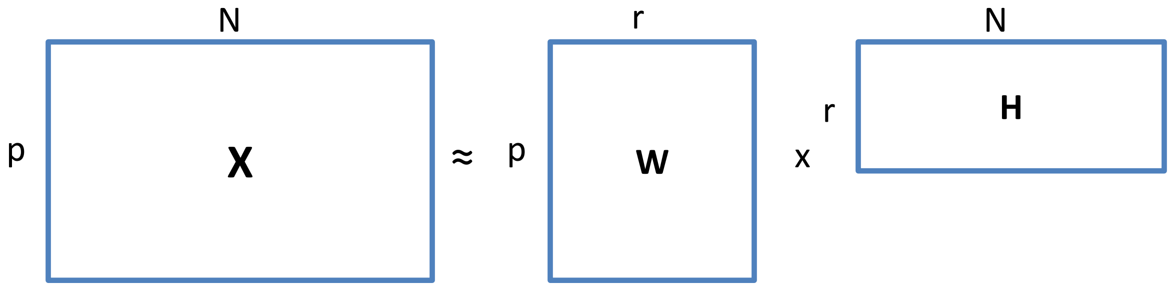


# Non-negative Matrix Factorization (Lee & Seung, Nature, 1999)

Consider p x N data matrix X with non-negative entries

Goal: find low rank approximation of X such that $\mathbf{X} \approx \mathbf{WH}$

- W: p x r, **non-negative**
- H: r x N, **non-negative**
- r << max(N, p)



Approach: $\mathbf{arg\,min}_{W,H}||X - WH||_F$ subject to W, H > 0

- Non-convex
- Not unique

# An Algorithm for Solving NMF (Lee & Seung, 1999)

- Objective: minimize divergence between X and WH

$$D(X\|WH)=\sum_{i,j}\left(x_{ij}\log\frac{x_{ij}}{w_{ij}h_{ij}}-x_{ij}+w_{ij}h_{ij}\right)$$

- Find local minimum by alternating two steps:

$$w_{ik}\leftarrow w_{ik}\frac{\sum_{j=1}^{p}h_{kj}x_{ij}/(\mathbf{WH})_{ij}}{\sum_{j=1}^{p}h_{kj}}$$

$$h_{kj}\leftarrow h_{kj}\frac{\sum_{i=1}^{N}w_{ik}x_{ij}/(\mathbf{WH})_{ij}}{\sum_{i=1}^{N}w_{ik}}$$

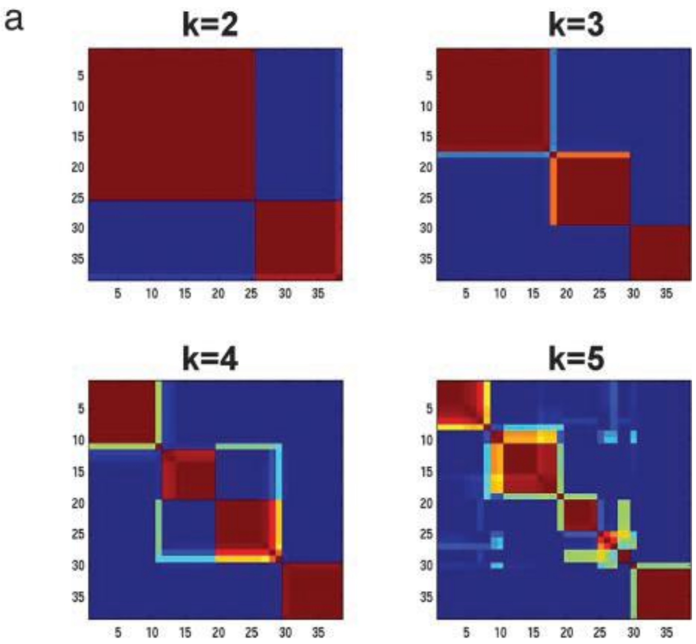
- Several other algorithmic variants known

# NMF Consensus Clustering

- NMF procedure is prone to random initialization

- Idea in Brunet et al. (2004): perform a number of runs and analyze consensus matrix, like in Consensus Clustering
  - Hierarchical clustering based on consensus matrix
  - Choose appropriate number of clusters based on silhouette index or something similar

**Example** (from Brunet et al.): clustering of leukemia patients based on gene expression data



# Summary

- Clustering is an exploratory technique
  - Examples:
    - identification of co-expressed genes
    - Identification of patient sub-populations
  - No prediction
  - Hard to validate
  - No claims about statistical significance of
    - Overall clustering structure
    - Differences between defined clusters
  - Statistical stability
- Hierarchical clustering, k-means, GMMs, NMF as examples
- Other frequently used methods in Bioinformatics
  - K-medoids (PAM)
  - Self Organizing Maps (SOMs)
- Consensus clustering as a means to address statistical instability

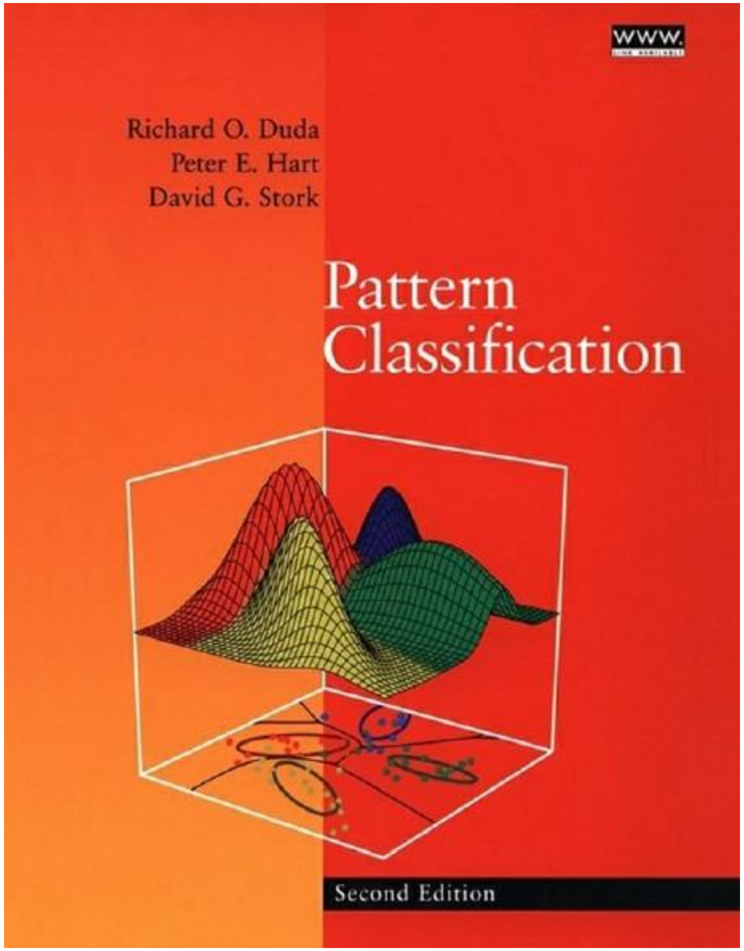
# More Applications in Bioinformatics

- Grouping of homologous sequences into gene families (evolutionary biology)
- Phyolgenetic tree reconstruction
- Multiple sequence alignment
- Biological or medical image analysis

# What you should know and being able to apply

- What is the purpose of clustering?
- Which clustering techniques exist (those which were covered here) and how do they work in principle?
- What are the pros and cons of the individual clustering algorithms discussed here? Which kind of clusters can they detect?
- How can we determine the quality of a clustering and the number of clusters?

# Literature



- Monti, S., Tamayo, P., Mesirov, J. & Golub, T (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning 52, 91–118.
- Daniel D. Lee and H. Sebastian Seung (1999). Learning the parts of objects by non-negative matrix factorization. Nature 401 (6755): 788–791
- Brunet, Tamayo, Golub, Mesirov (2004). Metagenes and molecular pattern discovery using matrix factorization. PNAS 101(12):4164–4169
- Verhaak, R. et al. (2010). Integrated Genomic Analysis Identifies Clinically Relevant Subtypes of Glioblastoma Characterized by Abnormalities in PDGFRA, IDH1, EGFR, and NF1. Cancer Cell 17, 98–110